Université Paris 1 Panthéon-Sorbonne
École doctorale de Philosophie

Thèse pour l'obtention du grade de docteur de l'Université de Paris-1

Spécialité : philosophie des sciences

Maël Pégny

Sur les limites empiriques du calcul

Calculabilité, complexité et physique

Préparée sous la direction de
M. le Professeur des Universités Jean-Baptiste Joinet
M. l'Ingénieur-Chercheur Alexei Grinbaum

Soutenue le 5 Décembre 2013

COMPOSITION DU JURY :
- Professor Jeffrey A. Barrett (rapporteur)
- M. le Directeur de Recherche Gilles Dowek (rapporteur)
- M^{me} la Professeure des Universités Anouk Barberousse
- M. le Chargé de Recherches H.D.R. Jean Lassègue

0.1 Résumé

Mots-clés : théorie de la connaissance, philosophie des sciences, philosophie de la physique, logique, calcul, informatique, calculabilité, complexité computationnelle, thèse de Church-Turing, calcul quantique, a priori, empirique.

Durant ces dernières décennies, la communauté informatique a montré un intérêt grandissant pour les modèles de calcul non-standard, inspirés par des phénomènes physiques, biologiques ou chimiques. Les propriétés exactes de ces modèles ont parfois été l'objet de controverses : que calculent-ils ? Et à quelle vitesse ? Les enjeux de ces questions sont renforcés par la possibilité que certains de ces modèles pourraient transgresser les limites acceptées du calcul, en violant soit la thèse de Church-Turing soit la thèse de Church-Turing étendue. La possibilité de réaliser physiquement ces modèles a notamment été au cœur des débats. Ainsi, des considérations empiriques semblent introduites dans les fondements même de la calculabilité et de la complexité computationnelle, deux théories qui auraient été précédemment considérées comme des parties purement a priori de la logique et de l'informatique. Par conséquent, ce travail est consacré à la question suivante : les limites du calcul reposent-elles sur des fondements empiriques ? Et si oui, quels sont-ils ? Pour ce faire, nous examinons tout d'abord la signification précise des limites du calcul, et articulons une conception épistémique du calcul, permettant la comparaison des modèles les plus variés. Nous répondrons à la première question par l'affirmative, grâce à un examen détaillé des débats entourant la faisabilité des modèles non-standard. Enfin, nous montrerons les incertitudes entourant la deuxième question dans l'état actuel de la recherche, en montrant les difficultés de la traduction des concepts computationnels en limites physiques.

0.2 Abstract

Keywords: epistemology, philosophy of science, philosophy of physics, logic, computation, computer science, computability, computational complexity, quantum computing, Church-Turing thesis, a priori, empirical.

Recent years have seen a surge in the interest for non-standard computational models, inspired by physical, biological or chemical phenomena. The exact properties of some of these models have been a topic of somewhat heated discussion: what do they compute? And how fast do they compute? The stakes of these questions were heightened by the claim that these models would violate the accepted limits of computation, by violating the Church-Turing Thesis or the Extended Church-Turing Thesis. To answer these questions, the physical realizability of some of those models - or lack thereof - has often been put at the center of the argument. It thus seems that empirical considerations have been introduced into the very foundations of computability and computational complexity theory, both subjects that would have been previously considered purely a priori parts of logic and computer science. Consequently, this dissertation is dedicated to the following question: do computability and computational complexity theory rest on empirical foundations? If yes, what are these foundations? We will first examine the precise meaning of those limits of computation, and articulate a philosophical conception of computation able to make sense of this variety of models. We then answer the first question by the affirmative, through a careful examination of current debates around non-standard models. We show the various difficulties surrounding the second question, and study how they stem from the complex translation of computational concepts into physical limitations.

0.3 Remerciements

Comme tout œuvre de l'intellect, ce travail de thèse est le fruit de rencontres fructueuses, de coups de main occasionnels, et de soutiens durables. Si la responsabilité des erreurs qu'il contient ne doit revenir qu'à un seul homme, la responsabilité de ses qualités pourrait bien revenir à plusieurs, dont l'auteur officiel ne sera pas toujours le plus important. C'est dans cette perspective que je tiens à remercier :

mes directeurs de thèse, Jean-Baptiste Joinet et Alexei Grinbaum, pour la liberté qu'ils m'ont laissée, pour leur patience face à mes errements, et avant toute chose, pour avoir réussi à me supporter pendant quatre ans ;

toute l'équipe Larsim du C.E.A., Étienne Klein, Vincent Bontems, Gilles Cohen-Tannoudji, et Issam Ibnouhsein, pour leur gentillesse, pour leur accueil ;

Gilles Dowek, pour ses encouragements, pour les longues discussions à Rio ;

Pablo Arrighi, pour son intérêt, pour son énergie ;

Olivier Bournez, pour sa disponibilité, pour son goût de la discussion ;

Nachum Dershowitz, pour les mêmes raisons ;

Scott Aaronson, pour sa faculté à changer des questions confuses en réponses enthousiasmantes ;

Marc Bagnol, et tous les membres du Groupe de Travail en Logique de l'E.N.S., pour les heures tardives de travail en commun ;

Luiz Carlos, Bruno Lopes Vieira, Fernanda Lobo et Cecilia Reis Englander Lustosa, pour leur accueil à Rio de Janeiro ;

Adrien Guatto, pour tout et n'importe quoi ;

et tous ceux que j'oublie, par un singulier défaut de la mémoire qui, quant à lui, n'a qu'un seul responsable.

Table des matières

Introduction

> *The theory of computation has traditionally been studied almost entirely in the abstract, as a topic in pure mathematics. This is to miss the point of it. Computers are physical objets, and computations are physical processes. What computers can or cannot compute is determined by the laws of physics alone, and not by pure mathematics.* David Deutsch, *The Fabric of Reality*[1]

Si l'on tord un fil de fer en une courbe fermée, que l'on plonge et que l'on extrait cette courbe d'un bain d'eau savonneuse, il se forme rapidement une bulle de savon adhérente au bord du fil de fer. Cette bulle de savon évolue rapidement vers une configuration stable, et forme une surface dont les propriétés géométriques varient en fonction du tracé de la courbe.

Selon les principes de la physique, la bulle de savon ne peut atteindre une configuration stable que lorsqu'elle a formé la surface minimisant l'énergie. Dans des conditions de pression ordinaire, la surface minimisant l'énergie se trouvera être la surface minimale adhérente à la courbe.

Il semble naturel de dire que la bulle de savon résout un problème géométrique : elle calcule la surface minimale adhérente à une courbe donnée. Si cette description ne choque pas l'usage, sa compréhension philosophique précise pose de nombreux problèmes.

La notion de calcul évoque spontanément une manipulation discrète de signes, rendue familière à chacun par la pratique du calcul papier-crayon. Elle évoque également de nos jours les tâches effectuées par les machines numériques programmables qui nous entourent. Mais ces machines exécutent le calcul par une manipulation de signaux discrets. Comme nous le verrons (voir section 1.3.3), ces deux conceptions du calcul sont, à un niveau d'abstraction pertinent, foncièrement identiques : les machines numériques programmables sont une automatisation du calcul papier-crayon.

Dans tous ces cas de figure, l'exécution du calcul est fondamentalement comprise comme un processus *symbolique*. Une telle conception semble immédiatement inadaptée au calcul effectué par la bulle de savon. Ici comme ailleurs, il ne faut pas prédiquer de la chose ce qui appartient à son mode de représentation. Si la bulle de savon obéit à une équation donnée, il est illu-

1. [81], 98

soire de lui attribuer les manipulations de symboles que nous effectuerions pour résoudre une telle équation. L'écart entre ces deux sens de la notion de calcul est si grande, qu'on peut douter qu'il faille bien employer le même mot. Il existe pourtant une analogie fonctionnelle entre ces deux usages : le calcul papier-crayon, comme la bulle de savon, sont pour nous des moyens de connaître la solution d'un problème, ou, pour employer un vocabulaire plus courant dans la communauté des informaticiens, d'exécuter une tâche. En modélisant l'évolution de la bulle de savon par une équation, le physicien accomplit un geste analogue à la spécification d'un programme par un informaticien : il attribue l'exécution d'une tâche précise à un processus donné. En mesurant l'état du système lorsqu'il a atteint une configuration stable, le physicien lit l'état du calculateur après qu'il soit entré dans un état d'arrêt : en d'autres termes, il lit le résultat du calcul. À titre d'hypothèse de départ, à tout le moins, il ne semble pas que la comparaison de ces deux processus -l'évolution d'une bulle de savon, l'exécution d'un calcul papier-crayon- soit le fruit d'une simple confusion linguistique.

La pratique du calcul papier-crayon, loin d'être un exemple paradigmatique, ne serait donc qu'un moyen parmi d'autres d'exécuter un calcul. Nous n'avons pris l'exemple de la bulle de savon qu'à des fins pédagogiques, et il ne constitue en aucun cas une occurrence isolée. La plus ancienne machine de calcul connue, la machine d'Anticythère, n'est pas une automatisation du calcul papier-crayon. Elle constitue un exemple d'un type de machines qui fut l'objet d'une étude intense, et de nombreuses implémentations de la fin du XIX^e siècle à la fin des années 1930, avant le triomphe des machines numériques : les machines analogiques. Celles-ci consistent en une exécution du calcul non par une manipulation de symboles discrets, mais par des processus continus [2]. L'intuition d'une conception non-symbolique du calcul est donc extrêmement ancienne. Néanmoins, il nous semble que la conception philosophique du calcul a été dominée par une compréhension de ce dernier comme processus symbolique, aux dépens d'autres modalités. La prise de conscience de la pluralité interne de la notion de calcul doit permettre un renouveau profond de l'approche philosophique de cet objet, dont nous allons à présent esquisser les enjeux sous forme d'un programme de recherche.

La nécessité d'un réexamen des catégorisations philosophiques du calcul est rendue plus urgente encore par certains courants de la recherche informatique contemporaine. Ces trois dernières décennies ont vu l'explosion de l'attention consacrée au modèles de calcul différents du calcul symbolique, qu'on les qualifie de modèles non-standard, calcul naturel, ou calcul non-

2. Découverte près des côtes de l'île d'Anticythère en 1901, la machine est datée de -87 av. J.C., mais sa conception est probablement beaucoup plus ancienne. Elle consiste en un système de roues dentées tournant sur plusieurs axes, permettant le calcul de dates d'événements astronomiques.

conventionnel [3]. Une attention toute particulière a notamment été consacrée à un modèle, le calcul quantique, qui est devenu l'objet d'une sous-discipline à part entière, mêlant physiciens et informaticiens.

Cet accroissement de l'intérêt pour des conceptions originales du calcul a permis une résurgence de questions fondationnelles en logique et en informatique. La question fondamentale posée par ces modèles non-standard peut intuitivement être formulée de la manière suivante : constituent-ils juste une autre manière d'effectuer les mêmes calculs, ou peut-on dire qu'ils permettent d'effectuer des calculs différents ?

Cette question peut être formulée plus rigoureusement. Un modèle de calcul bien défini peut se voir attribué un *pouvoir expressif*, soit un ensemble de problèmes décidables au titre de ce modèle. Dans certains cas, un modèle permet aussi de définir des mesures de complexité, et donc de définir la difficulté des problèmes calculables au titre de ce modèle. La théorie de la complexité permet ainsi de distinguer, au sein des problèmes calculables au titre du modèle, une classe de problèmes dits *faisables* (*tractables*), *faciles*, ou objets d'un calcul *efficace*. La détermination de la classe des problèmes décidables efficacement au titre d'un modèle constitue le *pouvoir calculatoire*(*computational power*) d'un modèle. La théorie de la calculabilité, et la théorie de la complexité computationnelle, permettent donc de définir ce qu'on peut appeler des limites du calcul : elles nous permettent de spécifier ce qui est calculable, et ce qui est calculable efficacement au titre d'un modèle.

Notre théorie actuelle du calcul, si on entend par là la théorie de la calculabilité et la théorie de la complexité computationnelle, ne se borne pas à l'étude d'une constellation de modèles disparates. Son unité est fondée sur deux principes computationnels : la thèse de Church-Turing, et la thèse de Church-Turing étendue. La première affirme que tout ce qui est calculable est calculable par une machine de Turing. La seconde affirme que tout ce qui est calculable efficacement est calculable efficacement par une machine de Turing. Les deux thèses énoncent deux limitations fondamentales du calcul.

Ces deux limitations, néanmoins, ont toutes deux été énoncées pour des modèles du calcul inspirés par le calcul papier-crayon. Les arguments en faveur de ces thèses ne s'appliquent pas immédiatement à des modèles inspirés par une conception du calcul profondément différente. L'ensemble des tâches effectuables par un calcul est-il l'ensemble des tâches effectuables par un calcul papier-crayon, et si oui, pourquoi ? Les mêmes questions se posent naturellement à l'égard de l'ensemble des tâches calculables efficacement.

Il existe dans la littérature scientifique de nombreux modèles visant à violer soit la thèse de Church-Turing, soit la thèse de Church-Turing étendue. L'intérêt porté au calcul quantique dépend ainsi largement de sa capacité présumée à violer la thèse de Church-Turing étendue. La possibilité de re-

3. Nous traduisons ici les expressions anglaises *non-standard models, natural computing* et *unconventional computing.*

penser les limites fondamentales du calcul à la lumière de nouveaux modèles est donc devenu l'objet d'un débat scientifique intense ces dernières années.

Pour le philosophe, les arguments échangés lors de ce débat sont tout aussi remarquables que les positions qu'ils sont censés appuyer. Parmi les critiques récurrentes adressées aux modèles prétendant violer la thèse de Church-Turing ou sa forme étendue, il existe le reproche de ne pas être physiquement réaliste. On peut même dire qu'en règle générale, le problème essentiel posé par de tels modèles n'est pas tant leur cohérence, ou leur bonne définition, que la possibilité de les implémenter. En discutant l'implémentabilité de certains modèles, le débat sur les limites du calcul parait ne plus être borné à ce qui est mathématiquement ou logiquement possible, mais à ce qui est empiriquement possible : les limites fondamentales du calcul dépendraient ainsi de considérations empiriques.

Cette dernière assertion constitue un paradoxe pour la compréhension ordinaire du calcul. Le calcul est ordinairement considéré à la fois, d'un point de vue pratique, comme une partie essentielle de la praxis des mathématiques pures et de toute discipline mathématisée, et, d'un point de vue théorique, comme l'objet de la science informatique. Au premier abord, le calcul appartient donc au domaine de la connaissance *a priori*. Pour une science empirique comme la physique, le calcul est un outil de représentation, mais en aucun cas un objet d'étude. Si les limites du calcul dépendent de considérations empiriques, il devient possible d'envisager, comme le fait le physicien D. Deutsch dans notre citation liminaire, que ce soit la physique qui fonde en dernier recours les limites du calcul.

L'enjeu des débats autour de la thèse de Church-Turing et de sa forme étendue n'est donc pas uniquement la valeur de vérité de ces propositions, mais leur statut épistémologique. Peut-on considérer ces hypothèses comme des propositions empiriques [4] ? Et si oui, peut-on articuler un ensemble d'hypothèses empiriques caractérisant les contraintes qu'elles feraient peser sur nos théories physiques ? On appellera *interprétation empirique des limites du calcul* la position consistant à défendre que les limites ultimes du calcul dépendent d'hypothèses empiriques. Cette position se scinde en une *interprétation empirique de la calculabilité* et une *interprétation empirique de la complexité*, dont nous verrons qu'elles posent des problèmes autonomes.

4. Dans l'intégralité de ce travail, sauf indication contraire, nous emploierons le qualificatif d'empirique dans son acception la plus libérale, qui comprend aussi bien des propositions de sens commun, des lois expérimentales, des rapports d'observation que les principes les plus fondamentaux de notre théorie physique. Le qualificatif d'empirique devra donc être compris par opposition à celui d' *a priori* ou de mathématique, et non de théorique. Nous emploierons occasionnellement l'expression « état de choses » pour désigner la référence d'une proposition empirique quelconque. Selon une telle convention, il existe un état de choses correspondant au principe d'équivalence, ou au principe de conservation de l'énergie, comme il existe un état de choses correspondant à un rapport météorologique. Cette convention doit être prise comme une simple commodité terminologique, et non comme une prise de position en faveur de quelque philosophie du langage que ce soit.

Enfin, nous parlerons de la *signification empirique* des principes computationnels, pour désigner les contraintes que ces principes, s'ils venaient à être considérés comme des principes empiriques, feraient peser sur notre théorie physique.

Cette interaction entre calcul et physique se retrouve également à un niveau d'abstraction supérieur, celui de la conception des langages de programmation. La réflexion sur les modèles non-standard a pu ainsi mener à la conception de nouveaux langages de programmation adaptés à ces modèles (voir par exemple les langages mentionnés dans la section 8.4.2). L'intérêt philosophique de la conception de ces langages est renforcé par l'émergence, durant ces dernières décennies, d'un paradigme commun à l'informatique et à la théorie de la démonstration : le paradigme preuves-programmes. Ce paradigme constitue une nouvelle compréhension de la nature et de la fonction de la logique, puisqu'il consiste à affirmer qu'une preuve doit être fondamentalement comprise comme un programme, une formule comme un type, une logique comme un langage de programmation. Dans cette nouvelle optique, la logique ne vise pas seulement à capturer le raisonnement, mais aussi à fournir une description théorique des propriétés du calcul (voir les introductions de [143] et de [218],[142], [155], [154], [111] et [112] pour une introduction technique). La recherche naissante en informatique quantique mène ainsi à un intérêt renouvelé, dans son intensité comme dans ses visées, pour les logiques quantiques, conçues à nouveaux frais comme des langages de programmation pour les ordinateurs quantiques.

Ce renouvellement scientifique est d'un intérêt d'autant plus vif pour le philosophe, que les logiques quantiques ont déjà été l'objet d'une grande attention philosophique. Initié dans les années 30 par les mathématiciens J. von Neumann et G. Birkhoff [29], le travail en logique quantique a tout d'abord été proposé comme un nouveau formalisme plus adapté à la physique quantique. Elle avait ensuite fait l'objet d'un commentaire philosophique développé par les philosophes américains W.V.O. Quine [198] et H. Putnam [196], [197]. L'existence de la logique quantique y fut conçue comme une preuve par l'exemple que le choix d'une logique pouvait, à tout le moins en droit, être influencé par des considérations d'origine empirique, et ne devait donc pas être considéré comme une question purement *a priori*. Le renouveau de l'activité scientifique autour des logiques quantiques invitent à un réexamen à nouveaux frais de ces positions philosophiques.

Enfin, il nous parait manifeste que ce programme de recherche devra déboucher sur une interrogation dépassant le simple cadre de la philosophie des sciences spéciales, pour aboutir à la philosophie générale de la connaissance. L'ensemble de ces réflexions mêlant physique, logique et calcul invite à un retour sur le sens et la pertinence de la catégorie de connaissance *a priori*, en particulier lorsqu'elle est appliquée à la notion de calcul et à la logique. La catégorie d'*a priori* se retrouve en effet à tous nos niveaux de réflexion. Plusieurs philosophes contemporains ont ainsi souligné que, si on applique

la distinction entre *a priori* et *a posteriori* non seulement à une proposition, mais aussi aux moyens par lesquels on parvient à la connaissance de cette proposition, on peut admettre la possibilité de principe de propositions *a priori* connues par des moyens *a posteriori* [99], [37]. Le philosophe des sciences E. Sober a ainsi remarqué que la bulle de savon constituait un moyen *a posteriori* de résoudre une instance du problème de Plateau, sur la surface minimale adhérente à une courbe fermée [214]. Nous prolongerions cette remarque en soulignant qu'il ne s'agit nullement là d'une simple curiosité, mais au contraire d'un cas très fréquent dans l'histoire des moyens de calcul. Les résultats de calcul constituent un exemple majeur de propositions *a priori* qu'on peut connaître à la fois par un moyen *a priori* -le calcul papier-crayon- et un moyen *a posteriori* -les machines analogiques et autres modèles non-standard.

Un potentiel renouvellement de la réflexion sur l'*a priori* peut aussi être trouvé à notre deuxième niveau d'analyse, celui de l'interprétation empirique des limites du calcul. Il s'agit là d'un véritable changement de statut épistémologique de certaines propositions, et au-delà, parce que ces propositions se trouvent aux fondements de l'étude théorique du calcul, de disciplines entières. Enfin, à notre troisième niveau d'analyse, la réflexion sur les langages de programmation pour l'informatique quantique, et sur les langages de programmation pour les modèles non-standard en général, invitent à jeter un regard neuf sur les problématiques liées au pluralisme logique, et aux vertus théoriques attribuées à une logique. Pour la logique quantique orthodoxe de J. von Neumann et G. Birkhoff, ainsi que pour leurs commentateurs philosophiques comme W.V.O. Quine et H. Putnam, le choix d'une logique différente de la logique classique, dusse-t-il rester au stade purement spéculatif, pouvait prendre en compte des considérations venues d'une science empirique, comme les difficultés de l'interprétation de la mécanique quantique, et la conception d'un formalisme mathématique plus adapté à cette nouvelle physique. Dans la nouvelle interprétation des logiques quantiques *qua* langages de programmation, les facteurs considérés seront la meilleure adéquation à la programmation pour des machines fondées sur la mécanique quantique.

L'approche mènera naturellement à un croisement des réflexions philosophiques sur la notion d'*a priori*, en particulier en philosophie de la logique et des mathématiques, qui sont très présentes dans la littérature américaine contemporaine, et des réflexions provenant de scientifiques, physiciens ou informaticiens, ayant effectué un retour critique sur leur pratique et sur les avancées récentes de l'étude scientifique du calcul (D. Deutsch [82], J. Barrow [20], S. Aaronson [1]). Du côté philosophique, le sens et la pertinence de la notion d'*a priori*, en particulier dans la philosophie des mathématiques, ont été vivement débattus, certains allant jusqu'à proposer l'abandon total de la notion (P. Maddy [165], P. Kitcher [151], [150]), tandis que d'autres prenaient sa défense (M. Friedman [107], [105], [106], [38]). Du côté scientifique, l'in-

tervention de considérations physiques dans des questions fondamentales de l'étude du calcul, comme celles de ses limites, ou la conception de nouvelles logiques, mène naturellement à interroger le statut *a priori* de ce domaine scientifique. Le débat sur la notion d'*a priori*, et le débat spécifique sur le calcul, sont ainsi voués à s'éclairer mutuellement.

En résumé, notre programme de recherche porte sur trois enjeux essentiels : la possibilité d'exécuter le calcul par des moyens autres que symboliques ; l'éventualité que ces moyens permettent de repousser les limites connues du calcul, en calculant d'autres tâches, ou en les calculant plus efficacement ; enfin la possibilité d'une nouvelle description théorique du calcul, à travers de nouveaux langages de programmation, ou logiques. Le rôle de la notion d'*a priori* dans notre compréhension du calcul peut être examiné sous un angle nouveau pour chacun de ses enjeux.

De ce vaste programme de recherche, la présente thèse ne pourra réaliser qu'une partie. Si une telle restriction a bien sûr des causes contingentes, elle se justifie également par des raisons philosophiques. La première est la prégnance, dans le discours des acteurs scientifiques eux-mêmes, d'un changement de perspective épistémologique sur leur activité de recherche, et leur inscription disciplinaire. Plusieurs chercheurs ayant participé activement à l'émergence et au développement du calcul quantique, tels que D. Deutsch, P. Shor et S. Aaronson, ont tous pu affirmer avoir pris conscience de l'existence de fondements physiques de la calculabilité et de la complexité. S. Aaronson et D. Deutsch ont également articulé la possibilité d'une rétroaction de la théorie du calcul sur la théorie physique elle-même, par la formulation d'hypothèses physiques douées de sens computationnel. Le sentiment d'inadéquation de nos catégorisations usuelles du calcul, et l'interrogation sur d'éventuelles interprétations empiriques de ses limites, loin d'être le fruit d'un commentaire *post factum* du philosophe, fait partie des thématisations réflexives d'une science en train de se faire, comme le montre les intitulés des colloques et des ouvrages, individuels ou collectifs (voir par exemple [64],[44]).

À notre connaissance, tant par le biais des publications que des conversations avec de nombreux chercheurs, une telle conscience n'existe pas, par exemple, dans la communauté travaillant sur les logiques et langages de programmation quantiques. Outre des contingences sociologiques, une telle différence peut être justifiée par une difficulté philosophique réelle. La conception d'un langage de programmation constitue assurément un degré d'abstraction supplémentaire par rapport à la conception d'un nouveau modèle de machine, et ce degré d'abstraction constitue un éloignement supplémentaire des contraintes de l'implémentation. Le sens en lequel un langage de programmation peut être dépendant de la physique régissant la machine exécutant les programmes en est d'autant plus difficile à déterminer. En outre, la sophistication mathématique des travaux de ce domaine constitue un obstacle objectif, et non simplement subjectif, à leur catégorisation philosophique,

tant le travail philosophique sur l'informatique théorique est encore dans son enfance.

À l'opposé du spectre des options stratégiques, une étude qui aurait été d'emblée centrée sur un concept philosophique sophistiqué et doué d'une très longue histoire, comme la notion de connaissance *a priori*, aurait couru le risque de se perdre dans les méandres des conceptualisations philosophiques existantes, et de ne pouvoir prendre langue avec la pratique scientifique contemporaine qu'en bout de course. Pour emprunter un terme de l'ingénierie, à une telle approche *top-down*, nous avons préféré une approche *bottom-up*, qui part des fulgurances conceptuelles isolées issues de la pratique scientifique pour les assembler, à terme, en une totalité philosophique cohérente. Dans une telle perspective, seule une approche centrée sur les enjeux de la calculabilité et de la complexité promettait la conciliation de la proximité avec la pratique scientifique, propre à la philosophie des sciences spéciales, et la volonté d'interroger les catégorisations philosophiques fondamentales de ces mêmes sciences, propre à la philosophie générale de la connaissance.

Ces justifications faites, il nous reste à exposer la marche que nous allons suivre. Nous traiterons successivement des problématiques liées à la calculabilité et à la complexité. La première raison de cette progression est contingente : les littératures traitant de la calculabilité et de la complexité se sont développées de manière autonome, peu de travaux abordant les deux questions de front. La deuxième est plus profonde, et peut justifier *a posteriori* la première. Malgré la présence d'arguments analogues, et de nombreuses ressemblances de surface, chaque problématique révèle à plus ample examen des enjeux propres. Ces deux raisons ont justifié un traitement séparé, laissant à notre conclusion le soin de souligner convergences réelles et divergences profondes.

Nous commencerons par distinguer, comme il est devenu répandu dans la littérature récente, deux formes de la thèse de Church-Turing (partie I), la forme algorithmique et la forme empirique. La première correspond aux intentions historiques des auteurs de cette thèse, et porte sur les limites du pouvoir expressif des procédures effectives (chapitre 1). La seconde porte sur les limites du calcul effectué par un système physique (chapitre 2). Nous examinerons les objections faites à cette distinction, en prenant un soin particulier à examiner les arguments niant que la thèse de Church-Turing empirique constitue une proposition empirique autonome de la thèse algorithmique. Toutes ces objections nous paraîtront insuffisantes, mais leur examen nous mènera à distinguer plus précisément les enjeux de la thèse empirique.

Avant de poursuivre sur cette thématique, nous ouvrirons une parenthèse sur certaines objections faites à l'endroit de la thèse de Church-Turing algorithmique, indépendemment des considérations liées à la thèse empirique (partie II). La thèse de Church-Turing algorithmique ne capturerait que le pouvoir expressif d'une forme particulière de calcul, pour des raisons purement conceptuelles, et non empiriques. Si ces objections se révélaient valides,

la thèse algorithmique constituerait une mauvaise base de comparaison pour évaluer les limites ultimes du calcul par un système physique, puisqu'elle ne porte que sur une forme restreinte du calcul qu'on peut concevoir avant même de prendre en compte des modèles inspirés par la Nature. Ce serait donc une erreur majeure que de considérer un énoncé technique portant sur une sous-forme particulière du calcul avec un énoncé fondamental sur les limites ultimes de la calculabilité par une procédure effective. Le premier ensemble d'objections porte sur la difficile généralisation de la thèse algorithmique aux fonctions réelles. Ces difficultés montreraient que la thèse de Church-Turing algorithmique n'exprime les limites de la calculabilité non seulement que pour les procédures effectives, mais aussi que pour les fonctions entières (chapitre 3). Le second porte sur son application à une famille récente de modèles de calcul, dits interactifs, et reproche à la thèse algorithmique d'être prisonnière d'une conception fonctionnelle du calcul (chapitre 4). Même si nous adopterons encore une fois un regard critique sur ces objections, elles nous permettront de poser de nombreux jalons conceptuels qui nous seront utiles par la suite.

Dans la partie *III*, nous aborderons de front la question de la vérité de la thèse de Church-Turing empirique, et de sa signification empirique. Pour ce faire, nous articulerons, dans la continuité des travaux de G. Piccinini, une conception épistémique du calcul qui s'applique à toutes les formes de calcul, et légitime ainsi la comparaison des modèles les plus divers (chapitre 5).

Dans notre ultime chapitre sur les questions de calculabilité (chapitre 6), nous procéderons à une revue raisonnée des modèles d'hypercalcul, afin de pouvoir discuter à la fois de la valeur de vérité de la thèse empirique et de sa signification empirique.

Dans notre dernière partie, nous étendrons notre champ d'investigation aux questions de complexité. Nous étudierons dans un premier temps les difficultés conceptuelles soulevées par la définition des concepts premiers de la théorie de la complexité computationnelle (chapitre 7). Nous examinerons enfin l'interprétation empirique de la complexité, les enjeux spécifiques qu'elle soulève, et ses conséquences pour notre vision de la complexité (chapitre 8).

Première partie

Les formes de la thèse de Church-Turing

Chapitre 1

La thèse de Church algorithmique

Introduction

Notre premier chapitre va être consacré à la thèse de Church-Turing algorithmique. Celle-ci correspond aux intentions théoriques historiques d'A. Church et A. Turing, et constitue le référent usuel, sinon explicite, des expressions « thèse de Church »ou « thèse de Church-Turing », dans les littératures logique et philosophique. La distinction explicite de deux formes de cette thèse, et l'addition conséquente d'un adjectif pour identifier ces formes, constituent des innovations très récentes. Celles-ci ne pourront être justifiées que durant notre deuxième chapitre, lorsque nous introduirons la deuxième forme et justifierons sa distinction d'avec la première. Le lecteur gardera donc à l'esprit que nous traiterons à l'occasion de questions historiques en employant une terminologie qui ne l'est pas, puisqu'elle est née de développements théoriques ultérieurs à la période d'énonciation de la thèse.

Dans un premier temps, nous énoncerons cette thèse algorithmique et commenterons en détail sa formulation. Nous examinerons ensuite la fonction théorique historique de cette thèse, et passerons en revue les arguments usuels en sa faveur. Enfin, nous commenterons les travaux récents de l'école de Y. Gurevich, qui prétendent fournir une démonstration de cette thèse.

1.1 Une première formulation intuitive

En première approche, la thèse de Church-Turing est un énoncé portant sur l'ensemble des fonctions calculables par un algorithme :

Thèse de Church-Turing Algorithmique. *Toute fonction calculable par un algorithme est récursive.*

Cette première formulation appelle plusieurs commentaires. Le premier est que cette thèse, de par son caractère informel, n'admet pas de formu-

lation parfaitement standard. En lieu et place des « fonctions calculables par un algorithme », on pourra tout aussi bien lire « fonctions calculables par une procédure effective », « une procédure mécanique », « une procédure combinatoire finie », « une procédure finitaire », ou tout simplement « fonctions calculables ». Il serait vain de formuler de fines distinctions entre ces différentes dénominations. Toutes sont censées exprimer le même concept intuitif. Puisque c'est aujourd'hui le terme le plus fréquemment employé dans les communautés mathématique et informatique, et qu'il a l'avantage de se prêter à la formation d'un adjectif, j'emploierai donc uniquement le terme d'« algorithme ». En outre, dans le cadre de notre analyse, il est souhaitable d'éviter de désigner l'objet de la thèse de Church-Turing par l'expression « procédure mécanique », puisqu'elle crée une ambiguïté indésirable avec la notion de calcul par une machine, dont on va verra qu'elle forme une notion distincte.

On pourra objecter que la notion de « procédure effective »est parfois considérée comme une espèce du genre « algorithme » [1]. Le terme d'algorithme peut ainsi désigner n'importe quelle procédure de calcul, tandis que le terme de procédure effective évoque des contraintes plus précises, que nous allons détailler ci-dessous. En outre, l'existence de procédures de calcul qui ne soient pas des procédures effectives, mais qui doivent néanmoins être considérées comme des procédures « réalistes », parce qu'elles permettent d'obtenir la connaissance d'un résultat, est au coeur de notre travail. Le terme d'algorithme peut ainsi être employé pour désigner ce type de procédures de calcul, et il nous arrivera de l'employer en ce sens, lorsque cet emploi est bien implanté dans l'usage [2]. Nous emploierons également l'expression « procédure de calcul », lorsque nous viserons à la plus grande généralité. Mais notre but n'est pas ici de recouvrir tous les emplois du concept d'algorithme, mais seulement de trouver un adjectif naturel pour désigner la première forme de la thèse de Church-Turing. Comme l'expression de « thèse effective »nous parait maladroite, nous nous tournerons donc vers le qualificatif d'algorithmique, et considérerons dans ce contexte qu' « algorithme »et « procédure effective »sont des expressions synonymes, comme c'est aussi souvent le cas dans l'usage. Nous emploierons à l'occasion les expressions de « calcul effectif »et de « calculabilité effective »pour désigner le champ d'application de la thèse algorithmique.

Le second problème interprétatif posé par cette thèse est l'absence de définition rigoureuse de la notion d'algorithme ou procédure effective. En l'absence d'une telle définition, la thèse de Church-Turing ne peut être mathématisée. En l'état, elle ne peut donc ni être prise comme axiome, ni être

1. On trouve par exemple une telle présentation dans ([125], 5-6) et ([40], 137).

2. Ce sera notamment le cas lorsque nous discuterons de l'algorithme de Kieu et de l'algorithme de Shor, dont nous verrons qu'ils ne respectent pas les contraintes propres aux procédures effectives.

démontrée comme un théorème [3]. C'est ce qui justifie qu'on parle d'une thèse, et non d'une proposition de Church-Turing.

Si le concept d'algorithme n'admet pas à l'heure actuelle de définition rigoureuse consensuelle, on peut énoncer à son sujet certaines conditions informelles. Avant de poursuivre notre discussion de la thèse de Church-Turing algorithmique, il nous semble désirable d'énoncer ces conditions, car elles se montreront utiles dans la suite de notre travail (voir notamment les chapitres 2 et 3).

Comme le remarque à juste titre D. Knuth [152], la signification du concept d'algorithme, dans son acception la plus vague et la plus intuitive, est similaire à celles des concepts de *routine*, *méthode* ou *recette*. Le terme évoque une suite d'instructions, permettant de résoudre systématiquement un type spécifique de problèmes. Lorsqu'on discute de la thèse de Church-Turing, la notion d'algorithme est soumise aux contraintes supplémentaires suivantes [4] :

1. *Description formelle finie.* Un algorithme est descriptible par une suite finie d'instructions de longueur finie, écrites dans le respect de la syntaxe d'un langage formel à alphabet fini, appelée *programme* [5]. Un langage formel permettant la rédaction de programmes est appelé *langage de programmation.*
2. *Entrées.* Un algorithme peut admettre des entrées. Les entrées sont des quantités appartenant à un ensemble spécifié d'objets. Elles peuvent être données avant l'exécution de l'algorithme, ou livrées de manière dynamique pendant son exécution.
3. *Uniformité.* Un algorithme définit une procédure qui s'applique uniformément à toutes les instances de la tâche considérée.
4. *Définition.* Un algorithme ne produit pas de résultat lorsqu'il est appliqué à une entrée pour laquelle la tâche qu'il exécute n'est pas définie.
5. *Sorties.* Un algorithme a une ou plusieurs sorties. Les sorties sont des quantités appartenant à un ensemble spécifié d'objets, qui entretiennent une relation mathématique déterminée aux entrées.
6. *Terminaison.* Lorsqu'un algorithme est appliqué à une entrée pour laquelle il est défini, son exécution prend un nombre fini d'étapes.
7. *Temps réel fini.* L'exécution d'un algorithme doit prendre un intervalle de temps réel fini.

3. On verra cependant qu'il existe des tentatives en ce sens (voir section 1.4).

4. Cette liste de conditions est fortement inspirée par les listes proposées par D. Knuth [152] et J. Copeland [65].

5. Les termes de « programme »et d'« algorithme »sont fréquemment employés comme des synonymes, en particulier dans la littérature informatique. Nous préférerons maintenir une distinction entre les deux concepts, pour refléter l'intuition courante, quoique très difficile à traduire en termes rigoureux, qu'un même algorithme, compris comme une entité mathématique abstraite, peut être rédigé en différents programmes dans différents langages de programmation. Une position similaire est adoptée par D. Knuth dans [152].

8. *Effectivité.* Indépendemment des contraintes de temps et d'espace mémoire, un être humain doit pouvoir exécuter un algorithme étape par étape, avec exactitude, à l'aide d'un crayon et de papier.
9. *Automatisme.* L'exécution des étapes d'un algorithme ne requiert aucune ingéniosité et aucune intelligence.

L'ensemble de ces conditions constitue une contrainte significative par rapport à l'usage, en ce sens qu'elles excluent des méthodes mathématiques qui sont couramment appelées « algorithme. »La condition de terminaison exclut les algorithmes comme l'algorithme d'Euclide pour trouver la commune mesure de deux segments, qui ne termine pas si les segments sont incommensurables. Elle exclut également les processus réactifs, qui interagissent continûment avec leur environnement [6]. Elle implique qu'un algorithme ne boucle que si la tâche qu'il exécute n'est pas définie pour l'entrée à laquelle il est appliqué. Si on la conjoint à la condition de définition, on peut en déduire immédiatement qu'un algorithme boucle si et seulement si il est appliqué à une entrée pour laquelle la tâche qu'il exécute n'est pas définie. Par exemple, on attend d'un algorithme pour la division qu'il boucle si on lui impose de diviser un nombre par zéro, et seulement dans ce cas.

La troisième difficulté d'interprétation provient de la référence à un modèle particulier de calcul. En lieu et place des « fonctions récursives », on pourrait tout aussi bien parler des « fonctions calculables par une machine de Turing »ou « fonctions Turing-calculables », des « fonctions λ-définissables », des « fonctions calculables par une machine de Post », des « fonctions calculables par une machine à registres », et de bien d'autres modèles de calcul encore. Tous ces modèles ont été démontrés équivalents, au sens où ils permettent l'expression du même ensemble de fonctions.

La thèse de Church-Turing peut donc être réinterprétée comme un énoncé portant sur le pouvoir expressif des modèles de calcul :

Thèse de Church-Turing algorithmique (modèle de calcul). *tout modèle du calcul effectif a un pouvoir expressif au mieux égal à l'ensemble des fonctions récursives.*

Une telle reformulation ne constitue pas un gain en rigueur. Elle va tout d'abord au-delà des équivalences démontrées entre modèles de calcul existants, pour affirmer l'impossibilité de concevoir un modèle de calcul doué d'un pouvoir expressif supérieur aux modèles connus. La définition du concept de « modèle de calcul », ensuite, pose autant de problèmes que celle d'algorithme, et l'on verra que son emploi est encore plus polysémique (voir notamment chapitre 4).

Le quatrième problème est la détermination du sens de la modalité dans l'adjectif « calculable ». L'interprétation usuelle est que l'objet de la thèse de Church-Turing est une calculabilité « en principe », abstraite des limitations

6. Pour plus de détails sur ce point, voir section 4.

contingentes qui peuvent rendre impossible l'exécution d'un calcul dans la pratique : est considérée comme calculable toute fonction pour laquelle il existe un algorithme permettant son évaluation. Pour mieux comprendre cette interprétation « en principe », il faut retourner à la fonction théorique originelle de la thèse de Church-Turing algorithmique.

1.2 Fonction théorique originelle de la thèse de Church-Turing, et sens des résultats d'indécidabilité

Par « fonction théorique originelle », nous entendons ici le premier objectif théorique que la formulation de la thèse de Church-Turing était censé atteindre. Une proposition théorique telle que cette thèse peut servir bien d'autres fins que celles pour laquelle elle fut initialement conçue. On peut ainsi dire que la thèse de Church-Turing fut l'une des (très) nombreuses étapes théoriques menant à la conception des machines et langages de programmation modernes. Mais ceci ne pouvait être la fin poursuivie par ceux qui l'ont énoncée, puisqu'ils étaient ignorants de cette évolution à venir.

D'un point de vue historique, la principale motivation de la thèse de Church-Turing provenait de problématiques liées aux fondements des mathématiques. Pour le comprendre, il faut remarquer le lien unissant calculabilité d'une fonction donnée et décidabilité d'une proposition donnée au sein d'un système déductif formel. Celui-ci était déjà remarqué par A. Church dans son article historique de 1936, où il affirmait l'existence de deux approches naturelles pour la définition des fonctions effectivement calculables ([56], 258) :

1. Une fonction est effectivement calculable ssi il existe un algorithme permettant de calculer ses valeurs.
2. Une fonction entière unaire F est effectivement calculable ssi pour tout $n \in \mathbb{N}$, pour tout $m \in \mathbb{N}$, la formule $F(n) = m$ est prouvable.

C'est cette perspective qui permet de relier les travaux de Church et Turing à une conjecture de D. Hilbert. Dans un ouvrage écrit en collaboration avec W. Ackermann ([11], 73-74) [7], D. Hilbert formula une conjecture portant, dans une terminologie moderne, sur la décidabilité de toute formule logique du calcul des prédicats égalitaire de premier ordre, conjecture devenue célèbre sous le nom d'*Entscheidungsproblem* (problème de la décision).

7. Nous traduisons l'extrait mentionné :

> Le problème de la décision est résolu, lorsqu'est connue une procédure permettant, pour une expression logique donnée, de décider en un nombre fini d'étapes de sa satisfiabilité ou validité. La solution du problème de la décision est d'une importance fondamentale pour la théorie de tous les domaines dont les propositions peuvent être développées sur la base d'un nombre fini d'axiomes.

L'ambition originale d'A. Church et d'A. Turing n'était donc pas tant de capturer la notion de « fonction calculable par un algorithme », mais plutôt de capturer celle de « prouvabilité dans un système formel » : il se trouve que d'un point de vue technique les deux questions sont équivalentes. Ceci leur permettait ensuite de démontrer rigoureusement la solution négative de l'*Entscheidungsproblem*, comme l'annonçaient les titres originaux de leurs articles : 'An unsolvable problem of elementary number theory' pour A. Church, et 'On computable numbers with an application to the Entscheidungsproblem' pour A. Turing.

La solution négative de l'*Entscheidungsproblem* a également des conséquences au-delà des seules problématiques de fondements des mathématiques. Dans sa célèbre conférence du congrès de Paris de 1900 [8], le mathématicien D. Hilbert a énoncé un ensemble de problèmes fondamentaux pour le siècle à venir. Parmi ses 23 problèmes, le dixième portait sur la conception d'un algorithme permettant de déterminer l'existence d'une solution rationnelle à une équation diophantienne à coefficients rationnels quelconque [9]. Ce n'est que sept décennies plus tard que Y. Matiyasevich a démontré l'indécidabilité du problème, en montrant son équivalence au problème de l'arrêt [168]. Cette démonstration historique constitue une preuve par l'exemple que les problèmes de décidabilité n'intéressent pas que les logiciens et informaticiens, et touchent aussi à des problèmes de « vraies »mathématiques, quelque soit le sens qu'on attribue à cette dernière expression.

Après cette brève mise en perspective historique, il nous faut préciser le sens de ces résultats par plusieurs remarques. La fonction originelle de la thèse de Church-Turing n'est pas de définir la notion d'algorithme, mais de déterminer l'extension du concept de « fonction calculable par un algorithme ». Une seule et même fonction peut être calculée par divers algorithmes. Un modèle de calcul peut exprimer l'ensemble des fonctions calculables, sans pour autant exprimer l'ensemble des algorithmes [10].

8. Voir [132], et sa traduction anglaise partielle dans [93].

9. Une équation diophantienne est une équation à coefficients et solutions entières, ou plus généralement à coefficients et solutions rationnelles.

10. Cette distinction est notamment bien soulignée par Y. Gurevich dans [125], 6-7 :

> There is also a rather common misunderstanding that Turing defined the notion of algorithm, albeit restricted to symbolic sequential algorithms. Let us restrict attention to such algorithms for a moment. Suppose that your computation model (e.g. a programming language) is Turing complete. Does mean that the model allows you to express all algorithms? Not necessarily. Turing machines simulate faithfully only the input-output behavior of algorithms. But there may be much more to algorithms than their input-output behavior. Turing completeness does not mean algorithmic completeness. It means only that, for every Turing computable function *f*, the language allows you to program an algorithm that computes *f*.

Pour illustrer cette idée, on peut rappeler que le λ-calcul sur les entiers de Church permet de calculer la fonction prédécesseur, mais il ne permet pas de programmer un algorithme calculant cette fonction en un nombre constant d'étapes de calcul [184].

Pour savoir si un problème est décidable, il suffit d'exhiber un algorithme idoine. Ainsi Löwenheim a pu démontrer la décidabilité du calcul des prédicats de premier ordre monadique en 1915[164], bien avant l'énoncé de la thèse de Church-Turing. Pour montrer qu'un problème n'est pas décidable, il faut en revanche donner une extension mathématique précise au concept de « fonction calculable par un algorithme », afin de montrer que le problème considéré ne peut appartenir à cet ensemble.

La décidabilité d'un problème est une question portant sur l'existence d'une méthode mathématique pour résoudre ce problème. Elle ne porte pas sur la capacité à résoudre dans la pratique toutes les instances de ce problème. La question de la calculabilité d'une fonction est ainsi distincte de celle de la possibilité pratique d'évaluer cette même fonction en tous ses arguments. Trois arguments permettent de justifier le choix d'une telle interprétation abstraite de la calculabilité :

1. *Existence de fonctions calculables à domaine infini.* Seule une telle compréhension permet de parler de calculabilité d'une fonction à domaine infini. Dans la pratique, les fonctions à domaine infini ne seront jamais évaluées en tous leurs arguments. Si l'on omettait de séparer les questions de calculabilité des questions de faisabilité dans la pratique, on devrait donc renoncer à parler de la calculabilité d'une fonction à domaine infini. Ceci aurait deux conséquences indésirables. La première serait l'exclusion de l'ensemble des fonctions calculables de candidats aussi naturels que les opérations arithmétiques élémentaires, le logarithme ou l'exponentielle, pour citer seulement quelques exemples. La seconde serait la trivialisation de la théorie de la calculabilité, puisque toutes les fonctions de domaine fini sont calculables.

2. *Existence d'entrées trop grandes pour être manipulées dans la pratique.* De nombreuses fonctions, y compris des fonctions de domaine fini, admettent des entrées de taille trop grande pour pouvoir même être écrites dans la mémoire d'un calculateur. L'addition de deux nombres de 10^{150} digits est ainsi pratiquement impossible. Si l'on prend donc en compte l'espace-mémoire nécessaire à la simple inscription d'une entrée, il existe des fonctions qui ne seront jamais, dans la pratique, évaluées en tous leurs arguments.

3. *Exigences astronomiques en ressources.* Le troisième argument consiste à prendre en compte les exigences en ressources de l'exécution du calcul, et à séparer strictement les questions de complexité des questions de calculabilité. On considère ainsi une fonction comme calculable, bien que l'algorithme optimal possède des exigences en coût impossibles à satisfaire non seulement dans notre horizon technologique présent, mais pour toute technologie envisageable. Ainsi, L. Stockmeyer et A. R. Meyer ont pu démontré que la solution d'un problème de décision simple pouvait requérir un circuit de 10^{125} portes [219]. Même

si chaque porte était de la taille d'un proton, un tel circuit ne pourrait être contenu dans l'Univers connu. S'il n'est pas évident d'articuler un argument de physique théorique prohibant rigoureusement une telle construction, il serait abusif de parler d'une « simple »impossibilité pratique, comme si un progrès technologique envisageable pouvait contourner un tel problème. Une fonction peut donc être considérée calculable, même si les exigences en ressources du meilleur algorithme rendent l'exécution de ce calcul à jamais impossible.

La définition abstraite de la calculabilité permet donc de distinguer la question mathématique de l'existence d'un algorithme, de la question empirique du calcul faisable, c'est-à-dire des calculs pouvant être réellement exécutés. Un problème peut parfaitement être qualifié de décidable, même si nombre de ses instances ne seront jamais résolues dans la pratique. La question de l'existence d'un algorithme doit être distincte de la possibilité d'exécuter cet algorithme sur toutes les entrées possibles, ce qui justifie la définition abstraite de la calculabilité.

Cette dernière remarque ne signifie pas que la question de l'existence d'un algorithme soit totalement distincte de celle de la possibilité de son exécution. On ne qualifiera pas d'algorithme une méthode qui ne pourrait être exécutée sur aucune entrée, et ne permettrait donc de connaître la solution d'aucune instance du problème considéré. Dans la suite de notre travail, on désignera cette condition sous l'appellation de *contrainte d'implémentabilité* : *un algorithme doit être implémentable.* en affirmant qu'un algorithme doit être implémentable.

Qu'on nous permette de conclure ces remarques sur le sens des résultats d'indécidabilité par une longue citation du *Perspectives in computation* de Robert Geroch ([109], 37-38) :

> Note that the theorem [on the unsolvability of the halting problem] does not assert that there is a *specific* machine T and string S such that we will be unable to decide whether that T, run on S, halts. Indeed, we except that, given (T, S), we could, given enough time and ingenuity, determine whether halting occurs. What the theorem does assert is that there is no single algorithm that will correctly decide halting in *every* case, that is, for every (T, S). Here is a more poignant restatement of these observations. Imagine having the following job: Occasionally, a Turing machine T and a string S are brought to you, and you are to determine and report to your boss whether or not that machine, applied to that string, ever halts. In some cases -for example, a machine for which q_H never appears in the third column of the table; or for will all states in the third colum, are q_H- your decision will take but a few minutes. In other cases- for example, that in which there is a collection of machine states (i) from

which the machine cannot exit, (ii) such that q_H does not appear in the third column of any of these states, and (iii) into which the machine, by virtue of the given S, will enter-it might take hours. In more complicated cases it might take days... or even years. As you continue working in this job, you will build a repertoire of arguments for settling this question in specific cases. And you will note that you are continually adding new, ever more clever, arguments to your collection. At some point, you may ask yourself, "Will this job ever become routine? Will I ever reach the point at which I have developped all the arguments that are needed to solve these puzzles-the point at which no further originality will be required for this job?" These questions are answered by the theorem just stated: The answers are all "No."

1.3 Arguments en faveur de la thèse de Church-Turing

Comme on l'a vu ci-dessus, la thèse de Church-Turing est un énoncé informel, qui ne peut en l'état ni être démontré ni être pris comme axiome. On se trouve là face à un exemple singulier dans l'histoire des sciences, où un énoncé informel s'est vu accepté de manière consensuelle par la communauté mathématique, et placé aux fondements de l'informatique. Nous allons présenter les trois arguments usuels en faveur de cette cette thèse, avant de discuter leur importance respective.

1.3.1 Absence de contre-exemple, et statut épistémique de la thèse

Il n'existe aucun exemple de fonction intuitivement calculable qui ne soit pas calculable par une machine de Turing.

Cet argument embrasse deux idées intuitives distinctes. La première est une idée de naturalité en extension de la thèse de Church-Turing : toutes les fonctions courantes que l'on considérerait intuitivement comme calculables sont bien calculables au titre de cette thèse. Ainsi, les opérations arithmétiques élémentaires, l'exponentielle, le logarithme neperien, les fonctions trigonométriques usuelles et bien d'autres encore sont calculables par une machine de Turing. La seconde est l'absence historique de découverte d'un contre-exemple sophistiqué.

Bien évidemment, l'absence de contre-exemple n'est pas une preuve de l'inexistence d'un tel contre-exemple. Le temps écoulé sans trouver de contre-exemple ne renforce guère cet argument, tant l'histoire des sciences peut se déployer sur le temps long.

La notion de contre-exemple n'est pas ici rigoureusement définie, puisqu'on n'a pas de définition rigoureuse de l'expression « fonction calculable par un algorithme ». Cet argument repose donc sur l'intuition qu'on reconnaîtra un algorithme, s'il est exhibé devant nos yeux[11].

Ce dernier point est décisif pour comprendre le statut épistémique précis de la thèse algorithmique. Sans cette croyance en notre capacité à reconnaître un algorithme donné, celle-ci pourrait être taxée de vacuité, au sens où elle serait trop mal définie pour se voir attribuer une valeur de vérité. Sur la base de cette croyance, la thèse de Church-Turing algorithmique, si elle ne peut être démontrée, peut à tout le moins être réfutée par l'exhibition d'un contre-exemple. Il est donc légitime de lui attribuer une valeur de vérité.

Puisqu'elle possède une valeur de vérité, il est inadéquat de considérer la thèse algorithmique comme une définition implicite des termes intuitifs qu'elle contient, à savoir « fonction calculable par une procédure effective. »Une définition ne possède pas de valeur de vérité, et le *definiendum* soit ne possède pas de signification préalable, soit se voit vidé des significations qui lui étaient conférées par les usages antérieurs, pour se voir donner une nouvelle signification dans le contexte de la définition. La thèse de Church-Turing algorithmique attribue au contraire une extension mathématique précise à des termes conservant leur signification intuitive antérieure.

Si elle n'est pas une définition, la thèse algorithmique ne peut être considérée comme une simple hypothèse. On pourrait certes être tenté de lui attribuer ce statut, puisque, tout comme une hypothèse empirique, elle n'est pas en l'état démontrable mais peut être réfutée par un contre-exemple. Mais une telle vision ne permettrait pas de comprendre la fonction historique de cette thèse, à savoir attribuer une extension mathématique précise à une notion intuitive. La meilleure qualification possible du statut épistémique de la thèse nous parait être celle de *reconstruction rationnelle*. Par ce terme, nous désignons une proposition théorique visant à substituer, dans un certain contexte d'usage scientifique, une expression rigoureusement définie à une expression intuitive, démunie de définition explicite, afin d'en préciser la signification et/ou l'extension. Le produit de la reconstruction rationnelle n'a pas pour fin de se substituer à tous les usages précédents de la notion intuitive. La reconstruction doit cependant permettre la substitution de l'expression rigoureusement définie à l'expression intuitive dans un certain sous-ensemble d'usages pertinents pour le contexte scientifique considéré. Pour ce faire, la notion rigoureusement définie doit comprendre dans son extension un sous-ensemble des objets de l'extension de la notion intuitive, comprenant un certain nombre de cas paradigmatiques. La notion intuitive, si elle aura

11. Je tiens à remercier Gilles Dowek pour m'avoir fait remarqué ce point. Il est également bien exprimé par G. Boolos ([41], 20) :

> Presumably, we are capable of seeing *in particular cases*, that sets of instructions are effective.

une signification relativement vague comme toute notion du langage naturel, doit cependant être suffisamment claire pour qu'on puisse en reconnaitre un exemple. Ces deux dernières propriétés garantissent qu'on puisse attribuer à une reconstruction rationnelle une valeur de vérité, puisqu'il est possible de reconnaitre un contre-exemple à la proposition.

La thèse de Church-Turing algorithmique est une reconstruction rationnelle permettant de substituer à la notion intuitive de « fonction calculable par un algorithme »la notion rigoureuse de « fonction calculable par une machine de Turing »ou toute autre expression démontrablement équivalente, dans le contexte scientifique de la logique mathématique et de l'informatique. Cette reconstruction rationnelle ne porte que sur l'extension de la notion intuitive, et non sur sa signification. Elle admet un certain ensemble de cas paradigmatiques : une notion rigoureuse de fonction calculable au titre de laquelle les opérations arithmétiques, le logarithme ou les fonctions trigonométriques usuelles ne seraient pas des fonctions calculables ne constituerait pas une reconstruction rationnelle. La notion intuitive est en outre suffisamment claire pour qu'on puisse penser reconnaitre un exemple de fonction calculable par un algorithme qui ne serait pas une fonction récursive, si un tel cas se présentait.

Doit-on considérer une reconstruction rationnelle comme une hypothèse empirique ? La question n'est pas sans difficulté. La valeur de vérité d'une reconstruction rationnelle dépend de la possible découverte d'un contre-exemple, et de la capture d'une pratique informelle préexistante. Dans le cas particulier de la thèse algorithmique, sa valeur de vérité dépend de la découverte d'un algorithme calculant une fonction non-récursive, et, comme nous allons le voir ci-dessous (voir section 1.3.3), de la modélisation de la pratique informelle du calcul papier-crayon. Ces deux traits peuvent évoquer naturellement une hypothèse empirique. Mais on remarquera également que la découverte d'un contre-exemple dépend uniquement de notre ingénuité en algorithmique. Comme nous le verrons plus loin (voir chapitre 6), une physique nouvelle peut inspirer de nouveaux modèles de calcul, qui ne sont pas basés sur la compréhension effective d'une procédure de calcul, mais elle ne peut modifier notre compréhension de ce qu'est une procédure effective, qui est ancrée dans notre pratique du calcul papier-crayon. Contrairement à une hypothèse empirique, cette reconstruction rationnelle n'est donc pas susceptible d'être réfutée suite à l'acquisition d'informations nouvelles sur le monde empirique. Or c'est précisément l'idée que de nouvelles informations sur le monde empirique, exprimées notamment au travers de théories physiques, puissent inspirer de nouvelles procédures de calcul allant au-delà des limites fixées par la forme algorithmique, qui sera l'objet central de notre travail sur la forme empirique. La question du statut épistémologique des reconstructions rationnelles mériterait assurément plus ample examen dans le cadre d'une réflexion approfondie sur les catégories philosophies de « proposition *a priori* »et de « proposition empirique ». Pour le bien de ce travail,

nous nous contenterons de déclarer que la thèse algorithmique, en tant que reconstruction rationnelle, n'est pas une proposition empirique, parce que sa valeur de vérité ne dépend pas de possibles découvertes dans les sciences de la nature.

1.3.2 Convergence des modèles

Tous les modèles computationnels ont un pouvoir expressif au plus équivalent à celui des machines de Turing.

Cet argument est parfois aussi nommé *confluence des formalismes*, ou *argument de robustesse*. De nombreux modèles de calcul ont été conçus depuis l'époque de Church et Turing, et aucun ne permet d'exprimer plus de fonctions que les modèles historiques. Cet argument est d'autant plus fort que ces divers modèles sont de conception extrêmement variée, et qu'ils n'apparaissent pas intuitivement comme de simples variantes d'un même modèle. L'invariance de l'ensemble des fonctions calculables par changement de modèle constitue une forme d'évidence théorique inductive que l'on a affaire à une notion saine ou, pour employer un anglicisme courant, robuste.

L'argument sera encore renforcé par le point suivant. Les modèles historiques de la calculabilité, si variés fussent-ils, visaient tous à modéliser des algorithmes déterministes et séquentiels. L'essor de l'informatique au cours du siècle dernier a permis un développement foisonnant de la typologie des algorithmes, et donc des modèles de calcul, qui peuvent être classés en famille partageant des propriétés communes : parallélisme, non-déterminisme, interactivité, etc. L'argument de convergence est d'autant plus puissant qu'il s'applique non seulement à divers modèles, mais à diverses familles de modèles, dont certaines ont des propriétés très éloignées des intuitions initiales d'A. Church et A. Turing. Ce renforcement de l'argument pose cependant certaines difficultés, qui ne pourront être traitées avec rigueur qu'au chapitre 2.

Cet argument peut être jugé plus convaincant que le premier, dans la mesure où il s'appuie sur une série de résultats mathématiques, et non sur une simple intuition. Cependant, rien n'exclut la possibilité d'une erreur systématique. En outre, la notion de « modèle de calcul »souffre de la même absence de définition explicite que la notion d'« algorithme ». En l'absence d'une telle définition, il est difficile de dire si l'on a bien pris en compte tous les modèles de calcul. Or, il existe des modèles de calcul putatifs permettant de calculer plus que les machines de Turing. En toute rigueur, il faudrait donc parler de « tous les modèles computationnels *raisonnables* »dans la formulation ci-dessus. Nous aurons amplement l'opportunité de revenir sur cette question [12].

On pourrait objecter que le premier argument et le second ne sont pas

12. Voir 2.1 et chapitre 5, introduction.

parfaitement distincts. La rédaction rigoureuse d'un contre-exemple à la thèse de Church-Turing supposerait la conception d'un modèle de calcul, qui serait un contre-exemple à l'argument de convergence des modèles. Si l'on impose au contre-exemple de ne pas être présenté de manière intuitive en pseudocode, et d'être rigoureusement rédigé en un programme, cet argument est exact. Il ne retire cependant rien à la nécessité de distinguer les deux arguments. Même s'il n'existait qu'un seul modèle computationnel, on pourrait formuler l'argument « aucun contre-exemple connu »à condition de l'entendre dans le premier sens que nous avons distinct ci-dessus : il suffit de montrer que ce modèle ne permet pas d'exprimer une fonction intuitivement calculable. Ce n'est évidemment pas le cas pour l'argument de convergence des modèles [13].

1.3.3 Argument de modélisation

La machine de Turing constitue une modélisation naturelle et complète d'un homme calculant en suivant un algorithme.

Cet argument réfère spécifiquement au modèle des machines de Turing, et ne s'applique pas aux autres modèles [14]. Il fut utilisé pour la première fois par A. Turing lui-même, dans la section 9 de son article de 1936 [220]. Turing essayait de montrer que son modèle du calcul constituait une modélisation naturelle et complète d'un être humain calculant en suivant un algorithme : un calcul peut être effectué par un homme suivant un algorithme ssi il peut être exécuté par une machine de Turing. Il nous semble crucial d'exposer et de commenter l'analyse de Turing en détail, car sa discussion sera d'une grande importance dans la suite de notre travail (voir sections 3.2 et 6.6.) [15].

L'argument de Turing part du simple constat que le calcul est ordinairement effectué par l'écriture de symboles sur du papier par un calculateur humain [16]. La modélisation proposée par Turing s'appuie ensuite sur un postulat fondamental : l'état du calculateur, et donc le processus du calcul, dépend exclusivement de deux facteurs, à savoir les symboles observés sur la

13. Je tiens à remercier Luiz Carlos pour avoir attiré mon attention sur ce point, lors d'une présentation orale de ce travail à la Pontificia Universidade Catolica de Rio de Janeiro.

14. Les autres modèles historiques du calcul, comme le λ-calcul ou les fonctions récursives, furent certainement inspirés par la pratique humaine du calcul papier-crayon. Les cas de base des fonctions récursives, et la β-réduction du λ-calcul sont évidemment calculables par un homme. Mais l'abstraction de ces modèles empêche de formuler un argument montrant qu'ils modélisent complètement la pratique humaine du calcul. Le modèle de Turing, parce qu'il est moins abstrait, permet au contraire la formulation d'un tel argument.

15. Nous nous référerons avec précision au texte original d'A. Turing, afin d'en permettre un commentaire rigoureux. Puisque toutes nos citations proviennent, sauf contre-indication explicite, de la section 9, nous n'indiquerons pas systématiquement la page de référence de la citation.

16. "Computing is normally done by writing symbols on paper."

feuille de papier, et « l'état d'esprit »du calculateur [17].

Deux conditions de finitude sont ensuite exprimées sur l'ensemble des symboles [18] et l'ensemble des états [19]. Turing ne remarque pas que la seconde des hypothèses est moins évidente que la première. Il est sans danger de supposer qu'un modèle du calcul papier-crayon possède une signature finie. En revanche, affirmer que l'ensemble des « états d'esprit »du calculateur est fini est plus délicat, tant la notion d'état d'esprit est plus difficile à manier. Nous reviendrons sur ce point lorsque nous examinerons les justifications que Turing donne à ces hypothèses.

Turing remarque ensuite que le caractère bidimensionnel du calcul papier-crayon, qui s'effectue bien souvent dans le plan d'une feuille, est purement contingent, et qu'on pourrait tout aussi bien effectuer n'importe quel calcul de manière linéaire [20]. On peut donc, sans perte de généralité, réduire le processus de lecture du calculateur à la lecture linéaire de cases d'une bande de papier.

La modélisation du processus de calcul suppose également son analyse en étapes élémentaires. C'est durant ces étapes élémentaires que sont modifiés les deux paramètres spécifiant l'état du calculateur, à savoir les symboles observés et « l'état d'esprit »du calculateur [21]. Pour donner un contenu précis à cette notion d'étape élémentaire de calcul, il faut énoncer des limites sur les capacités d'observation du calculateur, et son action sur les symboles, pendant l'exécution d'une étape. Au cours d'une étape, le calculateur ne peut observer qu'un nombre borné de symboles. L'existence de cette borne impose une séquentialité des opérations au calculateur : puisqu'il ne peut lire autant de symboles que désiré à la fois, il doit opérer par observations successives [22]. Si l'on admet en outre que le calculateur ne peut agir que sur les symboles observés, on obtient immédiatement une borne sur le nombre de symboles modifiables par étape de calcul [23].

17. "We know the state of the system if we know the sequence of symbols on the tape, which of these are observed by the computer (possibly with a special order), and the state of mind of the computer."

18. "I shall also suppose that the numbers of symbols which may be printed is finite."

19. "We will also suppose that the number of states of mind which need be taken into account is finite."

20. "The two-dimensional character of paper is no essential of computation. I assume then that the computation is carried out on one-dimensional paper (...)."

21. "Let us imagine the operations performed by the computer to be split up into simple 'operations' which are so elementary that it is not easy to imagine them further divided. Every such operation consists of some change of the physical system consisting of the computer and his tape."

22. "We may suppose that there is a bound B to the number of symbols or squares which the computer can observe at one moment. If he wishes to observe more, he must use successive observations."

23. "The situation in regard to the squares whose symbols may be altered in this way is the same as in regard to the observed squares. We may, therefore, without loss of generality, assume that the squares whose symbols are changed are always 'observed' squares".

Une fois admise ses limitations sur la capacité de lecture des symboles du calculateur, il devient nécessaire de modéliser le processus de calcul comme une série de déplacements le long de la bande. L'amplitude de ce déplacement est elle aussi supérieurement bornée : les cases observées durant l'étape $n+1$ ne peuvent être distantes de plus de L cases de la plus proche des cases observées pendant l'étape n [24].

Avant d'achever la présentation de l'argument de modélisation, il convient de faire quelques commentaires sur certaines hypothèses faites implicitement par Turing sans perte de généralité. À aucun moment de son argumentation Turing ne se restreint, comme c'est le cas dans la plupart des présentations scolaires contemporaines, au paradigme « scan-one-cell-move-one-cell »(lecture d'une case par étape, déplacement d'une case entre deux étapes). Le nombre de cases lues par étape de calcul, et le nombre de cases traversées entre deux étapes, sont simplement supérieurement bornés, sans que cette borne se voit attribuer une valeur explicite. Pourtant, dans sa première présentation de sa machine à la section 1 de son article, Turing adopte les restrictions usuelles [25], et ne fait aucun commentaire supplémentaire lors de son travail de modélisation. On peut supposer, sans faire preuve d'une charité excessive à l'égard du jeune Turing, qu'il ait considéré comme évident que la restriction au paradigme « lecture d'une case-déplacement d'une case »est contingente, et peut se faire sans perte de généralité.

Même avec une lecture charitable, la position de Turing sur le nombre de symboles modifiable par étape élémentaire de calcul est moins claire. Il écrit en effet simplement :

> We may suppose that in a simple operation not more than one symbol is altered.

Alors qu'il est soucieux, tout du long de son argumentation, d'indiquer que certaines hypothèses sont faites sans perte de généralité, Turing ne précise pas si cette hypothèse-ci est dans ce cas. Il est donc difficile de savoir si Turing repousse toute pertinence d'un parallélisme borné pour les questions de calculabilité, et s'il impose ainsi la séquentialité de l'action sur les symboles sans perte de généralité. Turing pouvait être poussé à faire cette hypothèse sans plus de discussion pour la simple raison que la pratique ordinaire du calcul papier-crayon par un calculateur ne permet que la modification d'un symbole à la fois.

24. "Besides these changes of symbols, the simple operations must include changes of distribution of observed squares. The new observed squares must be immediately recognisable by the computer. I think it is reasonable to suppose that they can only be squares whose distance from the closest of the immediately previously observed squares does not exceed a certain fixed amount. Let us say that each of the new observed squares is within L squares of an immediately previously observed square."

25. "At any moment there is just one square, say the r-th, bearing the symbol $S(r)$ which is 'in the machine'. We may call this square the 'scanned square'.(...)The machine may also change the square which is being scanned, but only by shifting it one place to right or left."([220], 1)

La réflexion sur l'étape élémentaire de calcul permet de définir la transition entre états du calculateur :

> The simple operations must therefore include :
>
> (a) Changes of the symbol on one of the observed squares.
>
> (b) Changes of one of the squares observed to another square within L squares of one of the previously observed squares.
>
> It may be that some of these changes necessarily involve a change of state of mind. The most general single operation must therefore be taken to be one of the following :
>
> A. A possible change (a) of symbol together with a possible change of state of mind.
>
> B. A possible change (b) of observed squares, together with a possible change of state of mind.

L'état du calculateur après l'opération est univoquement déterminé par son état avant l'opération. Dans la terminologie moderne, la transition entre états est déterministe.

Une fois le calcul papier-crayon ainsi modélisé, il devient aisé de concevoir une machine capable d'exécuter un tel calcul[26]. On constate alors rapidement qu'une telle machine est identique sur le principe aux machines que Turing avait décrites au début de son travail[27]. Le tableau récapitulatif ci-dessous permettra au lecteur de saisir de manière synoptique cette correspondance entre caractéristiques du calcul papier-crayon et caractéristiques de la machine de Turing.

Avant de commenter l'importance relative accordée à cet argument de Turing par diverses lectures, il nous faut encore faire quelques remarques, qui nous seront d'une grande utilité par la suite (voir section 6.6). La première est que Turing ne fait pas que décrire certaines caractéristiques du calcul papier-crayon : il tâche aussi d'expliquer l'origine de certaines d'entre elles. Il explique ainsi la contrainte de finitude appliquée à la signature du calcul symbolique :

> I shall also suppose that the number of symbols which may be printed is finite. If we were to allow an infinity of symbols, then there would be symbols differing to an arbitrarily small extent. The effect of this restriction of the number of symbols is not very

26. "We may now construct a machine to do the work of this computer." Le lecteur remarquera que le terme *computer*, comme il était d'usage à l'époque, désigne ici un calculateur humain.

27. "The machines just described do not differ very essentially from computing machines as defined in §2, and corresponding to any machine of this type a computing machine can be constructed to compute the same sequence, that is to say the sequence computed by the computer."

Calcul papier-crayon	**Machine de Turing**
Caractère inessentiel de l'emploi du plan	Mémoire externe linéaire
Signature finie	Signature finie
État du calculateur dépendant des symboles observés, et de son état d'esprit	État de la machine spécifié par le couple (p, q), avec p l'état du processeur, et q le symbole lu par la tête de lecture
Lecture d'un nombre borné de symboles durant l'effectuation d'une étape	Tête de lecture balayant un nombre borné de cases à la fois (s.p.g. scan-one-cell)
Déplacement d'une distance bornée entre deux étapes	Déplacement de la tête de lecture d'un nombre borné de cases entre deux étapes (s.p.g. move-one-cell)
État du calculateur après une opération dépend uniquement de l'état avant l'opération	Transition fonctionnelle entre les états (déterminisme du calcul)

TABLE 1.1 – Calcul papier-crayon et Machine de Turing

> serious. It is always possible to use sequences of symbols in the place of single symbols.

L'argument de Turing doit être complété par l'introduction d'une borne supérieure sur la taille des cases employées pour écrire des symboles. Si l'on pouvait agrandir la case d'un facteur arbitrairement grand, il serait possible d'introduire une infinité de nouveaux symboles sans que ceux-ci deviennent arbitrairement ressemblants.

L'incapacité à distinguer des symboles arbitrairement ressemblants, tout comme la borne supérieure sur la taille de la case, se justifient en termes des limites des pouvoirs perceptifs humains. Les sens humains ne disposent pas d'un pouvoir de résolution arbitraire, et ne pourraient donc distinguer des symboles arbitrairement proches. La borne supérieure sur la dimension de la case relève d'un autre facteur perceptif, à savoir la capacité à embrasser du regard un objet perçu [28].

Turing applique un raisonnement similaire à la finitude de l'espace des états :

> We will also suppose that the number of states of mind which need be taken into account is finite. The reasons for this are of the same character as those which restrict the number of symbols.

28. En outre, des symboles arbitrairement grands deviendraient excessivement difficiles à effacer et à modifier.

> If we admitted an infinity of states of mind, some of them will be 'arbitrarily close' and will be confused.

Il est ici plus difficile de reconstruire l'argument sous-jacent. Un état d'esprit n'est pas un objet extérieur au calculateur, que celui-ci percevrait par l'intermédiaire de ses sens. Il est même difficile de définir par quels moyens on distingue un état d'esprit d'un autre, et en quel sens on peut les dire arbitrairement ressemblants, comme Turing le souligne lui-même par l'emploi de guillemets de précaution. On peut cependant avancer une interprétation heuristique. La finitude de l'ensemble des états du calculateur est justifiée par la finitude de la mémoire humaine, qui ne pourrait garder trace d'une infinité actuelle d'états distincts. Cette interprétation est renforcée par une allusion elliptique au début de la section 1 de l'article. A. Turing y affirme l'importance fondamentale des limitations de la mémoire humaine dans la justification de son modèle ([220], 231, nous soulignons) :

> We have said that the computable numbers are those whose decimals are calculable by finite means. This requires rather more explicit definition. No real attempt will be made to justify the definitions given until we reach §9. *For the present I shall only say that the justification lies in the fact that the human memory is necessarily limited.*

La finitude de la mémoire humaine justifierait ainsi la finitude du nombre des états d'esprit du calculateur. On peut aussi conjecturer qu'elle est le fondement ultime de l'incapacité à distinguer une infinité de symboles, puisque des sens arbitrairement précis seraient inutiles sans une mémoire permettant de retenir l'information ainsi acquise. Il est également possible de tenter une généralisation de ces considérations aux machines. L'existence d'une machine possédant une infinité d'états est précisément l'une des possibilités que nous discuterons, lorsque nous étudierons les modèles d'hypercalcul[29].

Malgré leurs difficultés d'interprétation, les tentatives d'explication d'A. Turing ont un double intérêt. Le premier est d'expliquer certaines caractéristiques du calcul papier-crayon en termes de limitations perceptives de l'être humain. Le second est la possible généralisation de ces arguments, et d'autres arguments du même type, au cas des machines de calcul. Il peut ainsi sembler raisonnable de supposer qu'une machine ne pourra ni distinguer une infinité de symboles, ni les encoder d'une manière arbitrairement dense, ni manipuler une mémoire arbitrairement grande. Néanmoins, rien ne garantit a priori que l'inscription, la mémorisation, la lecture et la manipulation de l'information ne peuvent être réalisées de diverses manières. La traduction de considérations appliquées au calcul papier-crayon en des considérations portant sur une machine, en utilisant par exemple un vocabulaire inspiré de la physique théorique, n'est donc pas une tâche aisée. Nous aurons l'occasion

29. Voir chapitre 6.

de revenir sur ce point fondamental ultérieurement, notamment lors de notre examen de la thèse de Church-Turing physique [30].

Tous ces arguments permettent de présenter la machine de Turing comme un modèle abstrait de la pratique humaine du calcul papier-crayon. Comme cette même pratique serait la référence implicite des expressions « calcul effectué en suivant un algorithme », ou « calcul effectué en suivant une procédure effective », il devient naturel de songer qu'un bon modèle de cette pratique permettra d'exprimer l'ensemble des fonctions calculables par un algorithme. Nous reviendrons plus loin sur ce point, car il pose d'importants problèmes d'interprétation (voir section 2.2).

Importance relative des arguments

Selon les présentations, ces trois arguments se voient accordés une importance relative différente. Dans de nombreux cours de calculabilité [31], l'argument de modélisation n'est pas même évoqué. Si une telle omission n'est le plus souvent pas justifiée, on peut en deviner les raisons. L'absence de contre-exemple et la robustesse d'une notion font partie des considérations usuelles du mathématicien, tandis que la capture d'une notion intuitive l'est moins. Les arguments faisant usage de telles notions peuvent donc sembler d'une plus grande valeur aux yeux des mathématiciens.

D'autres présentations, comme celles de P. Walder [235], N. Dershowitz et Y. Gurevich [77], ou encore W. Sieg [207], accordent une importance privilégiée à cet argument. D'un point de vue historique, on peut sans doute ajouter K. Gödel à cette liste. Ce dernier aurait montré peu d'enthousiasme pour la thèse originelle de Church, avant de se déclarer convaincu par la thèse de Turing [32]. Enfin, *last but not least*, Church lui-même reconnut la force de conviction de la présentation de Turing, et ce dès sa revue de l'article de 1936 [57] :

> As a matter of fact, there is involved here the equivalence of three different notions : computability by a Turing machine, general recursiveness in the sense of Herbrand-Godel-Kleene, and λ-definability in the sense of Kleene and the present reviewer. Of these, *the first has the advantage of making the identification with effectiveness in the ordinary (not explicitly defined) sense evident immediately-i.e. without the necessity of proving preliminary theorems.*

L'argument de modélisation aurait pour avantage d'être le seul argument permettant de répondre à la question suivante : pourquoi devrait-on considérer que la thèse de Church-Turing permet de capturer l'ensemble des

30. Voir section 6.1.

31. Par exemple, deux manuels de référence comme [63] et [41] présentent la thèse de Church-Turing sans mentionner l'argument de modélisation.

32. Pour plus de détails sur cette question historique, voir [72].

fonctions calculables par un algorithme ? Pour répondre à cette question, un recours à l'intuition semble nécessaire, et c'est ce que réalise l'argument de Turing.

Cet argument ne doit en aucun cas être confondu avec un argument affirmant que le modèle des machines de Turing serait plus « naturel ».Ce dernier qualificatif peut lui-même s'interpréter de deux manières différentes : les machines de Turing serait un langage de programmation privilégié, ou seraient à tout le moins le meilleur modèle pour poser des questions de calculabilité. Il est bien connu de tout programmeur et de tout étudiant en logique que la première interprétation est évidemment fausse : il est au contraire extrêmement pénible de programmer une machine de Turing. Pour la seconde, l'investigation théorique des modèles de la calculabilité ne se restreint pas aux machines de Turing. Mais comme le remarque explicitement A. Turing dans son article démontrant l'équivalence des machines de Turing et de la λ-définissabilité [221], c'est précisément la conjonction des arguments de modélisation et de convergence qui permet d'employer sans arrière-pensée les autres modèles de la calculabilité, qui ne jouissent pas d'un argument de modélisation en leur faveur. S'il n'est pas d'emblée évident que les fonctions λ-définissables capturent l'ensemble des fonctions calculables par un algorithme, il devient plus naturel de le penser via l'équivalence avec le modèle des machines de Turing. Si l'on considère par exemple le λ-calcul comme un meilleur modèle théorique[33], il devient ainsi possible de l'employer sans craindre de perdre contact avec la notion de procédure effective.

Permettre la formulation de l'argument de modélisation n'est pas le seul mérite du modèle de Turing. Le modèle de Turing joue notamment un grand rôle en théorie de la complexité, puisque c'est au sein de ce modèle qu'on a pu définir de manière naturelle la notion de pas de calcul. Le modèle des fonctions récursives et du λ-calcul ne permettent pas en l'état une telle définition. Nous aurons amplement l'occasion de revenir sur ce point, lorsque nous traiterons plus en détail des questions de complexité (voir partie *IV*, notamment la section 8.2).

1.4 Une démonstration de la thèse de Church-Turing ?

Durant ces trois dernières décennies, un groupe de recherche centré autour de Y. Gurevich, N. Dershowitz et U. Boker a prétendu produire une démonstration de la thèse de Church-Turing algorithmique. Cette démonstration induit-elle une modification du statut épistémique de la thèse algorithmique, en faisant d'elle un véritable théorème ? Quels nouveaux éclairages nous apporte-t-elle sur la signification de la thèse ? Pour répondre à ces questions, nous allons présenter succinctement l'approche de ce groupe, avant de

33. C'est d'ailleurs la position adoptée par Turing lui-même, sans qu'il précise les raisons pour lesquelles le λ-calcul serait un modèle préférable.

discuter la signification exacte des résultats techniques obtenus.

Cette approche part d'un travail d'axiomatisation de la notion d'algorithme séquentiel déterministe. Comme nous l'avons mentionné plus haut (voir section 1.1), la thèse de Church-Turing algorithmique ne constitue nullement une définition de la notion d'algorithme, même restreinte à la famille des algorithmes séquentiels et déterministes. Dans [77], N. Dershowitz et Y.Gurevich ont proposé d'axiomatiser la notion d'algorithme séquentiel déterministe par quatre propositions, qui sont présentées informellement de la manière suivante :

1. *Un algorithme détermine une suite d'états computationnels pour chaque entrée valide.*
2. *Les états computationnels sont des structures invariantes par isomorphisme.*
3. *Les transitions entre états sont régies par une description finie fixée.*
4. *Seules des opérations indéniablement calculables sont admises au sein des états initiaux.*

La première condition capture l'intuition élémentaire d'un processus séquentiel et déterministe. La deuxième condition énonce une contrainte sur la forme de représentation mathématique des états computationnels (pour plus de commentaires sur cette idée, voir section 6.6). La troisième capture, pour le cas particulier des algorithmes séquentiels déterministes, l'intuition préthéorique que nous avons appelée *description formelle finie.* La quatrième, enfin, sert à garantir que l'algorithme ne soit pas muni d'emblée d'oracles[34] résolvant des problèmes indécidables.

N. Dershowitz et Y. Gurevich défendent ensuite que tout algorithme séquentiel déterministe, défini comme un objet satisfaisant ces postulats, peut être simulé pas à pas par une *Abstract State Machine*, un modèle de calcul dû à Y. Gurevich [124], [36]. Comme ils démontrent également qu'une A.S.M. qui n'est pas d'emblée munie d'oracles non-récursifs ne peut calculer qu'une fonction récursive, ces résultats impliquent que la thèse de Church-Turing est respectée par tout algorithme séquentiel déterministe. Y. Gurevich et consorts ont ensuite travaillé à étendre leur approche à d'autres familles d'algorithmes, comme les algorithmes parallèles ou interactifs (voir [32], et références mentionnées).

Notre objectif présent n'est pas de discuter de la justesse conceptuelle des axiomes proposés par Gurevich et consorts, ni de discuter le détail de

34. On appelle en informatique « oracle »un dispositif exécutant une tâche donnée, sans que son fonctionnement interne ne soit spécifié. La notion d'oracle fut introduite par A. Turing dans sa thèse [222], pour pouvoir notamment étudier l'ensemble des fonctions qui deviendraient calculables si un problème indécidable donné était calculable : on parle alors de calculabilité relative. Un oracle est donc un moyen de calcul hypothétique, qui peut permettre le calcul d'une fonction non-récursive, sans que l'on puisse préciser la procédure permettant l'exécution de ce calcul. Nous parlerons dans ce cas de figure d'« oracle non-récursif. »

leur formalisation et de la preuve formulée sur cette base. Nous chercherons plutôt à répondre à la question suivante : si l'on admet la justesse de l'axiomatisation et de la preuve proposées, en quoi affectent-t-elle le statut épistémique de la thèse de Church-Turing algorithmique ?

L'idée de « démonstration »évoque naturellement un changement radical du statut épistémique d'une proposition. La thèse de Church-Turing algorithmique est susceptible d'être réfutée par un contre-exemple. L'adhésion à cette proposition, en tant que reconstruction rationnelle, comprend une part irréductible d'intuition. Si elle était démontrée, la thèse deviendrait un théorème. Elle ne pourrait plus être réfutée par la découverte d'un contre-exemple, et l'adhésion à cette proposition ne supposerait pas plus de recours à l'intuition que n'importe quelle proposition mathématique.

Une telle modification de statut épistémique n'est pas offerte par la démonstration de Y. Gurevich et N. Dershowitz. Celle-ci ne part pas d'une définition de la notion d'algorithme dans le langage de ZFC, qui permettrait de la considérer comme un objet mathématique ordinaire, dont les propriétés seraient démontrées à partir des axiomes de la théorie des ensembles. Cette démonstration ne peut donc être considérée comme une démonstration mathématique ordinaire, et elle ne confère pas le statut épistémique usuellement attribué à un théorème mathématique. Elle ne garantit pas contre l'existence d'un contre-exemple, et ne supprime pas la nécessité de faire recours à l'intuition pour adhérer à la thèse. Pour formuler cette démonstration, il est en effet nécessaire d'accepter une axiomatisation qui ne fait pas partie des postulats mathématiques usuels, et qui peut elle-même faire objet de débat. L'adhésion aux quatre axiomes permettant la formalisation de la notion d'algorithme séquentiel déterministe doit donc être considérée comme une nouvelle thèse, la thèse de Gurevich [35].

Nos dernières remarques ne doivent pas être interprétées comme une critique du travail de Y. Gurevich et consorts, mais plutôt comme une mise en garde contre sa possible mésinterprétation. Y. Gurevich se défend d'ailleurs très explicitement de prétendre donner une définition formelle définitive de la notion d'algorithme ([125], 4) :

> In our opinion, the notion of algorithm cannot be rigorously defined in full generality, at least for the time being. The reason is that the notion is expanding.
>
> Concerning the analogy of algorithms to real numbers, mentioned in §1, Andreas Blass suggested a better analogy : algorithms to numbers. Many kinds of numbers have been introduced throughout history : positive integers, natural numbers, rationals, reals, complex numbers, quaternions, infinite cardinals, infinite ordinals, etc. Similarly many kinds of algorithms have

35. Y. Gurevich présente lui-même son travail de formalisation comme une thèse [123],[125] et a intitulé ainsi un de ses articles sur le sujet [121].

> been introduced. In addition to classical sequential algorithms, in use from antiquity, we have now parallel, interactive, distributed, real-time, analog, hybrid, quantum, etc. algorithms. New kinds of numbers and algorithms may be introduced. The notions of numbers and algorithms have not crystallized (and maybe never will) to support rigorous definitions.

Lorsqu'ils proposent leurs axiomes, qu'ils prennent d'ailleurs grand soin de justifier et de discuter (voir [77], 307-321), Y. Gurevich et N. Dershowitz ne cherchent donc pas à capturer le *genus* « algorithme », avant de se restreindre aux *differentiae proximae* « déterministe »et « séquentiel ». Ils tendent uniquement à capturer la famille d'algorithmes déterministes et séquentiels. Les tentatives de généralisation postérieure doivent être lues comme des efforts pour capturer d'autres familles d'algorithmes, et non comme une tentative pour parvenir à une absolue généralité.

Pour reprendre une expression heureuse due à S. Grigorieff [36], on peut dire que la thèse de Gurevich implique la thèse de Church-Turing algorithmique. Mais on ne peut dire que le travail de Y.Gurevich et N. Dershowitz dispense de l'emploi de thèses. La thèse de Gurevich a un statut épistémologique similaire à celui de la thèse de Church-Turing. Elle peut être soutenue par un argument analogue à celui de la convergence des modèles, puisque tous les modèles séquentiels usuels sont démontrablement simulables pas à pas par une A.S.M..

Plutôt qu'un changement de statut épistémique, on peut dire que la démonstration de Y.Gurevich et N. Dershowitz constitue une tentative d'approfondissement de l'effort de reconstruction rationnelle qui est le propre de la thèse. Cet approfondissement passe par un travail de définition de la notion d'algorithme séquentiel, et par un étude de la simulation pas à pas des algorithmes. Le terme de « caractérisation »nous parait plus approprié que celui de démonstration pour désigner un tel travail [37].

Il ne prête pas aux confusions que nous venons de dissiper, et il permet de souligner le principal mérite d'un tel travail, à savoir celui de transformer une discussion de la thèse de Church-Turing en une discussion des quatre axiomes. Y. Gurevich et N. Dershowitz n'affirment d'ailleurs pas autre chose [38].

---

36. Voir son exposé [117], pour une excellente introduction aux problématiques étudiées par Y. Gurevich.

37. Nous retrouverons cette notion de caractérisation de la thèse de Church-Turing lorsque nous discuterons de la thèse physique (voir section 6.6).

38. ([77], 339) :

> Our axiomatization provides a small number of principles that imply Church's Thesis, and focuses attention on the axioms. Thus, to the extent that one might entertain the notion that there exist non-recursive effective functions, one must reject one or more of these postulates. The need for "continual verification" of the legitimacy of Church's identification of recursiveness with effectivity, to which Post referred, can now center around the universality of

Leur travail comprend en outre de nombreux aspects dignes d'attention philosophique, notamment la définition d'une calculabilité relative à la calculabilité d'un ensemble d'oracles ou fonctions initiales. Mais puisque nous avons vu qu'il ne modifie pas fondamentalement le statut épistémique de la thèse algorithmique, nous ne nous attarderons pas davantage sur ce sujet dans le cadre de cette thèse.

Conclusion

La thèse de Church-Turing algorithmique est une reconstruction rationnelle de la notion de « fonction calculable par une procédure effective. »Elle n'offre pas une définition de la notion de procédure effective ou d'algorithme, mais elle permet d'en fixer l'extension mathématique. Elle permet ainsi la démonstration de résultats d'indécidabilité, qui fut la fonction historique de son énoncé.

La thèse algorithmique doit être comprise comme une authentique proposition, douée d'une valeur de vérité, et non comme une définition implicite déguisée. Sa valeur de vérité dépend de notre ingénuité en algorithmique, et de notre expérience du calcul papier-crayon : aucune découverte dans les sciences de la nature ne pourrait l'affecter. Il serait donc tout aussi erroné de considérer cette thèse comme une hypothèse empirique.

La thèse de Church-Turing algorithmique est appuyée par plusieurs arguments, qui justifient son adoption sans garantir sa vérité. Parmi ses différents arguments, nous nous sommes particulièrement intéressés à ce que nous avons appelé l'argument de modélisation d'A. Turing. Celui-ci justifie la thèse algorithmique en montrant que la machine de Turing constitue un modèle satisfaisant du calcul papier-crayon. Nous avons vu qu'A. Turing cherchait à fonder les limites de son modèle dans des limitations perceptuelles et cognitives du sujet humain. Ces limitations ne font cependant pas de la thèse algorithmique une thèse de psychologie empirique, comme nous allons le voir au chapitre suivant. Elles semblent plutôt indiquer la voie d'un possible élargissement de cette fondation en des hypothèses physiques, même si leur formulation précise promet d'être difficile (voir section 6.6).

---

the individual axioms.

La même idée est répétée avec éloquence dans un article signé par Y. Gurevich seul ([125], 5) :

> None of the [...] known kinds of algorithms seem to threaten the thesis but the thesis has not been dispensed with and probably never will be. The question whether some algorithm of a currently unknown kind would allow us to compute a function from natural numbers to natural numbers that is not Turing computable remains open, possibly forever. And even if we had a satisfactory axiomatic definition of algorithms in full generality, the thesis would not be dispensed with. It would be just reduced to the first principles enshrined in the axioms.

La « démonstration de la thèse de Church-Turing »offerte par Y. Gurevich et consorts doit être interprétée avec précaution. Les axiomes proposés par Y. Gurevich et N. Dershowitz ne visent pas à capturer la notion d'algorithme en toute généralité, mais seulement la famille des algorithmes séquentiels et déterministes. Leur travail ne constitue donc pas une démonstration au sens où elle ne constitue pas un changement de statut épistémique de la thèse de Church-Turing, qui permettrait de se dispenser de thèses dans l'expression des limites de la calculabilité. Elle doit plutôt être vue comme un effort de caractérisation, qui se fonde sur une tentative de formalisation d'une sous-classe des algorithmes. Ces remarques ne constituent en aucun cas une critique du travail fourni par Y. Gurevich et consorts, dont bien des aspects mériteraient l'attention des philosophes.

Chapitre 2

La thèse de Church empirique

Introduction

Nous allons à présent introduire la forme empirique de la thèse de Church-Turing. Dans un premier temps, tout comme pour la forme algorithmique, nous énoncerons cette thèse et discuterons sa formulation. Nous donnerons également les premiers arguments justifiant sa distinction d'avec la forme algorithmique, et dénoncerons certaines sources de confusion, à la fois sémantiques et historiques. Nous discuterons dans un deuxième temps une troisième forme de la thèse de Church-Turing, dite forme psychologique, et nous défendrons qu'elle ne constitue pas une véritable proposition indépendante des deux premières formes. Dans un troisième et dernier temps, nous discuterons en détail certaines objections, que nous qualifierons de *scepticisme méthodologique*, qui tendent à nier que la thèse empirique soit une véritable proposition empirique.

2.1 Distinction entre deux formes de la thèse de Church-Turing

La thèse de Church-Turing algorithmique est parfois présentée comme une proposition portant sur les machines de calcul. On peut ainsi lire que la thèse de Church-Turing porterait sur les limites des calculs effectuables par un ordinateur [1]. Une telle présentation repose sur une confusion. En énumérant les raisons de ce dernier jugement, nous allons justifier du même coup la distinction de la forme algorithmique de la thèse de Church-Turing d'avec une autre forme de cette thèse, la forme empirique.

Avant d'entamer cette discussion, il nous faut préciser ce que nous entendons par « machine de calcul ». Il n'est pas possible de donner une définition scientifique générale de ce concept. Les différentes machines conçues dans la

1. Pour un florilège de citations d'écrits commettant une telle confusion, voir [67], 11-13.

recherche contemporaine relèvent de disciplines scientifiques hétérogènes et peuvent être décrites en termes physiques, chimiques ou encore biologiques[2]. Face à une telle diversité de stratégies d'implémentation, il est malaisé de formuler une terminologie standard précise, permettant d'embrasser dans son intégralité cette série ouverte de développements scientifiques. Nous nous contenterons d'une définition intuitive générale, dont la portée sera ensuite précisée par quelques *caveats*.

En première approche, on désigne ici par « machine de calcul », « machine implémentant une fonction », ou simplement « machine »tout dispositif réalisant le calcul à la place d'un homme. Compte donc comme machine tout dispositif dans lequel on peut encoder l'information représentée par les arguments, laisser traiter cette information par une série de transformations de l'état du dispositif, et récupérer l'information par une opération de lecture, tel que l'ensemble de ces opérations commute avec l'application de la fonction à son argument. Pour que l'appellation de « machine »soit légitime, l'homme doit au plus intervenir dans le traitement de l'information au moment de l'encodage de la donnée et de la lecture du résultat. L'observateur humain peut parfaitement intervenir pour permettre l'exécution des transformations du dispositif, tant qu'il n'a pas accès à l'information encodée, et qu'il ne peut donc sciemment en encoder une nouvelle ou en effacer. C'est le cas, par exemple, du calcul avec brins d'ADN (voir section 8.3), où l'opérateur humain doit intervenir dans le processus de transformation de l'information en opérant des manipulations chimiques, ou de modèles mécaniques rudimentaires, où l'être humain doit fournir de l'énergie au système par l'action d'une manivelle, d'un levier ou d'un poussoir. Notre emploi du terme « machine de calcul »correspond donc à l'usage des historiens du calcul[3], par opposition aux « instruments de calcul », comme le boulier, qui demande une lecture de l'information par le sujet humain.

Les machines numériques programmables, qui font aujourd'hui partie de notre quotidien, sont bien sûr des machines au titre de notre définition. Mais cette dernière embrasse des cas très différents de ces machines. Une machine est n'importe quel dispositif permettant d'évaluer au moins une fonction. Elle n'est pas nécessairement une implémentation d'une machine de Turing universelle, et elle n'est pas nécessairement programmable à l'aide d'un langage de programmation ordinaire.

À présent que nous avons précisé notre compréhension du concept de « machine de calcul », nous pouvons expliquer pourquoi la thèse de Church-Turing algorithmique ne porte pas sur les limites de telles machines. Le premier argument en faveur d'une telle position est d'ordre sémantique. Les notions d'algorithme et de machine de calcul étant sémantiquement distinctes, aucune raison immédiate ne permet de transformer les considérations por-

2. Voir chapitres 6 et 8, pour plus de détails.
3. Voir [167].

tant sur les limites des algorithmes en considération sur les capacités des machines. Pour citer le physicien David Deutsch ([78], 4) :

> (. . .) There is no a priori reason why physical laws should respect the limitations of the mathematical processes we call 'algorithms'(...).

Cette remarque suffit à justifier la distinction entre deux propositions, l'une portant sur les limites des algorithmes, l'autre portant sur les limites des machines.

Thèse de Church-Turing empirique. *Toute fonction calculable par une machine est récursive.*

Avant de procéder à l'examen de cette thèse, nous souhaiterions préciser l'origine des confusions si fréquentes entre les deux notions, et donner les raisons intuitives pour lesquelles il n'est pas évident de montrer que l'ensemble des fonctions calculables par un algorithme et l'ensemble des fonctions calculables par une machine sont identiques.

La première cause de cette confusion est vraisemblablement l'ambiguïté entourant les termes de « machine »et « procédure mécanique ». Cette ambiguïté est particulièrement forte dans le cas de l'expression « machine de Turing ». Celle-ci prête à confondre trois idées distinctes :

1. *La* machine de Turing, qui désigne le modèle mathématique du calcul proposé par A. Turing.
2. *Une* machine de Turing, qui désigne un programme, ou table d'instructions, écrit dans le langage de programmation défini par ce modèle.
3. *L'implémentation* d'une machine de Turing, qui désigne un dispositif concret, homme, instrument ou machine, permettant l'exécution de tels programmes, ou les programmes d'un autre langage de programmation universel. On dit ainsi fréquemment que nos ordinateurs contemporains sont des machines de Turing universelles, alors qu'on devrait dire plus rigoureusement qu'ils sont des implémentations d'une machine de Turing universelle.

À cause de cette ambiguïté terminologique, il est aisé de passer d'un propos portant sur les limites des machines de Turing, c'est-à-dire sur les limites des programmes, à un propos portant sur les limites des machines.

Le terme de « procédure mécanique »prête également à confusion. Dans certains contextes, il sera entendu comme un synonyme strict de « procédure effective »ou d' « algorithme ». Il permet de souligner l'absence d'intuition, d'ingéniosité et de créativité qui est censée faire partie de nos intuitions préthéoriques sur le concept de « procédure effective ». Dans d'autres contextes, il évoquera une procédure exécutée par un dispositif mécanique, sans intervention de l'homme.

Cette ambiguïté terminologique est renforcée par un point conceptuel délicat. Puisqu'un algorithme est censé être exécutable sans intuition, ingéniosité, ou créativité, la possibilité d'automatiser l'exécution peut être employée comme un test permettant de reconnaitre qu'un programme est bien rédigé, i.e. qu'il ne fait aucune référence implicite à l'intuition. On peut désigner cette dernière condition par le nom de *critère d'automatisabilité* : *un programme doit pouvoir être exécuté par une machine de calcul.* Ce critère s'applique en premier lieu aux programmes, mais puisque nous avons vu que tout algorithme doit pouvoir être rédigé sous forme d'un programme, il s'applique également aux algorithmes [4]. Puisqu'un algorithme est une procédure qui doit pouvoir être exécuté par une machine, il est tentant de penser que la thèse de Church-Turing porte sur les limitations des machines. Mais ce n'est pas parce que la thèse de Church-Turing porte sur les limitations de procédures de calcul qui sont automatisables qu'elle énonce les limites de toute procédure automatisable. Il reste à montrer que tout calcul exécutable par une machine est simulable par une machine de Turing.

Une bonne compréhension du critère d'automatisabilité est nécessaire pour éclaircir certaines sources historiques de confusion entre thèse algorithmique et thèse empirique. Certains propos d'A. Church et A. Turing peuvent en effet laisser accroire qu'ils avaient pour objectif d'énoncer une thèse sur les limites des machines de calcul. On trouve une telle interprétation sous la plume de M. Davis [72] :

> (. . .) there is no doubt that, from the beginning the logicians developing the theoretical foundations of computing were thinking also in terms of physical mechanism. Thus, as early as 1937, Alonzo Church reviewing Turing's classic paper wrote [4] :
>
> > [Turing] proposes as a criterion that an infinite sequence of digits 0 and 1 be 'computable' that it shall be possible to devise a computing machine, occupying a finite space and with working parts of finite size, which will write down the sequence to any desired number of terms if allowed to run for a sufficiently long time. As a matter of convenience, certain further restrictions are imposed on the character of the machine, but these are of such a nature as obviously to cause no loss of generality . . .
>
> Turing himself speaking to the London Mathematical Society in 1947 said [24] :

4. Ce point a déjà été souligné par D. Knuth ([152], 6) :

   (. . .) an algorithm must be specified to such a degree that even a computer can follow the directions.

Dans le contexte de ce passage, le terme de *computer* désigne manifestement une machine de calcul, et non un calculateur humain.

> Some years ago I was researching what now may be described as an investigation of the theoretical possibilities and limitations of digital computing machines. I considered a type of machine which had a central mechanism, and an infinite memory which was contained on an infinite tape. This type of machine seemed to be sufficiently general. One of my conclusions was that the idea of 'rule of thumb' process and 'machine process' were synonymous.

Referring to the machine he had designed for the British National Physics Laboratory, Turing went on to say :

> Machines such as the ACE (Automatic Computing Engine) may be regarded as practical versions of this same type of machine.

A. Hodges adopte une position similaire ([136], 245). A. Hodges s'est en effet engagé dans une controverse de nature historique avec J. Copeland. Ce dernier défend une vision similaire à celle que nous avons exposée ici, à savoir la nécessité philosophique de distinguer forme algorithmique et forme empirique de la thèse de Church-Turing, jointe à l'affirmation historique que seule la première forme était l'objet des travaux d'A. Church et A. Turing ([136], 250) [5].

En plus de citer l'extrait de la revue de l'article d'A. Turing mentionné par M. Davis, il cite contre cette vision un extrait de la revue par le même A. Church du travail d'E. Post. A. Church reproche à ce dernier de présenter la thèse algorithmique comme une simple « hypothèse de travail »(*working hypothesis*), alors que l'analyse de Turing permet de la justifier comme une définition, si l'on comprend par là un travail de capture d'une notion informelle [58] :

> To define effectiveness as computability by an arbitrary machine, subject to restrictions of finiteness, would seem to be an adequate representation of the ordinary notion, and if this is done the need for a working hypothesis disappears.

Sur la base de ces extraits, A. Hodges conclut qu'A. Church et A. Turing avaient d'emblée l'ambition de capturer la notion de « fonction calculable

5.

> Copeland's contention is specifically concerned with the meaning of what was formulated in 1936. He holds both that the Church-Turing thesis is true, and that physical machines may be capable of computing uncomputable functions. The only way to reconcile these statements is to assert that Church and Turing positively held in 1936 that their concept of effective calculation did not refer to machines. The historical record does not support this contention.

par une machine », et ne se limitaient aucunement au calcul effectuable par un homme exécutant des instructions :

> It appears that Church and Turing (and others, like Gödel and Newman) used the word 'machine' quite freely as a synonym for 'mechanical process', without clearly distinguishing the model of a mechanical process given by the human rule-follower.

A. Hodges affirme ensuite explicitement que dans la mesure où on peut établir leur point de vue, Church et Turing avaient un point de vue mécaniste sur le calcul effectif. Citant une formulation par A. Yao de la forme physique [244], A. Hodges affirme sans ambiguïté que les intentions historiques de Turing et Church, pour autant qu'on puisse les cerner, étaient clairement d'énoncer une proposition similaire :

> Yao regards the thesis not as a dogma but as a claim about physical laws which may or may not be true. Yao's careful words about what Church and Turing believed are fair : we cannot know quite what they thought, but the evidence points to a standpoint closer in spirit to Yao's than to Copeland's.

Pour plus de clarté, il nous semble nécessaire de séparer dans notre réponse le cas de Turing de celui de Church.

L'interprétation de la position d'A. Church est la plus délicate. En plus de la citation déjà mentionnée, A. Church reprend essentiellement la même position dans un article de 1940[59], note 15, 133-134) [6]. A. Church parle bien, dans sa critique d'E. Post, de machine « arbitraire »(*arbitrary machine*), ce qui semble bien indiquer l'intention de définir la calculabilité par n'importe quel type de machine. Cependant, une lecture plus attentive du contexte indique bien que le sens de ce qui deviendra la thèse de Church-Turing est la reconstruction rationnelle de la notion d'effectivité. Il parle en effet de « représentation adéquate de la notion ordinaire »(*adequate representation of the ordinary notion*). Ainsi, dans sa revue d'A. Turing ([57], 43) :

> It is thus immediately clear that computability, so defined, can be identified with (especially, is no less general than) the notion of effectiveness as it appears in certain mathematical problems (various forms of the Entscheidungsproblem, various problems to

6.

> Turing proves the equivalence of λ-definability and general recursiveness to a notion of computability whose definition, briefly stated, is as follows : A function ϕ is computable if it is possible to make a computing machine, with a finite number of parts of finite size, which will calculate $\phi(n)$ for any assigned n, printing intermediate calculations and the final result on a tape with which the machine must be supplied (no upper limit is placed on the time or on the length of tape required for a particular calculation). Actually, Turing imposes several further conditions on the computing machine, but these are more or less clearly nonessential.

> find complete sets of invariants in topology, group theory, etc., and in general any problem which concerns the discovery of an algorithm).

On pourrait alors attribuer à A. Church la vision selon laquelle « fonction calculable par une machine quelconque »serait une bonne compréhension de la notion de « fonction calculable par une procédure effective. »Mais il est possible de lui attribuer une position différente. L'emploi du terme de « critère »(*criterion*) suggère une position similaire à celle que nous venons d'exposer, selon laquelle l'automatisabilité est un critère pour reconnaitre une authentique procédure effective. Si l'emploi du terme « arbitraire »est certes propre à semer le trouble, il nous semble que cette dernière interprétation est plus cohérente au vue de l'ensemble du texte, et devrait donc être adoptée par charité intellectuelle.

L'analyse de la position d'A. Turing par A. Hodges, quant à elle, est malheureusement fondée sur une sélection très restreinte des citations. Il existe de nombreuses autres citations montrant l'attention accordée au pouvoir expressif du calculateur humain.

Parmi les pères de la calculabilité, A. Church, J. Herbrand, K. Gödel, A. Turing était le seul à posséder une formation d'ingénieur, et à avoir été impliqué dans la construction de machines de calcul numériques. Il avait donc une conscience aiguë de l'existence de modèles de calculateurs, comme les calculateurs analogiques, dont la nature est très différente de celle du calcul papier-crayon : cette conscience est bien illustrée par une conférence de 1947 où il présente l'*Automatic Computing Engine (ACE)*, un modèle de calculateur numérique sur lequel il a travaillé. Il commence ainsi sa conférence par une comparaison des performances des machines numériques et des machines analogiques ([228], 1). Juste après cette comparaison, il fait l'affirmation suivante sur le pouvoir expressif de son modèle (*idem*) :

> A good working rule is that the ACE can be made to do any job that could be done by a human computer (...).

Le calculateur numérique sur lequel a travaillé A. Turing est donc explicitement présenté par ce dernier comme une automatisation du calcul papier-crayon effectué par les êtres humains. A. Turing assimile ensuite explicitement son travail sur les limites de la calculabilité à un travail sur les limites des calculateurs digitaux (*idem*, nous soulignons) :

> Some years ago I was researching on what might now be described as an investigation of the *theoretical possibilities and limitations of digital computing machines*. (...) One of my conclusions is was that the idea of 'rule of thumb' process and 'machine process' were synonymous. The expression 'machine process' of course means one which could be carried out by the type of machine I was considering.

La traduction de l'expression anglaise *rule of thumb* est délicate, et décisive pour notre propos. Elle désigne le plus fréquemment une règle dont la pratique montre qu'elle aboutit au résultat désiré la plupart du temps, mais ne doit pas être prise comme une règle absolue. On la traduit donc bien souvent, selon les cas, par les expressions françaises *règle d'or, règle générale, règle empirique* ou encore *règle approximative*, toutes traductions manifestement inadaptées dans ce cas. En sachant que l'expression est également employée pour désigner les règles de calcul simplifié d'approximations qu'apprennent les étudiants, on devine que l'expression désigne ici un processus aveugle d'exécution d'instructions. Le *'rule of thumb' process* ne serait donc rien d'autre qu'une expression équivalente, dans la terminologie de Turing, à celle de procédure effective [7].

A. Turing dit donc ici qu'une des intuitions ayant mené à son modèle de 1936 est que le calcul effectué par un homme suivant aveuglément des instructions est la même chose que le calcul exécuté par un certain type de machines, les machines numériques. La formulation employée ne laisse guère de doute sur la restriction de son propos à un certain type de machines de calcul (*...the type of machines I was considering.*). En affirmant que les expressions « processus aveugle d'exécution d'instructions »et « processus mécanique »sont synonymes, A. Turing reformule l'argument de modélisation : un calcul est exécutable par une procédure effective ssi il est exécutable par une machine numérique, dûment modélisée par la machine de Turing.

Il ne s'agit aucunement ici d'une occurrence isolée, qui pourrait être mis en balance avec les citations prises par A. Hodges. Après avoir lu toutes les références mentionnées par A. Hodges, nous n'avons trouvé qu'un seul autre passage soutenant son interprétation [8], tandis que nous en avons trouvé nombre d'autres corroborant celui que nous venons de citer, dont acte ([223] in [68], 444) :

> The idea behind digital computers may be explained by saying

7. C'est aussi l'interprétation adoptée par J. Copeland dans ([68], 480).
8. On trouve ainsi dans la thèse d'A. Turing l'extrait suivant ([222] in [68], 150) :

> A function is said to be "effectively calculable" if its values can be found by some purely mechanical process. Although it is fairly easy to get an intuitive grasp of this idea, it is nevertheless desirable to have some more definite, mathematically expressible definition.(...) The author has recently suggested a definition corresponding more closely to the intuitive idea (Turing [1], see also Post [1]). It was stated above that "a function is effectively calculable if its values can be found by some purely mechanical process". We may take this statement literally, understanding by a purely mechanical process one which could be carried out by a machine.

Là encore, A. Turing parle de « machine »et de « processus purement mécanique »sans préciser aucunement s'il ne pense qu'à un type restreint de machines, ou s'il compte embrasser tout type de machines possibles. À la lumière du nombre de citations explicites que nous avons mentionnées, l'interprétation la plus probable est qu'il s'agit d'une simple omission.

> that these machines are intended to carry out any operations which could be done by a human computer. The human computer is supposed to be following fixed rules ; he has no authority to deviate from them in any detail.

On peut encore rajouter (*idem*, 445) :

> The reader must accept it as a fact that digital computers can be constructed, and indeed have been constructed, according to the principles we have described, and that they can in fact mimic the actions of a human computer very closely [9].

9. Ne souhaitant pas écraser le lecteur sous une pluie de références indigestes, j'arrête là les citations données en corps de texte. Puisqu'il s'agit là d'un sujet controversé parmi les historiens, nous donnons dans cette note l'ensemble des autres citations soutenant notre interprétation. On considérera dans un premier temps cet autre extrait de ([228], 383) :

> If we take the properties of the universal machine in combination with the fact that machines processes and rule of thumb are synonymous, we may say that the universal machine is one which, when supplied with the appropriate instructions, can be made to do any rule of thumb process.

Dans ([226], 414), A. Turing parle de son modèle comme des *Logical Computing Machines (L.C.M.s)*, et décrit ainsi leur pouvoir expressif, avec une référence implicite à la thèse algorithmique :

> It is found in practice that L.C.M.s can do anything that could be described as 'rule of thumb' or 'purely mechanical'. This is sufficiently well established that it is now agreed amongst logicians that 'calculable by means of an L.C.M.' is the correct accurate rendering of such phrases. There are several mathematically equivalent but superficially very different renderings.

Dans ([224], 1), cité dans ([68], 480) :

> Electronic computers are intended to carry out any definite rule of thumb process which could have been done by a human operator working in a disciplined but unintelligent manner.

A. Turing affime également qu'un être humain muni de papier et d'un crayon peut jouer le rôle d'une Machine de Turing Universelle. Mais si tel est le cas, alors le calcul papier-crayon dispose d'un pouvoir expressif au moins aussi grand que celui des machines numériques ([227], 416) :

> It is possible to produce the effect of a computing machine by writing down a set of rules of procedure and asking a man to carry them out. Such a combination of a man with written instructions will be called a 'Paper Machine'. A man provided with paper, pencil, and rubber, and subject to strict discipline, is in effect a universal machine.

De manière plus indirecte, une autre citation de Turing ([223]) peut être lue sous un angle favorable à notre interprétation. Il y dit en effet que dans sa perspective, le mécanisme et l'écriture sont une seule et même chose, ce qui semble indiquer que les machines envisagées par Turing sont des machines manipulant des symboles ([68], 460, nous soulignons) :

> Presumably the child-brain is something like a note-book as one buys it from the stationers. Rather little mechanism, and lots of blank sheets. (*Mechanism and writing are from our point of view almost synonymous.*)

On peut enfin citer un témoignage de R. Gandy, un ancien étudiant d'A. Turing. Celui-ci décrit les intentions de Church et Turing de la manière suivante ([108], 123-124) :

Avant de conclure sur ce point, il faut souligner un autre passage qu'A. Hodges omet de citer, bien qu'il puisse être lu en faveur de son interprétation. Cet extrait appartient à une des discussions de la possibilité de simuler l'activité du cerveau humain par un ordinateur numérique, qui ont fait de Turing l'un des pères fondateurs de l'Intelligence Artificielle ([225], 482, nous soulignons) :

> A digital computer is a universal machine in the sense that it can be made to replace any machine of a certain very wide class. It will not replace a bulldozer or a steam-engine or a telescope, but *it will replace any rival design of calculating machine, that is to say any machine into which one can feed data and which will later print out results.* In order to arrange for our computer to imitate a given machine it is only necessary to programme the computer to calculate what the machine in question would do under given circumstances, and in particular what answers it would print out. The computer can then be made to print out the same answers. If now some particular machine can be described as a brain we have only to programme our digital computer to imitate it and it will also be a brain. If it is accepted that real brains, as found in animals, and in particular in men, are a sort of machine *it will follow that our digital computer, suitably programmed, will behave like a brain.*

Dans le premier passage souligné, A. Turing affirme explicitement la capacité d'une Machine de Turing Universelle de simuler n'importe quel type de machine de calcul : il énonce donc la forme empirique de la thèse de Church-Turing. Dans le second, de manière quelque peu plus inchoative, il énonce une autre forme de la thèse de Church-Turing, que nous allons bientôt discuter (voir section 2.2.2) : la thèse de Church-Turing psychologique. Il y affirme en effet que la machine de Turing peut simuler tout calcul effectué par le cerveau humain. Mais cet énoncé date de 1951, quinze ans après la publication de son modèle de machine, et il se situe dans un contexte de discussion très différent. Il ne s'agit pas ici d'exposer la signification de son article de 1936, mais d'énoncer une nouvelle thèse sur les capacités de sa machine numérique à reproduire l'activité cérébrale. A. Turing peut, sur la base de ces extraits, être vu comme un précurseur des énoncés des formes empirique et psychologique. Mais la masse des citations examinées ne permet en aucun cas d'affirmer que son intention théorique initiale, en 1936, était d'énoncer de

---

Both Church and Turing had in mind calculation by an abstract human being using some mechanical aids (such as paper and pencil). The word "abstract" indicates that the argument makes no appeal to the existence of practical limits of time and space. The word "effective" in the thesis serves to emphasize that the process of calculation is deterministic - not dependent on guesswork- and that it must terminate after a finite time.

telles thèses. Bien au contraire, les données textuelles montrent de manière écrasante que son intention initiale était bien d'énoncer la thèse algorithmique. Cet excursion historique étant à présent clos, nous allons reprendre notre discussion conceptuelle de la distinction des deux formes.

Il est parfois trivial de montrer de deux notions qu'elles sont extensionnellement équivalentes, bien qu'elles soient sémantiquement distinctes. Il est ainsi courant en mathématiques d'introduire deux définitions distinctes d'un même objet, pour démontrer immédiatement qu'elles sont équivalentes. Même si celles-ci ne pourront être entièrement développées que lorsque nous étudierons la forme empirique pour elle-même (voir chapitre 6), nous souhaiterions introduire les raisons pour lesquelles il n'est pas trivial de montrer que l'ensemble des fonctions calculables par un algorithme et l'ensemble des fonctions calculables par une machine seraient identiques.

La première est que les arguments typiquement présentés en faveur de la thèse de Church-Turing algorithmique ne permettent nullement de soutenir une thèse portant sur les machines de calcul. L'argument de modélisation est clairement sans pertinence, puisqu'il vise à montrer que les machines de Turing modélisent le calcul papier-crayon, et non l'ensemble des processus empiriques permettant de réaliser un calcul. L'absence de contre-exemple connu n'est guère plus pertinente, puisqu'elle porte sur l'absence d'exemple d'algorithme permettant de calculer une fonction non-récursive, et non sur l'absence d'une machine de calcul permettant de calculer une telle fonction.

L'argument de convergence des formalismes ne permet pas non plus de considérer la forme algorithmique de la thèse de Church-Turing comme un énoncé portant sur les machines. Tout d'abord, la convergence entre des langages de programmation abstraits, dénué de toute référence explicite à une implémentation possible, comme le λ-calcul, et un modèle du calcul papier-crayon, comme les machines de Turing, ne constitue pas en argument sur la modélisation de toutes les machines de calcul possibles. Ensuite, et surtout, il existe comme nous le verrons des modèles de calcul pour lesquels cette convergence n'a pas été démontrée, et qui peuvent précisément prétendre être la modélisation d'une certaine classe de machines (voir chapitre 6). Cette absence de pertinence des arguments originaux en faveur de la thèse de Church-Turing algorithmique pour une discussion portant sur la forme empirique est bien résumée par G. Piccinini ([190], 738) :

> It doesn't take a detailed analysis to see that the original arguments have little to do with the general properties of physical systems. The view that CT pertains directly to what can be computed by physical systems makes no sense of the fact that most logicians and theoretical computer scientists accept CT on the grounds of one or more of the original arguments.

Pour les premiers modèles de la calculabilité, la question de leur implémentabilité était triviale. La table d'instructions d'une machine de Turing,

la β-réduction d'un λ-terme sont évidemment implémentables par un calcul papier-crayon. Cela est naturel, dans la mesure où, comme nous l'avons vu précédemment(voir section 1.3.3), ces modèles sont plus ou moins directement inspirés par la pratique du calcul papier-crayon. En revanche, la question de l'implémentabilité n'est nullement triviale dans le cas de la forme empirique de la thèse de Church-Turing. Comme nous le verrons en étudiant les modèles d'hypercalcul (voir chapitre 6), de nombreux modèles offrant la possibilité de calculer une fonction non-récursive comptent, parmi leurs opérations primitives, des opérations qu'il est impossible à un homme d'exécuter. Pour que de tels modèles permettent réellement la résolution de problèmes, il faut donc qu'ils soient implémentables par une machine. L'automatisation n'est donc pas qu'un simple test pour ces modèles, mais une condition nécessaire de leur crédibilité en tant que modèles de calcul. Les calculs effectués au titre de ces modèles se doivent d'être automatisables, sous peine de ne pas être implémentables du tout. La question empirique de l'implémentabilité est donc décisive dans la discussion de ces modèles, alors qu'elle était triviale pour les modèles historiques de la calculabilité par un algorithme. Savoir si une machine de calcul permettrait l'implémentation d'opérations impossibles à une machine de Turing, et donc, modulo la forme algorithmique, à tout algorithme, constitue une question non-triviale.

L'identification de la thèse de Church-Turing à une thèse sur les machines nécessite donc un argument, et ne peut être prise comme une évidence sémantique. À ma connaissance, presque tous les auteurs ayant pris soin de comparer les deux thèses de Church-Turing ont souligné qu'elles ne doivent pas être confondues (voir [108], [207], [67], [212], [78], [84]) [10].

Cette deuxième forme ne doit pas être confondue avec un de ses cas particuliers, la forme physique de la thèse de Church :

Thèse de Church-Turing physique. *Toute fonction calculable par un système physique est récursive.*

La forme physique doit être comprise comme un cas particulier de la forme empirique, dans la mesure où la machine en question n'est pas forcément décrite en termes physiques : on peut ainsi songer au calcul avec des brins d'ADN (voir section 9.3.1). Quant à savoir si une machine biologique pourrait calculer plus de fonctions qu'une machine décrite dans les termes de la physique fondamentale, ou si la physique théorique peut au contraire poser des limites fondamentales au traitement de l'information permettant l'implémentation d'un calcul, c'est une question empirique difficile, qui ne peut être tranchée *a priori* [11]. Il serait donc préférable de parler de « thèse de Church-Turing empirique », et de réserver le qualificatif de « physique »au

10. À ma connaissance, K. Arkoudas constitue la seule exception à cette règle (voir [15]).

11. La résolution de ce problème pourrait d'ailleurs constituer un nouvel enjeu du débat vénérable autour du réductionnisme en biologie.

cas particulier où cette machine est décrite en termes physiques. Cette façon de faire se démarque quelque peu de l'usage courant, où les expressions de « calcul par un système physique », « forme physique de la thèse de Church-Turing »et de « thèse de Church-Turing physique »sont déjà raisonnablement bien implantées. Bien que je ne souhaite pas multiplier *ad libitum* les appellations, il me semble que cet usage introduit une ambiguïté regrettable, et je distinguerai à présent la thèse de Church-Turing empirique de son cas particulier, la thèse de Church-Turing physique, qui s'applique aux machines décrites en termes physiques. Nous accorderons cependant une attention privilégiée à la forme physique, lorsque nous aborderons la question de la signification empirique des principes computationnels (voir chapitre 6, Introduction).

Ni la forme physique ni la forme empirique ne doivent être confondues avec la thèse M formulée par R. Gandy (voir [108], 125-126), qui porte sur les fonctions calculables par un certain type de machines, les *discrete mechanical devices* (pour plus de détails sur la thèse M de Gandy, voir section 6.6).

Pour éviter toute confusion, je parlerai donc de « thèse de Church-Turing algorithmique », de « thèse de Church-Turing empirique »et de « thèse de Church-Turing physique », ainsi que de « forme algorithmique », de « forme empirique »et de « forme physique »de la thèse de Church-Turing. On se référera aux fonctions calculables selon ces thèses, respectivement, comme les « fonctions calculables par un algorithme », les « fonctions calculables par une machine », et les « fonctions calculables par un système physique ».

Tout comme la converse de la forme algorithmique, la converse de la forme empirique – toute fonction récursive est calculable par une machine – doit aussi être tenue pour vraie. Les ordinateurs modernes sont des implémentations d'une machine de Turing universelle et ont donc la capacité de calculer l'intégralité des fonctions récursives. Cette capacité est attribuée *de jure*, c'est-à-dire en ignorant les problèmes pragmatiques, bien souvent totalement insurmontables, liés à la complexité en temps et en espace de certaines fonctions. Mais il en va de même pour la notion mathématique de calculabilité, qui ignore également toute limitation contingente liées aux ressources. Ce n'est donc pas un défaut particulier de la notion de « calculabilité par une machine »que d'ignorer ces difficultés pratiques.

2.2 La thèse de Church-Turing psychologique : problèmes d'interprétation

Il existe une troisième formulation de la thèse de Church-Turing : *Toute fonction calculable par un homme est calculable par une machine de Turing.* Cette dernière forme ne sera pas l'objet de notre présent travail. Il nous semble cependant nécessaire de l'énoncer, et d'en discuter la distinction d'avec les deux premières formes. Cela nous permettra de préciser l'interpré-

tation de celles-ci, et d'écarter certains malentendus.

Cette formulation admet deux acceptions distinctes. La première conçoit cette proposition comme un cas particulier de la forme empirique, distinct de la forme physique. On parle alors de *forme psychologique de la thèse de Church*. La seconde est une conséquence d'une interprétation de la forme algorithmique. Selon cette interprétation, les fonctions calculables par un algorithme sont les fonctions calculables par un homme. Par conséquent, si la forme algorithmique de la thèse de Church-Turing est vraie, il s'ensuit immédiatement que toute fonction calculable par un homme est calculable par une machine de Turing.

Ces deux acceptions ne sont pas nécessairement exclusives, comme nous allons le voir immédiatement. Nous allons tout d'abord nous intéresser à la seconde interprétation, avant de revenir à la première.

2.2.1 Comme interprétation de la forme algorithmique

Cette acception est défendue par plusieurs auteurs, notamment J. Copeland ([67], 11) [12] :

> (...) the Church-Turing thesis concerns the extent of effective methods. Putting this another way (and ignoring contingencies such as boredom, death, or insufficiency of paper), the thesis concerns what a *human being* can achieve when working by rote with paper and pencil. (...) Essentially, then, the Church-Turing thesis says that no human computer, or machine that mimics a human computer, can out-compute the UTM.

Nous venons de voir que cette interprétation est soutenue par un grand nombre de sources historiques. Celles-ci montrent que, dans l'esprit des pères de la calculabilité, les notions de « calcul effectué en suivant une procédure effective »ou de « calcul effectué en suivant un algorithme »était synonyme de celle de « calcul effectué par un homme suivant aveuglément des instructions, et utilisant papier et crayon. »Par ce simple argument sémantique, l'ensemble des fonctions calculables par un algorithme est exactement l'ensemble des fonctions calculables par un homme utilisant papier et crayon. La quête des limites de la calculabilité par un algorithme, et celle des limites

12. Cette position est également adoptée par D. Knuth [152] :

> An algorithm is also generally expected to be *effective*, in the sense that its operations must all be sufficiently basic that they can in principle be done exactly and in a finite length of time by someone using pencil and paper.

On peut également la retrouver sous la plume de W. Sieg (ce dernier emploie le terme de *calculator* pour désigner un être humain exécutant un calcul papier-crayon) ([207],1)

> To investigate effective calculability is to analyze symbolic processes *that can in principle be carried out by calculators.*

de la calculabilité par un homme utilisant papier et crayon, sont d'emblée confondues.

Si cette interprétation est littéralement correcte, elle est sujette à certains malentendus. Le premier est évidemment que la référence explicite au calcul papier-crayon est ici inessentielle. Non seulement l'homme peut calculer en utilisant d'autres instruments de calcul[13] que le papier et le crayon, mais il peut même se passer de tout outil. Il lui est parfaitement possible de calculer une addition par un calcul mental, en suivant par exemple les étapes prescrites par l'algorithme scolaire usuel en son esprit plutôt que sur une feuille de papier. Pour cette raison, il est désirable de rechercher une façon de désigner la pratique quotidienne du calcul par l'être humain sans faire référence à des instruments de calcul particulier. Nous allons revenir sur ce point à la fin de ce paragraphe.

L'interprétation de la thèse algorithmique comme une thèse sur les limites du calcul effectué par un homme prête cependant à un malentendu plus sérieux. Dans une acception forte, elle nous mènerait à envisager la forme algorithmique comme une proposition de psychologie empirique. Cette interprétation fut ainsi défendue par E. Post, dans son article original de 1936. Après avoir affirmé que l'identification entre la calculabilité effective et sa formulation doit être considérée comme une hypothèse de travail (*working hypothesis*), Post fait la remarque critique suivante :

> Actually the work already done by Church and others carries this identification considerably beyond the working hypothesis stage. But to mask this identification under a definition hides the fact that a fundamental discovery in the limitations of the mathematicizing power of Homo Sapiens and blinds us to the need of its continual verification.

La démonstration de l'équivalence de différentes formulations de la calculabilité ne ferait pas de cette identification un axiome ou une définition, mais une loi naturelle (*natural law*). Bien qu'E. Post ne précise pas de quelle discipline cette loi relèverait, il est clair qu'il s'agit de psychologie empirique. Outre la mention des limitations d'Homo Sapiens, E. Post affirme en effet que sa formulation ne vise pas seulement à l'expressivité logique, mais à la fidélité psychologique[14]. La thèse de Post, variante de la thèse de Church-Turing, est donc d'emblée présentée comme une proposition de psychologie empirique.

En nous rappelant les justifications proposées par Turing dans le cadre de son argument de modélisation, on pourrait affirmer, en soutien de la position

13. Voir section 2.1 pour la définition d'un instrument de calcul, par opposition à une machine de calcul.

14. "The writer expects the present formulation to turn out to be logically equivalent to recursiveness in the sense of the Gödel-Church development. Its purpose, however, is not only to present a system of a certain logical potency but also, in its restricted field, of psychological fidelity." ([193], 105)

d'E. Post, que les limites de la calculabilité sont fondées sur des limitations perceptives et cognitives *propres* à l'être humain utilisant papier et crayon. Nous allons à présent montrer en quoi cette vision psychologisante de la thèse de Church-Turing est erronée.

Comme nous l'avons mentionné plus haut (voir section 1.2), les questions de calculabilité portent sur l'existence d'un algorithme pour résoudre un problème donné. Le moyen employé pour exécuter cet algorithme est indifférent à ces questions. En particulier, les résultats de calculabilité sont, dans un sens que nous allons préciser immédiatement, indifférentes à la distinction que nous avons faite précédemment entre « instrument de calcul »et « machine de calcul ». Comme nous l'avons dit à la section précédente, les machines numériques programmables sont des machines de calcul, et comptent comme implémentation de machines de Turing. Mais un homme armé de papier et de crayon, ou secondé d'un boulier, suivant aveuglément la table d'instructions d'une machine de Turing, constitue lui aussi une implémentation d'une machine de Turing. La seule question pertinente, du point de vue de la théorie de la calculabilité, est de savoir s'il existe une table d'instructions d'une machine de Turing, ou programme, qui permette de résoudre le problème considéré. La démonstration des résultats d'indécidabilité s'appuie uniquement sur la démonstration de l'existence, ou de l'inexistence, d'une machine de Turing pour effectuer une tâche donnée. Les moyens particuliers employés pour l'exécution des instructions d'une telle machine -appelés parfois par les informaticiens « modèle de hardware »- sont inessentiels à l'établissement de ces résultats. Il serait donc dépourvu de sens d'accorder un privilège à un modèle de hardware particulier [15].

On pourrait objecter qu'une telle remarque, si elle décrit bien la manière dont sont démontrés les résultats de calculabilité une fois la thèse de Church-Turing algorithmique acceptée, ne porte pas sur les motifs fondamentaux de l'admission de cette même thèse. Or, comme nous l'avons vu durant notre discussion de l'argument de modélisation, la capacité de la machine de Turing à modéliser la pratique du calcul papier-crayon est justement un argument en faveur de la forme algorithmique. Si les raisons fondamentales menant à l'admission de la forme algorithmique sont les limitations perceptives et cognitives de la pratique humaine du calcul papier-crayon, alors il est toujours légitime de considérer la forme algorithmique comme une thèse de psychologie empirique, même si la référence à cette pratique disparaît ensuite dans l'usage de la théorie de la calculabilité.

---

15. Copeland écarte explicitement un tel malentendu, puisqu'il déclare ([67], nous soulignons) :

> Essentially, then, the Church-Turing thesis says that no human computer, *or machine that mimics a human computer*, can out-compute the UTM.

Cela n'empêche pas K. Arkoudas ([15], 3) de lui en faire reproche.

Comme nous l'avons vu lors de notre discussion des arguments présentés en faveur de la forme algorithmique, l'argument de modélisation, en sus d'être un argument et non une démonstration, n'est qu'un argument parmi d'autres. Son poids relatif peut être discuté. Mais admettons aux fins de la discussion que cet argument soit décisif. Son examen plus précis, loin de soutenir l'acception de la forme algorithmique comme une proposition de psychologie empirique, souligne encore l'absence de pertinence des considérations propres au sujet humain dans la discussion de la calculabilité. Les contraintes énoncées par Turing au sujet du calcul papier-crayon sont en effet très générales, et s'appliquent également à d'autres dispositifs possibles d'exécution d'un algorithme. La plupart de ces contraintes sont en effet des contraintes de finitude, appliquées

- au nombre d'états du processeur ;
- au nombre de symboles modifiés par étape ;
- au déplacement effectué entre étapes ;
- à la capacité du processeur à lire et écrire des symboles distincts.

Quant aux autres contraintes que sont le déterminisme du calcul, et l'absence de nécessité de l'emploi du plan, elles ne sont pas justifiées par Turing en termes délimitations du processeur humain. Rien dans l'argumentation de Turing ne permet donc de faire de la forme algorithmique une proposition portant spécifiquement sur les capacités humaines. Si les contraintes étudiées s'appliquent bien sûr à l'homme utilisant papier et crayon, elles ne lui sont pas spécifiques. Elles ne sont pas même spécifiques à des entités qu'on pourrait croire douées de psychologie, et s'appliquent tout aussi bien à d'autres modèles de hardware, y compris des machines de calcul. Si l'argument de modélisation formulé par Turing tire bien évidemment inspiration de la pratique du calcul papier-crayon, les conclusions formulées sont d'une portée bien plus large. Il serait donc erroné de voir en la forme algorithmique une thèse de psychologie empirique [16].

16. Ce point a été correctement souligné par K. Arkoudas ([15], 3, nous soulignons) :

> Historically, it is true that the fundamental constraints on algorithms (e.g., that every step should manipulate a finite amount of information, terminate in a finite amount of time, etc.) can be traced to corresponding human limitations, and that Turing explicitly referred to human computers as a means of analogy when he first introduced Turing machines (e.g., comparing the state of the machine to a human's "state of mind," etc.), and in some of his later writings as well. However, the constraints in question are extremely meager, and a moment's reflection will suggest that they are applicable not only to humans but to any type of *finitary* system or device that purports to compute symbolically, that is, any type of computer in which all of the following quantities are finite : number of components, duration of converging computations, memory used by converging computations, number of internal states (or program size) ; symbol-recognition sensitivity ; and precision of symbolic output. *There is nothing intrinsically human about these constraints.*

Appelons « calcul symbolique finitaire », un calcul effectué par une manipulation de symboles [17] soumise aux contraintes de finitude ci-dessus exprimées. Lorsqu'on désigne l'objet de la thèse de Church-Turing, et de l'argument de modélisation, il nous semble judicieux, afin d'éviter les confusions que nous venons de mentionner, de le désigner comme étant le calcul symbolique finitaire, plutôt que le calcul effectué par un homme. Plutôt que d'affirmer que les fonctions calculables par un algorithme sont les fonctions calculables par un homme, il est préférable d'employer la formulation suivante : *Les fonctions calculables par un algorithme sont les fonctions calculables par un calcul symbolique finitaire.*

Une telle formulation permet d'éviter la tentation psychologisante contenue dans une formulation en termes de calcul effectué par un homme. Si l'on admet la validité de l'argument de modélisation, on peut ensuite ajouter comme un simple corollaire que *le calcul effectué par un homme utilisant papier et crayon est un calcul symbolique finitaire. Les fonctions calculables par un homme utilisant papier et crayon sont exactement les fonctions calculables par un calcul symbolique finitaire.*

2.2.2 Comme cas particulier de la forme empirique

L'analyse des relations entre « fonction calculable par un algorithme »et « fonction calculable par un homme utilisant papier et crayon »doit être distincte d'une seconde interprétation de notre énoncé initial. Cette seconde interprétation porte sur une proposition indépendante de la forme algorithmique, qui peut être présentée comme un cas particulier de la thèse de Church-Turing empirique, couramment nommée « forme psychologique de la thèse de Church-Turing ».

Pour mieux comprendre comment un même énoncé peut recouvrir deux propositions distinctes, il faut voir que l'expression « calculable par un homme »peut recevoir d'autres interprétations que « calculable par un homme utilisant papier et crayon. »

Il faut tout d'abord distinguer en toute généralité ce qui est « calculable par un homme effectuant un calcul papier-crayon », de « ce qui est calculable par un homme effectuant un calcul mental. »Nous avons vu plus haut que le calcul papier-crayon peut être exécuté par un calcul mental. Ceci n'implique pas que tout calcul mental puisse être considéré comme la simple exécution en l'esprit d'un calcul papier-crayon. Une telle assimilation peut être opérée sans trop de risque lorsque le calcul mental est parfaitement conscient. Mais

17. Le terme de « symbole »peut ici poser un léger problème. Il serait naturel de réagir à notre dernière proposition en affirmant que les calculateurs modernes, à l'exception de machines réalisées à de pures fins d'exhibition, ne manipulent pas des symboles écrits sur le papier, mais des signaux électroniques. Je désignerai néanmoins ici par « symbole »tout encodage discret et finitaire de l'information. Avec cette conception élargie du symbole, il est permis de dire qu'un scribe humain et une machine numérique effectuent tous deux des calculs symboliques.

le calcul mental peut être un processus seulement partiellement conscient. Le sujet calculant lui-même peut éprouver des difficultés à décrire la procédure suivie lors de l'exécution de son calcul. Ainsi, lorsqu'on étudie les processus mentaux des prodiges calculatoires capables d'exécuter des calculs complexes en un temps très court, il n'est pas même évident d'établir que le sujet suit bien un algorithme. Lorsqu'on lui demandera de décrire son expérience mentale au moment où il réalise une tâche complexe, le sujet se trouvera parfois incapable de distinguer des étapes dans son processus de découverte du résultat, et le caractérisera comme une intuition immédiate. Il s'agit donc d'une question de psychologie empirique difficile, que de déterminer si tout sujet humain effectuant un calcul mental suit bel et bien un algorithme, même si on peut bien sûr concevoir que cela soit le cas (voir [137], 635). Quel que soit l'état d'avancement de la recherche en psychologie empirique à ce sujet, montrer que le calcul mental est essentiellement équivalent au calcul symbolique est un véritable résultat scientifique, et non une simple équivalence sémantique [18].

Ce qui est calculable par un homme effectuant un calcul papier-crayon doit également être distingué de ce qui est calculable par le cerveau humain, modélisé comme une machine à calculer. On peut dans ce cas considérer que l'être humain calcule, même lorsqu'il n'effectue pas consciemment des calculs, par calcul papier-crayon ou par calcul mental. Une telle approche modifie complètement l'estimation des capacités calculatoires de l'être humain. On pourra ainsi attribuer au cerveau humain une vitesse d'exécution bien plus grande que celle qu'atteint un calculateur humain, et le considérer comme une machine parallèle, tandis que notre expérience consciente du calcul semble séquentielle. Enfin, on pourra lui attribuer des propriétés de calculabilité différentes, puisque certains modèles du cerveau humain lui attribue la faculté de calculer des fonctions que nulle machine de Turing ne peut calculer (voir notamment [209],[210]). L'étude du cerveau humain conçu comme une machine de calcul et l'étude du calcul symbolique pratiqué par le sujet humain constituent *a priori* deux questions autonomes [19].

On peut considérer la forme psychologique de la thèse de Church comme un cas particulier de la forme empirique. Le cerveau y est en effet modélisé comme une machine de calcul traitant des entrées et rejetant des sorties, sans que la conscience humaine n'ait plus son mot à dire. Ce cas particulier de la thèse empirique doit être considéré comme distinct de la forme physique, tant que le cerveau humain n'a pas fait l'objet d'une modélisation physique

18. Le lecteur remarquera au passage que même s'il était établi qu'un certain prodige du calcul mental était capable d'effectuer une tâche sans suivre un algorithme, cela n'impliquerait pas qu'il soit capable d'effectuer une tâche qu'aucun algorithme ne peut effectuer. L'ensemble des tâches effectuables par un tel processus mental non-algorithmique pourrait fort bien être un sous-ensemble strict des tâches effectuables par une machine de Turing.

19. J. Copeland fait bien la distinction, puisqu'il dit bien que la possibilité d'un hypercalcul psychologique est une question empirique ouverte ([67], 17).

explicite. Mais en tout état de cause, la forme psychologique de la thèse de Church-Turing ne constitue pas une proposition indépendante des formes algorithmique et empirique.

2.3 Une véritable question empirique ?

La possibilité de calculer une fonction non-récursive a été baptisée « hypercalcul »(*hypercomputation*) par J. Copeland et D. Proudfoot [69], et « calcul supraTuring »par H. Siegelmann [209]. Il existe de fait une littérature vaste et diversifiée proposant des modèles de calcul permettant l'évaluation de fonctions non-récursives (voir chapitre 6), qu'on qualifiera par analogie de « modèle d'hypercalcul ». Dans l'état actuel de la recherche, tous ces modèles sont cependant des modèles purement théoriques, qui n'ont donné lieu à aucune implémentation. En première approche, la principale question posée par ces modèles est donc la suivante : *Est-il possible de bâtir une machine permettant l'implémentation d'un modèle d'hypercalcul*[20] *?* Une réponse positive à une question identique en substance a été appelée « thèse de l'hypercalcul »dans [101].

Il existe deux interprétations différentes de cette dernière question. La première consiste à considérer qu'il s'agit là d'une véritable question empirique, qui doit faire l'objet d'un travail théorique en physique, ou dans une autre science empirique : est-il empiriquement possible de bâtir une telle machine ? La seconde consiste à la considérer comme une question méthodologique : est-il méthodologiquement possible d'arriver à montrer qu'une machine implémente bien un modèle d'hypercalcul ?

Trois raisons nous imposent de débuter par l'examen de l'interprétation méthodologique. Tout d'abord, la question empirique suppose une réponse positive à la question méthodologique : s'il est impossible de démontrer qu'une machine implémente bien un modèle d'hypercalcul, alors on ne peut considérer l'existence de l'hypercalcul comme une véritable question empirique. En outre, il existe dans la littérature une position sceptique à l'égard de l'hypercalcul, qu'on qualifiera de *scepticisme méthodologique*. Cette position s'appuie explicitement sur une réponse négative à l'interprétation méthodologique : il est méthodologiquement impossible d'affirmer qu'une machine serait une implémentation d'un modèle d'hypercalcul. Il s'agit là d'une classe d'objections particulièrement puissantes, puisqu'elle entend disqualifier tout modèle d'hypercalcul, présent et à venir. Enfin, l'examen de l'interprétation méthodologique est essentiel pour notre questionnement philosophique, puisqu'il engage non seulement la valeur de vérité, mais aussi le statut épistémologique de la thèse de Church empirique. S'il est méthodologiquement

20. Pour le bien de la variété et de la brièveté, nous substituerons occasionnellement les expressions « implémentation d'un hypercalcul »et « machine hypercalculante »à l'expression plus longue d'« implémentation d'un modèle d'hypercalcul ».

impossible de démontrer qu'une machine implémente un modèle d'hypercalcul, la thèse de Church empirique est vraie. Cependant, elle ne constitue pas une véritable proposition empirique, mais seulement un point de méthode. Dans un tel cas de figure, les limites de la calculabilité, malgré ce que la formulation de la thèse de Church empirique aurait pu nous donner à croire, ne constitue pas une question empirique.

Pour toutes ces raisons, notre examen de la thèse de Church empirique débutera par l'examen des arguments relevant du scepticisme méthodologique.

2.3.1 Le problème de la vérification

Admettons qu'on examine un système physique donné S, pour savoir s'il est un candidat raisonnable au titre de machine hypercalculante. Comment déterminer que ce système calcule bel et bien une fonction non-récursive ? La première objection à une telle possibilité a été baptisée « problème de la vérification », par J. Copeland [66] : comment savoir que les valeurs données par ce système en sorties sont bien les valeurs d'une fonction non-calculable, alors qu'aucun algorithme ne permet de connaître les valeurs d'une telle fonction ?

Cette objection est cependant aisément écartée, car elle est fondée sur une incompréhension de la signification des résultats d'indécidabilité. Qu'une fonction ne soit pas calculable n'interdit ni de connaître un nombre arbitrairement grand de ses valeurs, ni d'en connaître une valeur donnée. Comme nous l'avons remarqué plus haut (voir section 1.2), un résultat d'indécidabilité porte sur l'inexistence d'une méthode mathématique uniforme pour résoudre un problème donné. L'inexistence d'un algorithme signifie que nous ne disposons pas d'une telle méthode : elle n'implique pas qu'on ne puisse pas résoudre des instances de ce problème. Par exemple, nous pouvons connaître de nombreuses valeurs de la fonction d'arrêt, bien que le problème de l'arrêt soit indécidable [21]. À chaque fois qu'une machine de Turing, codée par l'entier n, s'arrête sur une entrée donnée m, nous connaissons la valeur correspondante de la fonction d'arrêt : $h(n, m) = 1$. Une telle approche ne nous permet que de connaître un nombre fini, quoique non borné, de valeurs de la fonction d'arrêt. En revanche, en démontrant qu'une fonction est totale [22], on peut déduire une infinité de valeurs de la fonction d'arrêt. Par exemple, si n est le code d'une machine de Turing calculant la fonction successeur, on sait que, pour tout entier m, $h(n, m) = 1$.

Les deux précédents arguments ne fonctionnent que pour une réponse positive au problème de l'arrêt, mais il est aussi possible de calculer des

21. Ceci a déjà été remarqué par J. Copeland dans ([66], 256).

22. Une fonction f d'un ensemble E dans un ensemble F est totale ssi f est partout définie sur E. Si le domaine de définition de f est un sous-ensemble propre de E, la fonction est partielle.

valeurs de la fonction d'arrêt quand la réponse est négative. En démontrant qu'une fonction n'est pas définie en un de ses arguments, on peut en déduire un ensemble de valeurs de la fonction d'arrêt pour tous les programmes calculant cette fonction. Ainsi, pour tout entier n codant une machine de Turing calculant une fonction f, si l'on peut démontrer que f n'est pas définie en un de ses arguments m, alors on sait que $h(n, m) = 0$. Que l'on songe par exemple qu'on souhaite calculer la valeur d'une fraction dont le dénominateur est un polynôme en une variable x. Lorsque x prend pour valeur une racine m du polynôme, la fonction n'est pas définie. Pour toute machine de Turing calculant cette fraction codée par un entier n, on a donc $h(n, m) = 0$. On peut considérer le calcul des racines du polynôme comme une méthode pour trouver des valeurs auxquelles la fonction n'est pas définie, et pour lesquelles le problème de l'arrêt reçoit par conséquent une réponse négative.

Chacune des approches que nous venons de décrire constitue une méthode pour résoudre une classe d'instances du problème de l'arrêt. On peut les qualifier de méthodes particulières, en ceci qu'elles ne permettent pas de résoudre le problème dans le cas général, par opposition à la méthode générale qu'aurait constitué un algorithme résolvant le problème de l'arrêt. La démonstration de l'indécidabilité d'un problème interdit de trouver une méthode permettant de résoudre toutes les instances d'un problème : elle n'empêche pas de connaître la solution du problème pour un nombre indéfini d'instances.

L'indécidabilité d'un problème n'implique pas non plus qu'il existerait des instances de ce problème qu'on ne saurait résoudre par aucune méthode particulière. Soient f une fonction, p, p' des programmes. On note $exec(p, n)$ l'exécution du programme p sur une entrée n.

$$\neg\exists p \forall n((f(n) = y) \leftrightarrow (exec(p, n) = y)) \tag{2.1}$$

n'implique pas

$$\exists n \neg\exists p'((f(n) = y) \leftrightarrow (exec(p', n) = y)). \tag{2.2}$$

L'inexistence d'un algorithme pour calculer une fonction donnée n'interdit pas de connaître un nombre arbitraire de valeurs de cette fonction, et elle n'interdit pas même de connaître une valeur donnée de cette fonction : elle rend seulement la recherche de ces valeurs beaucoup plus difficile.

Il serait donc en droit possible, bien qu'il puisse être pratiquement très difficile, de vérifier pour un ensemble arbitrairement grand de valeurs que les valeurs données par la machine hypercalculante sont correctes. Ceci ne démontrerait pas que la machine en question calcule bel et bien une fonction non-récursive. Mais la situation de la machine hypercalculante est à cet égard strictement identique à celle d'une machine dont on chercherait à déterminer si elle calcule bien l'addition, en la calibrant sur des entrées dont la sortie

associée est déjà connue. La construction d'une machine hypercalculante, en tant que question empirique, n'est pas censée échapper aux paradoxes usuels du raisonnement inductif. Ce point crucial est bien illustré par la citation suivante de C. Cleland (voir [60], 223-224)[23] :

> One of the most common objections to the possibility of physical hypercomputation concerns the difficulty of verifying that a physical device has computed a Turing uncomputable function. But as I have argued elsewhere [3,5], the verification problem is not unique to hypercomputational devices. It afflicts all physical devices for computing functions. As an example, it is impossible to conclusively verify that my hand calculator computes basic arithmetical functions like addition. For my hand calculator will break down long before it finishes computing a total function like addition. If I am lucky and no electronic glitches have occurred during its remarkably short life my calculator will compute a partial function that is consistent with its computing addition. Yet this partial function is also consistent with my calculator's computing any one of an uncountably infinite number of other total functions that are not addition. On the other hand, I am often unlucky (particularly with cheap calculators !) since all physical devices are subject to physical perturbations that may cause them to malfunction. This underscores a frequently overlooked point. The claim that a physical machine (or, for that matter, human) computes a given function is an empirical hypothesis. Its plausibility ultimately depends upon physical considerations, both empirical and theoretical (e.g., probabilistic causal relations, counterfactual suppositions grounded in physical law), as well as mathematical considerations (e.g., identity relations among different arithmetical operations). (...) we cannot dismiss the possibility that we will someday have reasons for believing that some physical device computes a Turing uncomputable function that are just as good as the reasons that we currently have for believing that our hand calculators compute addition.

On ne saurait donc reprocher à une machine hypercalculante d'être soumise aux mêmes difficultés expérimentales que la construction d'une machine

23. C. Cleland emploie l'expression « problème de vérification » dans un sens différent de celui introduit par J. Copeland. Il ne s'agit pas là d'établir la correction des valeurs données par une machine hypercalculante, mais d'établir à partir des données de l'expérience que notre machine implémente bien un hypercalcul. La confusion est naturelle, dans la mesure où il s'agit dans les deux cas d'établir que la machine est bien l'implémentation d'un modèle d'hypercalcul. Les deux problèmes sont cependant logiquement distincts, dans la mesure où le problème de la vérification, tel qu'il est initialement formulé par J. Copeland, interdit même d'affirmer qu'une et une seule valeur donnée par la machine est la solution d'une instance d'un problème indécidable.

calculante respectant la thèse de Church-Turing empirique [24].

Le « problème de la vérification »est donc fondé sur une fausse interprétation des résultats d'indécidabilité et n'interdit nullement la possibilité d'une machine hypercalculante.

2.3.2 Le scepticisme méthodologique de M. Davis

La dépendance à la théorie

Le théoricien de la calculabilité M. Davis est le sceptique le plus fréquemment cité dans le débat sur l'hypercalcul [116]. L'essentiel des critiques de M. Davis ne sont pas dirigées à l'encontre d'un modèle particulier d'hypercalcul, mais contre l'intégralité du programme de recherche. Il n'hésite ainsi pas à affirmer qu'il n'existe pas de sujet de recherche tel que l'hypercalcul [25].

Même s'il ne qualifie pas lui-même sa position de cette manière, M. Davis peut être considéré comme un représentant du scepticisme méthodologique. Il refuse explicitement qu'on puisse considérer la thèse de Church-Turing comme une proposition empirique ([74], 1, nous soulignons) :

> (...) by the late 1990s a consensus had developed that the Church-Turing Thesis is indeed the basis of modern computing practice. *Statements could be found characterizing it as a natural law, without proper care to distinguish between the infinitary nature of the work of the logicians and the necessarily finite character of physical computers.*

Il emploie divers arguments au cours de sa critique, présentés de manière toujours informelle, et parfois succincte. Nous allons voir que ses arguments doivent parfois faire l'objet d'interprétations, afin de les discuter de la façon la plus charitable possible.

Avant d'entamer cette discussion, quelques remarques préliminaires sont nécessaires au sujet des modèles d'hypercalcul. Les critiques de M. Davis, tout en prenant parfois pour cible un modèle donné d'hypercalcul, sont censés montrer l'impossibilité de toute machine hypercalculante. Afin de voir si ces critiques ont bien toute la généralité voulue, il nous faut distinguer entre plusieurs classes de modèles d'hypercalcul.

Une machine hypercalculante serait donc la réalisation physique S d'un oracle : on devrait pouvoir spécifier la tâche exécutée par cette machine, sans pouvoir décrire son fonctionnement interne comme l'exécution d'une procédure effective [26]. Cet oracle pourrait prendre plusieurs formes :

24. Pour plus de détails sur cette question, voir section suivante.

25. C'est même explicitement le titre d'un de ses articles, *Why there is no such Discipline as Hypercomputation*[75].

26. Ce dernier raisonnement suppose d'admettre en hypothèse la correction de la thèse de Church-Turing algorithmique.

– *Oracle fonctionnel.* Soient $x_1, ..., x_n, y$ une suite de paramètres mesurables de S, tels que $y = f(x_1, ..., x_n)$. Si f n'est pas une fonction récursive, S peut être utilisé pour évaluer une telle fonction. Il suffirait pour cela que les paramètres $x_1, ..., x_n$ soient contrôlables avec suffisamment de précision pour sélectionner une entrée, et y décodable par une opération de mesure suffisamment précise pour distinguer une sortie.
– *Oracle numérique.* La valeur d'un paramètre y est égale à un réel non-calculable r. y pourrait être soit une constante, soit une variable dynamique, sur laquelle on disposera d'un contrôle suffisant pour permettre de la fixer à une valeur déterminée pendant le temps nécessaire à une opération de mesure. En mesurant la valeur de y avec une précision croissante, on sera ainsi capable de connaître la suite non-calculable des décimales de r.

Ces distinctions faites, admettons que nous soyons mis face à un système physique S. Comment pouvons-nous défendre, ou exclure, la possibilité qu'un tel système permette l'implémentation d'un hypercalcul ? Nous allons d'abord montrer avec M. Davis qu'une telle décision dépend d'une théorie physique d'arrière-plan, et ne peut être prise sur la base d'une simple induction empirique. Nous sommes donc réduits à une interaction purement phénoménologique avec le système. Notre question initiale devient donc : pouvons-nous conclure, sur la base de données purement phénoménologiques, que ce système permet de réaliser une tâche qu'aucune machine de Turing ne peut exécuter ?

Admettons cependant qu'elle dispose de suffisamment de contrôle sur ce système pour effectuer une série arbitrairement longue de mesures à valeurs rationnelles sur un paramètre mesurable x de ce système, obtenant ainsi une suite finie de résultats. Par une pure induction expérimentale, le physicien expérimentateur peut-il légitimement affirmer que sa suite de mesures constitue le début d'une suite de nombres non-Turing calculable ?

Le théoricien de la calculabilité M. Davis aborde une question similaire dans son article *Why there is no such thing as hypercomputation.* Son argument porte uniquement sur la calculabilité d'une suite de nombres. Sa portée initiale est donc restreinte aux modèles d'hypercalcul que nous qualifié d'« oracles numériques », mais on va voir que certains de ses arguments se généralisent aux autres modèles.

> The great day has arrived ! The world's first working hypercomputer is being unveiled. Here comes the output - guaranteed by the engineers to be a non-Turing-computable infinite sequence of natural numbers :
>
> 23 ; 5 ; 1267 ; 111 ; 59 ; 87654 ; 21 ; 1729 ; 88888881 ; etc.
>
> But wait ! No matter how long this goes on, we will see only a finite number of these outputs. Moreover any such finite sequence

> of natural numbers is the initial part of both computable and non-computable infinite sequences (in fact, infinitely many of each kind). Thus, no finite amount of data will suffice to distinguish the computable from the non-computable, and since we, as finite beings with finite lifetimes will only have access to a finite amount of data, no possible experiment could certify that a device is truly going beyond the Turing computable.

L'argument de Davis part d'un constat simple : la suite de nombres dont on disposera par un tel procédé sera nécessairement finie, car l'information acquise expérimentalement sur un système est nécessairement finie.

À partir de ce point, l'argument de Davis est formulé de manière plus elliptique, et nécessite un effort de reconstruction, à la fois pour en formuler les dernières étapes, et en esquisser la conclusion. Une fois admis que toute l'information acquise par interaction avec le système est une suite finie de nombres, l'argument de M. Davis passe par une étape purement mathématique : pour toute suite d'entiers de longueur finie, il existe une infinité de suites infinies dont elle est une sous-suite. Parmi cette infinité de suites, une infinité est calculable, et une infinité ne l'est pas [27]. Il n'existe aucune raison mathématique pour affirmer, sur la base de la connaissance d'une suite finie de nombres, que celle-ci est le segment initial d'une des suites plutôt qu'une autre. Par conséquent, il n'existe aucune raison mathématique pour affirmer qu'elle est le segment initial d'une suite non-calculable, plutôt que d'une suite calculable.

Pour résumer, *sur la base d'une suite finie de nombres, il n'existe aucune raison mathématique de considérer cette suite comme une sous-suite d'une suite non-calculable plutôt que d'une suite infinie calculable.*

L'argument de Davis est à double tranchant. D'une part, il affirme qu'aucun ensemble fini de données ne pourra, à lui seul, nous contraindre d'admettre qu'un système permet de calculer une suite non-calculable. D'autre part, il montre qu'il n'existe pas de raison purement mathématique pour exclure une telle possibilité, sur la base de ce même ensemble. D'un point de vue phénoménologique, la calculabilité d'une telle suite est une question sous-déterminée.

L'argument de M. Davis porte initialement sur des suites de nombres. Cependant, comme on va le voir immédiatement, son raisonnement se généralise au cas fonctionnel.

Il faut tout d'abord distinguer les oracles permettant le calcul d'une fonction de domaine dénombrable, de ceux permettant le calcul d'une fonction de

27. Le cardinal de l'ensemble des suites infinies Σ^ω sur un alphabet fini Σ n'est pas dénombrable : l'ensemble des suites infinies ayant comme segment initial une suite de longueur n sera donc également non-dénombrable. Il est aisé d'engendrer un ensemble infini de suites calculables incluant cette sous-suite de manière triviale. Leur cardinalité sera nécessairement dénombrable, puisque l'ensemble des machines de Turing est dénombrable. Il reste donc après soustraction cardinale une infinité de suites non-calculables.

domaine non-dénombrable. Je parlerai donc d'"oracle fonctionnel à domaine dénombrable", pour désigner un oracle calculant une fonction de domaine dénombrable, et d'"oracle fonctionnel à domaine non-dénombrable", pour désigner un oracle calculant une fonction de domaine non-dénombrable.

L'argument portant sur les suites de nombres se généralise immédiatement aux oracles fonctionnels à domaine dénombrable. Les consultations successives de tels oracles peuvent être modélisées comme une suite de couples entrée-sortie $(0, f(0)), (1, f(1)), ..., (n, f(n))$, dont on peut ensuite demander si elle est calculable.

Dans le cas d'un oracle fonctionnel à domaine non-dénombrable, le problème posé par M. Davis est sensible dès le niveau de la description d'un couple entrée-sortie. Les paramètres observables associés aux entrées et aux sorties devraient être décrits par des réels, dont l'expansion décimale pourrait être infinie. Chaque mesure finie ne pourra cependant nous donner accès qu'à une sous-suite finie de ces décimales. La simple affirmation que ces deux suites finies sont des sous-suites du couple $(x, f(x))$ constitue une proposition allant bien au-delà des contraintes exercées par des données finies sur l'ensemble des suites infinies possibles.

Le type d'arguments que nous venons de discuter sera familier à tout lecteur habitué des débats en philosophie de la connaissance. Lorsqu'il affirme qu'un système suit une loi phénoménologique, le physicien fait bien plus que noter les résultats opérationnellement accessibles, et opérer une déduction mathématique. Le simple fait d'interpoler une courbe continue à partir d'un ensemble de points discrets constitue une décision théorique complexe, dans laquelle rentrent les considérations usuelles de simplicité, d'élégance et de cohérence théorique. On peut y voir un cas particulier de l'idée philosophique générale de sous-détermination de la théorie par l'expérience, en prenant soin de préciser que la découverte d'une loi phénoménologique doit ici compter comme une décision théorique. Affirmer qu'un système permet l'implémentation d'un hypercalcul suppose donc l'existence d'un modèle théorique de ce système. Comme toute proposition d'une théorie empirique, celle-ci ne peut se réduire à une simple déduction mathématique opérée sur un ensemble de données. Bien que la formulation de M. Davis soit plus elliptique, on peut lui attribuer une vision analogue, puisqu'il affirme juste après le passage précédemment cité :

> So, on what basis can someone claim that some device is indeed a "hypercomputer" ? It can only be on the basis of a physical theory.

.

Si notre interprétation est correcte, seule une théorie nous permettrait de défendre des propositions allant au-delà de ce qui est mathématiquement déductible à partir de données finies. Toute affirmation selon laquelle un système pourrait fonctionner comme un oracle non-récursif est une telle

proposition, de par l'argument mathématique ci-dessus. Par conséquent, la proposition selon laquelle un système est un oracle non-calculable ne peut être qu'une proposition théorique. Un système physique ne peut être considéré comme une machine hypercalculante que relativement à une théorie physique d'arrière-plan T, et non par le fruit d'une simple induction.

La nécessité des contraintes opérationnelles

Si un système S est considéré comme une machine hypercalculante, la modélisation du système S dans la théorie T comprend nécessairement l'emploi de mathématiques non-calculables. Un paramètre mesurable sera représenté par un réel non-calculable, dans le cas d'un oracle numérique, et un phénomène dynamique sera modélisé par une fonction non-calculable, dans le cas d'un oracle fonctionnel. Comme nous allons le voir à présent avec M. Davis, cette condition n'est pas suffisante en toute généralité.

Admettons que nous disposions d'un tel système S. Pour qu'un tel système puisse être utilisé comme un oracle, il faut qu'il soit possible d'exploiter opérationnellement la non-calculabilité théoriquement présente dans ce système. M. Davis va donc déployer une deuxième phase de son argumentation, qui vise à montrer qu'un oracle non-récursif n'a guère de chance de jamais pouvoir être utilisé [28] ([73], 12) :

> Consider what would be involved in harnessing a putative non-computable physics. What is usually taken to be the ultimate test of a physical theory, agreement with measurement to the extent that instruments permit, would be of no use, because no finite amount of information can verify the value of an infinite precision real number. All experience suggests that every physical theory is accepted only provisionnally with every expectation of its eventual replacement. But for an useable oracle to be obtained, one would require absolute certainty that a real number by a particular theory will not have its value changed if and when the theory is upgraded. Finally, even if one knew that some such number is not computable, in order to use it as an oracle, one would also have to know its degree of unsolvability. If indeed "the

28. M. Davis emploie exactement les mêmes arguments, dans une formulation plus ramassée, dans [75]. Nous laissons le lecteur jugé par lui-même qu'il n'y a aucune variation substantielle dans l'argumentation de M. Davis dans ce passage :

> Such a theory would have to be certified as being absolutely correct, unlike any existing theory, which physicists recognize to be only an approximation to reality. Furthermore, the theory would have to predict the value of some dimensionless uncomputable real number to infinite precision. Finally, the specifications of the machine would have to guarantee that it exactly follows the demands of this supposed theory. Needless to say, nothing like this is even remotely on the horizon.

search is on" for such a number, one can only pity those engaged in this misguided enterprise.

L'argument de Davis est encore une fois présenté de manière quelque peu elliptique. Comme il a déjà été souligné dans [116], il faut remarquer que l'argumentation de Davis est indûment restreinte au cas des oracles numériques. Nous allons donc examiner les objections de M. Davis à l'égard des oracles numériques, avant de déterminer dans un deuxième temps si elles s'étendent aux oracles fonctionnels.

L'argument de M. Davis part du constat suivant. Tout d'abord, tout nombre dont l'expansion décimale est finie, ou rationnel dyadique, est calculable. L'expansion décimale d'un nombre non-calculable est nécessairement infinie. Pour qu'un oracle donnant la suite des décimales d'un nombre non-calculable puisse être utilisé pour effectuer un hypercalcul, il faut donc qu'il n'existe aucune borne supérieure à la précision à laquelle il nous donne cette expansion. Si l'oracle cesse de donner une réponse après n requêtes, tout ce qu'il nous permet de connaître en pratique est un nombre à expansion décimale finie, donc calculable.

Ce problème se transpose au cas d'un oracle implémenté par un système physique. Admettons l'existence d'une théorie T décrivant la valeur d'un paramètre x d'un système S par un réel non-calculable, $x = r$. Pour disposer d'un accès épistémique à l'expansion de r, il faut lire ses valeurs par des opérations de mesure du paramètre x. Or, si nombre de paramètres physiques sont décrits en théorie par des nombres à expansion décimale infinie, aucune théorie physique n'a été à ce jour validée à une précision arbitrairement grande. Le partisan de l'hypercalcul se doit donc de répondre à la question suivante : quand bien même une théorie T prédirait qu'un paramètre x d'un système S est décrite par un nombre non-calculable r, peut-on garantir que les prédictions de T sont valides à une précision arbitrairement grande ?

À ce stade, on voit donc déjà que l'existence d'une théorie T décrivant un paramètre x par un nombre non-calculable, si elle est une condition nécessaire de l'hypercalcul, n'en est pas une condition suffisante. Pour un modèle comme l'oracle numérique, il faut également prendre en compte d'autres contraintes. La premièreest de justifier que les prédictions de T sont correctes à une précision arbitraire.

M. Davis défend que cette contrainte ne pourra jamais être satisfaite : *il n'existera jamais de théorie physique permettant la formulation de prédictions arbitrairement précises sur une valeur de paramètre observable.* Le principal argument en faveur de cette thèse est un argument historique. À ce jour, aucune théorie arbitrairement précise n'a jamais existé dans l'histoire de la physique. Nos plus anciennes théories physiques, comme la mécanique newtonienne ou l'électromagnétisme classique, ont toutes été prises en défaut à un certain degré de précision. Nos meilleures théories fondamentales actuelles, la théorie quantique des champs et la relativité générale, sont in-

compatibles au degré de précision nécessaire pour formuler une théorie de la gravitation quantique, et devront céder la place à une nouvelle théorie unifiée. Toute notre expérience historique confirme donc l'idée qu'une théorie physique n'est valide qu'à une certaine approximation, approximation qui ne peut être précisément découverte qu'une fois la théorie invalidée.

Un tel argument historique, s'il justifie un scepticisme de bon aloi à l'égard de la possibilité de l'hypercalcul, ne saurait nullement constitué une réfutation de principe de cette possibilité. Qu'est-ce qui interdirait en principe l'émergence future d'une théorie dont les prédictions seraient arbitrairement précises ? Sans formuler une véritable réponse de principe, nous pouvons formuler plusieurs arguments qui tous soulignent l'implausibilité d'une théorie arbitrairement précise.

Tout d'abord, comme le remarque à juste titre M. Davis, le progrès de la physique ne constitue pas nécessairement en une évaluation de plus en plus précise de valeurs de paramètres décrites par des réels. Le progrès théorique peut mener à une reconception de la nature du paramètre mesuré, reconception qui peut elle-même mener à un changement de représentation numérique. Le paramètre autrefois décrit par un réel à expansion décimale infinie peut se révéler être, à un autre niveau d'analyse, un nombre à l'expansion décimale finie. Ainsi les poids atomiques des éléments furent représentés par des réels, que les chimistes tâchaient de mesurer avec une précision croissante. Les progrès de la chimie permirent de réaliser que ces mesures étaient en réalité effectué non pas sur des échantillons purs des éléments, mais sur des échantillons présentant un mélange de divers isotopes, dont les poids atomiques, conçus comme le nombre des protons et neutrons, devaient être décrits par des entiers (voir [73], 10). Ensuite, selon nombre d'approches en physique théorique contemporaine, certains paramètres mesurables, comme la distance et le temps, ne seront plus physiquement définies à une certaine échelle de précision, typiquement l'échelle de Planck. Le progrès de la physique ne nous mènerait donc pas à une évaluation toujours plus précise de ces paramètres. Il montrerait au contraire qu'il n'y aurait aucun sens à les évaluer au-delà d'une précision donnée.

Enfin, la possibilité de mesurer un paramètre à une précision arbitraire est contradictoire avec un principe actuellement débattu en physique, le principe holographique. Sans entrer dans les détails d'exposition de ce principe, sur lesquels nous aurons amplement l'occasion de revenir (voir section 6.5.3), ce principe impose à un système S contenu dans un volume fini V de ne contenir qu'une quantité finie d'information. Si l'on accepte un tel résultat, il devient impossible de croire qu'un paramètre du système S puisse être décrite fondamentalement par un réel non-calculable, puisque ce dernier ne peut être spécifié que par une quantité infinie d'information. Nous reviendrons ultérieurement sur ce point.

Tous les arguments que nous venons de présenter justifient un scepticisme robuste à l'égard des modèle d'oracles numériques. Dans quelle me-

sure peuvent-ils s'étendre au cas des oracles fonctionnels, que M. Davis omet d'examiner ?

De manière évidente, les arguments de Davis se généralisent aux oracles fonctionnels à domaine non-dénombrable. Chaque entrée et chaque sortie de tels oracles nécessitent en effet une infinité d'information pour être décrites. Le cas des oracles fonctionnels à domaine dénombrable est en revanche totalement différent. Comme nous le verrons (voir section 3.1), les éléments d'un domaine dénombrable peuvent être indicés par des entiers, qui peuvent eux-mêmes être représentés par des suites finies de bits. Pour spécifier l'entrée donnée à un oracle fonctionnel à domaine dénombrable, et pour en lire la sortie, il n'est pas besoin de croire que les prédictions de la théorie T valent à une précision arbitrairement grande, ou qu'il soit possible de réaliser une mesure arbitrairement précise. Plus précisément, le sceptique doit encore montrer que la théorie physique d'arrière-plan T qui permettrait d'implémenter un oracle fonctionnel impliquerait également la possibilité d'une mesure robuste d'une valeur réelle. Pour formuler un tel argument, le sceptique devrait donner un modèle théorique explicite de son oracle fonctionnel, ce qui ne peut guère être présenté comment un argument purement méthodologique.

En toute généralité, les objections soulevées par M. Davis ne passent pas au cas fonctionnel, et ne permettent donc pas d'exclure tous les modèles d'hypercalcul possibles.

De manière plus fondamentale, l'argumentation de M. Davis souffre d'une restriction injustifiée au cas des modèles d'hypercalcul analogiques. À aucun moment il ne donne une raison principielle pour laquelle tout modèle d'hypercalcul devrait encoder l'information non-calculable dans la valeur d'un réel, et pour laquelle aucun modèle discret d'hypercalcul n'est possible. Si l'on veut exclure la possibilité même d'un modèle réaliste d'hypercalcul, un tel argument sera cependant nécessaire, sous peine de confondre une propriété des modèles existants avec une propriété nécessaire de tout modèle possible. En outre, la position de Davis lui impose de fournir un argument de nature logique, mathématique, ou méthodologique pour interdire la possibilité d'un modèle d'hypercalcul discret. S'il était nécessaire de recourir à des hypothèses empiriques pour exclure une telle possibilité, il ne serait plus possible de défendre, comme le fait M. Davis, que la thèse de Church-Turing empirique ne constitue pas une véritable proposition empirique. L'argumentation de M. Davis est donc incomplète, et ne saurait justifié une interprétation méthodologique de la thèse empirique, et pas même un rejet de la possibilité d'une machine hypercalculante.

Comme nous l'avons vu ci-dessus, la caractérisation de la thèse de Church-Turing comme une proposition empirique reposerait selon M. Davis sur une confusion conceptuelle, et négligerait de prendre en compte le caractère infinitaire de nombreuses notions de calculabilité. Cette position s'appuie sur une croyance fondamentale sur les résultats de mesures physiques, à savoir que le résultat de la détermination de la valeur d'une grandeur physique

par un protocole de mesure donné est descriptible par une quantité finie d'information, bornée par la précision de ce même protocole. Cette proposition, si elle n'a pas le statut explicite d'un principe de la physique, constitue néanmoins une proposition fondamentale à laquelle l'immense majorité des physiciens souscrirait. C'est précisément ce qui fait la force de l'argumentation de M. Davis, et permet d'adopter une position sceptique à l'égard de la plupart des modèles d'hypercalcul. Néanmoins, cette proposition, si elle peut être prise comme principe, ou appuyée par un principe putatif comme le principe de densité finie de l'information, doit être comprise comme une proposition empirique. Le fait qu'on puisse formuler un argument historique en faveur de ce principe n'implique nullement qu'il ne doive pas être compris comme une proposition empirique, ni même qu'il ne puisse devenir un authentique principe physique. Un délai historique extrêmement long peut séparer l'acceptation d'une proposition dans la pratique expérimentale et sa thématisation théorique. Qu'on songe par exemple à l'existence relativement récente du second principe de la thermodynamique, comparée à l'intuition immémorialle que certains phénomènes macroscopiques sont irréversibles. De manière analogue, l'idée qu'il est impossible de mesurer une observable à une précision arbitraire fait partie de notre expérience historique de la physique. Cela n'exclut pas qu'un jour un principe fondamental puisse venir justifier ce constat commun d'un point de vue théorique (voir section 6.5). La position sceptique de M. Davis, quelle que soit sa force par ailleurs, ne peut être qualifiée de purement méthodologique, dans la mesure où elle s'appuie sur une telle proposition empirique.

L'analyse détaillée de l'argumentation de M. Davis conforte donc le caractère empirique de la thèse de Church-Turing physique. Par l'étude de cas des oracles fonctionnels, on voit également que la précision bornée des mesures ne suffit pas à fonder la croyance en cette thèse, et que l'analyse des hypothèses physiques sous-jacentes doit être poursuivie.

Note sur l'interprétation des positions de M. Davis

À cause d'une rédaction quelque peu elliptique, la détermination exacte des positions de M. Davis n'est pas sans difficulté. Il est notamment difficile de discerner la conclusion précise à laquelle Davis souhaitait arriver à la suite de son premier argument, portant sur la suite finie de résultats donnée par une machine hypercalculante. Comme nous allons le voir immédiatement, certains auteurs ont pu proposer une lecture différente de la nôtre. Dans cette brève note, que le lecteur peut sauter sans dommage à la première lecture, nous souhaitons préciser les raisons qui nous ont poussé à adopter l'interprétation que nous venons d'exposer.

Deux hypothèses de lecture s'offrent à nous. La première serait de lire l'argument comme une critique de principe de la possibilité de l'hypercalcul. Une telle interprétation ferait beau jeu de la symétrie des cas calculable et

non-calculable, et omettrait de remarquer qu'il s'agit là d'une situation banale en méthodologie des sciences expérimentales. Il est vrai que le lecteur peut être invité à une telle interprétation par deux traits de la rédaction des textes. Le premier est la radicalité du ton de ces articles. Comme nous l'avons déjà mentionné plus haut, M. Davis n'a ainsi pas hésité à affirmer qu'il n'existait pas de domaine de recherche digne d'être nommé « hypercalcul. »Le second est que dans l'article *The myth of hypercomputation* [73], M. Davis propose une autre version du même argument, où l'impossibilité de vérifier expérimentalement l'implémentation d'un modèle d'hypercalcul semble encore plus nettement affirmée. M. Davis cite ainsi avec approbation une correspondance privée de Dana Scott où celui-ci affirme (*ibid.*, p.12) :

> Suppose some quantum genius gave you an oracle as a black box. No finite amount of observation would tell you what it does and why it is not recursive. Hence, there would be no way to write an algorithm to solve an understandable problem you couldn't solve before !

À la suite de ce passage, M. Davis reprend cependant en substance les arguments sur la précision des prédictions théoriques que nous venons de discuter. Il est donc possible d'adopter une hypothèse de lecture alternative, selon laquelle l'argument de finitude des données de l'observation ne doit pas être conçu comme une objection méthodologique en soi contre l'hypercalcul. Il s'agit là plutôt d'une considération préliminaire, menant aux arguments portant sur la précision des prédictions théoriques. Dans l'état de rédaction des textes de M. Davis, il est difficile de trancher entre ces deux hypothèses, mais, au vu de l'analyse que nous avons proposée, il nous a paru plus charitable d'adopter la seconde. D'autres auteurs ont adopté la première interprétation, et ont critiqué M. Davis à ce titre, notamment J. Copeland. Après s'être référé à la citation de D. Scott faite par M. Davis, Copeland commente en effet ([66], 257-258) :

> The problem that Scott is pointing to appears, however, to be a particular case of the general problem posed by Kripke's Wittgenstein [34]-and this is everyone's problem, having nothing to do with non-recursiveness per se. To illustrate : suppose God hands you a black box, informing you only that the box is able to tell you all the values of some Turing-machine computable two-place function on the natural numbers. No finite amount of observation would reveal what the black box does. Plus or quus ? 'Quus' is Kripke's term for a 'bent' version of integer addition, behaving like normal addition over some initial segment of the number sequence and then deviating in some way. Does the box implement plus, or $quus_1$, or $quus_2$, or ... ? No matter how many pairs of numbers are tried out-finitely many-the box's behaviour will be consistent with many different rules. Engineers will want

> to know why we are required to treat the device as a black box. Why can't we take it apart and study it in the light of our best physical theories ? Once they have taken the plus/quus box apart and reassembled it, the engineers may tell us 'Given current physics, our best hypothesis is that this box will add any pair of non-negative integers (so long as you insert more tape whenever the red light goes on).' It is because we believe similar pronouncements by engineers that we entrust our savings, and in the case of air travel even our lives, to computers. The engineers' task is in principle no different in the case of Scott's box, assuming, as we may in this imaginary scenario, that the best quantum physics of their day is non-recursive.

Le lecteur se convaincra aisément, à la lecture de la citation offerte ci-dessus, que l'esprit de notre analyse est parfaitement identique à la sienne, ainsi qu'à celle de C. Cleland (voir citation dans la section précédente). Nous différons uniquement sur la nécessité herméneutique d'attribuer cette vision à M. Davis.

Conclusion

Dans ce second chapitre, nous avons défendu, aux côtés de nombre d'auteurs contemporains, la nécessité de distinguer la thèse de Church-Turing empirique de la thèse de Church-Turing algorithmique. Nous avons en revanche rejetté la nécessité de distinguer une forme psychologique de la thèse de Church-Turing. Cette dernière se révèle après analyse être soit une interprétation de la thèse algorithmique, soit une forme particulière de la thèse empirique. En discutant ce dernier point, nous avons pu préciser la relation exacte entre l'argument de modélisation de Turing et les limites perceptuelles et cognitives humaines. Bien qu'A. Turing justifie son modèle par des limitations qui sont inspirées des limitations d'un être humain effectuant un calcul papier-crayon, celles-ci doivent être comprises comme des contraintes de finitude, qui n'ont rien de particulier à l'espèce humaine. Par conséquent, les limitations soulignées par A. Turing valent toutes aussi bien pour des machines que pour des hommes, et la thèse de Church-Turing algorithmique ne doit en aucun cas être comprise comme une thèse de psychologie empirique. Nous avons résumé ce résultat en affirmant que la thèse algorithmique a pour objet le calcul symbolique finitaire, et qu'un être humain effectuant un calcul papier-crayon effectue un calcul symbolique finitaire.

Nous avons également défendu le statut proprement empirique de la thèse physique, contre certaines objections qui en feraient une simple proposition méthodologique. Dans un premier temps, nous avons montré que le problème dit de la vérification prenait sa source dans une confusion entre indécidabilité d'un problème et impossibilité de résoudre ses instances. Dans un second temps, nous avons examiné les arguments sceptiques de M. Davis, et montrer

que sa position était fondée sur la critique suivante : l'implémentabilité d'un modèle d'hypercalcul supposerait de croire à la validité d'une théorie physique à une précision arbitraire. La difficulté soulevée par M. Davis est réelle. Nous aurons l'occasion d'y revenir en détail, et nous montrerons même qu'elle s'applique à d'autres modèles que ceux examinés par M. Davis (voir chapitre 6). Mais elle ne suffit pas à établir que la thèse empirique ne constitue pas une authentique proposition empirique. Pour cela, il faudrait démontrer la nécessité de cette difficulté pour tout modèle, et la démontrer sans faire usage d'aucune hypothèse empirique. Pour que sa critique soit décisive, il aurait fallu à M. Davis une démonstration logique ou méthodologique de l'impossibilité d'un modèle discret d'hypercalcul. En l'absence d'une telle preuve, son argumentation souffre d'une confusion entre critique des modèles existants, et critique de principe de tout modèle d'hypercalcul concevable.

En réfutant ces critiques, nous avons défendu la légitimité du problème représenté par l'interprétation empirique de la calculabilité. Si aucune objection logique ou méthodologique n'empêche de considérer qu'un système implémente un hypercalcul, alors la question du respect de la thèse de Church-Turing physique par toute machine de calcul devient une question empirique. Modulo la thèse de Church-Turing algorithmique, c'est donc la question des limites ultimes de la calculabilité qui peut dépendre d'hypothèses empiriques.

Cette défense du caractère empirique de la thèse est cependant purement de principe. À ce stade de notre travail, nous avons seulement montré qu'il n'existe pas d'objection de principe, d'ordre logique ou méthodologique, contre la possibilité d'un modèle d'hypercalcul implémentable. En l'absence d'une telle objection de principe, nous sommes obligés de suivre l'intuition première, selon laquelle la thèse de Church-Turing empirique est bien une proposition empirique. Nous n'avons pas formulé d'argument positif en faveur d'une telle position, et nous n'avons pas encore donné une idée claire de la possible signification empirique d'une telle proposition. Mais avant de nous attaquer à ce problème, nous allons ouvrir une parenthèse portant sur certaines objections conceptuelles faites à la thèse algorithmique.

Deuxième partie

La thèse de Church-Turing algorithmique comme expression des limites du calcul effectif

Chapitre 3

La thèse de Church-Turing sur les réels

Introduction

Nous avons présenté la thèse de Church-Turing algorithmique comme un énoncé assignant les limites ultimes de la calculabilité effective. Dans les deux chapitres suivants, nous allons examiner deux perspectives critiques sur cette interprétation. Aucune de ses critiques ne vise à montrer que la thèse de Church-Turing algorithmique est fausse, mais que celle-ci, bien que correcte, ne capturerait qu'une forme particulière du calcul effectif. Ces critiques représentent évidemment un enjeu fondamental pour la discussion philosophique de la thèse de Church-Turing algorithmique, puisqu'elles en affectent la portée même. Si elles s'avéraient fondées, l'interprétation philosophique de la thèse de Church-Turing comme délimitation des limites des procédures effectives serait invalidée. En outre, la thèse de Church-Turing algorithmique constituerait une mauvaise base de comparaison pour estimer le pouvoir expressif de toute machine. Une machine pourrait alors calculer une fonction qui ne soit pas calculable par une machine de Turing, sans que cela constitue une violation des limites ultimes de la calculabilité effective, et donc une véritable révolution informatique.

Ces critiques doivent être bien distinctes de critiques que nous examinerons ultérieurement (voir chapitre 6), en ce qu'elles sont exclusivement de nature conceptuelle, et ne mettent pas en oeuvre, ou du moins pas de prime abord, des considérations empiriques. Elles méritent donc un traitement séparé. La première de ces critiques porte sur la généralisation de la thèse de Church-Turing aux fonctions de domaine non-dénombrables, typiquement les fonctions réelles. La seconde, inspirée par un paradigme appelé *calcul interactif*, conteste la réduction, implicite dans les énoncés usuels de la thèse de Church-Turing, de l'ensemble des tâches calculables à un ensemble de fonctions calculables.

L'extension de la notion de « fonction calculable par une procédure ef-

fective »à des domaines non-dénombrables pose un ensemble de problèmes conceptuels, parfois épineux. Ces problèmes sont parfois considérés comme révélateurs de limitations intrinsèques de la portée de la thèse de Church-Turing algorithmique : si la thèse de Church-Turing n'est bien définie que pour les fonctions de domaine dénombrables, et que sa généralisation aux domaines non-dénombrables est problématique, les limites de la calculabilité algorithmique ne seraient alors connues que pour un sous-ensemble particulier de fonctions.

Nous allons présenter en détail les différentes difficultés conceptuelles qu'affronte la généralisation de la théorie de la calculabilité aux domaines non-dénombrables, avant de discuter si ces difficultés justifient la position sceptique que nous venons d'articuler. Avant de rentrer dans le détail des différents problèmes, il convient, à des fins pédagogiques, d'en donner une brève présentation synthétique.

La principale difficulté rencontrée par la théorie de la calculabilité sur les domaines non-dénombrables est l'absence de convergence des modèles. Tout comme la calculabilité sur les fonctions entières, la calculabilité sur les fonctions réelles a fait l'objet de plusieurs modélisations distinctes, portées par des intuitions très différentes. En revanche, ces modèles n'ont pas présentés immédiatement les relations de simulation réciproque qui sont à la base de l'argument de convergence des modèles. L'un des principaux arguments en faveur de la thèse de Church-Turing est donc absent dans le cas réel [1].

La deuxième difficulté fréquemment rencontrée est l'existence d'intuitions conflictuelles sur le caractère naturel de la modélisation offerte par certains formalismes, en particulier le plus courant d'entre eux, l'analyse récursive. Ainsi, comme nous le verrons ci-dessous, les fonctions réelles calculables au titre de l'analyse récursive sont des fonctions continues. Cela est parfois présenté comme un argument contre l'analyse récursive, dans la mesure où certains peuvent considérer que certaines fonctions discontinues seraient manifestement calculables, et qu'un autre modèle, le modèle B.S.S., permet le calcul de fonctions discontinues. [2]

1. Ceci permet de souligner *a posteriori* la force de l'argument de convergence des formalismes : rien ne garantissait *a priori* que des formalisations diverses soient extensionnellement équivalentes.

2. On peut trouver un exemple d'une telle position chez J. Earman ([85], 125) :

> Church's thesis, or proposal as I would prefer to call it, says,
> **(CP1)** *The class of programmable or algorithmically computable functions of the integers is to be identified with the Turing computable functions.*
> I have no doubts about the adequacy of (CP1), especially as regards the originally intended application to Hilbert's decision problem.(...) Church's initial proposal (CP1) could be extended to functions of the reals by
> **(CP2)** *The class of programmable or algorithmically computable functions of the reals is to be identified with the Grzegorczyk computable functions.*
> However, (CP2) does not carry the conviction of (CP1) because Grzegorczyk's definition, though useful for providing results in analysis, is only one of various possible ways to generalize Turing computability to functions of

3.1 Procédure effective et calculabilité réelle : le problème de la représentation effective des éléments du domaine

Le premier problème conceptuel posé par les domaines non-dénombrables est celui de la représentation effective de leurs éléments par des chaînes de caractère. Les domaines dénombrables ont en effet la propriété remarquable que tous les éléments de leur domaine sont représentables, univoquement et exactement, par un élément de Σ^*, ensemble des mots finis sur un alphabet fini Σ. On dira qu'un objet est *finitairement représentable*, ou *à représentation finitaire*, si et seulement si il est ainsi encodable par un mot fini, et, par extension, les domaine et codomaine d'une fonction sont finitairement représentable, ou à représentation finitaire, ssi tous leurs éléments sont finitairement représentables. Ceci permet de définir une fonction effectivement calculable sur un domaine dénombrable f comme une application associant au nom d'un élément $x \in Dom(f)$ le nom de $y = f(x)$, la valeur de f en cet argument : $f :\subseteq (\Sigma^*)^n \longmapsto \Sigma^*$. Les objets et fonctions ne jouissant pas de cette propriété seront dits *à représentation infinitaire.*

Un domaine non-dénombrable n'est pas finitairement représentable, par un simple argument de cardinalité. L'ensemble des suites finies de symboles étant de cardinalité dénombrable, il ne peut y avoir d'injection d'un domaine non-dénombrable dans cet ensemble. Pour pouvoir encoder les éléments d'un domaine non-dénombrable, il est donc nécessaire de substituer à Σ^* l'ensemble Σ^ω des mots infinis sur un alphabet fini Σ. De prime abord, on peut songer à l'expansion décimale des nombres réels comme possible exemple d'un ensemble de noms utilisables non seulement pour $\mathbb{R}$, mais pour tout ensemble de cardinalité identique [3].

L'argument de cardinalité sert uniquement à prouver l'existence d'éléments à représentation infinitaire parmi les réels. Comme nous le verrons, il n'est nullement nécessaire que le domaine d'une fonction soit non-dénombrable pour que celle-ci ait une représentation infinitaire. Les réels calculables forment

---

the reals.

On peut trouver l'expression d'une position similaire sous la plume de T. Paul et G. Longo ([143], 178) :

> (...) de cette équivalence des formalismes, au coeur de la thèse de Church, il ne reste rien pour ce qui est de la calculabilité sur les réels : les modèles proposés, dans leur structure originaire, sont démontrablement différents, pour ce qui concerne l'expressivité calculatoire (les classes de fonctions définies).

T. Paul et G. Longo expriment cependant l'espoir qu'une notion standard de calculabilité sur les réels puisse émerger à l'avenir (*op. cit.*, p.180). On trouvera encore un exemple de position similaire dans ([156], 278).

3. Nous verrons plus loin que la représentation des réels par leur expansion décimale est en réalité problématique pour l'analyse récursive : nous ne prenons donc cet exemple qu'à titre pédagogique.

un domaine dénombrable, mais n'admettent pas de représentation finitaire : les domaines des fonctions des réels calculables dans les réels calculables sont donc à représentation infinitaire. Nous emploierons l'expression « calculabilité sur les réels »comme une expression générique, servant à désigner non seulement la calculabilité sur tout domaine non-dénombrable, mais de façon plus générale encore sur tout domaine à représentation infinitaire. Par ce choix terminologique, on souligne avant tout que le problème de la représentation effective des données est le problème conceptuel fondamental de la calculabilité sur les réels : il affecte à ce titre tout modèle raisonnable de la calculabilité sur les réels. Nous allons tout d'abord voir comment le modèle le plus étudié de la calculabilité sur les réels, à savoir l'analyse récursive définie par A. Gregorzczyk et D. Lacombe, résout ce problème. Nous étudierons ensuite ce modèle et les différentes critiques qui lui sont couramment faites, avant de passer à l'étude des autres modèles de la calculabilité sur les réels.

3.1.1 L'analyse récursive et ses critiques

Définitions et propriétés élémentaires

Le premier modèle historique de la calculabilité sur les réels, et le plus employé, est l'analyse récursive, fondée par les travaux de A. Grzegorczyk et D. Lacombe ([118], [119],[158]). Cette approche de la calculabilité sur les réels se fonde sur un travail d'effectivation des concepts de l'analyse classique, en tout premier lieu des concepts de fonction réelle et de nombre réel[4]. Nous reproduisons ici la présentation de ces définitions par M. Pour-El et J. Richards ([194], 13).

Une suite r_k de nombres rationnels est *calculable* s'il existe trois fonctions récursives a, b, s de $\mathbb{N} \longrightarrow \mathbb{N}$ telles que, pour tout k, $b(k) \neq 0$ et

$$r(k) = (-1)^{s(k)} \frac{a(k)}{b(k)} \tag{3.1}$$

Une suite r_k de nombres rationnels *converge effectivement* vers un nombre réel x s'il existe une fonction récursive $e : \mathbb{N} \longrightarrow \mathbb{N}$ telle que pour tout N :

$$k \geq e(N) \qquad \text{implique} \qquad |r_k - x| \leq 2^{-N}. \tag{3.2}$$

Un nombre réel x est *calculable* s'il existe une suite calculable r_k de nombres rationnels qui converge effectivement vers x[5].

4. On trouve une telle description chez Ker-i Ko([153], 1) :

> Recursive analysis studies effective computability in classical analysis; that is, it studies which mathematical notions and proofs are computable and which are not computable.

5. Le lecteur mathématicien aura remarqué que cette définition constitue une effectivation d'une des définitions usuelles des réels comme limite de suite de rationnels. Toutes les

Pour comprendre le caractère essentiel de la condition de convergence effective, considérons la somme suivante [110] :

$$K = \sum_{n=1}^{\infty} \alpha_n 3^{-n} \tag{3.3}$$

avec α_n la n-ième machine de Turing dans une énumération effective des machines de Turing fixée. Cette somme est manifestement convergente, et définit donc un nombre réel compris entre 0 et 1. Ce réel n'est pas calculable, car il donnerait une procédure immédiate pour résoudre le problème de l'arrêt : il suffirait de lire l'approximation rationnelle de K à la précision $\epsilon = 3^{-(n+1)}$. Il existe néanmoins un algorithme produisant une suite croissante convergeant vers K. On considère une machine de Turing universelle possédant une bande de lecture contenant un entier n :

1. Fixer un entier n.
2. Simuler les n premières machines de Turing pendant n étapes.
3. Si α_m, avec $m \leq n$ s'arrête en n étapes, fixer $\alpha_m = 1$ et $\alpha_m = 0$ sinon.
4. Incrémenter n. Retourner à 1.

Il est ainsi aisé de produire une suite d'approximations rationnelles de K de précision croissante. Le réel K n'en reste pas moins non-calculable, parce qu'il est impossible de déterminer, pour tout rationnel r_n appartenant à cette suite, à quelle précision ϵ il approxime K : ϵ ne peut être écrit comme une fonction calculable de n. Si α_1 s'arrête au bout de 1000000 d'étapes, notre approximation s'accroîtra soudainement d'un tiers. Si la suite de rationnels est bien calculable par une procédure effective, la convergence de cette suite n'est pas effective : il n'existe pas d'algorithme nous donnant, pour toute précision ϵ, à quel point de la suite une erreur bornée par ϵ a été atteinte. Pour qu'un réel soit calculable, il ne suffit pas qu'il existe un algorithme produisant une suite d'approximations de précision croissante : il faut encore que la précision de l'approximation soit elle-même calculable.

L'exemple que nous venons de donner ne constitue nullement une exception perverse, mais est au contraire l'instanciation d'une propriété plus générale. Soit A un ensemble récursif : le réel r défini par

$$r = \sum_{i \in A} 2^{-i} \tag{3.4}$$

est calculable. Si A est récursivement énumérable, et que f est une fonction récursive telle que $A = rng(f)$, alors la somme définie par

autres définitions de la notion de « réel calculable » obtenues par effectivation d'une définition mathématique des nombres réels sont équivalentes, et c'est là un trait remarquable de l'analyse récursive.

$$r' = \sum_{j \in Dom f} 2^{-f(j)} \tag{3.5}$$

converge toujours vers un réel r', mais celui-ci n'est pas calculable. r' est néanmoins la limite d'une série croissante monotone, dont les termes peuvent être engendrés par une machine de Turing : on dit que r' est calculable à gauche (*left-computable*). Si la suite est décroissante, on dit que r' est calculable à droite (*right-computable*). Un réel r calculable à gauche et calculable à droite est calculable (pour plus de détails, voir [48]).

Une fonction $f :\subseteq \mathbb{R} \longmapsto \mathbb{R}$ est calculable ssi

- (i) f est *séquentiellement calculable* : si $i \mapsto x_i$ est une suite calculable dans $dom(f)$ convergeant vers $x \in Dom(f)$, alors la suite calculable $i \mapsto f(x_i)$ converge vers $f(x)$.
- (ii) f est *effectivement uniformément continue* : il existe une fonction récursive $d : \mathbb{N} \longmapsto \mathbb{N}$ telle que pour tout $x, y \in Dom(f)$, et pour tout $n \in \mathbb{N}$

$$|x - y| \leq \frac{1}{d(n)} \qquad \text{implique que} \qquad |f(x) - f(y)| \leq 2^{-n}. \tag{3.6}$$

En termes intuitifs, cette définition affirme qu'une fonction réelle f est calculable s'il existe un algorithme qui, à la donnée d'une suite d'approximations x du domaine de f, associe une suite d'approximations de $f(x)$, et que cette approximation est elle-même calculable. Cette définition peut être étendue à des fonctions de plusieurs variables, et à des fonctions définies sur tout $\mathbb{R}$ (voir [194]). On qualifiera les fonctions calculables au titre de l'analyse récursive de fonctions Grzegorczyk-calculables [6].

Cette définition induit deux propriétés remarquables des fonctions réelles calculables, que nous commenterons bientôt plus en détail. La première est que la définition n'impose pas que l'argument d'une fonction réelle calculable soit un réel calculable. La seconde est que toutes les fonctions réelles calculables sont continues.

Les définitions de Grzegorczyk et Lacombe des notions de réel et de fonction réelle ont l'avantage d'être naturelles pour un analyste, puisqu'elles consistent essentiellement en l'effectivation de définitions standard en analyse. Elles ont néanmoins l'inconvénient de ne pas expliciter le caractère syntaxique du calcul effectué, comme le fait la définition en termes de machines de Turing des fonctions calculables de $\Sigma^* \longmapsto \Sigma^*$. Ceci est réalisé dans

6. Cet emploi se distingue de l'usage le plus courant dans les communautés logicienne et informaticienne, où on les qualifie simplement de « fonctions calculables », tant la prégnance du modèle de l'analyse récursive est importante. Cet écart par rapport à l'usage nous a paru nécessaire, dans la mesure où nous comparons différents modèles de calculabilité sur les réels et que nous nous référons à des intuitions préthéoriques sur la notion de calculabilité, notamment de calculabilité effective : l'emploi de la notion de « fonction calculable » pour désigner les fonctions calculables au titre d'un modèle particulier aurait donc été source de confusion.

une définition équivalente de K. Weihrauch, les machines de Turing de type 2 [7]. L'examen de cette définition va nous permettre de comprendre en quelle mesure l'analyse récursive constitue une extension de la théorie de la calculabilité au calcul d'approximations successives d'un résultat n'admettant pas de représentation finitaire.

Pour $k \geq 0$ et $Y_0, Y_1, ..., Y_k \in \Sigma^*, \Sigma^\omega$, on définit les fonctions calculables $f :\subseteq Y_1 \times Y_2 \times ... \times Y_k \longmapsto Y_0$ à l'aide d'une machine de Turing munie de k bandes de lecture, un nombre fini de bandes de travail, et une unique bande d'écriture sans possibilité d'effacement (*one-way input tape*). La configuration initiale de la machine est définie de manière naturelle, chaque bande de lecture i contenant le mot fini ou infini Y_i, toute case de la bande ne contenant pas un caractère du nom de l'entrée contenant le symbole B, ainsi que toutes les cases de toutes les autres bandes [8]. En revanche, par contraste avec les fonctions calculables définies sur un domaine dénombrable, il faut distinguer entre le cas où le résultat Y_0 est finitairement représentable, et celui où il ne l'est pas. Plus rigoureusement, soit $f_M :\subseteq Y_1 \times Y_2 \times ... \times Y_k \longmapsto Y_0$ la fonction calculée par la machine de Turing M ci-dessus définie. Pour tout $y_0 \in Y_0, y_1 \in Y_1, ..., y_k \in Y_k$, on définit :

1. Cas $Y_0 = \Sigma^*$: $f_M(y_1, ..., y_k) := y_0 \in \Sigma^*$, ssi M s'arrête sur l'entrée $(y_1, ..., y_k)$ avec y_0 sur la bande d'écriture.
2. Cas $Y_0 = \Sigma^\omega$: $f_M(y_1, ..., y_k) := y_0 \in \Sigma^\omega$, ssi M calcule indéfiniment sur l'entrée $(y_1, ..., y_k)$ et écrit y_0 sur la bande d'écriture.

On appelle ce type de machines de Turing, en reprenant la terminologie de Weihrauch, *machines de Turing de Type 2.*

On dira d'une fonction $f :\subseteq Y_1 \times Y_2 \times ... \times Y_k \longmapsto Y_0$ qu'elle est calculable ssi il existe une machine M de Type-2 qui la calcule. On notera que $f_M(y_1, ..., y_k)$ n'est pas définie si la machine M calcule indéfiniment mais n'écrit qu'un nombre fini de caractères sur la bande d'écriture.

Une telle définition pose d'évidents problèmes d'effectivité. Pour qu'une procédure puisse être qualifiée d'effective, il est nécessaire que les entrées et sorties puissent être lues, écrites et éventuellement modifiées. Dans le cas des fonctions à domaine dénombrable, cette liste de propriétés étaient trivialement vérifiées, puisque les éléments du domaine sont finitairement représentables. En revanche, il est impossible d'écrire ou de lire un mot de longueur infinie. Pour pouvoir définir une notion de fonction effectivement calculable pour les domaines non-dénombrables, il est nécessaire d'assouplir

7. Un rationnel est un objet de type 0, une fonction rationnelle un objet de type 1, une fonctionnelle associant une fonction rationnelle à une fonction rationnelle un objet de type 2. Un réel calculable r peut être représenté par une fonction récursive associant à un entier n une approximation rationnelle r_n de r. Une fonction calculable $f :\subseteq \mathbb{R}^n \longmapsto \mathbb{R}$ peut donc être représentée comme une fonctionnelle associant de telles fonctions, donc comme un objet de type 2.

8. Pour plus de détails, voir ([241], 14-15).

l'exigence d'écrire une représentation finitaire *exacte* de l'entrée et de la sortie, pour lui substituer une idée de représentation approchée. On demande donc de la machine de type 2 qu'elle lise ou écrive une suite de préfixes de la chaîne infinie de caractères représentant l'élément considéré du domaine ou du codomaine. En outre, il n'est plus possible d'exiger qu'une procédure effective finisse en un nombre fini d'étapes, si elle est appliquée à un élément du domaine de la fonction considérée : le calcul d'une chaîne infinie de caractères est au contraire voué à se poursuivre indéfiniment ([241],15-16) :

> Clearly, infinite inputs or outputs do not exist and infinite computations cannot be finished in reality. But *finite* computations on *finite* initial parts of inputs producing *finite* initial parts of the outputs can be realized on physical devices as long as enough time and memory are available.(..) Of course, Type-2 machines can be simulated by digital computers. Therefore, infinite computations of Type-2 machines can be *approximated* by finite physical computations with arbitrary precision. The restriction to one-way output guarantees that any partial output of a finite initial part of a computation cannot be erased in the future and therefore, is final. For this reason, models of computation with two-ways output would not be very useful.

En filigrane de cette citation de K. Weihrauch, on voit que, pour être étendue aux objets n'admettant pas de représentation finitaire, la notion de correction du calcul doit être assouplie. Dans le cas des domaines finitairement représentables, la correction de l'écriture du résultat est définie de la manière naturelle. En se restreignant sans perte de généralité aux machines de Turing disposant d'une bande d'écriture sans possibilité d'effacement, on dit que la machine de Turing M calcule le résultat de l'évaluation d'une fonction f en une entrée $(x_1, ..., x_k) \in Domf$ ssi il existe un n tel que après n étapes de calcul, la bande d'écriture contient le nom fini w associé à $y = f(x_1, ..., x_k)$, et le symbole B sur toutes ses autres cases, et entre dans son état d'arrêt H.

Dans le cas d'une fonction de domaine n'admettant pas de représentation finitaire, ce critère doit être redéfinie pour le cas $Y_0 = \Sigma^\omega$. Soit $f :\subseteq Y_1 \times Y_2 \times ... \times Y_k \longmapsto Y_0$ une fonction calculable, $(y_1, ..., y_k)$ le nom d'une entrée appartenant au domaine de définition de f, et $y_0 = f(y_1, ..., y_k) \in \Sigma^\omega$. La machine de Turing M calcule la valeur y_0 de f en l'argument $(y_1, ..., y_k)$ ssi :

1. *L'écriture du résultat commence.* Il existe une étape de calcul k_0 telle que M écrit le premier symbole de y_0.
2. *L'écriture du résultat se poursuit.* Pour tout k, si M a écrit n symboles au terme de k étapes de calcul,il existe $k' \geq k$ tel que M écrive le $n+1$-ième symbole au terme de k' étapes.
3. *Écriture de préfix* ou *Pas de retour en arrière.* Pour tout $k \geq k_0$, la chaîne de caractère écrite sur la bande est un préfix du mot infini

représentant le résultat [9].

L'exigence d'avoir une bande d'écriture sans possibilité d'effacement est ainsi justifiée par une exigence plus fondamentale, à savoir celle d'avoir, à toute étape du calcul, un préfix correct de la sortie. Les deux premières conditions garantissent récursivement le caractère finitaire, en termes du nombre d'opérations exécutées, du calcul nécessaire pour écrire une suite de symboles finie. La troisième condition garantit le caractère significatif des deux premières conditions, en imposant que la suite finie de symboles soit un préfix du résultat du calcul. Ces conditions sur la correction du résultat ne sont pas triviales, en ce qu'elles contraignent l'ensemble des représentations admissibles des entrées et sorties. L'écriture de l'expansion décimale d'un réel n'est ainsi pas admissible, parce qu'elle ne permettrait pas de définir la multiplication d'un réel comme une fonction calculable, tout en respectant les critères de correction exprimés ci-dessus [10].

Une définition de la notion de « fonction effectivement calculable »qui n'admet pas comme fonctions calculables une opération arithmétique élémentaire comme la multiplication ne présente guère d'intérêt pratique ou épistémologique. Pour sauvegarder la définition de Grzegorczyk-Lacombe, il est donc nécessaire de restreindre les représentations admissibles des entrées et des sorties afin d'éviter la réduction par l'absurde démontrée dans le cas de l'expansion décimale.

Pour reprendre les termes de K. Weihrauch, la définition du calcul d'une fonction réelle en analyse récursive dépend d'une propriété de finitude (*finiteness property*) : *Toute portion finie de l'output $f_M(p)$ est déterminée univoquement par une portion finie de l'input p.* On remarquera simplement que cette propriété de finitude est impliquée par les conditions énoncées ci-dessus. Dans la suite de ce chapitre, je désignerai donc pas l'expression « propriété de finitude », ces trois conditions précises, et considérerait la proposition de Weihrauch comme une synthèse intuitive de ces conditions.

Au vu de ces dernières définitions, l'analyse récursive doit être comprise comme une extension de la notion de calcul effectif au calcul d'une suite

9. Nous exprimons ici une condition de correction qui n'est pas donnée par K. Weihrauch, mais qui nous semble implicite dans sa présentation des concepts, et dans l'usage qu'il en fait dans ses démonstrations.

10. Supposons par l'absurde qu'il existe une machine de Turing de Type 2 calculant la fonction réelle $x \mapsto 3.x$ en représentation décimale. Cette machine doit donc écrire le résultat correct en acceptant comme entrée le nom **0.333...** de $\frac{1}{3}$. La machine doit donc écrire en sortie l'un des deux noms **0.999...** ou **1.0000...** Considérons le premier cas. Par la condition 1 de l'écriture d'un résultat correct, il existe un nombre k d'étapes de calcul pendant lesquels M opère avant d'écrire le préfixe **0.** sur la bande d'écriture. En k étapes de calcul la machine M ne peut lire qu'un nombre fini de symboles de la bande de lecture, soit le préfixe **0.w**. Soit l'entrée **0.w999...** représentant un nombre réel $z > \frac{1}{3}$. Sur cette entrée la machine M commencera également par écrire le préfixe **0.**. Comme $3.z > 1$, la machine ne peut satisfaire la condition 3 (contradiction). Un argument similaire s'applique à l'autre choix de représentant de la sortie. Par conséquent, la machine M ne peut exister.

d'approximations d'un résultat n'admettant pas de représentation finitaire. Cette interprétation mène à un assouplissement de la condition de terminaison que nous avons énoncée plus haut. Celle-ci spécifiait qu'un calcul effectif devait s'achever par l'écriture de résultat après un nombre fini d'étapes de calcul. Cette condition ne peut évidemment être satisfaite par les calculs décrits par l'analyse récursive. Plutôt que d'affirmer que l'analyse récursive ne décrit pas des calculs effectifs, il nous semble plus naturel d'affirmer que la condition de terminaison doit être assouplie, pour être remplacée par les trois conditions de finitude décrites plus haut.

Deux arguments plaident en faveur de cette position. Le premier est naturellement que les calculs décrits par l'analyse récursive sont exécutés par des machines de Turing, et en tant que tels obéissent aux conditions énoncées par A. Turing pour un modèle du calcul papier-crayon : cardinalité finie de la signature et de l'ensemble des états, nombre de symboles et de déplacements finis par étape de calcul. Ils sont donc exécutables par un calcul papier-crayon ou, comme nous avons vu plus haut (voir section 2.2.1), par un calcul symbolique finitaire. Le second est que la propriété de finitude constitue une condition naturelle de l'effectivité d'un calcul : un calcul effectif ne requiert qu'une quantité finie d'information en entrée, et une quantité finie d'opérations, pour produire un résultat fini, ou un préfix fini du résultat en sortie. Il n'y a aucune raison plaidant pour une restriction de la notion d'effectivité au cas des résultats finitairement représentables. En assouplissant la condition de terminaison du calcul en un nombre d'étapes fini, pour la remplacer par la propriété de finitude, on étend naturellement la notion de procédure effective au calcul sur les réels.

Pour rendre cette extension de la notion de procédure effective parfaitement naturelle, il faut encore préciser un mode de donation des entrées à la machine de Turing de Type 2. La définition de K. Weihrauch suppose que le nom infini d'un réel soit donné comme entrée à la machine de Turing de type 2, afin qu'elle puisse calculer la valeur de la fonction en l'argument représenté par ce nom. Par la propriété de finitude de Weihrauch, le calcul ne nécessite à aucune étape la donnée d'une infinité d'information. Le modèle recourt donc à une idéalisation sans nécessité logique et physiquement irréaliste. Il est donc souhaitable d'ajouter au modèle un mode de donation de l'entrée, qui permette de modéliser l'intuition selon laquelle l'entrée peut être fournie à l'algorithme au fur et à mesure du calcul [11]. Ceci est réalisé

11. On peut ainsi lire sous la plume de Ker-i Ko ([153], 3) :

> Since x is a type-1 function that does not have a finite representation, machine M cannot directly "read" its input x. Instead, we must provide a more complicated mechanism to allow machine M to access the information about the real number x. In our computational model, we use the oracle machine to formalize the communication between the machine M and the input real number x.

par le modèle des machines de Turing à oracle de Ker-i Ko, où le mode de donation de l'entrée est modélisé par une suite de requêtes à un oracle.

Une machine de Turing à oracle M est une machine de Turing ordinaire, munie de bandes de lecture, d'une bande d'écriture sans possibilité d'effacement et d'une bande de travail. On y ajoute une bande de travail distinguée, la bande de requête (*query tape*), et un état distingué d'appel à l'oracle. L'oracle est une boite noire capable de fournir la valeur d'une fonction $\phi : \Sigma^* \longmapsto \Sigma^*$ pour un argument arbitraire, par un mécanisme qui n'est pas spécifié dans le modèle. Munie de son oracle, la machine M remplace la chaîne w écrite sur la bande de requête par $\phi(|w|)$ en une étape, à chaque fois qu'un état d'appel d'oracle a été atteint. Intuitivement, la fonction calculée par l'oracle est la fonction de type 1 permettant de produire une suite convergente d'approximations de l'entrée désirée. L'entrée peut ainsi être fournie à la machine à la précision exigée au fur et à mesure du calcul.

Une machine de Turing à oracle M calcule une fonction réelle en suivant la procédure suivante [12] (Ker-i Ko) :

1. L'entrée x de f est donnée à M comme un oracle.
2. La précision exigée sur la sortie, 2^{-n} est donnée sous la forme d'un entier n comme entrée à M.
3. M exécute le calcul en deux étapes :
   - i) M calcule, à partir de la précision exigée sur la sortie 2^{-n}, la précision exigée sur l'entrée 2^{-m} ;
   - ii)M consulte l'oracle pour obtenir $\phi(m)$, tel que $|\phi(m) - x| \leq 2^{-m}$, et calcule à partir de $\phi(m)$ une sortie d tel que $|d - f(x)| \leq 2^{-n}$.

Une fonction $f :\subseteq \mathbb{R} \longmapsto \mathbb{R}$ est calculable s'il existe une machine de Turing à oracle telle que pour toute entrée $n \in \mathbb{N}$ et pour toute représentation admissible ρ, la machine associe à tout ρ-nom de $x \in Dom(f)$ donné comme oracle, un ρ-nom y tel que $\|y - f(x)\| \leq 2^{-n}$.

L'oracle peut simplement être une sous-routine calculant le réel calculable pris comme argument. Il peut aussi s'agir d'une sous-routine calculant une suite de rationnels convergeant vers un réel non-calculable. On verra que le formalisme des oracles peut également être utilisé pour modéliser une mesure physique (voir section 6.5.1).

Une fois les machines de Turing de Type 2 formalisées comme des machines à oracles, on peut affirmer que le calcul d'une fonction $f : \Sigma^\omega \longmapsto \Sigma^\omega$ Grzegorczyk-calculable, bien qu'il fasse emploi de mathématiques infinitaires dans sa modélisation, ne requiert à aucun moment la lecture ou manipulation d'une infinité d'information.

12. La présentation rigoureuse de la définition de Ker-i Ko nécessiterait d'introduire le formalisme des fonctions de Cauchy, que Ker-i Ko emploie pour des raisons liées à la définition de la complexité en analyse récursive. Puisque l'introduction de ce formalisme serait sans intérêt pour nos préoccupations présentes, nous nous contenterons d'une présentation informelle.

Critiques de l'analyse récursive : la continuité des fonctions Grzegorczyk-calculables

Avant d'aborder les problèmes soulevés par la comparaison de l'analyse récursive et des autres modèles de calculabilité sur les réels, il nous faut étudier une critique spécifique de l'analyse récursive. Pour en donner une première présentation intuitive, cette critique consiste en un reproche de manque de naturalité : la définition de la calculabilité au titre de l'analyse récursive exclurait certaines fonctions intuitivement calculables. Nous avons vu plus haut que la calculabilité sur les fonctions entières ne souffrait pas un tel reproche : toutes les fonctions entières intuitivement calculables sont calculables par une machine de Turing. Si un tel argument se révélait être bien défini et justifié, cela constituerait donc un problème spécifique de l'analyse récursive.

La critique prend sa source dans une propriété forte des fonctions Grzegorczyk-calculables : toute fonction calculable est continue. Cette propriété de continuité exclut la calculabilité de certaines fonctions réelles simples, comme la fonction échelon, $s(x) = c$ si $x < x_0$, et d sinon, avec c, d et x_0 des réels calculables, et la fonction partie entière, qui sont évidemment discontinues.

Le caractère bien posé de cette objection est cependant discutable : en quel sens ces fonctions sont-elles intuitivement calculables ? Plusieurs raisons peuvent être évoquées pour soutenir ce point de vue. En sus d'être bien définies, le graphe de ces fonctions est d'un tracé facile. En outre, la calculabilité de ces fonctions peut sembler intuitive, dans la mesure où il est aisé de calculer leurs valeurs pour de très nombreux exemples : peu de fonctions sont aussi simples à calculer que la fonction partie entière. De tels arguments intuitifs manquent cependant la difficulté propre à la calculabilité sur un domaine non-finitairement représentable. L'égalité à un réel calculable est triviale à décider dans une infinité de cas, mais cela n'empêche pas qu'elle ne soit pas décidable au titre de l'analyse récursive, parce qu'elle nécessiterait une infinité d'information. De manière analogue, le calcul des valeurs d'une fonction discontinue au voisinage d'un point de discontinuité x nécessite de distinguer une valeur égale à x d'une valeur x' arbitrairement proche de x pour déterminer le premier symbole du nom de $f(x')$.

Au-delà d'intuitions discutables, la critique de manque de naturalité peut s'enraciner dans la possibilité d'appliquer des notions de calculabilité plus faibles à ces fonctions.

Comme le remarque K. Weihrauch ([241], 7, 121), la fonction partie entière, si elle n'est pas Grzegorczyk-calculable, a la propriété remarquable suivante : il existe un algorithme permettant d'associer à toute suite de rationnels convergeant effectivement vers x une suite croissante de rationnels convergeant vers $f(x)$. Les valeurs de la fonction sont donc calculables à gauche. La même intuition qui peut, de prime abord, pousser à considérer

comme calculable un réel pour lequel on peut produire une suite d'approximations, ignorant ainsi la condition de convergence effective, peut pousser à qualifier de calculable une fonction comme la fonction partie entière.

Il serait byzantin de chercher à déterminer s'il est plus intuitif de parler de calculabilité avec ou sans la propriété de convergence effective. Comme le remarque J. Earman, la définition d'un réel calculable provoque une scission des intuitions : il est intuitif d'appeler « calculable »un réel dont on peut produire une suite d'approximations, mais il est également intuitif de penser qu'on doit être capable de déterminer la précision à laquelle on approxime une valeur. La correction du modèle de l'analyse récursive ne peut être appuyée par de telles intuitions conflictuelles, ni être réfutée par elles.

Si la continuité des fonctions Grzegorczyk-calculables ne peut être vue comme un défaut de l'analyse récursive, c'est parce qu'elle découle, comme le remarque justement K. Weihrauch [13], de la propriété de finitude. Or, comme nous venons de le voir, cette propriété est essentielle pour généraliser nos intuitions fondamentales sur l'effectivité à des objets n'admettant pas de représentation finitaire. La continuité ne peut donc être vue comme une propriété contingente du modèle de l'analyse récursive, dont une variante pourrait et devrait se passer, mais comme une conséquence naturelle de la généralisation de l'approche effective aux objets n'admettant pas de représentation finitaire.

3.1.2 La définition de Markov et les fonctions des réels calculables dans les réels calculables

La présente section vise essentiellement à la complétude doxographique, et peut aisément être sautée en première lecture. Elle porte sur une autre extension de la calculabilité effective aux réels, limitée quant à elle aux réels calculables : la calculabilité de Markov [14].

Soit $\mathbb{R}_c$ l'ensemble des réels calculables, et une représentation fixée $\rho : \mathbb{R}_c \longmapsto \Sigma^\omega$. Une fonction $f :\subseteq \mathbb{R}_c \longmapsto \mathbb{R}_c$ est *Markov-calculable* si et seulement s'il existe une machine de Turing M calculant la fonction $g :\subseteq \mathbb{N} \longmapsto \mathbb{N}$ qui associe à tout code d'une machine de Turing calculant le ρ-nom d'un réel $x \in Dom(f)$ le code d'une machine de Turing calculant le ρ-nom de $f(x)$. La fonction récursive g doit satisfaire un critère de correction naturelle : si n et m sont les codes de machines de Turing calculant une même fonction f, alors $g(n) = g(m)$.

La restriction aux réels calculables d'une fonction Grzegorczyk-calculable est Markov-calculable. Il suffit de faire exécuter par une machine de Turing universelle la machine calculant l'entrée x, et d'utiliser la suite d'approxi-

13. Pour la démonstration, voir ([241], 6, 30).

14. Il ne s'agit bien sûr pas là de l'auteur des chaînes du même nom, qui vécut à une époque (1856-1922) bien antérieure aux commencements de la théorie de la calculabilité, mais de son fils unique, Andreï Andreïevitch Markov (1903-1979).

mations obtenues pour calculer $f(x)$, pour obtenir un algorithme de Markov. En revanche, il n'est pas évident que toute fonction Markov-calculable puisse être vue comme la restriction d'une fonction Grzegorczyk-calculable aux réels calculables. Si l'on conçoit un réel calculable comme une fonction rationnelle, la calculabilité de Markov demande en entrée une description de toute la fonction, par le biais d'un algorithme, tandis que la calculabilité de Grzegorczyk ne requiert que quelques approximations de l'entrée pour calculer quelques approximations de la sortie(pour plus de détails, voir ([157], 304-305). Il n'est donc pas surprenant que la converse soit fausse en toute généralité, même si elle peut être vraie si certaines conditions sont satisfaites sur le domaine de la fonction.

L'ensemble des réels calculables forme un ensemble dénombrable, puisque chaque réel calculable est associé à au moins une machine de Turing, et que l'ensemble des codes de machines de Turing est récursivement énumérable. Comme nous l'avons déjà souligné en introduction, l'existence d'une représentation finitaire des fonctions et la cardinalité dénombrable de leur domaine sont deux questions distinctes. Du point de vue de la théorie de la calculabilité, la différence essentielle entre fonctions des réels dans les réels et fonctions des réels calculables dans les réels calculables n'est pas tant la cardinalité de leur domaine, que la possibilité d'engendrer la représentation de leurs éléments par une procédure effective.

Formulée par le fondateur de l'école russe d'analyse constructive, cette définition est inspirée par une philosophie intuitionniste, qui exclut des objets non-constructifs comme les réels encodant un oracle non-récursif. Une approche intuitionniste de la calculabilité n'a pas vocation à être démontrée équivalente à une approche classique, ni même à constituer un sous-domaine bien défini de l'approche classique. Par conséquent, il n'est guère surprenant que l'analyse récursive et l'analyse constructive à la Markov ne constitue pas des modèles équivalents, et aucun auteur, à notre connaissance, ne le pose comme problème. La calculabilité à la Markov ne constitue donc pas un enjeu dans une discussion de la thèse de Church-Turing sur les réels.

3.2 Les modèles analogiques

Nous allons à présent examiner les modèles de calculabilité sur les réels dits *analogiques*. La notion de *modèle analogique* n'admet pas de définition générale rigoureuse. Elle désigne de manière générique un modèle où un ou plusieurs paramètres sont décrits par des variables continues, qu'il s'agisse du temps, de l'espace ou des états du processeur. L'une des propriétés fondamentales de ce type de modèles est que les nombres réels n'y sont plus considérés comme des chaînes de caractères, mais comme des quantités en elles-mêmes [15].

15. Ce dernier point est bien souligné par C. Moore dans ([171],1, nous soulignons) :

Ces modèles sont d'une nature différente de ceux que nous venons de présenter et de discuter [16]. Comme nous venons de le voir, l'analyse récursive, comme la calculabilité de Markov, constituent des formes d'extension de la notion de procédure effective au calcul d'approximations d'un objet n'admettant pas de représentation finitaire. La notion de procédure effective réfère à un processus de calcul que nous avons qualifié de *symbolique* et *finitaire*. Le calcul est *symbolique* au sens où

1. Entrées et sorties sont représentées par des chaînes de caractères sur un alphabet fini Σ.
2. L'exécution du calcul consiste en la lecture et la modification de chaînes de caractères.

Le calcul effectué selon une procédure effective est ensuite *finitaire*, au sens où, durant toute étape de son exécution concrète, les chaînes de caractères retenues en mémoire par le calcul sont de longueur finie, et les opérations effectuées en une étape obéissent aux conditions de finitude définies par A. Turing. Une telle conception symbolique et finitaire du calcul évoque naturellement la pratique historique du calcul par un homme muni de papier et d'un crayon.

En revanche, les modèles analogiques de calculabilité sur les réels ne décrivent en aucune façon un calcul symbolique finitaire : l'exécution du calcul n'y consiste pas en la lecture et manipulation de chaînes de caractères finies. Ils ne sont donc pas exécutables en tant que tels par un être humain muni de papier et d'un crayon. L'évolution dynamique d'une machine analogique ne peut être vue comme une implémentation des étapes d'une procédure effective, bien qu'elle implémente une fonction. Afin de mieux voir ce dernier point, prenons un exemple. Pour des raisons de complexité, on a pu ainsi choisir d'utiliser directement, pour calculer les solutions de l'équation de Navier-Stokes, des systèmes hydrodynamiques gouvernés par cette équation, comme des souffleries (voir [95], 11-12). On peut donc dire qu'une ligne de courant implémente le calcul d'une solution de l'équation de Navier- Stokes. Néanmoins, il serait vain de chercher à distinguer dans une ligne de courant

---

> (...) to discuss the physical world (or at least its classical limit) in which the states of things are described by real numbers and processes take place in continuous time, we need a different theory : a theory of analog computation, where states and processes are inherently continuous, and *which treats real numbers not as sequences of digits but as quantities in themselves.*

Le même point de vue est exprimé par L. Blum, M. Shub et S. Snale ([34], 3) :

> (...) we view a real number not as its decimal (or binary) expansion, but rather a mathematical entity as is generally the practice in numerical analysis.

16. Dans la littérature informatique, on emploie fréquemment les expressions « modèle analogique »et « modèle de calcul sur les réels »de manière interchangeable. La reprise de cet usage aurait cependant été dommageable à la rédaction de ce paragraphe, dans la mesure où il efface les distinctions conceptuelles que nous essayons ici de souligner.

une étape où elle calcule une dérivée partielle par rapport au temps, ou la divergence d'une quantité donnée. Il serait encore plus vain de se demander selon quelles instructions d'une machine de Turing elle effectue ces calculs.

Puisque ces modèles ne visent pas à formaliser le calcul symbolique finitaire, l'éventuelle démonstration de leur équivalence avec l'analyse récursive prend un sens nouveau. Comme nous l'avons déjà souligné (section 1.1), les premiers modèles historiques du calcul visaient tous à capturer la notion de « fonction calculable par une procédure effective ». La démonstration de leur équivalence signifiait donc que ces modèles constituent des formalisations équivalentes d'une même notion de calcul intuitive. Les modèles analogiques et l'analyse récursive, *a contrario*, constituent deux classes de modèles animés par des visées théoriques distinctes : ils ne visent pas à formaliser la même notion intuitive de calcul. La démonstration de l'équivalence d'un modèle analogique et de l'analyse récursive porte donc sur la capacité d'une procédure effective à effectuer une simulation entrées-sorties des calculs effectués par un modèle analogique, et réciproquement. On montrerait ainsi que deux conceptualisations intuitivement distinctes du calcul aboutissent à un pouvoir expressif identique.

Les problèmes de la calculabilité sur les réels que nous avons rencontré jusqu'à présent pouvaient être décrits comme des problèmes internes à des tentatives d'extension de la notion de calcul effectif aux domaines à représentation infinitaire. Les difficultés rencontrées avec les modèles analogiques sont d'une autre nature : il s'agit des difficultés rencontrées à montrer l'équivalence, en termes de pouvoir expressif, entre le calcul effectif sur les réels et une conception différente du calcul sur les réels.

3.2.1 Le G.P.A.C.

Le General Purpose Analog Computer (G.P.A.C.) est le modèle analogique le plus connu. Il fut conçu durant les années 1930 par C. Shannon comme une modélisation et une généralisation des calculateurs analogiques existants à cette époque, notamment le *Differential Analyzer* mis au point par V. Bush. Bien qu'il ait dû faire l'objet de plusieurs retouches théoriques par la suite [211], il est toujours considéré comme un modèle robuste des calculateurs analogiques existants.

Un G.P.A.C. est intuitivement constitué par un circuit composé de boites noires reliées entre elles, appelées « unités analogigues »(*analog units*). Chacune de ces unités calcule une fonction primitive, dont les entrées sont paramétrisées par une variable continue. Chacune de ses boites noires est moralement conçue pour être implémentée par un dispositif physique modélisé par un formalisme continu, dont les entrées sont des quantités continues paramétrisées par le temps réel. À l'aide d'un amplificateur opérationnel, d'un condensateur et d'une résistance, on peut ainsi bâtir un montage simple tel que la tension en sortie $V(t)$ soit égale à l'intégrale de la tension en entrée

$U(t) : V(t) = \frac{-1}{RC} \int_0^t U(t)dt$. Deux mesures de la tension nous permettent donc de réaliser une intégration. Le G.P.A.C. est constitué de quatre unités réalisant les opérations primitives suivantes : la production d'une constante réelle k, l'addition, la multiplication, et l'intégration de deux valeurs u et v [17]. C. Shannon a démontré que ces unités permettaient le calcul de nombreuses fonctions usuelles, comme les polynômes, l'exponentielle, les fonctions trigonométriques et leurs inverses. Le pouvoir expressif du G.P.A.C., d'après les premiers résultats de C. Shannon, serait cependant strictement inférieur à celui de l'analyse récursive. C. Shannon avait en effet démontré que le G.P.A.C. pouvait calculer l'ensemble des fonctions différentiellement algébriques [18]. La fonction $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$ est Grzegorczyk-calculable, mais elle n'est pas différentielle algébrique : elle n'est donc pas générable par un G.P.A.C.. Cette première estimation du pouvoir expressif du G.P.A.C. est à la base de la croyance courante, selon laquelle les différents modèles de la calculabilité sur les réels ne sont pas équivalents.

Cette estimation du pouvoir expressif du G.P.A.C. a cependant été critiquée dans un article récent [42]. Elle impose en effet de calculer une fonction $f(t, x)$ en temps réel t, une contrainte à laquelle l'analyse récursive n'est nullement soumise. O. Bournez, M. Campagnolo, D. Graça et E. Hainry ont donc proposé de remplacer la première définition de la calculabilité au titre du G.P.A.C. par une notion de calcul approché, qui rende la comparaison avec l'analyse récursive plus naturelle.

Pour ce faire, ils ont employé une caractérisation alternative du pouvoir expressif du G.P.A.C., due à D. Graça et J.F. Costa [211]. Une fonction y est générable par un G.P.A.C. ssi elle est composante de la solution $y = (y_1, ..., y_n)$ de l'équation différentielle $y' = p(y, t)$ où p est un vecteur de polynômes.

Une fonction $f :\subseteq \mathbb{R}^n \longmapsto \mathbb{R}$ est calculable par le G.P.A.C. [19] par approximations s'il existe une équation différentielle ordinaire polynômiale avec n composantes $y_1, ..., y_n$ admettant les conditions initiales $x_1, ..., x_n$ telle que, pour les composantes particulières g et ϵ, on ait $\lim_{t\to\infty} \epsilon(x_1, ..., x_n, t) = 0$ et $|||f(x_1, ..., x_n, t) - g(x_1, ..., x_n, t)||| \leq \epsilon(x_1, ..., x_n, t)$.

En termes intuitifs, on considère un système dynamique $y' = p(y, t)$ avec la condition initiale x. Pour tout x, la composante g du système approche $f(x)$ avec une approximation bornée supérieurement par une autre composante ϵ qui s'annule à l'infini.

17. L'unité d'intégration associe aux valeurs d'entrée u et v, fonctions du temps, la sortie w avec $w(t) = u(t)v'(t)$ et $w(t_0) = \alpha$.

18. Une fonction $f(x)$ est différentielle algébrique ssi ses dérivées satisfont une équation polynômiale à coefficients rationnels $P(x, f(x), f'(x), ..., f^k(x)) = 0$.

19. Nous reprenons ici la distinction employée par Bournez et alii entre « fonction générable par un G.P.A.C. », qui désigne la première conception de la calculabilité au titre de ce modèle, et « fonction calculable par un G.P.A.C. »ou « fonction G.P.A.C.-calculable », qui désigne leur propre conception du calcul par approximations.

Avec cette définition de la G.P.A.C.-calculabilité, Bournez et alii ont montré que l'analyse récursive et le G.P.A.C. sont des modèles équivalents.

Ce résultat appelle plusieurs commentaires. Dans le modèle original de Shannon, aucune condition n'est imposée sur les constantes et les conditions initiales de l'équation différentielle ordinaire considérée : celles-ci peuvent être décrites par des réels non Grzegorczyk-calculables. Dans ce cas de figure, il est très aisé de démontrer que le G.P.A.C. permet le calcul de fonctions non-Grzegorczyk-calculables. Bournez et alii ont donc choisi de restreindre l'étude du pouvoir expressif du G.P.A.C. au cas où les conditions initiales et constantes sont Grzegorczyk-calculables.

Une fonction $f : [a, b] \longmapsto \mathbb{R}$ est G.P.A.C.-calculable ssi il existe des polynômes calculables $p : \mathbb{R}^{n+1} \longmapsto \mathbb{R}^n$; $p_0 : \mathbb{R} \longmapsto \mathbb{R}$ et $n-1$ valeurs réelles Grzegorczyk-calculables $\alpha_1, ..., \alpha_{n-1}$ telles que

1. $(y_1, ..., y_n)$ est la solution du problème de Cauchy $y' = p(y, t)$ avec les conditions initiales $\alpha_1, ..., \alpha_{n-1}, p_0(x)$ initialisé au temps $t_0 = 0$.
2. Il y a $i, j \in \{1, ..., n\}$ telles que $\lim_{t\to\infty} y_j(t) = 0$ et $|f(x) - y_i(t)| \leq y_j(t)$ pour tout $x \in [a, b]$ et tout $t \in [0, +\infty)$.

.

En termes intuitifs, le travail de Bournez et alii démontrent que le G.P.A.C. et l'analyse récursive sont des modèles équivalents, à deux conditions : bien définir la calculabilité au titre du G.P.A.C. afin d'effectuer une comparaison sensée, et considérer des conditions initiales calculables. Si la première condition est parfaitement naturelle, la seconde manque en l'état de justification théorique fondamentale. Pourquoi l'état initial du dispositif, modélisé comme un système dynamique, qui sert à l'exécution d'un calcul analogique serait-il nécessairement descriptible par des valeurs réelles calculables ? Par cette question se trouvent introduites, dans l'évaluation du pouvoir expressif d'un modèle analogique, des considérations empiriques, portant sur notre capacité à préparer un système dans un état donné.

Un point similaire a été soulevé par D. Graça et F. Costa dans leur étude critique du G.P.A.C.. Le modèle ne suppose à l'origine aucune contrainte sur les fonctions continues du temps réel pris comme entrées par les unités analogiques. Cependant, la bonne définition de certaines fonctions exigent la différentiabilité continue des fonctions prises en entrée ([211], 8, nous soulignons) :

> (...) from now on, we will always assume that the inputs are continuously differentiable functions of the time. And if the outputs of all units are defined for all $t \in I$; where I is an interval, then we will also assume that they are continuous in that interval. This is needed for the following results and *may be seen as physical constraints to which all units are subjected.*

On remarquera que de tels problèmes ne se posent guère pour l'analyse récursive. L'étude des modèles analogiques introduit donc des considérations

d'un type nouveau dans l'évaluation du pouvoir calculatoire d'un modèle, dont nous allons trouver d'autres exemples dans les sections à venir.

3.2.2 Les modèles B.S.S. et RAM réelle

Il existe d'autres modèles analogiques, qui aboutissent à une évaluation différente de la calculabilité sur les réels : ils permettent de calculer des fonctions non-Grzegorczyk-calculables. Mais à la différence du G.P.A.C., dont l'implémentabilité n'est pas problématique, ces modèles ne sont pas implémentables. Leur discussion et critique exigent donc la prise en compte de considérations empiriques, que nous ne pourrons pleinement développer que lorsque nous aborderons la thèse de Church-Turing physique et les modèles d'hypercalcul (voir chapitre 6). Il est cependant nécessaire d'évoquer dès à présent ces modèles, pour permettre une discussion complète des problèmes liés à la non-équivalence des modèles en calculabilité réelle. Nous allons donc ici présenter les modèles les plus abstraits, qui nous permettront de repousser à plus tard l'examen de détail des considérations empiriques : le modèle Blum-Shub-Snale, dit modèle B.S.S., le modèle RAM réelle (*real-RAM model*), et le modèle des fonctions $\mathbb{R}$-récursives de C. Moore.

Au-delà de leurs différences, qui sont sans pertinence pour notre présent sujet, les modèles B.S.S. et RAM réelle partagent certains caractéristiques remarquables : nous allons donc commencer par leur examen.

Le modèle RAM réelle tout comme le modèle B.S.S. peuvent être présentés comme des *flowcharts*[20] agissant sur des suites de registres $N_0, ..., N_1$ pour nombres entiers et une suite de registres $R_0, ..., R_1$ pour nombres réels. Le lecteur notera que chaque registre n'est pas censé contenir une approximation du réel, mais bien le nombre réel lui-même, soit une infinité actuelle d'information. Le modèle autorise ensuite les opérations arithmétiques élémentaires sur ces réels, normalement à coût unitaire. Il permet aussi des embranchements conditionnés sur la comparaison entre deux réels contenus dans deux registres R_i et R_j. Ces modèles supposent donc d'emblée la possibilité d'une opération impossible au titre de l'analyse récursive, où la comparaison de deux nombres réels est semi-décidable. Puisque cette opération est primitive, ils ne donnent aucune indication sur la manière dont une telle opération pourrait être exécutée.

Toute comparaison du pouvoir expressif de ces modèles est difficile, en ce qu'il n'existe pas de relation d'inclusion entre les fonctions calculables au titre du modèle RAM réelle et les fonctions Grzegorczyk-calculables. Le modèle RAM réelle permet le calcul de l'escalier de Gauss, mais ne permet pas le calcul de fonctions simples comme la racine carré ou l'exponentielle. Les

20. Un *flowchart* est une présentation diagrammatique d'une procédure : le terme n'a pas à notre connaissance de traduction standard en français. La présentation en *flowchart* n'est adaptée qu'à une procédure en temps discret. Les modèles B.S.S. et RAM réelle ont ainsi la particularité d'être des modèles à temps discret et à espace continu.

propriétés globales des fonctions calculables au titre de ces modèles sont également très différentes. Comme nous l'avons vu, les fonctions Grzegorczyk-calculables sont continues, tandis que la plupart des fonctions RAM réelle calculables sont discontinues, chaque embranchement $R_i < R_j$ non-trivial provoquant un point de discontinuité. Enfin, il est impossible de permettre au modèle RAM réelle de calculer toutes les fonctions calculables au titre de l'analyse récursive simplement en ajoutant un ensemble fini de fonctions primitives, comme l'exponentielle et la racine carrée. Il demeurera toujours des fonctions Grzegorczyk-calculables qui ne seront pas calculables au titre du modèle RAM réelle [21].

Il n'est cependant pas évident de considérer que l'absence d'équivalence entre les modèles à RAM réelle et B.S.S. d'une part, et l'analyse récursive d'autre part, constituent de graves problèmes conceptuels pour la théorie de la calculabilité sur les réels, voir le signe d'une impossibilité d'une généralisation de la thèse de Church-Turing aux modèles à domaine non-dénombrable. Pour qu'une comparaison entre deux modèles puisse poser problème, il faut encore comparer ce qui est comparable. Il n'y a aucune comparaison à faire entre le modèle RAM réelle et B.S.S. et l'analyse récursive, en ce sens que tous les modèles permettant le stockage et la manipulation d'un nombre réel ne constituent pas d'authentiques modèles de calcul. Les raisons justifiant cette prise de position ne pourront devenir pleinement explicites qu'au moment de notre discussion sur les conditions pour un modèle raisonnable, et les problèmes soulevés par les modèles d'hypercalcul (voir chapitre 6, notamment section 6.2). Les problèmes immédiats posés par de tels modèles sont bien résumés par la citation suivante de K. Weihrauch ([241], 262) :

> Real-RAMs cannot be realized by physical machines, that is, they are unrealistic, for the following reason : in a finite amount of time every physical information channel can transfer only finitely many bits of information and every physical memory is finite. Since they are uncountably many real numbers, it is impossible to identify an arbitrary real number by a finite amount of information. Therefore, it is impossible to transfer an arbitrary real number to or from a computer in a finite amount of time or to store a real number in a computer.

Notre compréhension intuitive de l'effectivité, et du fonctionnement d'une machine de calcul concrète, postule que le stockage et la manipulation de quantité infinie d'information sont impossibles [22]. Face à cette impossibilité, les modèles RAM réelle et B.S.S. décrivent des calculs qui ne peuvent être

21. Pour plus de détails, voir ([241], 260-265).

22. Une position similaire sur le modèle à RAM réelle est exprimée par Ker-i Ko ([153], 5) :

> (...) it is apparent that no physical implementation of this model is possible.

exécutés, et ne permettent donc pas d'accéder à la connaissance de leur résultat. Tout modèle de calcul violant cette impossibilité supposée doit être compris soit comme un modèle notionnel ne décrivant aucune procédure exécutable dans la pratique, soit comme un modèle de portée révolutionnaire, voué à nous emmener au-delà des capacités de la machine de Turing. La première branche de l'alternative ne fait pas partie de notre objet d'étude ; la seconde constituera l'objet du chapitre 6. Ceci n'implique aucunement que ces modèles soient dépourvus d'intérêt : ils peuvent être employés à profit dans l'étude des calculs sur les réels en représentation à virgule flottante, ou dans l'étude de certains problèmes de complexité (voir ([241], 262-264), et [33]).

3.2.3 Les fonctions $\mathbb{R}$-récursives de Moore

Les fonctions $\mathbb{R}$-récursives de C. Moore ont été conçues comme une extension aux fonctions réelles de la définition récursive des fonctions calculables[23].

Une fonction $\mathbb{R}^m \longmapsto \mathbb{R}^n$ est $\mathbb{R}$-récursive si elle peut être engendrée à partir des fonctions constantes 0 et 1, à l'aide des opérateurs suivants : si f et g sont définies, alors la fonction h est bien définie par les opérations suivantes :

1. *Composition.* $h(\vec{x}) = f(g(\vec{x}))$.
2. *Récursion différentielle* ou *Intégration.* $h(\vec{x}, 0) = f(\vec{x}), \partial_y h(\vec{x}, y) = g(\vec{x}, y, h(\vec{x}, y))$. De manière équivalente, on peut écrire

$$h(\vec{x}, y) = f(\vec{x}) + \int_0^y g(\vec{x}, y', h(\vec{x}, y'))\mathrm{d}y' \tag{3.7}$$

3. *μ-récursion* ou *Détection du zéro.* $h(\vec{x}) = \mu_y f(\vec{x}, y) = inf\{y | h(\vec{x}, y) = 0\}$, avec inf sélectionnant le y de plus petite valeur absolue, et, s'il existe deux y de même valeur absolue, la valeur négative par convention.
4. Les fonctions vectorielles sont définies ssi leurs composantes sont définies.

L'intégration est un analogue naturel de la récursion primitive : au lieu de définir $h(\vec{x}, y+1)$ en fonction de $h(y)$, y et $\vec{x}$, on définit $\partial h/\partial y$ en termes de $h(y)$, y et $\vec{x}$. La fonction solution peut ne pas être unique, et elle peut également diverger. Par simplicité, on suppose donc que h n'est définie que lorsque l'équation admet une solution unique et finie incluant le point $h(\vec{x}, 0) = f(\vec{x})$.

Tout comme dans le cas partiel récursif, l'opérateur μ a pour fonction de détecter la plus petite valeur de y telle que $h(\vec{x}, y) = 0$. L'existence de réels négatifs impose cependant de modifier la définition de manière adaptée :

23. Le passage qui suit est un résumé de ([171], 4-5). Le lecteur avide de détails pourra consulter avec profit l'article original, d'une grande qualité d'exposition.

on ordonne donc les réels par leur valeur absolue, et, pour deux valeurs de valeur absolue égale, on choisit la valeur négative par convention. En outre, si une infinité de zéros s'accumulent au-dessus d'une valeur y, l'opérateur μ choisit cette valeur, même si elle n'est pas elle-même un zéro. De manière équivalente, si $[a, b]$ est le plus grand intervalle fermé contenant 0 tel que $f(y) \neq 0$, μ retourne la plus petite valeur absolue d'entre a et b, ou, si $a = -b$ la valeur négative entre a et b.

C. Moore a également choisi d'inclure dans ses cas de base les fonctions vectorielles. Cela n'est pas nécessaire pour les fonctions récursives partielles, puisqu'un vecteur d'entiers naturels peut être encodé par un entier. S'il existe des fonctions $\mathbb{R}$-récursives de $\mathbb{R}^n$ dans $\mathbb{R}$, elles ne sont pas différentiables, ce qui interfère avec l'opération d'intégration.

À partir de ces définitions, il est aisé de montrer que l'équivalent du dernier cas de base des fonctions récursives partielles, à savoir les projections, est définissable. De nombreuses fonctions ordinaires sont également définissables, comme les opérations arithmétiques élémentaires, l'exponentielle et le logarithme, les fonctions trigonométriques usuelles, x mod y. On peut également définir une nouvelle notion de réel calculable :

Un nombre $x \in \mathbb{R}$ est $\mathbb{R}$-récursif ssi il existe une fonction $\mathbb{R}$-récursive f telle que $x = f(0)$.

La notion de calculabilité ici employée est totalement différente de celle imaginée par A. Turing, et employée couramment en analyse récursive. Un réel calculable n'est pas un nombre dont la représentation infinie peut être séquentiellement engendrée par un algorithme avec une précision calculable, mais une quantité exacte, engendrée par un processus intrinsèquement analogique. Il n'est donc guère surprenant que certains nombres $\mathbb{R}$-récursifs ne soient pas Grzegorczyk-calculables.

L'ensemble des fonctions $\mathbb{R}$-récursives n'est pas non plus équivalent à l'ensemble des fonctions Grzegorczyk-calculables. On peut en effet trouver parmi les fonctions $\mathbb{R}$-récursives des fonctions discontinues comme la fonction δ de Kronecker, la fonction valeur absolue et la fonction échelon. On remarque que la définition d'une fonction $\mathbb{R}$-récursive discontinue nécessite l'emploi de l'opérateur μ.

Le travail original de C. Moore souffre de nombreuses difficultés mathématiques, touchant à la bonne fondation et la bonne définition de certaines de ses constructions, et des classes de fonctions définies grâce à elles[24] (voir [173], [172], [53], [54] pour différentes discussions de ces problèmes). Nous nous discuterons pas ici de ces difficultés, et nous nous pencherons plutôt sur les divers traits du modèle, signalés explicitement par C. Moore, qui mènent à douter de son caractère réaliste :

1. *Robustesse au bruit.* L'opérateur μ permet de détecter n'importe quel

24. Nous tenons à remercier O. Bournez pour nous avoir signalé l'existence de ces différents problèmes.

zéro d'une fonction, y compris si un tel zéro est isolé et crée une discontinuité, comme pour une fonction constante partout sauf en une unique valeur y, où elle prend la valeur 0 . L'implémentation d'un tel opérateur semble à première vue empiriquement impossible, puisqu'elle nécessiterait une protection absolue contre tout bruit.

2. *Nombre infini d'appels d'une fonction.* C. Moore ([171], 9) remarque que le calcul d'une intégration $h = f \int g$ nécessite l'exécution d'une boucle for : for $y' = 0$ to y do $\partial_y h = g(\vec{x}, y, h(\vec{x}, y))$. Le calcul de $h = \mu_y f$ nécessite l'exécution d'une boucle while : while $f(\vec{x}, y) \neq 0$ and $f(\vec{x}, -y) \neq 0$ do $\partial_y f = 1$ if $f(\vec{x}, -y) = 0$, then return(-y) else return(y). Comme dans le cas discret usuel, s'il n'existe pas de telle valeur y, le programme ne sort jamais de la boucle while et h n'est pas définie.

   Pour leur exécution, les boucles for et while nécessitent un nombre d'appel infini indénombrable des fonctions g et f. Si le temps d'exécution de ces fonctions n'est pas infinitésimal, comme une définition raisonnable du temps de calcul devrait le supposer, l'exécution de ces boucles prendra donc un temps infini.

3. *Divergence des ressources.* Quand bien même le problème précédent serait surmonté, le modèle des fonctions $\mathbb{R}$-récursives affronterait encore d'autres problèmes liés à l'opérateur de minimisation μ. Celui-ci permet en effet de définir l'opérateur η, qui détecte l'existence d'un zéro d'une fonction $\mathbb{R}$-récursive arbitraire.

$$\eta_y f(\vec{x}, y) = \begin{cases} 1 & \text{si} \quad \exists y f(\vec{x}, y) = 0, \\ 0 & \text{si} \quad \forall y f(\vec{x}, y) \neq 0. \end{cases} \tag{3.8}$$

   D'un autre point de vue, $\eta_y f(\vec{x}, y)$ est la fonction caractéristique de l'ensemble des vecteurs $\vec{x}$ tels que $\mu_y f$ est bien défini.

   Il est aisé de montrer que si $f(\vec{x}, y)$ est $\mathbb{R}$-récursive, $\eta_y f(\vec{x}, y)$ est $\mathbb{R}$-récursive :

$$\eta_y f(\vec{x}, y) = 1 - \delta(\mu_y[f(\vec{x}, g^{-1}(y))(y - g(\infty))] - g(\infty)) \tag{3.9}$$

   avec g, nommée « fonction de compression », n'importe quelle fonction de $[-\infty, \infty]$ dans un intervalle borné $[g^{-1}(-\infty), g^{-1}(-\infty)]$ par exemple $tan^{-1}(x)$:

$$\eta_y f(\vec{x}, y) = 1 - \delta(\mu_y[f(\vec{x}, tan(y))(y - \frac{\pi}{2})] - \frac{\pi}{2}) \tag{3.10}$$

   L'opérateur η_y est d'une grande puissance, puisqu'il permet de restreindre toute fonction partielle $h = \mu_y f$ à son domaine de définition,

faisant d'elle une fonction totale, $h_{tot} = (\mu_y f)(\eta_y f)$ [25].

Le problème est que l'inverse de la fonction g (dans l'exemple $tan(x)$), doit diverger lors du parcours de l'intervalle compact (dans l'exemple $[-\frac{\pi}{2}, \frac{\pi}{2}]$). Si, comme on peut s'y attendre, les valeurs prises par cette fonction sont implémentées dans un calculateur analogue par des valeurs d'une quantité observable, cela signifierait que ces quantités observables divergeraient pendant une évolution dynamique en temps fini. Une quantité observable étant censé prendre des valeurs finies mesurables, un tel comportement est le signe d'un modèle irréaliste (voir section 6.1, pour plus de développements sur cette idée) [26].

Tout comme les modèles B.S.S. et RAM réelle, le modèle des fonctions $\mathbb{R}$-récursives souffre d'un défaut d'implémentabilité. À ce titre, il ne peut être qualifié d'authentique modèle de calcul, même si cela ne lui retire pas tout intérêt théorique [27]. Nous reviendrons plus en détails sur les différentes critiques du caractère réaliste du modèle lorsque nous étudierons les modèles analogiques d'hypercalcul (voir section 6.2).

Conclusion

Pour conclure, il nous semble nécessaire de critiquer le caractère bien fondé de notre problème initial. L'absence d'équivalence entre modèles de la calculabilité sur les réels ne constitue pas en soi un problème unique et bien posé pour l'étude des limites de la calculabilité. Il faut distinguer deux problèmes, qui appellent des réponses autonomes.

Le premier est l'extension de la notion de procédure effective à des objets n'admettant pas de représentation finitaire. Nous avons vu que celle-ci est réalisée de manière naturelle par l'analyse récursive, à condition de désacraliser la condition de terminaison du calcul en temps fini, et de la remplacer par la condition de finitude de K. Weihrauch. La continuité des fonctions Grzegorczyk-calculables ne doit pas être considérée comme problématique, dans la mesure où elle découle immédiatement de la condition de finitude, et par elle de nos intuitions les plus fondamentales sur la nature d'une procédure effective. La critique de l'analyse récursive, selon laquelle celle-ci souffrirait d'un défaut de naturalité en extension, n'est pas valide. L'extension de la notion de procédure effective aux domaines non-dénombrables provoque une

25. Cela n'implique pas que l'opérateur puisse restreindre toute fonction $\mathbb{R}$-récursive partielle à son domaine de définition, et résoudre ainsi le problème de l'arrêt. Contrairement au cas des fonctions récursives partielles ordinaires, l'opérateur μ n'est pas la seule source de partialité pour les fonctions $\mathbb{R}$-récursives : l'opérateur d'intégration peut aussi engendrer des fonctions partielles.

26. C. Moore se demande ([171],8) si les deux derniers problèmes ne sont pas équivalents. Notre analyse montre clairement qu'il n'en est rien : même si le premier problème était résolu, le deuxième se poserait toujours.

27. Dans [211], D. Graça et J.F. Costa ont pu ainsi étudié la classe des fonctions $\mathbb{R}$-récursives générables par un G.P.A.C..

scission de notre intuition sur les propriétés concernées, qui rend son emploi inepte.

Le second est la question de la simulation des modèles analogiques par l'analyse récursive, autrement dit la simulation d'un processus de calcul intrinsèquement continu par un calcul symbolique finitaire. Cette deuxième question, pour être bien posée, exige que la première question ait été résolue. Elle suppose également qu'un grand soin soit pris pour effectuer des comparaisons douées de sens entre le pouvoir expressif des modèles considérés. La définition dans [42] d'une notion de calcul approché pour le G.P.A.C. en est un exemple. Enfin, et surtout, cette question dépend d'une variété de considérations empiriques pesant sur l'implémentabilité de certains modèles analogiques, comprenant notamment les problèmes de robustesse face au bruit, la capacité à préparer le calculateur dans un état initial donné, ou encore l'impossibilité d'effectuer une étape de calcul en un temps arbitrairement court. D'un point de vue fondationnel, cette deuxième question est donc inséparable de l'examen de la thèse de Church-Turing empirique, et nous ne pourrons la traiter en détails qu'au moment de procéder à cet examen (voir chapitre 6, en particulier la section 6.2).

Il n'existe donc pas de problème propre à l'extension de la thèse de Church-Turing aux domaines non-dénombrables. Il existe une extension réussie de la notion de procédure effective à de tels domaines, réalisée par l'analyse récursive d'A. Grzegorczyk et D. Lacombe, et une variété de modèles analogiques, qui, outre certains problèmes conceptuels spécifiques, doivent affronter les difficultés propres à la thèse de Church-Turing empirique.

Chapitre 4

La conception interactive du calcul

Introduction

Pour le philosophe intéressé par l'étude du concept de calcul, la thèse de Church-Turing algorithmique serait particulièrement digne d'intérêt, parce qu'elle permettrait de délimiter avec rigueur l'ensemble des problèmes calculatoires (*computational problems*) ou tâches calculatoires (*computational tasks*).

Cependant, les différentes formulations de la thèse de Church-Turing algorithmique que nous avons examinées ne font pas emploi du concept de tâches calculatoires. Au lieu de cela, elles font emploi du concept mathématique de fonctions. D'un point de vue philosophique, il semble *a priori* légitime de s'interroger sur le privilège ainsi accordé aux fonctions, et de poser la question suivante : toutes les tâches calculatoires sont-elles descriptibles comme le calcul d'une fonction ? La théorie de la calculabilité ne fut pas d'emblée présentée comme une théorie des fonctions calculables. Ainsi, au début de son article de 1936, A. Turing affirmait ne donner de statut privilégié à aucun objet en particulier. Il présente ainsi son choix de prendre comme objet d'étude les nombres calculables comme une pure simple commodité, équivalent d'un point de vue théorique à n'importe quel autre choix d'objet pour la théorie de la calculabilité [220] :

> The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable *numbers*, it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates, and so forth. The fundamental problems involved are, however, the same in each case, and I have chosen the computable numbers for explicit

treatment as involving the least cumbrous technique. [1].

De prime abord, cette question peut se voir donner une réponse technique simple. S'il est légitime de théoriser la calculabilité sous la forme de la calculabilité des fonctions, c'est parce que la calculabilité sur d'autres objets mathématiques est toujours définie par référence à la calculabilité d'une fonction :

- Un ensemble E est calculable ssi sa fonction caractéristique χ_E est calculable.
- Un prédicat P est calculable ssi la fonction caractéristique de l'ensemble des éléments qui satisfont ce prédicat est calculable.
- Une relation binaire R est calculable ssi la fonction qui à x associe l'ensemble des y tels que xRy est calculable.
- Une équation est calculable ssi ses fonctions solutions sont calculables [2].

Une fois qu'on a ainsi constaté que la calculabilité des autres objets mathématiques est toujours définie en termes de calculabilité de fonctions, il devient naturel de présenter la théorie de la calculabilité comme une théorie des fonctions calculables. Cependant, cet état de fait de la théorie de la calculabilité ne constitue pas une réponse théorique de fond à notre question initiale. Dans la littérature récente, plusieurs voix se sont élevées pour affirmer que la calculabilité ne devait pas être réduite à la calculabilité des fonctions. En son état actuel, la théorie de la calculabilité ne décrirait pas les limites du calcul en toute généralité, mais uniquement les limites du calcul descriptible par une fonction, ou calcul fonctionnel. La compréhension de certaines tâches étudiées dans l'informatique contemporaine nécessiterait un changement de point de vue sur la nature des calculs effectués, qui serait mieux saisie par de nouveaux modèles de calcul, les modèles interactifs. Les informaticiens P. Wegner et D. Goldin ont à notre connaissance été les avocats les plus déterminés de cette conception (voir notamment [239], [238], [114]). Les informaticiens J. Wiedermann et J. van Leeuwen ont soutenu à plusieurs reprises des positions analogues, même si l'on verra que leur propos se démarque nettement de celui de P. Wegner et D. Goldin sur certains points cruciaux [232], [231], [233]. D'autres défenses de thèses similaires peuvent encore être trouvées dans [215].

Précisons immédiatement que cette conception ne vise nullement à remettre en cause la thèse de Church-Turing algorithmique sous sa forme originale. La validité de cette dernière n'est nullement remise en cause par le « paradigme interactif », défendu par P. Wegner et consorts. Les modèles de

1. Le lecteur notera au passage deux naïvetés initiales de Turing, qui ignore et les difficultés associées au choix de la représentation décimale, et les difficultés particulières à définir la notion de fonction réelle calculable.

2. Le lecteur peut également revoir la définition du concept de « réel calculable »que nous avons donnée à la section 3.1.1, pour voir qu'elle fait explicitement emploi de fonctions récursives.

calcul interactifs ne sont pas censés calculer des fonctions qu'une machine de Turing ne pourrait calculer : ils sont censés permettre d'effectuer des calculs qui ne peuvent être décrits en termes fonctionnels. Il ne s'agit pas de remettre en cause l'idée que les machines de Turing capturent adéquatement le calcul fonctionnel, mais de remettre en cause l'idée que tout calcul est un calcul fonctionnel.

Ce point semble devoir être souligné, tant il nous a été manifeste, notamment dans plusieurs conversations privées et débats sur des forums scientifiques, qu'il n'est pas toujours bien compris que les problèmes posés par une telle position ne sont pas les mêmes que ceux posés par une négation pure et simple de la validité de la thèse de Church-Turing algorithmique, ou même par une négation de la thèse de Church-Turing physique. Il est vrai qu'une telle confusion a été favorisée tout d'abord par certains éléments de langage des papiers de P. Wegner et D. Goldin, qui ont parlé dans un de leurs intitulés de « briser le mythe de la thèse de Church-Turing »[240] ; ensuite, et surtout, par l'ambiguïté entourant leur emploi des concepts de « pouvoir expressif »et « problème calculatoire ». Nous reviendrons sur ce dernier point, qui est évidemment essentiel au débat sur leur position. Néanmoins, dans l'exposé le plus récent de leur conception [114], D. Goldin et P. Wegner sont parfaitement explicites dans leur distinction des différents enjeux. Ils commencent en effet par distinguer la thèse de Church-Turing d'une deuxième proposition, baptisée « thèse de Church-Turing forte » : *Tout calcul effectif peut être effectué par une machine de Turing.* Cette thèse constitue une proposition plus forte que la thèse de Church-Turing originale, qui ne porte que sur les fonctions. Les arguments en faveur de la thèse de Church-Turing algorithmique ne pèsent en faveur de cette nouvelle thèse que si l'on admet une autre proposition, que Goldin et Wegner baptisent *conception mathématique* (*mathematical worldview*) : *Tous les problèmes calculatoires sont descriptibles en termes de fonctions* (*All computable problems are function-based*([114], 4). Ils affirment ensuite que toutes leurs critiques sont dirigées à l'encontre de cette dernière proposition, et non de la thèse de Church-Turing algorithmique en elle-même. Si leur rhétorique a pu prêter à confusion, la position de D. Goldin et P. Wegner est sur le fond parfaitement claire.

La présentation détaillée des modèles de calcul interactif, et des arguments présentés contre la thèse de Church-Turing algorithmique sur la base de discussions de ces modèles, déborderait amplement du cadre de ce travail. Il existe en effet de nombreux modèles de calcul interactif, aux ambitions théoriques diverses, et une pluralité de critiques de la thèse de Church-Turing réalisées sur cette base [3]. Puisque nous allons adopter une posture critique face à ces positions, il nous faut signaler l'existence d'une littérature déve-

3. P. Wegner a par exemple défendu son point de vue dans de nombreux articles, dont la publication s'étale sur une quinzaine d'années. À notre connaissance, le plus ancien article de cet auteur sur ce sujet date en effet de 1997 [238].

loppée consacrée à la critique de ce point de vue (voir [195], [90], [62]) [4]. Sans viser à l'exhaustivité, ni à la synthèse de la littérature existante, nous préférerons présenter des arguments qui sont, à notre connaissance, originaux. Nous commencerons par une brève synthèse des arguments présentés en faveur d'une rupture du calcul interactif avec le paradigme fonctionnel de la calculabilité. Nous montrerons ensuite en deux temps l'insuffisance de ces arguments. Dans un premier temps, nous montrerons que ces arguments, dans leur formulation usuelle, s'appliqueraient tout aussi bien à l'analyse récursive, bien que ce modèle ne viole nullement la thèse de Church-Turing algorithmique. Nous tâcherons ensuite de voir si une version modifiée de ces arguments, qui capturerait plus adéquatement la singularité du calcul interactif, ne pourrait avoir plus de succès. En soulignant que c'est avant tout le caractère imprédictible de l'interaction avec l'environnement qui fait la singularité du calcul interactif, nous montrerons, par une présentation des travaux de J. Wiedermann et J. van Leeuwen, que ce trait est difficile à concilier avec la thèse d'un pouvoir expressif supérieur du calcul interactif. Nous conclurons en discutant en quoi les thèses de P. Wegner et D. Goldin ne nous semblent pas, en l'état actuel, être bien formulées.

4.1 Arguments en faveur de la conception interactive

La conception interactive du calcul a émergé dans un corpus de travaux récents dans la littérature informatique. Ceux-ci visent à modéliser certaines tâches calculatoires fréquentes dans la pratique, telles que celles exécutées par les systèmes d'exploitation, les serveurs, les traitements de texte, les systèmes embarqués ou les interfaces graphiques. Les tâches effectuées par ces logiciels ou systèmes [5] présentent plusieurs caractéristiques essentielles parmi les suivantes :

1. *Interaction avec un environnement imprédictible.* La machine doit interagir avec un flot imprédictible d'information provenant de son environnement. L'environnement peut être constitué par des utilisateurs, un réseau de machines, ou même l'environnement physique réel dans le cas des systèmes embarqués. C'est naturellement ce trait fondamental qui justifie l'emploi de l'appellation de « calcul interactif ».
2. *Absence de borne temporelle sur le processus.* Les systèmes considérés ne sont pas censés s'arrêter après avoir exécutés une tâche. Un serveur ou un opérateur système sont conçus pour opérer pendant un temps

4. En particulier, pour une discussion détaillée d'arguments basés sur les modèles du π-calcul, du \$-calcul et des algorithmes évolutifs, voir ([62], 201-215).

5. Dans le vocabulaire informatique, un système désigne de manière informelle un ensemble de logiciels et/ou de matériels fonctionnant ensemble pour permettre l'exécution d'une tâche.

indéfini. Leur fonction centrale n'est pas de mener à terme une tâche donnée, mais de maintenir un certain comportement par rapport à leur environnement.

3. *Évolution dynamique.* Le système ne constitue pas une entité fixée une fois pour toute, comme un algorithme est fixé avant toute exécution d'un calcul. Le système, par son interaction avec son environnement, évolue au cours du temps, à au moins deux titres :
   - *Apprentissage.* Le système apprend au fur et à mesure de son histoire, dans la mesure où il conserve une mémoire de ses travaux passés, et utilise cette mémoire dans son comportement subséquent.
   - *Mises à jour.* Le système peut subir de nombreuses mises à jour, à la fois matérielles et logicielles.

Ces différentes caractéristiques rendent intuitivement difficile la modélisation de ces systèmes, ou à tout le moins de l'intégralité de l'activité de ces systèmes, par des machines de Turing. Pour reprendre une expression due à D. Goldin et P. Wegner, le calcul par une machine de Turing (ou par tout autre modèle universel) est une boite noire [114]. Suite à la lecture de l'entrée, aucune communication n'a lieu entre la machine et son environnement. Par conséquent, toute l'information nécessaire à l'exécution du calcul doit être contenue dans l'entrée et dans le programme. Ce trait essentiel du calcul fonctionnel interdit de modéliser l'interaction imprédictible avec un environnement, et l'évolution dynamique qui peut en résulter.

À ce premier grief essentiel à l'égard du modèle des machines de Turing, D. Goldin et P. Wegner en ajoutent parfois d'autres, qui sont censés montrer son inadéquation pour les processus en question. Le premier est la finitude des entrées et des sorties : les entrées et sorties du calcul peuvent être représentées par une chaîne de caractères finie sur un alphabet fini. Une telle représentation des entrées et des sorties est évidemment impossible pour une machine interagissant indéfiniment avec un flux d'entrées, et produisant un flux indéfini de sorties.

Le second est la terminaison : de manière générale, le calcul par une machine de Turing est voué à terminer en temps fini, sauf dans le cas où la fonction n'est pas définie sur l'entrée considérée. Les calculs effectués par un serveur, en revanche, n'ont pas vocation à terminer. Pour reprendre une expression due à J. Wiedermann et J. van Leeuwen, les calculs effectués par une machine interactive n'ont pas pour fin ultime de produire une sortie donnée, mais de maintenir un certain schème d'action-réaction à l'égard de leur environnement ([232], 1).

Cette liste de griefs n'est cependant pas constituée de propositions indépendantes. Le trait le plus fondamental du calcul interactif est l'interaction avec un environnement pendant un temps indéfini. C'est cette propriété qui interdit d'encoder toute l'information pertinente pour le calcul dans une entrée de taille finie, et qui interdit de décrire la tâche effectuée comme la

production d'une sortie de taille finie.

Comme le formulent P. Wegner et D. Goldin, la conception fonctionnelle du calcul ne permet pas de répondre à la question : qu'est-ce qu'un système d'exploitation calcule ? Face à cette incapacité de la machine de Turing à modéliser tous les calculs, il deviendrait naturel de demander un élargissement de paradigme, pour comprendre ces tâches si fréquentes dans l'informatique contemporaine ([240], 3-4).

P. Wegner et D. Goldin argumentent également en faveur de la plus grande expressivité de la conception interactive, en montrant que le calcul fonctionnel peut être compris comme un cas particulier de calcul interactif. Pour le comprendre, il est nécessaire de présenter succinctement le modèle particulier du calcul interactif dû à ces auteurs, les Machines de Turing Persistantes (*Persistent Turing Machines*, *P.T.M.s*).

Dans ce modèle, on considère des flux infinis d'entrées/sorties (*I/O streams*) $(i_1, o_1, i_1, o_2, ...)$ où o_k est calculé à partir de i_k. Chaque pas de transition $(s, i) => (s', o)$ est un calcul de machine de Turing complet. Mais o_k précède et peut influencer la i_{k+1}-ième entrée. L'existence d'une mémoire persistante permet ainsi un couplage entrée-sortie. Le calcul d'une machine de Turing est un calcul de machine persistante privée de sa mémoire persistante [6].

Cette inclusion comme un cas particulier du modèle de Turing dans un modèle interactif permet de justifier à la fois que le calcul interactif soit considéré comme plus expressif, et que la thèse de Church-Turing algorithmique demeure vraie dans son domaine de validité propre.

4.2 Critiques de la conception interactive

4.2.1 L'analogie avec l'analyse récursive

Tous les arguments employés par P. Wegner et D. Goldin nous semblent cependant insuffisants. Pour justifier cette critique, nous allons voir que tous ces arguments, à une exception près, s'appliquent tout aussi bien à un modèle dont on ne peut défendre -et dont personne, à notre connaissance, ne défend- qu'il ait un pouvoir expressif supérieur aux machines de Turing : l'analyse récursive.

Comme nous l'avons vu lors de notre présentation de ce modèle, ni la représentation finitaire des entrées et sorties, ni la terminaison ne doivent être considérées comme des propriétés essentielles du calcul effectif. C'est la condition de finitude de Weihrauch, plus faible, qui doit être considérée

6. Il s'agit là d'une analyse propre au modèle du calcul interactif proposé par D. Goldin et P. Wegner. Ceux-ci ne prétendent en aucun cas capturer le calcul interactif dans toute sa généralité par le modèle des P.T.M.s, et ils envisagent la possibilité de modèles encore plus expressifs incluant le leur comme un cas particulier (voir [239]).

comme essentielle[7]. L'absence de ces propriétés au sein d'un modèle de calcul ne peut donc en aucun cas être considérée comme une rupture avec le paradigme du calcul fonctionnel.

En outre, nous avons vu que la réception de l'entrée par la machine de Type-2 peut être représentée sur un mode dynamique, où les approximations successives de l'entrée dans la représentation choisie sont données au processeur au fur et à mesure. Dans cette perspective, on peut même affirmer que l'interaction avec un environnement n'est pas étrangère au modèle des machines de Type-2, et donc au calcul effectif, puisqu'on peut considérer que c'est l'environnement qui fournit à la machine les approximations successives de son entrée.

L'analogie avec l'analyse récursive permet aussi de dénoncer l'insuffisance de l'argument portant sur le pouvoir expressif. La modélisation du calcul suppose l'emploi de certains objets mathématiques. Un modèle donné peut employer dans son langage, pour modéliser tel trait du calcul, des objets dont un autre modèle n'emploie que des cas particuliers : cela n'implique pas que le premier modèle ait un pouvoir expressif supérieur au second. Les fonctions $f :\subseteq \Sigma^* \longmapsto \Sigma^*$ calculées par une machine de Turing sont un cas particulier des fonctions $g :\subseteq \Sigma^\omega \longmapsto \Sigma^\omega$ calculées par une machine de Turing de type 2. Cela ne signifie pas que le pouvoir expressif de l'analyse récursive soit supérieur à celui des machines de Turing.

Si nombre des arguments employés en sa faveur sont insuffisants, la thèse de P. Wegner et D. Goldin n'est pas pour autant réfutée. Dans le cas de l'analyse récursive, l'interaction avec l'environnement est prédictible, et obéit à une spécification donnée, à savoir donner les approximations successives d'un réel. Ce qui est à tout le moins censé faire la spécificité du calcul interactif, c'est le caractère imprédictible de l'interaction avec l'environnement, qui interdit de donner au calcul une spécification a priori, et donc interdit *a fortiori* de représenter le processus de calcul comme l'exécution d'une fonction.

4.2.2 Résultats théoriques et véritable pouvoir expressif : la *Web Turing Machine* de Wiedermann et van Leeuwen

Nous allons voir, en examinant le travail de J. Wiedermann et J. van Leeuwen, que l'imprédictibilité de l'environnement ne suffit pas à attribuer un pouvoir expressif supplémentaire à un modèle interactif. Dans celui-ci, l'écart entre modélisation théorique du calcul et véritable pouvoir expressif

7. On remarquera que J. Wiedermann et J. van Leeuwen imposent à un de leurs modèles du calcul interactif, les *Interactive Turing Machines*, une condition analogue à la condition de finitude de Weihrauch. Après avoir spécifiés que leur machine peut lire en son port d'entrée un symbole vide, signifiant l'absence d'information nouvelle, ils énoncent la condition de délai fini (*finite delay*) suivante : pour toute entrée non-vide, la machine doit produire une sortie non-vide en un temps fini ([232], 95). La seule différence entre cette condition et la condition de finitude est l'absence d'imposition d'une relation fonctionnelle entre l'entrée et la sortie.

attribuable au modèle est encore plus sensible. Les calculs effectués par leur modèle du calcul interactif sont décrits par des classes de complexité non-uniformes [8] sans pour autant qu'on puisse considérer ce modèle comme un modèle d'hypercalcul.

L'ambition initiale de Wiedermann et van Leeuwen est d'oeuvrer à la compréhension de ce qu'ils nomment des systèmes interactifs évolutifs (*evolving interactive systems*). Ces systèmes peuvent recevoir une série indéfinie de mises à jour matérielles et logicielles [9]. Les calculs effectués en réseaux, notamment sur Internet, sont l'objet naturel de cette modélisation : un réseau de machines peut ainsi se voir ajouter de nouvelles machines, ou voir ses machines mises à jour en cours de fonctionnement.

Wiedermann et Leeuwen définissent la *Web Turing Machine, WTM*, comme un modèle abstrait du calcul sur Internet. Une *WTM* est un ensemble fini de sites en interaction variant dans le temps. La cardinalité de l'ensemble, les programmes des machines appartenant à cet ensemble, ainsi que le délai de réception d'un message peuvent varier dans le temps de manière complètement imprédictible. On suppose uniquement qu'il existe une échelle de temps discrète uniforme au sein d'une *WTM*, modélisée par $\mathbb{Z}^+$, l'ensemble des entiers naturels positifs. Une *WTM* est caractérisée par le triplet $G = (\alpha, \delta, \mu)$ tel que :

- $\alpha : \mathbb{Z}^+ \longmapsto 2^{\mathbb{Z}^+}$ est une fonction d'adressage, qui à chaque instant $t \geq 0$ associe l'ensemble fini des adresses de sites en réseau au temps t. G consiste donc à l'instant $t \geq 0$ des $\alpha(t)$ sites de l'ensemble $S_t = \{M_i | i \in \alpha(t)\}$ où M_i est le site d'adresse i.
- $\delta : \mathbb{Z}^+ \times \mathbb{Z}^+ \longmapsto \Sigma^*$ est la fonction d'encodage qui à tout temps $t \geq 0$ et à toute adresse $i \in \alpha(t)$ associe l'encodage $\langle M_i \rangle$ des sites M_i à cette adresse à cet instant.
- $\mu : \mathbb{Z}^+ \times \mathbb{Z}^+ \times \mathbb{Z}^+ \longmapsto \mathbb{N}$ est la fonction de transfert de message qui à tout site émetteur i et à tout site récepteur j, pour $i, j \in \alpha(t)$ et l'instant $t \geq 0$ associe la durée du message de transfert de i à j.

Une *description* d'une *W.T.M.* est donnée par l'ensemble des encodages de l'intégralité de ses sites à l'instant t. Une *configuration* d'une *W.T.M.* au

8. Intuitivement, la propriété de non-uniformité permet d'utiliser un algorithme différent pour chaque instance. Une famille de circuits booléens $C_1, ..., C_n$ est non-uniforme ssi la description de chaque circuit pour une longueur n ne peut être engendrée par une machine de Turing. De manière équivalente, on peut définir une classe de complexité non-uniforme comme la classe de problèmes appartenant à une classe donnée C, ssi l'on fournit un conseil de longueur $F(n)$ pour chaque instance de longueur n. On note alors la classe $C//F$. Par exemple, la classe $P//Poly$ est la classe des problèmes solubles en temps polynômial par une machine de Turing, ssi l'on fournit un conseil polynômial en n pour chaque instance de longueur n. Une procédure de calcul non-uniforme n'est évidemment pas effective, et une classe de problèmes non-uniforme peut inclure strictement l'ensemble des fonctions récursives, comme c'est le cas pour notre exemple $P//Poly$, et les classes caractérisant le pouvoir expressif de la WTM.

9. Cette propriété est absente du modèle des Machines de Turing Persistantes.

temps t consiste en la liste de la configuration de ses sites à cet instant.

Intuitivement, chaque site connecté au réseau est occupé par une machine de Turing et la donnée écrite sur sa bande d'entrée. À chaque instant t, la machine de Turing peut effectuer une étape de calcul, envoyer ou recevoir un message à ou d'une autre machine. En mode hors connexion, la fonction de transition peut être altérée, modifiant la machine de Turing occupant le site. Une machine peut être connectée ou déconnectée à tout instant : les valeurs des fonctions α et δ enregistrent ces diverses modifications de la composition du réseau.

Rien n'impose que les fonctions α, δ et μ soient récursives. L'évolution du réseau dans le temps est imprédictible, et on ne peut donc présupposer l'existence de relation fonctionnelle récursive entre le temps et les machines connectées, entre les adresses et la description des machines occupant ces adresses, ou encore entre les adresses et le délai de réception d'un message. Pour tout instant $t \geq 0$, les fonctions peuvent seulement être décrites par une table finie.

À chaque instant et à chaque site, la WTM lit un symbole en entrée et produit un symbole en sortie (le symbole pouvant être le symbole vide). La WTM calcule donc un morphisme associant à un ensemble de flux finis ou infinis de symboles en entrée des flux similaires en sortie. Le résultat du calcul peut dépendre du contenu des messages reçus et de leur temps d'arrivée. Mais le contenu des messages étant le fruit d'un calcul interne à un site, et le temps de réception étant déterminé par la fonction μ, cela n'introduit pas d'indéterminisme dans le calcul d'une WTM : pour un ensemble de flux donné, le résultat d'un calcul est uniquement déterminé (pour plus de détails, voir [233], 13).

La caractérisation du pouvoir expressif de ces WTM donne un résultat paradoxal, puisqu'il est défini par des classes de complexité non-uniforme, dont certaines ont un pouvoir suprarécursif. Ces classes de complexité sont définies relativement à un autre modèle de machine interactive que nous allons présenter succinctement. Une machine de Turing Interactive est une machine de Turing munie d'un port d'entrée et d'un port de sortie, afin de permettre un échange indéfini d'information avec son environnement. À chaque étape de calcul, la machine de Turing lit un symbole de son port d'entrée et écrit un symbole dans son port de sortie, les symboles lus et écrits comprenant le symbole vide λ. On impose à une ITM de respecter une *condition de délai fini* : la machine doit réagir à toute entrée non-vide en un temps fini par l'écriture d'un symbole non-vide. On déduit immédiatement de cette condition qu'un flux d'entrées infini entraîne l'existence d'un flux de sorties infini.

Comme le remarquent J. Wiedermann et J. van Leeuwen, une machine interactive peut en principe être simulée par une machine de Turing pendant un intervalle de temps fini arbitraire : il suffit pour cela d'encoder la suite finie des entrées reçues sur la bande d'entrée de la machine. Cette simulation

ne peut être réalisée qu'*a posteriori*, tandis que la machine interactive est censée interagir avec son environnement en temps réel. Mais la présence d'interaction en temps réel ne joue pas de rôle dans l'estimation du pouvoir expressif.

On définit ensuite les machines de Turing interactives avec oracles de manière analogue aux machines de Turing avec oracles. La fonction oracle est une fonction $f : \mathbb{Z}^+ \longmapsto \Sigma^*$. On ajoute à la machine un état distingué de requête de l'oracle, et une bande de conseil pour échanger avec l'oracle. À l'instant $t \geq t_1$, la machine entre dans son état de requête et reproduit sur la bande de conseil l'argument T_1 écrit sur sa bande d'entrée à l'instant t_1. La valeur $f(T_1)$ est alors substituée à T_1 sur la bande de conseil en une étape. Comme dans le cas des machines de Turing simples, rien n'oblige la fonction de conseil à être récursive, et les classes de complexité non-uniformes ainsi définies peuvent définir un pouvoir expressif suprarécursif. Comme il est d'usage pour les machines de Turing, la longueur du conseil donné par l'oracle peut être bornée par une fonction récursive de la taille de l'entrée. Les entrées étant ici des flux infinis, on doit moralement se restreindre à la suite des symboles écrite à l'instant t. Le temps t étant une borne supérieure sur le nombre de symboles lus, on borne les fonctions de conseil par des fonctions calculables du temps t. On définit ainsi les classes de complexités $ITM - C/F$ comme l'ensemble des fonctions calculables par une $ITM - C$ machines utilisant une fonction oracle issu de l'ensemble F.

Il n'est pas très surprenant que toute WTM puisse être simulée par une ITM/A. L'ITM peut enregistrer sur ses bandes l'ensemble des sites et leur état de connection, tout en stockant dans sa fonction de conseil les valeurs des trois fonctions α, δ, et μ. L'ITM peut ainsi mettre à jour les sites de manière séquentielle. De manière plus inattendue, J. van Leeuwen et J. Wiedermann ont démontré la simulation inverse : toute ITM/A peut être simulée par une WTM. L'idée de base de la simulation est de faire simuler le comportement de l'ITM/A par un site déterminé de la WTM. L'emploi de la fonction de conseil f est simulé par l'envoi d'un message au site $j \in \alpha(t)$, qui est programmé pour répondre à tout message à tout moment par l'envoi d'un message encodant $f(j)$. Puisqu'à tout moment $j < t$, l'ITM ne peut avoir consulté la fonction oracle qu'un nombre fini de fois, le nombre fini et non-borné des sites suffit pour encoder toute valeur désirée de la fonction. Pour reprendre une image de J. van Leeuwen et J. Wiedermann, les sites fonctionnent comme des pages Web qui sont consultées pour connaître une valeur de la fonction oracle ([231], 7).

Au titre de cette simulation, on serait tenté d'affirmer que la WTM dispose d'un pouvoir expressif suprarécursif. Cette conclusion serait cependant discutable, car comme le signalent explicitement Leeuwen et Wiedermann, l'existence d'une telle simulation n'implique pas qu'on puisse utiliser un réseau de machines pour évaluer une fonction non-récursive :

> A challenging question still remains unanswered : could one indeed make use of the super-Turing potential of the underlying machines to one's advantage ? Can one solve certain concrete undecidable problems by such machines ? The answer is, (un)fortunately, no. From a practical point of view our results mean that the corresponding devices cannot be simulated by standard TM's working under a classical scenario. This is because the evolving interactive machinery develops in an unpredictable manner, by a concurrent unpredictable activity of all agents operating the sites.

C'est pour la même raison qu'une WTM n'est pas simulable par une machine de Turing standard, et qu'elle ne peut être utilisée pour effectuer un calcul suprarécursif : son caractère imprédictible. On voit comment la modélisation de tels processus imprédictibles peut donner l'impression superficielle d'un pouvoir calculatoire supérieur, qui n'a de fonction que théorique. Ce pouvoir calculatoire n'est attribué aux machines qu'à l'infini du temps, et ne représente donc aucune information qu'un utilisateur pourrait acquérir en un temps fini [10]. La caractérisation mathématique des *Web Turing Machines* par des classes de complexité non-uniforme ne constitue donc pas une indication qu'on puisse employer un réseau de machines comme Internet pour réaliser un calcul suprarécursif.

Cette dernière remarque est bien entendu valable pour un modèle particulier. Elle illustre cependant le gouffre qui peut exister entre modélisation théorique d'un calcul et évaluation de son pouvoir expressif réel. Elle montre également une difficulté essentielle à toute la critique de la thèse de Church-Turing algorithmique basée sur le calcul interactif : comment exploiter une interaction imprédictible avec un environnement comme une source de pouvoir expressif supplémentaire ?

Conclusion

Notre question initiale -toutes les tâches calculatoires sont-elles descriptibles par des fonctions ?- est-elle bien posée ? Il est aisé de voir que, si aucune borne n'est mise à l'emploi des termes « tâche calculatoire », « problème

10. On lit ainsi dans un rapport technique de J. van Leeuwen et J. Wiedermann (le terme « global TM »y est synonyme de « Web Turing Machine »([231], 13) :

> [...]global TMs seem to present an interesting case : at each point of time they have a finite description, but when seen in the course of time, they represent infinite non-uniform sequences of computing devices, similar to non-uniform circuit families. Thus, what makes them non-fitting under the notion of traditional algorithms is their evolution in time. This gives them the necessary infinite dimension that boosts their computational power beyond that of standard Turing machines.

computationnel », « pouvoir expressif »ou même simplement « calcul », la réfutation de la conception fonctionnelle du calcul deviendra bien trop aisée, et par là dépourvue de tout apport théorique. Le théoricien de la complexité S. Aaronson a ainsi remarqué avec ironie qu'une machine de Turing augmenté d'un grille-pain (*Toaster-enhanced Turing Machine*) est assurément capable d'exécuter des tâches qu'aucune machine de Turing n'est capable d'exécuter [6]. Ira-t-on cependant prétendre qu'une telle machine constitue un nouveau paradigme calculatoire, plus expressif que celui de la machine de Turing ? Non pas, parce qu'aucun informaticien n'irait prétendre que griller du pain constitue en soi une tâche calculatoire.

Il semble donc nécessaire d'énoncer certaines contraintes sur les concepts de tâche, problème et pouvoir expressif, afin d'éviter de trivialiser la question [11]. Il semble en particulier nécessaire d'énoncer des contraintes indépendantes permettant de qualifier de manière non-triviale un processus donné de « tâche calculatoire », avant d'affirmer que ce processus n'est pas modélisable comme le calcul d'une fonction.

Comme le remarquent à juste titre D. Goldin et P. Wegner ([114], 23), un auteur comme D. Knuth discute la notion d'algorithme d'une manière qui exclue d'emblée toute interaction en temps réel avec l'environnement, et garantit ainsi le caractère fonctionnel du calcul [12].

Existent-t-il des traits naturels et féconds de la notion de calcul, qui seraient ignorés par sa compréhension comme procédure effective, et seraient révélés par la conception interactive ? D. Goldin et P. Wegner n'offrent pas de réponse élaborée à cette question, sinon un appel à l'évidence de l'exemple, en l'occurrence celui d'un système de pilotage automatique ([114], 22, nous

11. Ce point fut remarqué dès les prémisses de ce débat sur le calcul interactif ([195], 2).

12.

> In his definition of algorithms, Knuth was consistent with the mathematical function-based foundations of the theory of computation. He explicitly specified that algorithms are closed ; no new input is accepted once the computation has begun : "An algorithm has zero or more inputs, i.e., quantities which are given to it initially before the algorithm begins." Knuth distinguished algorithms from arbitrary computation that may involve I/O. One example of a problem that is not algorithmic is the following instruction from a recipe (Knuth 1968) : "toss lightly until the mixture is crumbly." This problem is not algorithmic because it is impossible for a computer to know how long to mix ; this may depend on external dynamically changing conditions such as humidity, that cannot be predicted with certainty ahead of time. In the function-based mathematical worldview, all inputs must be specified at the start of the computation, preventing the kind of feedback that would be necessary to determine when it's time to stop mixing. The problem of driving home from work is also among those problems that Knuth meant to exclude. Knuth's detailed discussion of algorithmic computation ensured their function-based behavior and guaranteed their equivalence with TMs (...).

soulignons) :

> The driving example represents an empirical proof of the claim that interactive computation is more expressive than function-based computation, i.e., it can solve a greater range of problems. *However, to accept this claim, one has to broaden one's notion of a problem beyond what is prescribed by the mathematical world-view.* Driving home from work, queuing jobs within an operating system, or document processing, are all legitimate computational problems on par with finding common divisors or choosing the next move on a chess board [13].

Tout terme scientifique n'a certes pas à recevoir de définition explicite. Néanmoins, comme nous l'avons vu plus haut, la thèse de Church-Turing algorithmique s'appuie sur l'intuition que nous sommes capables de reconnaitre une procédure effective lorsque nous en rencontrons un exemple. Néanmoins, il est possible d'énoncer quelques conditions informelles sur la notion de procédure effective qui, si elles ne constituent pas une définition rigoureuse, montrent que la notion d'algorithme n'est pas vide avant d'être l'objet de la thèse de Church-Turing, et que celle-ci ne peut donc être considérée comme une définition implicite de la notion. Dans le cas des notions de « problème calculatoire »et de « tâche », nous ne disposons ni de cette capacité de reconnaissance d'un cas positif, ni de ces contraintes informelles mais significatives. Comme il est manifeste dans les emplois que nous venons de citer, ces notions peuvent finir par désigner toute tâche confiée à un ordinateur, ou toute fonction incombant à un logiciel particulier. Si l'on considère l'immense variété des activités dans lesquelles sont employés les ordinateurs, un tel emploi est d'une généralité qui confine à la vacuité. Pour mieux le faire voir, il suffira de faire remarquer qu'à ce titre, « gérer une comptabilité »et « moduler un stimulateur cardiaque »comptent également comme des tâches calculatoires. On peut certes concéder ce point, mais sans que cela ait une quelconque pertinence pour la thèse de Church-Turing. La critique de la thèse de Church-Turing offerte par D. Goldin et P. Wegner, et plus précisément la critique de ce qu'ils appellent la conception mathématique, ne peut à proprement parler être qualifiée de fausse. Dans son état actuel, elle souffre cependant d'une certaine vacuité.

Comme le remarque S. Aaronson, le modèle des machines de Turing n'est pas destiné à modéliser n'importe quelle interaction d'un ordinateur avec

13. On peut encore retrouver la même perspective dans la citation suivante ([114], 21, nous soulignons) :

> The question "what do operating systems compute?" has been a conundrum for the adherents of the mathematical worldview, since these systems never terminate, and therefore never formally produce an output. Yet *it is clear that they do compute*, and that their computation is both useful and important to capture formally.

son environnement. À cette juste remarque nous souhaiterions ajouter qu'il est illégitime d'exiger du modèle de la machine de Turing qu'il modélise tout et n'importe quoi. On trouve malheureusement dans la littérature des critiques du modèle de Turing, et de la thèse de Church-Turing, qui partent manifestement du postulat que tout processus naturel doit être modélisé par une machine de Turing. Une idée de ce type est ainsi exprimée dans [161] :

> On the front of biological and cognitive processes, Church-Turing Thesis appears as inadequate, too. Such systems actually show features very far from those required by Turing computability : they are evolutionary, adapting and self-organising, so processes do not "halt", they show intrinsic emergence, are dynamically goal-oriented, modify their codes in relation to the context (semantic appropriation of information), process superposed patterns of continuous information and manage noise.

Face à ce type d'énoncés, il semble nécessaire de rappeler que, bien que le calcul soit fort utile dans la modélisation de nombreux processus naturels, et que les approches computationnelles sont vouées à se répandre dans de nombreuses sciences, le processus du calcul lui-même, tel qu'il peut être modélisé par un modèle de calcul universel, n'a pas à reproduire trait pour trait chaque processus qu'on pourrait souhaiter modéliser. De même, l'incapacité à modéliser tel ou tel processus par une fonction ne compte pas comme argument contre la formulation fonctionnelle de la thèse de Church-Turing algorithmique, car celle-ci n'a jamais eu pour objet une telle modélisation.

En l'état actuel, les arguments provenant de l'analyse des modèles interactifs du calcul ne constituent pas une critique de la thèse de Church-Turing. Ceci ne retire pas à ces modèles toute portée philosophique. L'idée que certaines tâches de l'informatique invitent à la création de nouveaux paradigmes de programmation, et que la modélisation théorique du calcul puisse y rompre avec une description fonctionnelle du calcul nous semble digne d'attention. Mais il s'agit là d'un énoncé portant sur les vertus théoriques d'un langage de programmation, et non sur l'évaluation de son pouvoir expressif.

Troisième partie

Pour la thèse de Church-Turing empirique

It always bothers me that, according to the laws as we understand them today, it takes a computing machine an infinite number of logical operations to figure out what goes on in no matter how tiny a region of space, and no matter how tiny a region of time. How can all that be going on in that tiny space? Why should it take an infinite amount of logic to figure out what one tiny region of space/time is going to do? So I have often made the hypothesis that ultimately physics will not require a mathematical statement, that in the end the machinery will be revealed, and the laws will turn out to be simple, like the chequer board with all its apparent complexity. But this speculation is of the same nature as those other people make - 'I like it', 'I don't like it',- and it is not good to be too prejudiced about these things.

Richard Feynman, *The character of physical law*[14]

14. [97]

Chapitre 5

Conditions pour un modèle de calcul raisonnable

> *Computability should constitute of both consistency and also implementability.*

T.D. Kieu, *Reply to Andrew Hodges*[1]

Introduction : la notion de modèle de calcul non-standard

Une large partie des questions de calculabilité et de complexité que nous allons aborder dans la suite de ce travail dépend de l'examen de ce que nous appellerons des « modèles de calcul non-standard. »Avant de commencer notre examen de la thèse de Church-Turing empirique, nous allons préciser ce qu'on entendra ici par cette expression, en commençant par « modèle de calcul. »

La notion de « modèle de calcul »fait partie de ces quelques notions centrales de l'informatique, tout comme « algorithme », « processus », « comportement », qui sont sans cesse employées dans la communauté, sans se voir donner de définition rigoureuse. Comme nous le verrons (voir chapitre 6), il est difficile d'attribuer une telle absence de définition rigoureuse à un manque de travail sur les fondements de l'informatique : il se peut qu'un tel concept n'aie pas vocation à se voir donner une définition formelle. Sans prétendre formuler une telle définition, nous allons ici énoncer quelques contraintes qui nous seront utiles par la suite.

En première approche, un modèle de calcul est une modélisation mathématique permettant de décrire rigoureusement :

1. La suite des instructions composant la procédure de calcul.
2. L'exécution de cette procédure sur une entrée donnée.

1. *Ibid.*, p.7.

La première condition est moralement équivalente à l'affirmation qu'un modèle de calcul doit permettre de définir un langage de programmation. La seconde est moralement équivalente à l'existence d'une sémantique opérationnelle pour ce langage de programmation. L'équivalence est morale, au sens où de pour de nombreux langages de programmation, il n'existe pas de sémantique opérationnelle proprement définie. Celle-ci est remplacée par la définition d'un compilateur, qui va permettre l'exécution des instructions après compilation dans le langage-machine [2].

En outre, le modèle M doit être suffisamment bien défini pour permettre de déterminer l'ensemble des fonctions calculables au titre de ce modèle. Il est également souhaitable qu'il puisse permettre la quantification des ressources nécessaires à l'exécution du calcul, afin d'étudier ses propriétés de complexité. Mais cela peut constituer une difficulté en soi, bien distincte de la contrainte de définir l'ensemble des fonctions calculables au titre du modèle. Comme nous l'avons déjà remarqué plus haut (voir section 1.3.3), le λ-calcul, dans sa présentation historique, est un modèle bien défini de la calculabilité, quoique la β-réduction ne puisse passer pour une définition raisonnable d'une étape de calcul.

Enfin, il est naturel d'exiger qu'un modèle de calcul soit implémentable, c'est-à-dire que toutes les procédures de calcul qu'il permet de définir soient implémentables. Mais cette condition d'implémentabilité est complexe, et doit faire l'objet d'une discussion propre.

On appelle parfois « modèles de calcul non-standard » [3] des modèles de hardware décrivant une implémentation originale des portes logiques booléennes. Certains de ces modèles sont de véritables candidats au développement industriel, tandis que d'autres poursuivent des fins simplement théoriques ou ludiques. Au lieu des circuits électroniques usuels, on peut ainsi songer à bâtir des portes logiques à l'aide de photons, de fluides sous pression, de dominos, de boules de billard ou même de crabes vivants ([120],[96], [102]). À un niveau théorique, ces modèles de calcul ne présentent pas de véritable originalité : ils constituent simplement des modèles de hardware inhabituels pour les portes logiques des circuits classiques. Puisque les circuits classiques sont un modèle de calcul aux propriétés de calculabilité et de complexité bien connues, ces modèles ne peuvent engendrer aucune surprise dans ces domaines.

Par « modèles de calcul non-standard », j'entendrai donc ici non pas de tels modèles, mais des modèles d'une nature théorique plus ambitieuse. Les modèles que nous allons considérer emploient un traitement de l'information radicalement différent de celui des modèles démontrés équivalents

2. Je tiens à remercier M. Bagnol pour m'avoir fait remarqué ce point.

3. Le terme est inspiré d'une traduction de l'expression anglaise *unconventional computational models*, ou encore *unconventional computing*. Cependant il se démarque de certains usages de ce terme, comme nous allons le voir immédiatement. L'expression *non-standard models* a déjà été employée par E. Blakey dans [31].

aux machines de Turing. Pour reprendre la terminologie définie ci-dessus (voir section 2.2.1), ces modèles ne définissent pas une forme du calcul symbolique finitaire. Leurs propriétés de calculabilité et de complexité sont, à l'heure de la rédaction de cet écrit, soit totalement inconnues, soit l'objet de discussions scientifiques en cours. C'est en ce sens qu'on peut les qualifier de « non-standard ». Parmi ces modèles non-standard, il se trouve bien évidemment des modèles d'hypercalcul.

L'étude des propriétés de calculabilité et de complexité au sein des modèles de calcul non-standard pose deux problèmes particuliers. Le premier de ces problèmes est la définition du pouvoir expressif et des ressources computationnelles au sein de ces modèles. Il est en effet souvent impossible de transposer les définitions usuelles à ces modèles originaux. Sans de telles définitions, il est impossible de comparer rigoureusement les propriétés calculatoires de ces modèles non-standard à celles des modèles connus [4].

Le deuxième problème est la question de l'implémentabilité de ces modèles. Comme nous l'avons précédemment mentionné (voir section 2.1), les modèles historiques de la calculabilité étaient trivialement implémentables par un calcul papier-crayon. Nous verrons qu'au contraire certains des modèles de calcul non-standard permettent des manipulations de l'information strictement impossibles pour un tel calcul. S'ils ne sont pas automatisables, ces modèles ne seront donc pas implémentables. Par conséquent, ces modèles décrivent des calculs qui ne peuvent être exécutés que par des machines.

Les modèles de calcul implémentables sont parfois informellement qualifiés de modèles « réalistes »ou « raisonnables »et par défaut, nous emploierons ces termes comme synonymes (voir section 8.1). On peut considérer que la thèse suivante comme une reformulation de la thèse de Church-Turing empirique :

Thèse de Church-Turing empirique (Modèle raisonnable). Tous les modèles de calcul raisonnables ont un pouvoir expressif au plus équivalent aux machines de Turing.

On pourra dire, de manière équivalente, que tout modèle de calcul raisonnable est simulable par une machine de Turing.

Face à la division des modèles de calcul en standard et non-standard, et face à la grande diversité interne aux modèles non-standard, l'unité philosophique de notre sujet peut légitimement être mise en doute. Lorsqu'on dit d'un homme muni d'un boulier, et d'une machine analogique, qu'ils calculent, emploie-t-on bien la même notion [5] ? Comment définir la notion de calcul de façon à la fois substantielle, et indépendante des particularités d'un modèle

4. Nous avons déjà rencontré ce problème lorsque nous avons discuté du pouvoir expressif du modèle G.P.A.C., qui compte comme modèle non-standard au titre de notre définition. Nous le rencontrerons à nouveau au niveau des questions de complexité (voir chapitre 8).

5. Cette question est notamment soulevée par J. Earman (voir [85], 124).

donné ? Afin d'apaiser cette inquiétude philosophique, nous allons consacrer notre prochain chapitre à l'énoncé de conditions sur la notion de modèle de calcul raisonnable, qui permettra de préciser ce qu'on attend d'un modèle de calcul bien défini, avant de poser la question de son implémentabilité. Nous disposerons ainsi d'une conception philosophique unifiée du calcul, qui pourra s'appliquer aussi bien aux modèles standard qu'aux modèles non-standard, et garantira notre analyse philosophique des errements menaçant toute discussion d'un concept sur l'emploi duquel aucune contrainte précise ne vient peser. Le terrain sera alors prêt pour une discussion des questions d'implémentabilité des modèles d'hypercalcul, leurs conséquences pour l'interprétation empirique de la calculabilité, et la signification empirique de la thèse de Church-Turing.

Nous commencerons par discuter la conception épistémique du calcul, articulée dans des travaux récents par le philosophe G. Piccinini. Nous verrons que cette conception, après quelques aménagements, constitue la compréhension générale du calcul que nous recherchons. Nous défendrons ensuite cette conception contre diverses objections, en considérant tout d'abord la position de J. Earman, et une critique fréquente contre le critère d'implémentabilité. Nous illustrerons enfin la fécondité de cette conception, en montrant comment elle permet de distinguer diverses problématiques mêlant physique et calcul, qui sont fréquemment confondues dans la littérature.

5.1 Les conditions de G. Piccinini

Dans un article récent [190], le philosophe G. Piccinini a proposé, afin de préciser le sens de la thèse de Church-Turing empirique, l'énoncé d'une série de conditions sur la notion de « fonction calculable par un système physique[6] ». Outre leur grand intérêt pour la discussion des modèles d'hypercalcul, ces conditions sont justifiées par une conception globale du calcul, qu'on qualifiera de *conception épistémique.* Selon cette conception, le calcul est un processus nous permettant d'atteindre la connaissance représentée par la solution d'instances de problème, ou la valeur de fonctions en leurs arguments. Avant de discuter et défendre cette conception, nous allons expliquer son contenu en exposant, commentant et amendant les différentes contraintes articulées par G. Piccinini.

Piccinini énonce tout d'abord une contrainte globale :

> ***Utilisabilité.*** Pour qu'un processus physique compte comme calcul, il doit pouvoir être utilisé par un observateur fini pour obtenir les valeurs désirées de la fonction.

6. G. Piccinini ne distingue pas la forme physique de la thèse de Church-Turing comme un cas particulier de la forme empirique. Lorsqu'il parle des fonctions calculables par un système physique, cela doit donc être compris comme l'ensemble des fonctions calculables par une machine, dans notre terminologie.

Il énonce ensuite une série de sous-contraintes sur le processus physique effectuant un calcul, qui se doit d'être :

1. *Exécutable.* L'observateur fini doit être capable de découvrir quelle tâche est effectuée par le dispositif, de lire entrées et sortie.
2. *Uniforme.* Le processus employé doit être identique pour tout argument de la fonction/entrée de la tâche.
3. *Automatique.* Le processus doit pouvoir être exécuté sans intuition, ingénuité, invention ou conjectures.
4. *Fiabilité.* Il doit être possible d'obtenir, avec au moins une certaine probabilité [7], des résultats corrects à l'aide du protocole considéré.

Nous allons voir que les différentes contraintes articulées par G. Piccinini découlent de deux exigences théoriques distinctes. La première est que la notion de « fonction calculable par un système physique »soit un analogue théorique pertinent de la notion de « fonction calculable par une procédure effective. » [8] Les quatre contraintes que nous venons d'énoncer sont d'ailleurs toutes introduites par analogie théorique avec la notion de procédure effective, qui est censée jouir de propriétés analogues.

La seconde exigence, plus implicite dans le propos de G. Piccinini, est la prise en compte des contraintes d'implémentabilité. Celles-ci sont pour l'essentiel discutées dans le cadre de la justification de la contrainte d'exécutabilité, qui se voit ainsi articulée en de nombreuses sous-contraintes. Nous traiterons par conséquent l'exécutabilité en dernier, après avoir discuté les trois autres contraintes.

La première est la condition d'uniformité. Cette condition permet de voir sur un exemple que les considérations d'implémentabilité et de bonne analogie théorique sont indépendantes. Pour le comprendre, imaginons qu'un dispositif permette de calculer la fonction addition jusqu'à 100. Ce dispositif ne peut compter comme une implémentation de la fonction addition. Pour qu'un dispositif implémente une fonction de domaine infini, il est nécessaire qu'il soit applicable, au moins en théorie, à un nombre non-borné d'arguments de cette fonction. Dans le cas contraire, il est impossible de considérer un tel dispositif comme l'analogue physique d'un algorithme calculant l'addition : un algorithme s'applique uniformément à toute entrée de la fonction,

7. Nous précisons ici quelque peu la formulation de G. Piccinini, qui écrivait plus simplement que le processus doit produire des résultats corrects au moins parfois (*at least some of the time*, ([190], 741).

8. L'analogie avec les contraintes pesant sur la notion de procédure effective est énoncée très explicitement par G. Piccinini ([190], 741, nous soulignons) :

> a finite observer can follow an effective procedure because an effective procedure is executable, automatic, uniform, and reliable. *By analogy*, in order to be usable by a finite observer, a physical process must be executable, automatic, uniform, and reliable.

tandis qu'il sera ici nécessaire d'inventer un nouveau dispositif pour calculer certaines valeurs de la fonction donnée. Cependant, rien n'interdit qu'un tel dispositif soit implémentable. La contrainte d'uniformité est donc bien distincte de toute considération d'implémentabilité, et vise à garantir une analogie théorique correcte entre les notions de « fonction effectivement calculable »et de « fonction calculable par un système physique ».

La formulation de la contrainte d'automaticité par G. Piccinini nous semble critiquable, en ce qu'elle n'offre pas un bon analogue de la contrainte homonyme pour les procédures effectives. La contrainte d'automaticité est tout d'abord introduite pour les procédures effectives, avec la formulation usuelle que nous avons discutée à la section 1.1[9]. G. Piccinini formule ensuite la condition suivante pour les processus physiques ([190], 23) :

> An *automatic* physical process is a physical process that runs until it produces its results if it halts, or else it continues indefinitely as long as it doesn't break down.

Dans le cas d'une procédure effective exécutée par un calculateur humain, la condition d'automaticité est nécessaire pour préciser que le dit calculateur doit se borner à une exécution aveugle des instructions de l'algorithme, qui ne doit être en aucun cas corrigée ou supplémentée par un recours à l'intuition ou à l'invention. Lorsqu'elle est appliquée à un système physique, cette condition peut être retranscrite en affirmant qu'une fonction calculable par un système physique est une fonction calculable par une machine, au sens que nous avons défini plus haut (voir section 3.1). Dans le cas d'une tâche effectuée par une machine, l'observateur se contente d'encoder les entrées et de lire les sorties, et n'a jamais accès aux transformations de l'information opérées durant l'exécution du calcul, qui sont réalisées par la dynamique propre au système considéré. Une machine est par définition automatique, au sens où elle ne peut introduire aucune intuition ou invention dans sa propre dynamique.

La formulation de G. Piccinini n'offre pas d'analogue satisfaisant à la condition d'automaticité pour les procédures effectives, et doit être remplacée par la condition suivante :

> *Automaticité.* L'observateur fini ne doit avoir accès à l'information traitée par le processus physique qu'au stade de la lecture des entrées et des sorties.

La formulation de la condition de fiabilité nous semble encore plus critiquable, car elle rassemble sous un même chapeau deux idées distinctes. Cette condition est ainsi introduite pour les procédures effectives ([190], 736) :

9. G. Piccinini écrit en effet ([190], 7) :

   > The procedure requires no intuitions about the domain (e.g., no intuitions about numbers), no ingenuity, no invention, and no guesses.

.

> *Reliability* : when the procedure terminates, the procedure generates the correct value of the function for each argument after a finite number of primitive operations are performed.

Cette condition est ensuite reformulée pour s'appliquer aux processus physiques ([190], 741) :

> a *reliable* physical process is one that generates results at least some of the time and, when it does so, its results are correct.

L'analogie de surface entre ces deux formulations dissimule des différences importantes. Dans le cas des procédures effectives, il faut distinguer deux sens en lesquels on peut dire que la procédure engendre un résultat correct. Le premier désigne une propriété de la procédure effective elle-même, à savoir sa correction. Les procédures effectives peuvent ainsi être l'objet de démonstrations de correction [10], qui visent précisément à montrer que la suite, parfois excessivement longue et complexe, des instructions qui les composent permet bien d'arriver au résultat voulu dans tous les cas de figure.

Le second sens porte, non sur la procédure effective elle-même, mais sur l'exécution de cette procédure. Il s'agit ici de savoir si l'exécution du calcul par un dispositif concret, qui peut évidemment être un sujet humain armé d'un papier et d'un crayon, n'a pas produit d'erreur. Ces deux sens sont distincts, en ce qu'il est parfaitement possible de faire des erreurs dans l'exécution d'une procédure démontrée correcte. La condition de fiabilité énoncée par G. Piccinini confond donc deux problèmes distincts, la correction mathématique de la procédure utilisée pour calculer la fonction, et la fiabilité de son exécution par l'implémentation envisagée.

Il serait donc souhaitable de distinguer deux conditions en lieu et place de la condition de fiabilité énoncée par G. Piccinini. La première condition serait une condition de correction. Elle porterait sur la correction de la description mathématique de la procédure envisagée pour résoudre un problème donné.

> *Correction.* La description mathématique de la procédure de calcul peut faire l'objet d'une démonstration de correction, montrant qu'elle permet, pour toutes les instances du problème, de produire le résultat correct avec au moins une certaine probabilité.

La seconde condition serait plus proprement appelée condition de fiabilité. Elle porte quant à elle sur le processus physique destiné à implémenter la procédure décrite mathématiquement, et assure qu'il existe un processus permettant l'implémentation fiable de la procédure, et l'obtention du résultat correct garanti par la condition de correction, au moins une proportion raisonnable de fois. Cette deuxième condition serait plus avantageusement

10. Dans la terminologie usuelle, on parle plutôt de preuves de correction de programmes, le terme de procédure effective étant largement tombé en désuétude. Mais cette différence terminologique n'a pas d'importance pour notre propos.

présentée comme une sous-condition de la condition générale d'exécutabilité, puisqu'elle porte bien sur l'exécution concrète de la procédure.

> *Fiabilité*. Un processus physique implémentant l'exécution de la procédure doit permettre de produire, avec au moins une certaine probabilité, le résultat correct souhaité.

La contrainte d'exécutabilité est justifiée naturellement, une fois qu'on admet la conception épistémique du calcul exprimée par la contrainte générale d'utilisabilité. Si les procédures décrites par le modèle de calcul n'existent que dans l'abstraction mathématique, et ne peuvent être exécutées sur des entrées particulières, il est impossible de les employer pour connaître la solution d'instances de problèmes. À partir du moment où l'on désire faire exécuter le calcul par une machine, et qu'on renonce donc à l'emploi du calcul mental, l'exécution du calcul suppose l'existence d'un dispositif concret dans lequel on va pouvoir encoder l'entrée, faire exécuter les opérations nécessaires à la procédure de calcul, et récupérer le résultat par une opération de lecture. En outre, comme nous le verrons dans notre revue des modèles d'hypercalcul (voir chapitre 6), certaines procédures supposent l'exécution d'opérations impossibles à réaliser par calcul mental ou par calcul papier-crayon. L'implémentabilité de tels modèles dépendra donc crucialement de l'existence de systèmes jouissant des propriétés empiriques nécessaires à l'implémentation de telles opérations, et c'est précisément cette existence qui, dans le cadre des modèles d'hypercalcul, sera l'objet du débat. La contrainte d'exécutabilité est donc la porte par laquelle entrent les considérations d'implémentabilité dans la discussion des modèles d'hypercalcul. L'ensemble des sous-contraintes de la contrainte d'exécutabilité peuvent donc être lues comme des contraintes d'implémentabilité.

Considérons la liste de sous-contraintes énoncées par G. Piccinini, avec un résumé des justifications offertes en leur faveur :

1. *Lisibilité des entrées et sorties*. Les entrées et sorties doivent être lisibles. Cela signifie qu'elles doivent être encodées dans des quantités observables mesurables au degré nécessaire d'approximation.
2. *Tâche définissable indépendamment du processus de calcul*[11]. La tâche mathématique exécutée doit être définissable indépendamment du processus physique la réalisant.

   Cette deuxième condition est également justifiée par analogie théorique avec la notion de procédure effective. La tâche effectuée par une procédure effective est définissable indépendamment de l'algorithme qui la résout. En règle générale, il s'agit d'une fonction définie comme application par une formule du langage de ZFC au premier ordre. Sans une claire distinction entre la définition de la tâche par une formule du langage mathématique, et l'existence d'une procédure effective la

11. Nous traduisons par cette périphrase l'expression anglaise *process-independent rule*.

calculant, il n'est pas possible de se demander si une fonction est ou non calculable, et s'il existe plusieurs algorithmes pour la calculer [12]. De manière analogue, la contrainte d'indépendance de la définition de la tâche affirme la nécessité de définir la tâche mathématique effectuée par un dispositif empirique, indépendamment de la description de ce même dispositif.

Cette contrainte n'est pas triviale, en ce sens qu'elle permet d'éliminer certains processus physiques qui ne peuvent jouer le rôle d'implémentation d'un modèle de calcul. Une suite aléatoire de nombres n'est pas engendrable par une machine de Turing. Il existe cependant des processus quantiques permettant d'engendrer une telle suite aléatoire (voir [52]). De tels processus ont parfois été présentés comme « dépassant les capacités des machines de Turing. »Cette dernière affirmation est correcte au sens que nous venons de préciser. Mais un tel dispositif ne doit pas être compris comme l'implémentation d'un modèle d'hypercalcul. Il est en effet impossible de spécifier quelle fonction mathématique serait calculée par un tel processus. Par conséquent, si l'on peut bien dire qu'un tel processus accomplit quelque chose qu'aucune machine de Turing ne peut accomplir, ce quelque chose ne peut être décrit comme un calcul. Une telle remarque conceptuelle n'affecte évidemment en rien l'implémentabilité d'un tel processus.

3. *Répétabilité.* Le protocole doit être répétable. Cette condition est défendue par G. Piccinini sur la base d'un argument épistémologique, qui la rattache à la condition générale de fiabilité : un processus qui ne serait pas répétable ne serait pas fiable épistémologiquement. La répétabilité est une des conditions de la fiabilité, dans la mesure où un test basique de fiabilité est de vérifier qu'on obtient bien le même résultat lorsqu'on exécute le même calcul.
4. *Initialisabilité* [13]. L'observateur doit être capable d'initialiser le calculateur sur une entrée de son choix.

12. Ces considérations peuvent sembler triviales à toute personne ayant joui d'une éducation aux mathématiques supérieures. Il faut cependant rappeler qu'elles n'ont rien d'une évidence, ni d'un point de vue pédagogique, ni d'un point de vue historique. Si l'on demande à une personne dépourvue d'une telle éducation ce qu'est l'addition, il est fort probable qu'elle réponde par un exemple d'application de l'algorithme scolaire usuel, et non par la définition d'une certaine application $Add : \mathbb{N}^2 \longmapsto \mathbb{N}$. Il est particulièrement difficile au néophyte en mathématiques de distinguer une fonction de l'algorithme standard la calculant, lorsqu'il ne connaît aucun autre algorithme. Sans sombrer dans des spéculations historiques qui dépassent nos connaissances, il ne nous semble pas *a priori* impossible que des mathématiciens historiques aient pu commettre des confusions analogues.

13. Nous traduisons par ce barbarisme le terme anglais *settability*, qui peut lui-même être considéré comme un barbarisme scientifique, principalement employé en informatique. Le verbe anglais *to set* devant être lui-même traduit, dans un tel contexte, par « mettre à zéro »ou « initialiser », il nous a semblé qu'une telle traduction aurait à tout le moins le mérite de la précision et de la concordance des niveaux de langue, à défaut d'avoir celui de l'élégance.

Il nous semble nécessaire de renforcer cette condition, en ajoutant que l'observateur doit être capable, au moins en droit, d'encoder n'importe quelle entrée du domaine de la fonction calculée dans le système physique considéré. Dans le cas contraire, s'il existait par exemple une borne à la taille des entrées qu'on pouvait encoder comme entrée dans un tel dispositif, il ne serait pas possible de dire légitimement qu'un tel processus permet de calculer la fonction considérée, mais seulement la restriction de cette fonction à un sous-ensemble propre de son domaine (voir section 1.2).

5. *Constructibilité physique*. Un calculateur ne doit pas exister que sur le papier, mais il doit être réalisé empiriquement. La condition de constructibilité ne signifie pas que le système considéré doit être bâti *from scratch*, à partir de composants élémentaires, comme on l'imagine spontanément lorsqu'on songe à une machine bâtie par des ingénieurs, ainsi que le montre G. Piccinini ([190], 744) :

> Suppose that firing n pulses of energy into a sunspot reliably and repeatedly causes $f(n)$ pulses of radiation to be emitted, for some nontrivial f. The system that includes the sunspots as well as the apparatus for emitting and receiving energy pulses counts as physically constructible in the present sense.

Nous allons à présent commenter cette liste de sous-contraintes de manière linéaire.

G. Piccinini confond, en les rassemblant sous le chapeau *Lisibilité des entrées et sorties*, deux conditions distinctes. La lisibilité des entrées et des sorties ne doit en effet pas être confondue avec l' existence d'un état d'arrêt opérationnellement identifiable. L'observateur fini doit non seulement être capable de lire une valeur encodée dans le système, mais il doit en outre être capable de l'interpréter sémantiquement comme représentant le résultat du calcul, comme le remarque bien G. Piccinini lui-même ([190], 742) :

> For an output to be readable in the intended sense, the computing system must have a recognizable halting state. As a counterexample, consider a machine that purportedly computes characteristic function f. Suppose that for any argument of f, the machine is guaranteed to output its correct value at some future time. The machine gives its output as a one or a zero on a display. Suppose further that at any time, the value on the display may change from a one to a zero or vice versa and there is no known time after which the result of the computation is definitive. It may seem that the output is readable, because one can see whether it's a zero or a one. But it is not readable in the relevant sense, because the user never knows whether she is reading the desired value of f.

Une fois cette sous-contrainte bien distincte de la sous-contrainte de lisibilité, sa formulation exacte peut faire objet de discussion. G. Piccinini considère implicitement les calculs finitaires, qui doivent s'arrêter sur une valeur de sortie correcte et faire entrer la machine dans un état d'arrêt, et ignorent les calculs infinis comme ceux que nous avons pu rencontrer dans notre présentation de l'analyse récursive. Les machines de Turing de type-2, comme tout dispositif qui produirait une série d'approximations rationnelles d'un nombre réel, ne peuvent entrer dans un état d'arrêt. Néanmoins, il est crucial de pouvoir distinguer une étape intermédiaire du calcul de la production d'une approximation rationnelle, qui doit être considérée comme un préfix du résultat. Le lecteur se souviendra que, dans le cas des machines de Turing de type 2, cette distinction était assuré par l'existence d'une bande d'écriture du résultat sans possibilité d'effacement (*one-way output tape*). À défaut d'un état d'arrêt à proprement parler, la machine doit donc disposer d'un *état opérationnellement distinguable d'écriture d'un préfix du résultat* , sous peine d'être inutilisable.

Les sous-contraintes de définissabilité autonome de la tâche exécutée, de répétabilité, d'initialisabilité et de constructibilité nous semblent toutes convenablement justifiées par G. Piccinini. Cependant, la classification des contraintes de définissabilité autonome de la tâche exécutée et de constructibilité comme des sous-contraintes de l'exécutabilité est problématique.

La possibilité de définir la tâche exécutée indépendamment du processus physique employé pour l'exécuter est une contrainte conceptuelle distincte de l'exécutabilité : lorsque cette contrainte n'est pas satisfaisante, il n'est pas légitime de désigner le processus physique envisagé comme l'exécution d'un calcul. Elle permet de garantir une bonne analogie théorique avec la calculabilité usuelle par une procédure effective, puisqu'elle maintient la distinction entre la fonction calculée et l'algorithme la calculant. Elle permet ainsi d'exclure de la discussion certains processus physiques, bien qu'ils soient parfaitement implémentables, parce qu'ils sont sans pertinence pour la question de la calculabilité par une machine. On évite ainsi de sombrer dans la confusion conceptuelle menant à dire qu'un processus aléatoire viole la thèse de Church-Turing physique, parce qu'il « fait quelque chose qu'une machine de Turing ne peut pas faire », alors qu'un tel processus ne peut compter comme l'exécution d'un calcul. Il n'est donc pas pertinent de classer cette contrainte comme une sous-contrainte de l'exécutabilité, et il nous parait plus juste de la classer parmi les contraintes garantissant une bonne analogie théorique avec la calculabilité par une procédure effective, comme les contraintes d'uniformité, d'automaticité, et de correction.

L'identification de la constructibilité comme une sous-contrainte indépendante de l'exécutabilité nous semble également problématique. La notion de *constructibilité* de Piccinini ne se distingue pas de la notion d'*implémentabilité* que nous avons introduite plus haut (voir section 1.1), et dont on a vu qu'elle est moralement impliquée par la contrainte d'exécutabilité. À ce titre,

on peut considérer les différentes sous-contraintes de lisibilité des entrées et sorties, d'existence d'un état opérationnellement distinguable d'écriture d'un préfix du résultat, de répétabilité et d'initialisabilité comme des sous-contraintes détaillant la notion d'implémentabilité ou de constructibilité. L'existence de protocoles de mesure permettant de lire entrées et sorties, et de les identifier comme tels, la capacité à réinitialiser et à répéter la procédure sont toutes des questions opérationnelles portant sur l'implémentabilité du calcul. Il y a donc quelque chose de trompeur à présenter la constructibilité ou implémentabilité comme une sous-contrainte autonome de l'exécutabilité.

De manière analogue, la contrainte de fiabilité, bien distincte de la contrainte de correction, ne peut être présentée comme une contrainte autonome. La fiabilité est une propriété de l'exécution de la procédure de calcul par une machine, et en tant que telle elle se rattache à la contrainte d'exécutabilité. Par contre, tout comme la contrainte de constructibilité, il est difficile de la présenter comme une sous-contrainte autonome : la fiabilité est une qualité qu'on peut et doit exiger, par exemple, des processus de lecture des entrées ou des sorties, ou de l'initialisation, comme de toute partie du processus de l'exécution susceptible d'être sujette à des erreurs. La répétabilité, comme nous l'avons vu plus haut, est présentée par G. Piccinini lui-même comme un moyen de garantir la fiabilité de l'implémentation du processus.

Il serait donc plus élégant de dire que la contrainte d'exécutabilité implique la contrainte globale de trouver une implémentation fiable de la procédure de calcul. Cette contrainte globale d'implémentation fiable est ensuite détaillée par les sous-contraintes de lisibilité, d'initialisabilité, répétabilité, et d'existence d'un état opérationnellement distinguable d'écriture d'un préfix du résultat.

Pour résumer le fruit de notre analyse, comparons la liste originale de contraintes énoncées par G. Piccinini à celle, légèrement amendée, que nous défendons. G. Piccinini distinguait quatre contraintes fondamentales, d'*uniformité*, d'*automaticité*, de *fiabilité* et d'*exécutabilité*. Cette dernière contrainte est détaillée en quatre sous-contraintes : *lisibilité des entrées et des sorties*, *définissabilité indépendante du processus et de la tâche*, *répétabilité et initialisabilité*. Pour notre part, nous distinguerons les cinq contraintes fondamentales suivantes : *uniformité*, *automaticité*, *correction*, *définissabilité indépendante du processus et de la tâche*, *exécutabilité*. En constatant que la contrainte d'exécutabilité implique de trouver une implémentation fiable de la procédure de calcul, nous détaillons les sous-contraintes suivantes sur la notion d'implémentabilité fiable : *lisibilité des entrées et des sorties*, *existence opérationnellement distinguable d'un état d'écriture d'un préfix du résultat*, *répétabilité* et *initialisabilité*.

Ces contraintes sont manifestement nécessaires : un processus physique qui échouerait à satisfaire l'une d'entre elles ne constituerait pas une implémentation fiable d'un calcul. Le caractère suffisant de ces contraintes est

plus sujet à débat, et bien plus difficile à établir. La liste des sous-contraintes détaillant l'implémentabilité fiable peut en effet être considérée comme ouverte : de nouveaux débats, suscités par de nouveaux modèles non-standard, pourront mener à la formulation de contraintes supplémentaires. À ce titre, la formulation de la contrainte de constructibilité physique comme une sous-contrainte indépendante de l'exécutabilité obscurcit la discussion, dans la mesure où cette contrainte peut jouer le rôle de fourre-tout dans lequel seront dissimulées toutes les sous-contraintes encore inexplicitées.

S'il est impossible d'établir le caractère suffisant de cette liste, il nous sera possible d'établir son efficacité, en montrant qu'elle suffit à tout le moins à exclure nombre de modèles contemporains d'hypercalcul (voir chapitre 6). Elle permet également de légiférer sur l'usage de l'expression « modèle de calcul », et d'en dénoncer certains usages courants comme tendancieux, et d'autres comme frauduleux. Ainsi, la présentation des modèles B.S.S., RAM réelle et des fonctions $\mathbb{R}$-récursives de Moore comme des modèles de calcul, sans plus de commentaires, est tendancieuse. Elle omet de préciser que, dans la mesure où leur implémentabilité est très douteuse, il est bien possible qu'aucun calcul ne puisse jamais être effectué à partir de tels modèles. La présentation des machines à oracles non-récursifs comme un modèle de calcul est en revanche frauduleuse. Leur description ne contient en effet aucune explication de la procédure respectueuse des contraintes de Piccinini qui permettrait de calculer l'oracle, et n'exprime donc qu'une calculabilité purement hypothétique.

5.2 Sur le critère d'implémentabilité : deux poids, deux mesures ?

Par le truchement de la contrainte d'exécutabilité, l'implémentabilité devient donc une condition nécessaire pour tout modèle de calcul raisonnable. Puisque cette condition, en tant qu'elle introduit des considérations empiriques dans la discussion des limites du calcul, sera particulièrement importante dans la suite de ce travail, il nous semble nécessaire d'examiner une objection fréquemment formulée à son endroit. À notre connaissance, cette objection est apparue dans le contexte des discussions autour du caractère raisonnable de certains modèles d'hypercalcul. Comme nous allons le voir (voir chapitre 6), les modèles d'hypercalcul sont fréquemment la cible de critiques basées, implicitement ou explicitement, sur le critère d'implémentabilité. De nombreux modèles ont ainsi été critiqués soit parce qu'ils ne seraient pas implémentables pour des raisons de principe, soit parce qu'on n'aurait aucune idée claire d'une stratégie réaliste d'implémentation.

Face à ces critiques, certains partisans de l'hypercalcul ont répondu par une contestation de la légitimité du critère d'implémentabilité (voir, par exemple, [116], [139]). Leur argument part du constat courant que le modèle

des machines de Turing n'est pas implémentable. Celui-ci suppose en effet l'existence d'un nombre infini de cases, ou mémoire discrète infinie, pédagogiquement représentée comme une bande infinie de papier. Une telle mémoire infinie n'a jamais été implémentée. On pourrait s'interroger sur les raisons de principe interdisant une telle implémentation. Aux fins de la discussion, nous nous contenterons ici d'adopter la sagesse usuelle selon laquelle une telle mémoire n'est pas implémentable. Lorsqu'on dit d'un ordinateur donné qu'il constitue une implémentation d'une machine de Turing universelle, on considère en réalité une implémentation d'un automate fini.

À partir de cette hypothèse, il serait inapproprié de faire de l'implémentabilité une condition nécessaire du caractère raisonnable d'un modèle de calcul, puisqu'un modèle reconnu raisonnable par tous, à savoir la machine de Turing, n'est pas implémentable. Lorsqu'on reproche à un modèle de calcul non-standard de ne pas être implémentable, on pratiquerait indûment un système « deux poids, deux mesures »dans l'appréciation des modèles de calcul.

Cet argument est cependant fondé sur une mésinterprétation du rôle de la mémoire infinie discrète dans le modèle de Turing. Une interprétation correcte de ce trait du modèle est essentielle à une discussion générale des questions d'implémentabilité.

S'il termine, un calcul effectif ne demande qu'un nombre fini d'étapes de calcul, et donc de cases-mémoire, avant d'atteindre son résultat. La machine de Turing respecte ce trait élémentaire de l'effectivité, et aucune exécution d'une machine de Turing ne requiert l'usage d'un ensemble infini actuel de cases-mémoire. En revanche, il n'existe aucune borne fixée à la quantité de mémoire nécessaire pour exécuter une machine de Turing quelconque.

L'implémentation du modèle de la machine de Turing ne nécessite donc pas l'implémentation d'une mémoire infinie actuelle, mais seulement d'une d'une mémoire finie non-bornée. Il est ainsi parfaitement possible de définir les machines de Turing sans faire usage d'un ensemble infini de cases-mémoire. Il suffit de considérer une machine capable d'engendrer une nouvelle case-mémoire à chaque étape de calcul, en fonction des exigences en mémoire du calcul exécuté ([98], 89).

Pour que le modèle de la machine de Turing soit considéré comme implémentable, il n'est donc nul besoin de pouvoir implémenter une mémoire discrète infinie, mais uniquement de pouvoir faire croître indéfiniment le nombre de cases disponibles. L'impossibilité d'implémenter une mémoire infinie ne pèse en aucun cas sur la question de l'implémentabilité de la machine de Turing. L'emploi d'un ensemble infini actuel de cases au sein de ce modèle doit être interprété comme une représentation mathématique idéalisée de l'absence de borne supérieure sur le nombre de cases, et non comme une condition nécessaire de l'implémentation.

Même si cette précision pare l'argument sous sa forme courante, elle ne le réfute pas en toute généralité. Elle ne montre pas en effet que la machine de

Turing ainsi comprise soit implémentable. Notre question originale se réduit donc à la question suivante : est-il possible de faire croître indéfiniment la mémoire disponible pour une machine ? Cette question doit évidemment être lue comme une question de principe. Elle ne porte pas sur les limitations contingentes d'espace mémoire que subissent nos machines, mais sur des limitations fondamentales, ancrées dans la physique théorique, que subira toute machine.

Sans entrer dans une discussion théorique approfondie, nous nous contenterons pour le moment de constater que l'attitude dictée par la sagesse usuelle se renverse face à cette nouvelle formulation : autant il est usuel de dire qu'il est impossible d'implémenter une mémoire infinie, autant il est habituel de considérer qu'on peut faire croître indéfiniment la taille d'une mémoire. À l'heure actuelle, il n'existe aucune raison théorique de principe permettant de borner l'espace mémoire qu'on peut fournir à un ordinateur. En termes imagés, si dans la pratique chaque machine concrète possède une mémoire limitée, il n'existe aucune borne sur le nombre de barres de RAM qu'on peut lui ajouter. Si on admet la légitimité de notre manière de poser la question, il devient manifeste que la machine de Turing est un modèle implémentable.

Contrairement à ce qu'affirmait l'argument des partisans de l'hypercalcul, imposer le critère d'implémentabilité ne constitue donc pas un système indu de « deux poids, deux mesures. »La machine de Turing est bien implémentable, à condition de bien comprendre et le modèle de la machine de Turing, et ce que l'implémentabilité signifie. Il est donc parfaitement légitime d'appliquer le même critère d'implémentabilité à des modèles de calcul non-standard, y compris des modèles d'hypercalcul.

Avant de poursuivre notre discussion de la conception épistémique, nous souhaiterions tirer de notre discussion une leçon générale sur la relation entre limitations des ressources, théorie de la calculabilité et implémentation. Il faut tout d'abord souligner que l'inexistence de bornes supérieures en certaines ressources est nécessaire pour que la théorie de la calculabilité ait une pertinence pour l'étude des capacités des machines. Par exemple, la théorie de la calculabilité devient triviale si elle est appliquée à des calculs bornés en mémoire. Soit un automate disposant de n cases-mémoire, pouvant encoder 2^n bits d'information. Une telle machine ne peut calculer que des fonctions de domaine fini. Toute fonction de domaine fini est simulable par une machine de Turing : il suffit d'écrire une table d'instructions associant la sortie désirée à chaque entrée.

Des considérations analogues s'appliquent au temps de calcul. On peut démontrer que toute machine calculant en temps borné ne peut calculer qu'un sous-ensemble strict des fonctions calculables.

L'existence de bornes supérieures sur les ressources consommées par une machine trivialise donc les questions de calculabilité. On peut cependant défendre que toute machine concrète donnée est soumise à de telles limitations de ressources. Pour que la théorie de la calculabilité ne soit pas dépourvue de

toute pertinence pour les implémentations pratiques, il convient de préciser qu'on peut discuter de l'implémentabilité d'un modèle, tant qu'il n'existe pas de borne supérieure de principe sur ces ressources.

Lorsqu'on discute des propriétés de calculabilité de certaines machines, il faut donc comprendre ces dernières comme des *abstracta* et non des *concreta*. Toute machine concrète sera soumise à des bornes supérieures sur de nombreuses ressources pertinentes, comme le temps, l'espace mémoire, ou encore l'énergie, et ses propriétés de calculabilité seront triviales. La théorie de la calculabilité n'est pas pertinente pour une machine concrète donnée, mais pour une classe de machines concrètes dont les ressources pertinentes peuvent croître sans bornes. La théorie de la calculabilité a pour objet les limitations de cette classe idéale de machines, et non les limitations d'une machine concrète donnée. Elle ne peut donc s'appliquer aux implémentations qu'à l'aide d'une saine dose d'idéalisation, qui n'est limitée que par les bornes ultimes offertes par la physique théorique.

Nos dernières remarques doivent être comprises dans une perspective heuristique. Les résultats de calculabilité à espace ou à temps borné sont en effet obtenus au sein d'un modèle à temps et à espace discrets. Il n'est pas évident de les généraliser à d'autres modèles utilisant une représentation continue de l'espace et du temps, comme nous le verrons plus loin (voir section 8.2). Nous nous contenterons donc pour le moment de formuler l'hypothèse heuristique suivante : *si l'existence d'une borne supérieure sur une ressource R employée par un modèle implique que son pouvoir expressif est strictement inférieur à celui des machines de Turing, et si l'existence d'une telle borne est fondée en principe, on dira que le modèle est* trivial en R. Pour qu'un modèle de calcul soit Turing-complet, et a fortiori pour qu'il compte comme modèle d'hypercalcul, il faut montrer que ce modèle n'est pas trivial [14].

Sur ce point, notre position diverge de celle de G. Piccinini. Après avoir formulé une objection similaire à celle que nous venons de discuter, celui-ci écrit en effet (*ibid.*, p.745) :

> All that CT says is that any function that is intuitively computable is computable by some Turing machine. Computability by Turing machines is the upper bound on what CT deems intuitively computable. And acting as an *upper bound* does not require being physically constructible.

Comme nous venons de le montrer, il n'est pas requis de concéder aux objecteurs du critère d'implémentabilité que les machines de Turing ne sont pas implémentables. En outre, nous avons également défendu (voir section 1.2) que le critère d'implémentabilité faisait partie de nos intuitions préthéoriques sur ce qui est « intuitivement calculable. »Il serait donc erroné d'affirmer que

14. Nous verrons que des problèmes similaires se posent en théorie de la complexité, notamment pour le calcul quantique (voir section 8.3.2).

le modèle de Turing, ou les autres modèles universels historiques de la calculabilité, n'ont que faire des questions d'implémentabilité.

Un partisan de G. Piccinini pourrait prendre sa défense avec l'argument suivant. S'il est bien nécessaire que *certaines* machines de Turing soient implémentables pour que le modèle ne soit pas complètement irréaliste, il n'est pas nécessaire que toutes le soient, pour que le modèle puisse jouer sa fonction de borne supérieure de la calculabilité. Mais cette défense devient intenable si on revient sur les raisons de la soi-disante non-implémentabilité du modèle de Turing. Si l'on considère que toute machine de Turing nécessitant une quantité de mémoire non-bornée n'est pas implémentable, alors toute machine de Turing calculant une fonction de domaine infini sera exclue. Seules les machines de Turing calculant des fonctions de domaine fini seront implémentables. Dans un tel cas de figure, comme nous venons de le rappeler, la théorie de la calculabilité est triviale. Une telle position reviendrait donc à priver la théorie de la calculabilité de toute pertinence pour le calcul implémentable.

5.3 Défense de la conception épistémique du calcul : contre les objections de John Earman

Nous allons à présent discuter la conception *épistémique* du calcul par un système physique défendue par G. Piccinini, qui se trouve au fondement de la liste de contraintes que nous venons de discuter. Nous allons défendre cette conception, en montrant qu'elle seule permet de bien poser la question de la validité de la thèse de Church-Turing empirique.

La conception épistémique consiste à défendre qu'une procédure de calcul doit être une procédure nous permettant d'accéder à une connaissance mathématique, typiquement sur les valeurs d'une fonction, ou les solutions d'instances d'un problème. De cette conception fondamentale découle la contrainte d'utilisabilité énoncée par G. Piccinini : un processus physique ne peut compter comme calcul que s'il peut être exploité par un observateur fini pour accéder à une telle connaissance mathématique. L'intérêt d'une telle conception est qu'elle permet de mettre au coeur de l'étude de la thèse de Church-Turing empirique la question de l'accès épistémique à l'information théoriquement contenue dans un système.

Cette conception n'est cependant pas dépourvue d'opposants. Dans [85], le philosophe de la physique J. Earman nie explicitement que l'accessibilité épistémique soit une composante essentielle de la théorie de la calculabilité, ou, pour employer notre terminologie préférée, de la notion de modèle de calcul raisonnable. Il plaide à l'inverse pour une théorie généralisée de la calculabilité, formalisée selon une perspective formulée par R. Montague [169]. Nous ne nous attarderons pas sur les détails techniques de cette conception. Pour l'heure, il est suffisant de dire qu'au titre de la calculabilité généralisée,

n'importe quel système dynamique déterministe et invariant par translation dans le temps peut être conçu comme un calculateur. La calculabilité à la Turing, tout comme les modèles analogiques usuels, peuvent être vus comme de simples cas particuliers de cette compréhension générale de la calculabilité, obtenus par spécialisation des lois L et des variables par lesquelles le calculateur généralisé est décrit [15].

Le rejet par J. Earman du critère d'accessibilité épistémique n'implique pas une ignorance complète des questions d'implémentabilité :

> (...) the worry arises that the sense of computability involved is so general as to be useless ; for if L is allowed to range over every kind of deterministic law then presumably few if any function will fail to be generalized computable. The worry can be assuaged by emphasizing that the generalized concept of computability is relativized to laws L. The subject is given content by proving results about what functions are and are not generalized computable relative to what laws, especially the kind of laws encountered in mathematical physics [16].

La théorie de la calculabilité est ainsi simultanément généralisée et relativisée. La généralisation augmente considérablement la portée de la théorie, mais la relativisation aux lois admissibles en physique permettraient d'éviter le reproche d'irréalisme, et par là de trivialisation de la théorie.

Voyons à présent les raisons justifiant la position de J. Earman. Puisque l'interprétation du texte n'est pas ici sans difficulté, qu'on nous permette de le citer *in extenso*. À la fin de son argumentaire, J. Earman met en scène un dialogue avec un objecteur imaginaire, qui défend précisément, entre autres critiques, une interprétation épistémologique de la calculabilité [17] :

> (...) your Big Computer in the Sky does not make outputs available as information in the sense of symbols printed on paper and the like. *Response.* I view this aspect of computation as an engineering problem and not as a part of the analysis of computability per se.
>
> *Objection 2.* You are missing the point. The key idea is that a computer accepts and then outputs information. The exact definition of 'information' is not at issue here *except in so far as it necessarily involves symbolic representation.* Thus the maxim, 'No computation without symbolic representation', which is grossly flouted in your notion of generalized computability. *Response.* A definition of computable function might but need not make use of a coding of values by symbols. Whenever possible it seems best to characterize computability, digital or analogue, without

15. *Ibid.*, p.123.
16. *Ibid.*, p.123.
17. *Ibid.*, pp.123-124, nous soulignons.

> reference to coding; for while no one can doubt that the standard Turing machine coding - a string of m consecutive '1' to code the positive integer m - is an effective coding, the general problem of distinguishing effective from non-effective codings is equivalent to the problem of deciding when a function is effectively computable, the very problem at issue. (It should be obvious, for example, that if the coding of integers is not effective, then any set of integers could be enumerated by a Turing machine.) [...] *Objection 3.* What we want of a representation is that it enables us to access the information, and the standard systems of symbolic representation, so conspicuously lacking in your characterization of generalized computability, are designed to guarantee such success. *Response.* Epistemic access is an important issue. [...] But, to repeat, I do deny that computability is an epistemological concept. Turing computability can be presented in a purely abstract fashion, avoiding questions of representation and epistemic access; and just as the mathematical theory of Turing computability can be developped independently of these questions, so can the theory of generalized computability.

Il nous semble devoir distinguer deux arguments au sein de ce dialogue. Le premier serait qu'une définition de la calculabilité ne peut pas faire référence au codage sans tomber dans un cercle vicieux, car l'effectivité de la fonction de codage elle-même fait partie du problème censé être résolu par cette définition.

Cet argument fait référence à un problème posé pour la première fois par R. Montague. D'un point de vue historique, il existe deux énoncés distincts de la thèse de Church-Turing. Le premier, dû à Turing, choisit de donner une définition syntaxique des fonctions calculables comme des fonctions $f :\subseteq \Sigma^* \longmapsto \Sigma^*$. Le second, dû à Church et son élève Kleene, les définit de manière abstraite comme des fonctions $g :\subseteq \mathbb{N}^p \longmapsto \mathbb{N}$ [18]. Pour que ces définitions soient équivalentes, il faut qu'il existe une représentation des entiers par des chaînes de caractères, $\rho :\subseteq \mathbb{N} \longmapsto \Sigma^*$, ainsi qu'une représentation des chaînes de caractères par les entiers, $\rho' :\subseteq \Sigma^* \longmapsto \mathbb{N}$: l'existence de ces représentations est garantie par des fonctions de codage usuelles, comme par exemple la gödelisation.

Il est en outre nécessaire de déterminer la classe des représentations admissibles, car certaines représentations permettraient de violer la thèse de Church-Turing algorithmique. Soit D l'ensemble des entiers codant des machines de Turing, et m le code d'une entrée. Soit l'injection associant à (n, m)

18. Nous avons rencontré le même phénomène dans la définition des fonctions réelles calculables au titre de l'analyse récursive : celles-ci peuvent être définies abstraitement, à la Grzegorczyk et Lacombe, comme des fonctions $f :\subseteq \mathbb{R} \longmapsto \mathbb{R}$, ou syntaxiquement, à la Weihrauch, comme des fonctions $g :\subseteq \Sigma^\omega \longmapsto \Sigma^\omega$.

une chaîne de caractère représentant $2nm$, si n est le code d'une machine de Turing s'arrêtant sur l'entrée m, et une chaîne de caractères représentant $2nm + 1$, si n est le code d'une machine de Turing qui ne s'arrête pas sur l'entrée m. Il suffit alors d'appliquer la fonction de parité aux sorties de cette injection pour résoudre le problème de l'arrêt.

La réaction immédiate et naturelle est d'exclure une telle représentation parce qu'elle n'est pas effective. Cependant, en prononçant une telle exclusion, on emploie une notion intuitive d'effectivité d'une fonction $\rho :\subseteq \mathbb{N} \longmapsto \Sigma^*$, pour laquelle la thèse de Church-Turing n'est pas définie. Pour étendre la notion de fonction calculable à d'autres domaines qu'à ceux mentionnés dans les formulations de Church ou de Turing, on est donc contraint de recourir à une notion intuitive d'effectivité, dont la formulation de la thèse de Church-Turing était justement censée nous dispenser. Ce problème a récemment fait l'objet d'une nouvelle étude par N. Dershowitz et U. Boker, visant à formuler la thèse de Church-Turing pour des domaines dénombrables arbitraires [19].

Si nous comprenons bien la formulation quelque peu elliptique de J. Earman, celui-ci argumente que l'existence du problème de Montague justifie une définition abstraite de la calculabilité, qui ne fasse pas mention de la représentation symbolique des entrées et sorties. Dans le cas contraire, la définition ferait mention d'une représentation des données en chaîne de caractères sans pouvoir justifier de son effectivité, ce qui rendrait la définition inutile d'un point de vue fondationnel.

L'argument repose cependant sur une confusion. Il accorde un statut privilégié à la définition abstraite de la calculabilité à la Church, alors que le problème de Montague se pose de manière équivalente que l'on choisisse une définition abstraite ou une définition syntaxique de la calculabilité. Le problème de Montague n'est aucunement lié à l'introduction d'une représentation symbolique des données, mais à la nécessité de définir des représentations permettant d'étendre la définition des fonctions calculables à d'autres domaines que celui pour lequel elles furent originellement définies, comme par exemple des graphes ou des matrices. Il se pose donc tout aussi bien qu'on adopte une définition syntaxique ou une définition abstraite, qu'on considère des représentations des entiers dans les mots, ou des mots dans les entiers.

Le second argument est que, comme le montre la définition abstraite à la Church, la calculabilité a pu être définie sans que soit fait mention de conditions d'accessibilité épistémique dans sa formulation. Si on ainsi pu définir l'ensemble des fonctions calculables sans expliciter de conditions d'accessibilité épistémique, il n'y aurait pas de raison d'en faire un prérequis de toute théorie de la calculabilité. Cet argument est cependant aisément écarté. Les débats historiques autour de la définition des fonctions calculables ne né-

19. Pour plus de détails sur la formulation du problème de Montague, et pour la solution proposée par U. Boker et N. Dershowitz, voir [39].

cessitaient pas l'énoncé de condition d'accessibilité épistémique, parce que la question de l'accès épistémique est trivial dans le cas du calcul papier-crayon : il suffit de lire les symboles écrits sur le papier pour accéder aux données, aux étapes intermédiaires du calcul, et au résultat. Mais l'absence de pertinence d'une question pour un modèle du calcul papier-crayon n'implique pas l'absence de toute pertinence pour une compréhension généralisée du calcul. Nous l'avons déjà vu au sujet de certains modèles analogues, et nous le reverrons au sujet de plusieurs modèles d'hypercalcul : nombre de modèles échouent à être d'authentiques modèles de calcul, par absence d'accès épistémique à leurs entrées et sorties (voir chapitre 6). En ce sens, l'absence de condition d'accessibilité épistémique précise peut être conçue comme une faiblesse des modèles originaux, et non comme une preuve de la contingence de cette condition.

L'absence de mention de la représentation symbolique dans la définition de l'ensemble des fonctions calculables ne signifie pas que, d'un point de vue philosophique, le concept de calculabilité ne soit pas un concept épistémique, et que la question de la représentation symbolique n'aie aucun rôle à jouer dans la bonne compréhension de ce concept. Si elle n'est pas évoquée au niveau de la définition des fonctions calculables, la question de la représentation des données doit être évoquée dans la formulation d'un modèle réaliste. Si le lecteur se reporte à notre section sur l'analyse récursive (voir section 3.1), il verra que la définition des fonctions calculables au titre du modèle est une question indépendante de la question des représentations admissibles des éléments du domaine, qui exclut par exemple la représentation décimale. Mais la question de la représentation effective des données par des symboles est au coeur des motivations conceptuelles des définitions, puisque c'est elle qui impose de définir la calculabilité sur les réels comme une calculabilité de suites d'approximations. En outre, comme nous l'avons vu dans notre discussion de l'argument de modélisation (voir section 1.3.3), un Turing désireux de s'abstraire des contingences de la pratique du calcul papier-crayon discutait déjà des conditions d'accessibilité épistémique, comme lorsqu'il discute de la capacité du calculateur à distinguer les symboles. En suivant la lettre des définitions, J. Earman en trahit donc l'esprit.

Outre l'insuffisance de ses critiques explicites de la conception épistémique, la position de J. Earman souffre de la réduction, implicite dans son propos, de la question de l'implémentabilité à celle du caractère admissibles des lois gouvernant la dynamique du calculateur. Nous allons donner un contre-exemple à cette réduction, en montrant l'existence d'une classe de systèmes physiques, les systèmes sensibles aux conditions initiales, qui obéissent à des lois déterministes admissibles, mais ne peuvent nullement être utilisés pour implémenter un calcul. Là encore, on verra que la question de l'accessibilité épistémique de l'information théoriquement offerte par les lois du système est décisive pour la discussion de l'implémentabilité. Il nous faut pour cela définir les applications sensibles aux conditions initiales.

Pour simplifier l'exposition, on se restreindra ici aux cas des applications discrètes :

$$\text{Une application } f : V \longrightarrow V \text{ est sensible aux conditions initiales ssi} \quad (5.1)$$
$$\exists\delta\ \forall x \in V\ \forall\epsilon\ \exists y \|x - y\| < \epsilon \Longrightarrow \exists n \|f^n(x) - f^n(y)\| > \delta \quad (5.2)$$

Dans tout intervalle centré sur x, on trouvera donc au moins un point satisfaisant la condition énoncée [20].

Pour que la sensibilité aux conditions initiales se manifeste au niveau opérationnel, il ne suffit pas que la fonction décrivant la dynamique du système soit sensible aux conditions initiales : le système doit encore satisfaire deux conditions :

- il faut que δ soit supérieur à l'ϵ décrivant notre meilleure précision de mesure. Autrement, il serait impossible de distinguer expérimentalement les deux trajectoires de l'espace des phases données par $f^n(x)$ et $f^n(y)$.
- Il faut que la valeur de n à partir de laquelle la condition est satisfaite soit raisonnable. Si n était égal à trois fois l'espérance de vie de l'Univers, la sensibilité aux conditions initiales ne mènerait à aucune imprédictibilité *en pratique*, même si la première condition était satisfaite.

À notre connaissance, rien dans notre théorie physique ne garantit que ces deux conditions soient satisfaites. La sensibilité aux conditions initiales est une propriété mathématique qui peut faire l'objet d'une démonstration théorique. Les deux conditions ci-dessus sont en revanche des conditions empiriques qui sont vérifiées par l'expérience. De nombreux systèmes sont ainsi décrits par des systèmes d'équations sensibles aux conditions initiales, comme les doubles pendules ou certains systèmes météorologiques.

Pour notre présent sujet, l'essentiel est de bien voir que ces systèmes ne peuvent être exploités pour implémenter le calcul des lois dynamiques les décrivant. À quelque précision qu'on fixe l'observable x comme entrée du calculateur, il existera un temps n tel que la mesure de la valeur de x pourra donner deux valeurs distinguables à la précision δ de notre mesure. Bien que la dynamique du système soit décrite par une fonction, et soit donc théoriquement strictement déterministe, la relation entre les entrées et les sorties opérationnellement distinguables n'est pas même fonctionnelle. Lancer sur

20. Le concept de sensibilité aux conditions initiales est souvent confondu avec un concept plus fort, celui de chaos déterministe. Cette dernière notion admet tout d'abord plusieurs définitions non-équivalentes entre elles. Ensuite, dans sa définition topologique, une application est chaotique sur V, ssi elle a les propriétés de sensibilité aux conditions initiales, de transitivité topologique et de densité de l'ensemble des points périodiques dans V. La sensibilité aux CI n'est donc qu'une des conditions définissant une application chaotique (pour plus de détails, voir [83]). Un autre terme parfois employé de manière synonyme avec la sensibilité aux conditions initiales est celui de *système fortement non-linéaire*.

la même valeur opérationnelle, le système pourra rendre deux sorties opérationnellement distinctes. L'observateur sera donc incapable de savoir quelle valeur de sortie doit compter comme la valeur de notre fonction en son argument. Pourtant, les lois sensibles aux conditions initiales sont non seulement parfaitement admissibles d'un point de vue théorique, mais sont en outre numériquement simulables : l'ensemble des trajectoires admissibles peut être engendré par un ordinateur. Si la question de l'implémentabilité devait être réduite à celle de l'admissibilité des lois, alors on devrait compter comme calculateur des systèmes qui ne nous permette de résoudre aucun problème mathématique, pas même celui des solutions de leur propre dynamique.

Voyons à présent comment des considérations d'accessibilité épistémique permettent d'exclure de tels systèmes de l'ensemble des modèles raisonnables. Il convient de noter que les considérations d'accessibilité épistémique ne se réduisent pas à la prohibition d'une mesure arbitrairement précise. Pour employer un système déterministe comme calculateur, il n'est nullement besoin d'employer une mesure de précision arbitraire, mais uniquement de pouvoir, à partir d'un encodage de la valeur initiale à une précision δ, pouvoir connaître la valeur finale à une même précision δ. Pour reprendre une terminologie inspirée par notre examen de l'analyse récursive (voir section 3.1), il faut que le calcul effectué à l'aide du système obéisse à une condition de finitude renforcée : toute portion finie de la sortie à une précision ϵ doit dépendre d'une portion finie de l'input définie à une précision $\delta \geq \epsilon$. C'est précisément ce que nous interdit la sensibilité aux conditions initiales. D'après notre théorie, il existera toujours une valeur temporelle au-delà de laquelle la valeur de la sortie en deçà de ϵ dépendra des valeurs au-delà de ϵ. S'il est impossible d'utiliser un système sensible aux conditions initiales comme machine de calcul, c'est parce que la production d'une sortie opérationnellement distinguable à l'aide de notre meilleure précision de mesure dépendra de valeurs de l'entrée qui nous seront épistémiquement inaccessibles avec cette même précision. La question de l'accessibilité épistémique est donc bien au coeur des raisons qui permettent d'exclure une telle classe de systèmes comme des modèles de calcul raisonnables.

Contre J. Earman, nous conclurons donc qu'une théorie de la calculabilité, et notamment une discussion de la thèse de Church-Turing empirique, ne peut exclure les considérations d'accès épistémique, sous peine de sombrer dans la trivialité. La conception épistémique du calcul défendue par G. Piccinini est donc la seule à même d'assurer la bonne position des questions soulevées par la thèse de Church-Turing empirique.

5.4 Distinction de la thèse de Church-Turing empirique d'avec d'autres problèmes analogues

Nous venons de voir que, si l'on s'en tient à la conception épistémique du calcul, la thèse selon laquelle tous les systèmes physiques calculent des fonctions récursives est assurément erronée, dans la mesure où il n'est pas possible d'exploiter tout système physique pour réaliser un calcul. Cette thèse ne doit donc en aucun cas être confondue avec la thèse de Church-Turing physique. Celle-ci ne consiste nullement en l'affirmation que tous les systèmes physiques calculent uniquement des fonctions récursives, mais plutôt en celle que, si un système physique peut être utilisé comme une machine de calcul, alors il calculera des fonctions récursives [21]. Nous allons voir dans la présente section que la formulation de la conception épistémique a également l'avantage de permettre de distinguer la thèse de Church-Turing empirique d'autres thèses avec lesquelles elle est parfois confondue.

Une autre question peut être aisément confondue avec la thèse de Church-Turing empirique : les équations utilisées par nos théories physiques sont-elles calculables [22] ?

Cette question peut elle même être interprétée en deux sens. Elle peut porter sur les équations de la physique considérées *in abstracto*. En ce sens la question a d'ors et déjà reçue une réponse négative. M. Pour-El et I. Richards ont en effet démontré que l'équation d'onde admet des solutions non-calculables avec des conditions initiales calculables (voir [194]). Il est cependant bien connu que les équations de la physique, si elles sont considérées *in abstracto*, peuvent avoir des solutions que les physiciens s'accordent à considérer comme dépourvues de sens physique. Si elles nous disent quelque chose sur les propriétés mathématiques de nos équations, ces solutions ne nous disent donc rien sur ce qui est possible dans notre monde physique. L'exemple de M. Pour-El et I. Richards est dans ce cas. Comme le remarque en effet R. Penrose ([188], 244), un champ physique raisonnable est supposé être deux fois différentiable, ce qui n'est pas le cas des conditions initiales considérées par Pour-El et Richards. Pour que les solutions des équations soient douées de sens physiques, les physiciens requièrent que ces équations et leurs conditions initiales satisfassent certaines conditions, comme celle évoquée par R. Penrose.

Il existe une deuxième interprétation du problème de la calculabilité des équations de la physique : existent-t-il des solutions non-calculables d'équations de la physique qui soient physiquement raisonnables ? Ou peut-on au contraire prouver que toute équation satisfaisant des conditions raisonnables ne peut admettre de telles solutions ? À notre connaissance, cette question

21. Cette confusion est notamment commise par T. Paul et G. Longo dans [143].

22. Un système d'équations est calculable ssi, étant données des conditions initiales calculables, ses fonctions solutions sont calculables.

est ouverte à l'heure de l'écriture de ces lignes [23]. Cette question est rendue d'autant plus difficile que l'établissement d'une liste exhaustive des conditions auxquelles une équation doit être soumise pour être douée de sens physique, ainsi que la mathématisation de ces conditions, peuvent être des tâches théoriques très ardues.

S'il s'avérait qu'il existait des systèmes, dont la dynamique est décrite par une équation admettant des solutions physiquement raisonnables mais non-calculables, cela n'impliquerait nullement que de tels systèmes puissent être utilisés pour implémenter un modèle d'hypercalcul. L'exemple envisagé par M. Pour-El et I. Richards souffre de problèmes qui sont déjà familiers à ce stade de notre travail, comme la nécessité de réaliser des mesures de précision arbitraire pour pouvoir exploiter opérationnellement la non-calculabilité présente dans le modèle théorique. L'existence de solutions non-calculables de l'équation régissant la dynamique d'un système doit donc être conçue comme une condition nécessaire, mais non suffisante, pour qu'un tel système puisse être conçu comme l'implémentation d'un modèle d'hypercalcul. La question de la calculabilité des équations de la physique, dans sa deuxième interprétation, peut donc être vue comme un sous-problème de la question de l'existence d'un modèle raisonnable d'hypercalcul. Elle peut aussi être simplement vue comme une question portant sur les outils mathématiques permissibles en physique théorique (voir [110]).

Il existe encore d'autres problématiques unissant physique théorique et calculabilité, comme la possibilité d'implémenter une machine de Turing universelle dans un système physique. Il existe plusieurs tentatives en ce sens. C. Moore a ainsi pu encoder une machine de Turing universelle dans une particule classique se déplaçant dans un potentiel tridimensionnel composé de miroirs plans et paraboliques [170]. La tentative la plus connue est cependant celle de E. Fredkin et T. Toffoli, qui ont démontré la possibilité théorique d'implémenter une machine de Turing universelle à l'aide d'un système composé de boules de billards idéalisées s'entrechoquant de manière parfaitement élastique, et de murs parfaitement solides [102]. Toutes ces tentatives demeurent cependant purement théoriques, car elles considèrent des systèmes fortement idéalisés. L'existence d'une implémentation robuste d'une machine de Turing universelle par un système dynamique reste donc une question ouverte. En ignorant ces difficultés aux fins de la discussion, on peut remarquer que de tels systèmes jouissent de propriétés théoriques remarquables : certaines propriétés de leur comportement à long terme deviennent incalculables. Pour le voir, il suffit de remarquer que E. Fredkin et T. Toffoli ont défini un état d'arrêt opérationnellement distinguable dans leur système. L'indécidabilité du problème de l'arrêt se traduit donc en l'in-

23. R. Penrose conjecture ainsi la calculabilité des équations connues de la physique, tout en admettant que cette conjecture ne constitue pas autre chose qu'un avis éclairé ([188], 231).

décidabilité de certaines propriétés dynamiques de long terme pour un tel système.

Certains auteurs ont pu en conclure hâtivement que ces mêmes propriétés devenaient imprédictibles [24]. Une telle assimilation entre imprédictibilité et indécidabilité repose sur la confusion, que nous avons déjà précédemment dénoncée (voir section 2.3.1), entre problème et instance de problème. S'il n'existe pas d'algorithme permettant de décider la propriété dynamique en question en toute généralité, rien n'interdit de la résoudre dans certains cas. Une telle remarque n'aura en soi rien de surprenant pour le lecteur familier de la pratique de la physique, qui saura que les physiciens peuvent se montrer experts dans la résolution de cas particuliers d'équations qu'ils sont incapables de résoudre dans le cas général. Si l'indécidabilité de certains problèmes dynamiques peut représenter une nouveauté, c'est parce qu'elle ferait de cet état de fait de la pratique un état de droit : il sera mathématiquement impossible aux physiciens de faire mieux que de trouver des solutions particulières à un problème dynamique général, ce qui compliquerait sensiblement le travail d'extraction de prédictions testables du modèle mathématique. Il s'agirait là d'un trait théorique intéressant pour tout philosophie de la relation entre mathématiques et sciences empiriques. En revanche, cela ne contribue aucunement au débat sur la possibilité d'un modèle raisonnable d'hypercalcul : ces systèmes ne peuvent en aucun cas être utilisés pour implémenter un modèle d'hypercalcul, puisqu'ils sont justement des implémentations d'une machine de Turing universelle.

Conclusion

Dans ce chapitre, nous avons résolument opté en faveur d'une version amendée de la conception épistémique du calcul, proposée pour la première fois par G. Piccinini. Nous avons défendu cette conception contre diverses objections, en montrant notamment la nécessité de la condition d'implémentabilité, et plus particulièrement celle d'accès épistémique aux entrées et aux sorties. Enfin, nous avons montré que cette conception jouait le rôle qu'on est en droit d'attendre d'une définition philosophique, à savoir celui de censurer certains usages, de distinguer les questions indépendantes, et d'éviter de sombrer dans la confusion conceptuelle.

Ce cadre conceptuel étant posé, nous allons pouvoir procéder à une revue des modèles d'hypercalcul. Avant de commencer, il importe de rappeler les trois objectifs poursuivis par cet exercice. Le premier porte sur la valeur de

24. On peut trouver un exemple d'une telle confusion chez C. Bennett [27] :

> (...) for [a dynamical system] to be undecidable means that essential aspects of its long-term behaviour -such as whether a trajectory ever enters a given region- though determined, are unpredictable even from total knowledge of the initial condition.

vérité de la thèse de Church-Turing empirique : les modèles d'hypercalcul existants offrent-ils des pistes sérieuses pour violer cette thèse, ou justifient-ils au contraire une attitude sceptique à l'égard de l'hypercalcul ? Le second porte sur la signification empirique que l'on voudrait accorder à cette proposition : quelles contraintes ferait-elle peser sur la physique ? Cette deuxième question est indépendante de la première. Quelle que soit notre position sur la valeur de vérité de la thèse empirique, elle nous permet de déterminer à quoi nous engage notre acceptation ou notre refus de cette thèse. Le troisième est de voir si le débat autour des modèles d'hypercalcul permet de formuler un argument positif en faveur de l'interprétation empirique de la calculabilité. Nous avions en effet conclu, à la fin de notre second chapitre, qu'on pouvait accorder sur le principe que la thèse empirique était bien une proposition empirique, au sens où il n'existait pas d'objection logique ou méthodologique à une telle interprétation. Mais il manquait encore un argument positif permettant de soutenir cette même interprétation. Nous verrons si le débat sur les modèles d'hypercalcul nous permet de formuler un tel argument.

Chapitre 6

Une revue raisonnée des modèles d'hypercalcul

Introduction

Jusqu'à présent, nous avons parlé de la thèse de Church-Turing empirique en toute généralité. Nous allons maintenant restreindre notre attention à la thèse de Church-Turing physique, pour deux raisons. La première provient de notre interrogation sur la possibilité de principe d'une machine hypercalculante. La physique théorique a l'avantage d'offrir une compréhension claire de ce qu'est une possibilité de principe : est possible en principe tout modèle cohérent avec les postulats de nos meilleures théories fondamentales. De plus, si on adopte une position réductionniste, la physique théorique doit à terme fournir les limites ultimes du traitement de l'information auxquelles sera soumis tout processus empirique. Les limites données par la physique doivent en tout état de cause être examinées en priorité.

La seconde raison provient de notre volonté d'extraire la signification empirique des principes computationnels. Nous souhaitons extraire des hypothèses empiriques précises, à partir d'une revue raisonnée des modèles d'hypercalcul. Pour atteindre, cet objectif, il nous faut distinguer entre deux types de modèles. Nous dirons qu'un modèle est *physiquement défini* ssi *l'ensemble des opérations constituant la procédure de calcul est défini dans les termes d'une théorie physique*. Un modèle qui n'est pas physiquement défini sera qualifié de *notionnel*[1].

Les modèles notionnels consistent le plus souvent en une variation mathématique sur les hypothèses décrivant une procédure effective. On pourra ainsi envisager des machines de calcul exécutant une infinité d'étapes en un temps réel fini, des programmes de taille infinie, de mémoires encodant une

1. Le qualificatif de « notionnel », contrairement à celui de « physiquement défini »est déjà employé dans la littérature, sans être jamais explicitement défini (voir ainsi [101], [216]). Capturer et éclaircir l'intuition comprise dans cet emploi informel fait partie des fins de notre définition.

infinité d'information, ou de processeurs calculant une infinité d'opérations en parallèle. Si l'on exclut la possibilité d'implémenter de tels modèles, on peut considérer une telle exclusion comme une hypothèse empirique. Cependant, une telle approche empêcherait de distinguer la signification empirique précise de la thèse de Church-Turing empirique. Mémoire, étape de calcul, processeur ou parallélisme sont des notions informatiques. Pour leur conférer un véritable contenu empirique, il faut les définir en termes physiques, en donnant un modèle explicite des différents traits du modèle dans une théorie physique donnée. Cet effort de modélisation est aussi un processus de traduction des concepts informatiques en concepts physiques. S'il est naturel de considérer que l'exclusion de ces modèles notionnels repose sur des hypothèses empiriques, nous verrons qu'il est impossible d'articuler une signification empirique précise des limites de la calculabilité sans ce processus de traduction.

Notre deuxième raison de nous concentrer sur la thèse physique rejoint en dernier recours la première. Pour pouvoir exclure ou accepter un modèle notionnel sur une base principielle, il est nécessaire de définir physiquement ce modèle en des termes de physique fondamentale, afin de voir si son implémentation est compatible avec les principes fondamentaux de nos meilleures théories. Que ce soit pour déterminer la valeur de vérité de la thèse de Church-Turing empirique, ou pour extraire sa signification empirique, il faut se restreindre aux modèles physiquement définis, et donc à la thèse de Church-Turing physique.

La distinction entre modèle notionnel et modèle physiquement défini ne recoupe nullement la distinction entre modèles implémentables et modèles impossibles à implémenter. La machine de Turing, bien qu'elle soit parfaitement implémentable, n'est pas un modèle physiquement défini. L'information n'y est pas explicitement encodée dans l'état d'un système, l'opération de lecture de la mémoire n'est pas définie comme une mesure, la dynamique permettant d'altérer le contenu d'une case-mémoire et de déplacer la tête de lecture le long de cette mémoire n'est pas explicitée. Cela n'affecte en rien l'implémentabilité du modèle, puisqu'il n'est point besoin d'une modélisation physique si explicite pour réaliser que les instructions d'une machine de Turing sont parfaitement exécutables par un calcul papier-crayon. Par contraste, le modèle des circuits quantiques est un modèle physiquement défini. L'information est encodée dans l'état de systèmes quantiques, l'exécution du calcul y est modélisée par une suite de transformations unitaires, et le calcul s'achève par une opération de mesure. En revanche, il n'est à l'heure actuelle pas certain que ce modèle soit implémentable pour plus de quelques centaines de qubits (voir section 8.2).

Définir physiquement un modèle ne signifie pas non plus fixer une stratégie d'implémentation. Le modèle des circuits quantiques est implémentable par de multiples stratégies, en choisissant par exemple comme systèmes des photons ou des ions piégés. Ce qui compte est que les systèmes considérés

jouissent des bonnes propriétés théoriques pour implémenter la procédure décrite par l'algorithme quantique. Un modèle physiquement défini est décrit en termes théoriques, tout comme un modèle notionnel. Ce qui fait sa spécificité, c'est le langage théorique exact dans lequel il est décrit, qui ne comprend que des termes définis dans une théorie physique.

Pour répondre à nos deux questions, nous examinerons les critiques de l'implémentabilité des modèles d'hypercalcul, et tâcherons de dégager les hypothèses physiques aux fondements de ces critiques. Nous verrons qu'aucun des modèles d'hypercalcul décrits dans la littérature ne peut être le point de départ d'un programme d'implémentation. Par induction sur les critiques des différents modèles, nous tâcherons donc d'isoler les propositions les plus générales permettant d'asseoir une défense de la thèse de Church-Turing physique.

Puisque nous viserons l'exhibition des hypothèses communes, nous ne prétendrons pas à la plus grande minutie dans l'exposé des détails de chaque modèle. Bien que l'exhaustivité soit très dure à atteindre, il existe déjà plusieurs revues de qualité embrassant un grand nombre de modèles, auxquelles nous reporterons le lecteur avide d'explorer la singularité de chacun d'eux (voir [190], [71], [181]).

Nous commencerons par examiner un modèle notionnel, afin d'illustrer les problèmes soulevés par la traduction des notions informatiques en termes physiques. Nous nous consacrerons ensuite à l'examen de modèles physiquement définis, avant d'explorer plus en détails les hypothèses communes qui ressortiront des critiques de ces modèles.

Nous examinerons ensuite les tentatives de caractérisation de la thèse de Church-Turing physique, initiées par les travaux de R. Gandy, pour en comparer les résultats avec ceux de notre revue critique.

6.1 Un modèle notionnel : les machines de Turing accélérantes

Notre premier modèle présente un double intérêt. Tout d'abord, il permet de révéler certaines difficultés que posent les définitions standard de la complexité en temps et en espace, et leurs rapports au temps et à l'espace consommés par une machine concrète exécutant un calcul. Ensuite, il permet d'illustrer les difficultés qu'il y a à fonder la critique d'un modèle notionnel sur des hypothèses empiriques précises.

Le temps de calcul est usuellement défini comme le nombre maximal d'opérations nécessaires pour résoudre une instance en fonction de sa taille. Cette appellation semble naturelle, dans la mesure où il est intuitif de considérer qu'un plus grand nombre d'opérations nécessitera un intervalle de temps réel plus important pour être exécuté par une machine. Une telle définition est donc à la fois abstraite et pertinente pour l'implémentation.

Elle est abstraite parce qu'elle ne prend pas en compte les progrès du matériel permettant d'améliorer le temps d'exécution d'une opération donnée. Elle est pertinente pour l'implémentation parce que le nombre d'opérations sera un facteur déterminant du temps d'exécution réel du calcul. S. Arora et B. Barak [16], pour illustrer ce point, comparent les performances de l'algorithme scolaire usuel pour la multiplication d'un entier a par n, avec celles d'un algorithme particulièrement grossier, consistant à additionner $a\ n-1$ fois à lui-même. En exécutant le premier algorithme sur des instances de taille suffisamment importantes, un être humain armé de papier et de crayon pourrait sans difficulté battre en temps réel les meilleurs machines actuelles, si elles utilisaient le second algorithme. Non seulement l'efficacité de l'algorithme est un véritable facteur dans la détermination du temps réel d'exécution, mais elle peut être plus importante que tous les facteurs technologiques pertinents.

Une telle intuition repose sur une hypothèse rarement explicitée dans les manuels, à savoir l'impossibilité d'effectuer une infinité d'opérations en un intervalle de temps réel fini. Pour le comprendre, supposons qu'une machine donnée puisse exécuter une étape de calcul en un temps arbitrairement petit. Pour fixer les idées, supposons qu'une telle machine exécute la première étape d'un calcul en une unité de temps, puis la deuxième en une demi-unité, la troisième en un quart... de telle manière que la n-ième étape de calcul soit exécutée en un temps égal à $\frac{1}{2^{n-1}}$ unités. Ce modèle spéculatif est connu sous le nom de machines de Turing accélérantes (*Accelerating Turing Machines (A.T.M.s)*), parfois appelé plus familièrement 'effet Zénon'[65]. Comme le remarquait déjà Hermann Weyl, le pouvoir calculatoire d'une telle machine surpasserait tout modèle connu, puisqu'elle permettrait d'exécuter [242] :

> an infinite sequence of distinct acts of decision within a finite time ; say, by supplying the first result after $\frac{1}{2}$ minute, the second after another $\frac{1}{4}$ minute, the third $\frac{1}{8}$ minute later than the second, etc. In this way it would be possible ... to achieve a traversal of all natural numbers and thereby a sure yes-or-no decision regarding any existential question about natural numbers.

En termes modernes, un tel modèle de calcul permettrait de résoudre toute question Σ_1^0 ou Π_1^0. Notre conception de la calculabilité est donc fondée sur l'intuition qu'une telle machine n'est pas réaliste, et qu'il est impossible d'exécuter une infinité d'opérations en un intervalle de temps fini. Si une telle performance était possible, la théorie de la calculabilité telle que nous la connaissons s'effondrerait, puisque tout ensemble récursivement énumérable serait récursif. En outre, la théorie de la complexité n'aurait plus de sens pour un tel modèle, puisque tous les problèmes calculables seraient solubles en un temps unique de deux unités de temps. L'appartenance à différentes classes de complexité n'aurait donc plus de pertinence pour l'évaluation de l'intervalle de temps réel pris par une machine pour exécuter un calcul. Si les résultats de complexité en temps permettent d'estimer le temps réel d'exécu-

tion du calcul, c'est parce qu'aucune machine n'est censée violer l'hypothèse qu'un tel modèle est impossible [2].

Cette hypothèse est donc au fondement de la pertinence pour l'implémentation de la définition usuelle du temps de calcul, à tout le moins pour les modèles à temps discret. L'exécution d'une infinité d'opérations en un intervalle de temps fini étant nommée « supertâche »(*supertask*), on peut dire que les théories de la calculabilité et de la complexité sont fondées sur l'hypothèse que les supertâches sont impossibles.

Peut-on dire que l'implémentation d'une supertâche est interdite par nos principes physiques fondamentaux ? À notre connaissance, tous les travaux réalisés sur la question tendent à répondre par l'affirmative, mais pas forcément pour les mêmes raisons. Une étude des différentes tentatives pour fonder physiquement l'impossibilité des supertâches va nous permettre d'illustrer un point général, à savoir la difficulté qu'il y a à expliciter des hypothèses physiques spécifiques derrière la plausible impossibilité de certaines performances calculatoires.

Le premier argument consiste à souligner que l'exécution d'une étape de calcul par une A.T.M. prendra rapidement un intervalle de temps inférieur au temps de Planck, $t_P = \sqrt{\frac{\hbar G}{c^5}} = 5,39121 \times 10^{-44} s$, avec c la vitesse de la lumière dans le vide, G la constante gravitationnelle et $\hbar$ la constante de Planck. L'objection à l'égard du modèle peut alors prendre deux formes. On peut simplement souligner que le temps de Planck est le temps minimal auquel on puisse conférer une signification physique au sein de nos théories fondamentales actuelles. Il est donc impossible de formuler, au sein de ces théories, un modèle physiquement défini qui permettrait de décrire une étape de calcul s'exécutant en un temps inférieur au temps de Planck. Un tel argument interdit donc de définir un modèle d'A.T.M. à l'aide de nos théories actuelles, et assombrit les perspectives de formuler un tel modèle dans un futur proche. Cependant, il ne constitue pas un argument de principe, fondé sur une hypothèse empirique, contre l'existence d'un tel modèle. C'est ce qu'offre la deuxième forme de l'argument. Celui-ci, appuyé sur certains courants de la théorie physique contemporaine, affirme l'impossibilité de principe de définir un intervalle de temps inférieur au temps de Planck dans la future théorie de la gravité quantique. L'exécution d'une étape de calcul, comme tout événement physique, ne pourrait donc se dérouler en un temps arbitrairement petit. Un tel argument est cependant plus spéculatif, comme toute affirmation sur l'avenir de la physique à l'échelle de Planck.

Un deuxième argument consiste à s'appuyer sur les limitations posées par la relativité restreinte sur la vitesse de propagation de l'information. Il consiste à supposer que l'exécution d'une étape de calcul implique le déplacement d'un système doué de masse, comme le curseur se promenant sur la bande de la machine de Turing, ou une communication entre deux parties

2. Voir section 8.2, pour plus de détails sur cette question.

distinctes du calculateur. Le déplacement d'un système doué d'une masse, comme la communication d'information, sont censés être bornés par la vitesse de la lumière c au sein de la théorie de la relativité restreinte.

Comme J. Earman le remarque dans ([86], 104), l'argument de la limite de vitesse c, déjà employé par P. Benacerraf et H. Putnam en 1983 [26], n'est pas suffisant pour exclure de tels modèles [3]. Il ne suffit pas de dire que les parties mécaniques ne peuvent se déplacer à cette vitesse, mais il faut aussi montrer que la distance qu'elles doivent parcourir ne peut rapetisser si vite qu'elle permette de passer sous c. Même si une telle machine est cinématiquement possible, elle pourra néanmoins être exclue sur la base de considérations dynamiques [4].

Pour pouvoir formuler des arguments dynamiques, il importe de fournir une modélisation plus explicite des opérations de calcul. Un troisième argument, formulé par S. Aaronson [3], va ainsi faire usage de la relativité générale, et se base sur l'étude du coût en énergie d'une opération de calcul. On suppose qu'une implémentation d'A.T.M. emploierait des photons, puisque ceux-ci sont le moyen le plus rapide de transporter de l'information [7]. La croissance en vitesse des opérations est modélisée par l'augmentation en fréquence de photons.

La critique de l'implémentabilité du modèle est alors fondée sur l'équation relativiste générale du rayon de Schwarzschild. Celle-ci qui décrit le rayon-limite d'une sphère pouvant contenir la masse M d'un corps avant que celui-ci ne s'effondre en un trou noir :

On utilise donc l'équation :

$$r = \frac{2GM}{c^2} \tag{6.1}$$

avec $G = 6.67384 \times 10^{-11} m^3 kg^{-1} s^{-2}$ la constante gravitationnelle, et M la masse du corps considéré.

On cherche à déterminer l'énergie-masse des photons en fonction de la fréquence des photons, décrite par l'équation

$$E = h\nu \tag{6.2}$$

3. J. Earman n'évoque pas directement dans ce passage les Machines de Turing Accélérantes, mais la lampe de Thomson, un autre modèle notionnel censé réaliser une infinité d'opérations en un intervalle de temps fini. Mais les arguments sont strictement identiques dans les deux cas.

4.

> A demonstration is needed to rule out as a kinematic possibility that the operation of the device is arranged so that with each successive step the distance the parts have to move (as in ordinary stroll from a to b) shrinks sufficiently fast that the bound c is never violated. Of course, even if the device can be shown to pass muster at the kinematic level, it may still fail to satisfy necessary conditions for a dynamically possible process (...).

avec E l'énergie du système, m sa masse, v la fréquence et $h = 2\pi\hbar$, $\hbar = 1.0545 \times 10^{34} joule - sec$.

Un photon étant dépourvue de masse, on convertit l'énergie en un moment par l'équation relativiste :

$$E^2 = p^2c^2 + m^2c^4 \tag{6.3}$$

avec E l'énergie du système, m sa masse, v la fréquence et $h = 2\pi\hbar$, $\hbar = 1.0545 \times 10^{34} joule - sec$ la constante de Planck.

On a donc déduit qu'un photon dont la fréquence excéderait $10^{43} Hz$ aurait une énergie de $10^{9} joule$. La distance parcourue par un photon en $10^{43} s$ serait de l'ordre de $10^{-35} m$. Si on fixe cette longueur comme rayon de Schwarzschild pour déterminer la masse associée, $10^{-11} kg$, et qu'on convertit cette masse en énergie, on obtient une énergie limite de l'ordre de $10^{6} joule$. Tout photon d'une fréquence supérieure à $10^{43} Hz$ aura donc une énergie-masse supérieure à celle autorisée par l'équation du rayon de Schwarzschild, et entraînera la formation d'un trou noir.

La conclusion de l'argumentation dépend alors d'une position en physique théorique, selon laquelle l'information contenue dans un trou noir serait inaccessible à un observateur extérieur (voir [162], 15 et section 6.5.3, pour les difficultés soulevées par cette idée). Quand bien même on admettrait que notre système continue à fonctionner comme un calculateur tout en subissant un effondrement gravitationnel, l'information du résultat serait donc inatteignable pour l'utilisateur, violant la condition de lisibilité des sorties. L'argument a cependant la faiblesse de dépendre d'une prise de position dans des discussions en cours en physique théorique.

La deuxième faiblesse de cet argument est l'absence de raison principielle pour l'emploi de photons dans une implémentation d' A.T.M.. Certes, on peut arguer que l'onde électromagnétique est le moyen le plus précis à notre disposition permettant d'effectuer des mesures directes de temps et de distance[7]. Il n'y a cependant pas d'argument fondamental selon lequel l'implémentation d'une A.T.M. devrait impliquer des mesures *directes* de distance ou de temps. Le choix des photons relève donc d'une décision dont on ne peut, dans l'état actuel de la discussion, déterminer si elle peut être justifiée d'un point de vue principiel, ou si elle relève d'une simple stratégie d'implémentation.

Une troisième approche, formulée cette fois au sein d'un modèle purement quantique, le théorème de Margolus-Levitin, peut être utilisé pour exclure la possibilité des supertâches.

Le modèle de Margolus-Levitin s'appuie sur une définition simple et naturelle du nombre d'opérations effectuables par un système physique en une unité de temps, à savoir le nombre d'états opérationnellement distinguables que ce système peut traverser en une unité de temps. En partant du constat que la théorie quantique est notre meilleure modélisation du comportement

fondamental de la matière, on choisit de décrire ces états en termes quantiques. Il existe alors un critère pour qualifier deux états d'opérationnellement distinguables par une mesure projective : deux états quantiques $|\phi\rangle$ et $|\psi\rangle$ sont distinguables ssi ils sont mutuellement orthogonaux, $|\langle\phi|\psi\rangle| = 0$.

Le travail de Margolus et Levin prend sa source dans une interprétation de la relation d'incertitude de Heisenberg en énergie-temps :

$$\Delta E \Delta t \geq \hbar \tag{6.4}$$

Celle-ci n'est pas lue comme signifiant qu'un intervalle de temps Δt est nécessaire pour mesurer l'énergie E d'un système dans un état ψ à une précision ΔE, mais plutôt qu'un état quantique avec un écart-type ΔE sur son énergie aura besoin d'un temps $\Delta t = \frac{\pi\hbar}{2\Delta E}$ pour évoluer vers un état orthogonal à ψ. Cette interprétation ne fixe en elle-même aucune limite sur le temps nécessaire, puisque ΔE peut être arbitrairement large.

N. Margolus et L. Levitin ont réussi à démontrer des résultats analogues pour l'énergie maximale et l'énergie moyenne d'un système [162], [166]. On considère un espace des états de dimension infinie dénombrable. Un état quantique arbitraire $|\phi\rangle$ peut être représenté comme une superposition d'états propres de l'énergie :

$$|\phi\rangle = \sum_{n=0}^{\infty} c_n |E_n\rangle \tag{6.5}$$

On considère que les états d'énergie sont numérotés de telle manière que les valeurs E_n associées aux états $\{E_n\}$ soient croissantes, et on fixe le zéro d'énergie à E_0.

On note $v_\perp$ la vitesse à laquelle un système peut traverser l'ensemble des états mutuellement orthogonaux accessibles. On montre alors, si E_{max} est l'énergie maximale accessible au système,

$$v_\perp \geq \frac{E_{max}}{\hbar} \tag{6.6}$$

Si au lieu d'un maximum d'énergie, on fixe une énergie moyenne E on a

$$v_\perp \geq \frac{2E}{\hbar} \tag{6.7}$$

Le résultat de N. Margolus et L. Levitin appelle plusieurs commentaires. Le premier est que toute évolution du système par une succession arbitraire d'états distinguables y est considéré comme un calcul, bien qu'une telle succession puisse ne correspondre à aucune spécification. La borne supérieure exprimée constitue donc une borne *a fortiori* sur le nombre d'opérations effectuables par le système. Nous continuerons donc à parler d'opérations effectuables par le système, bien que cette expression doive être prise *cum grano salis.*

En outre, la conception de l'étape de calcul comme transition entre deux états est marquée par des intuitions provenant de l'expérience du calcul symbolique finitaire. Dans un calcul symbolique finitaire, réaliser une étape de calcul implique de remplacer un symbole par un symbole parfaitement distinguable [5]. Ainsi, dans un alphabet binaire, une opération élémentaire consisterait à passer d'un symbole 0 à un symbole 1. Si l'on généralise cette conception, une opération élémentaire effectué par un système quantique à deux niveaux, ou qubit, constituerait à passer de l'état de base $|0\rangle$ à l'état de base orthogonal $|1\rangle$. La définition de N. Margolus et L. Levitin généralise une telle vision à un système à dimension infinie dénombrable. Mais le modèle le plus courant du calcul quantique, le modèle des circuits quantiques, ne s'appuie pas sur une telle définition d'une étape de calcul élémentaire : celle-ci ne mène pas nécessairement d'un état quantique à un état orthogonal [6]. Montrer l'impossibilité des A.T.M.s, c'est montrer l'impossibilité d'exécuter une infinité d'étapes en un temps fini, mais seulement pour une certaine conception de l'étape élémentaire de calcul.

Le second est que ce résultat ne prohibe pas immédiatement la possibilité d'exécuter en un temps fini un nombre d'opérations infini : il borne le taux d'opérations par l'énergie du système. Pour que le taux d'opération puisse diverger, il faudrait donc que l'énergie du système diverge. Les raisons qui interdisent à un système d'implémenter une supertâche sont donc exactement les raisons qui lui interdisent d'avoir une énergie divergente.

Peut-on autant dire que la théorie de la calculabilité des modèles à temps discret est fondée sur une hypothèse empirique ? Pour que l'exécution d'une A.T.M. soit possible, il ne suffit pas qu'un nombre arbitrairement grand d'opérations soit exécutable en un intervalle de temps fini, mais qu'une infinité actuelle d'opérations soit exécutée. Une machine capable d'exécuter un nombre arbitrairement grand d'opérations par unité de temps - par exemple, n'importe quelle valeur d'une tour d'exponentielle- bouleverserait certainement notre conception de la complexité calculatoire, mais elle laisserait notre théorie de la calculabilité inchangée. Une telle machine exécuterait toujours un nombre fini d'opérations en un temps fini, et ne constituerait donc pas l'implémentation d'une A.T.M..

Notre théorie de la calculabilité n'exige donc pas l'existence d'une borne supérieure sur le nombre d'opérations exécutables par unité de temps, ou taux de traitement borné de l'information [7], hypothèse sur laquelle nous aurons l'occasion de revenir (voir section 8.2). Elle exige seulement que le

5. On peut dire que la seule contrainte structurelle pesant sur une signature Σ est que ses symboles soient deux à deux distinguables par un observateur doué d'une précision d'observation finie.

6. Nous aborderons cette conception plus en détail durant notre étude des problématiques de complexité (voir section 8.2).

7. Cette dénomination est une traduction de l'expression anglaise *maximum rate of information processing* employée par S. Lloyd ([163], 2).

nombre d'opérations exécutées en un intervalle de temps fini soit fini. Le théorème de Margolus-Levin nous montre à tout le moins que, sous certaines hypothèses, une telle condition est satisfaite tant que l'énergie du système considéré est finie. Doit-on qualifier cette dernière proposition d'empirique ?

Pour répondre à cette dernière question, il est nécessaire de réaliser un bref excursus sur le statut des divergences en physique. Au cours d'un raisonnement théorique, l'emploi de quantités divergentes, notamment par un passage à la limite, est extrêmement fréquent. On peut ainsi considérer un intervalle de temps, une distance, une masse ou une quantité de particules infinies. Dans le cadre d'une prédiction sur la valeur d'une quantité observable, la pratique de la physique est soumise au principe informel suivant : les prédictions de valeurs de quantités observables résultant sur des divergences sont dépourvues de sens physique [8]. Un tel principe est rendu naturel, par la difficulté qu'il y aurait à conférer un sens opérationnel à une mesure de valeur infinie. À ce titre, le rejet des divergences des quantités observables en physique ne peut pas même être qualifié de principe empirique, mais plutôt de principe méthodologique : il est une condition de possibilité du sens d'une prédiction empirique.

Le modèle de Margolus-Levitin est donc critiquable à plus d'un titre. La définition proposée d'une étape élémentaire de calcul est fondée sur des intuitions issues de la pratique du calcul symbolique finitaire, qui ne se généralisent pas à tous les modèles de calcul. Elle pêche également par une restriction des opérations de mesure quantiques aux opérations de mesure projective de von Neumann. Enfin, sur la seule base de ce modèle, il n'est donc pas même évident de qualifier les hypothèses à la base du rejet des supertâches d'empiriques. C'est un précepte méthodologique général, l'interdiction des divergences dans les prédictions de valeurs de quantités observables, qui exclut la possibilité d'une divergence du taux de traitement de l'information [9]

Nos dernières remarques illustrent une difficulté plus générale de la discussion des A.T.M.s, et par delà de tous les modèles notionnels. L'exhibition de propositions physiques précises interdisant l'implémentation de ces modèles suppose un processus de substitution d'un modèle physiquement

8. Par l'expression « prédiction de valeurs d'une quantité observable », j'entends ici d'authentiques prédictions censées aboutir à un test en laboratoire, et non une prédiction théorique qui n'est pas systématiquement soumise à une telle exigence. Ainsi, il est fréquent en théorie quantique des champs d'obtenir des prédictions divergentes avant renormalisation.

9. On peut donc conclure que le modèle des A.T.M.s est rejetté sur la base d'un argument méthodologique. Mais un tel type d'arguments ne doit pas être confondu avec les arguments du scepticisme méthodologique que nous avons étudiés plus haut (voir section 2.3.2). Ces arguments visaient à établir l'impossibilité de principe de tout modèle d'hypercalcul. Un argument méthodologique fondé sur l'existence de divergences établit l'impossibilité de principe d'un modèle physiquement défini donné. Il ne peut donc être formulé qu'au cas par cas.

défini au modèle original, processus qui peut être vu comme une traduction des notions informatiques abstraites dans les termes d'une théorie physique. Comme le montre nos exemples d'argumentations, cette traduction n'est ni nécessairement aisée, ni nécessairement univoque. La difficulté provient des choix de modélisation et de définition de concepts, dont l'absolue nécessité eu égard au modèle notionnel de départ peut faire l'objet de débats : on l'a vu pour l'emploi de photons à haute énergie dans le modèle de S. Aaronson, ou la définition d'une étape élémentaire dans le modèle de Margolus-Levitin. Selon le choix opéré, les propositions mises aux fondements de la critique du modèle peuvent varier. Ainsi on invoque des arguments issus de la relativité générale pour le modèle d'Aaronson, tandis qu'on s'appuie sur des propriétés purement quantiques, et un argument méthodologique, dans le modèle de Margolus-Levitin. Dans l'état actuel de la littérature, il n'est pas possible d'identifier un ensemble défini d'hypothèses comme la raison interdisant l'implémentabilité d'un modèle notionnel comme les A.T.M.s. Il existe un faisceau d'arguments légitimant une position de scepticisme robuste à l'égard de l'implémentabilité d'une supertâche, mais non une authentique démonstration théorique, qui révélerait les raisons précises d'une telle impossibilité.

De telles difficultés ne sont pas propres aux A.T.M.s, et menacent en droit tout modèle notionnel. On pourra certes arguer que l'ambiguïté des résultats obtenus pour les A.T.M.s n'est qu'une contingence de l'état actuel de la littérature. Pour obtenir des résultats définitifs, il faudra attendre d'avoir pu formuler des définitions physiques consensuelles des concepts fondamentaux de l'informatique, comme « mémoire », ou « opération élémentaire ». Mais cette réponse, en elle-même parfaitement légitime, soulève un problème plus profond d'une recherche de la signification empirique de la thèse de Church-Turing partant de l'examen de modèles notionnels.

Un modèle notionnel d'hypercalcul, sans aucune inspiration physique, peut être conçu comme une variante des modèles historiques du calcul effectif. S'il viole les hypothèses décrivant une procédure effective, il est donc encore conçu dans les catégories du calcul effectif. Or il n'y aucune raison que tous les modèles physiques fondamentaux d'une machine de calcul puisse être décrit dans ces catégories. M. Stannett a ainsi proposé une classification des modèles d'hypercalcul basée sur la violation de propriétés fondamentales des machines de Turing [216]. Il fait ensuite la remarque suivante sur cette forme de classification (*ibid.*, p.152) :

> I've suggested that Turing machines owe their power to four basic properties, and have considered varying each of these in turn. This is by no means a complete analysis of hypercomputation and its causes. There is no *a priori* reason why every feasible machine should necessarily be considered a variant of the Turing machine, and perhaps models will be created that owe nothing

to the principles considered here.

En cherchant à fonder en termes physiques l'impossibilité d'un modèle notionnel d'hypercalcul, notre recherche se condamnerait à concevoir le calcul dans les catégories conçues pour les procédures effectives. Le processus de définition du calcul en termes physiques n'a pas à se réduire à un processus de transcription littérale des catégories informatiques usuelles en termes physiques. Nous allons voir en examinant les modèles physiquement définis que la prédiction de M. Stannett est en réalité déjà réalisée. La physique théorique a déjà inspiré la conception de modèles d'hypercalcul qui ne peuvent être rigoureusement décrits dans les termes conçus pour de simples variantes des hypothèses d'effectivité, comme les supertâches. Une étude de la signification empirique de la calculabilité qui demeurerait centrée sur les modèles notionnels courrait donc un double risque : celui d'une indécision dans ses conclusions tout d'abord, et celui d'une restriction indue des possibilités envisagées, ensuite.

Par conséquent, nous allons à présent nous concentrer sur les modèles que nous avons qualifiés de *physiquement définis*, afin de voir s'ils permettent d'arriver à des conclusions à la fois plus tranchées et plus profondes.

6.2 Les modèles analogiques d'hypercalcul

Nous avons déjà présenté certains modèles analogiques (voir section 3.2), et discuter les problèmes soulevés par leur éventuelle implémentation. Cependant, tous les modèles d'hypercalcul analogiques que nous avons évoqués jusqu'à présent étaient des modèles notionnels. Il existe cependant des modèles d'hypercalcul physiquement définis. Nous allons présenter un de ces modèles, et conclure sur le potentiel des modèles analogiques à devenir des modèles d'hypercalcul.

Dans [43], O. Bournez et M. Cosnard considèrent un modèle nommé système de miroirs (*mirror system*). Un système de miroirs est composé d'un ensemble fini de miroirs et d'un rayon lumineux. La trajectoire du rayon lumineux est déterminé par les lois de l'optique géométrique. Un système est dit planaire parabolique si chacun des miroirs est soit plan soit parabolique.

Le calcul est effectué par une particule se réfléchissant sur les miroirs. La suite des états du système est donné par la suite des intersections de la trajectoire de la particule avec une section fixe du plan. L'intuition essentielle du modèle est la suivante. Soit P la section du plan $P = \{(x, y, 0)|(x, y) \in [0, 1]^2\}$ dans l'espace $(0, x, y, z)$. On construit alors un système de miroirs S tel que, si une particule p intersecte P perpendiculairement en un point $(x, y, 0)$ dans la direction $z > 0$, alors p intersectera nécessairement P perpendiculairement au point $(x', y', 0)$ dans la direction $z > 0$, avec $(x', y') = f(x, y)$. O. Bournez et M. Cosnard ont ensuite démontré que l'ensemble des fonctions f calculables au titre de ce modèle était équivalent à celui cal-

culé par un modèle analogique d'hypercalcul notionnel, le *analog two-stack automaton*([43], 21-24).

Encore une fois, « physiquement défini »ne doit nullement être confondu avec « réaliste », et ce modèle souffre des mêmes problèmes de précision arbitraire que ceux que nous avons déjà rencontrés lors de notre discussion de la calculabilité sur les réels et des arguments de M. Davis (voir sections 2.3.2 et 3.2) [10]. Nous reviendrons sur la signification précise de cette exigence de précision arbitraire ultérieurement (voir section 6.5).

Entre-temps, le modèle d'O. Bournez et M. Cosnard permet d'illustrer un point essentiel, à savoir qu'un modèle analogique physiquement défini souffre des mêmes difficultés qu'un modèle notionnel. Sa puissance est censée provenir d'un encodage infiniment dense de l'information dans les variables du système, ce qui impose une exigence de précision arbitraire à l'utilisateur fini du calculateur.

Les modèles de calcul analogiques, qu'ils soient notionnels ou physiquement définis, peuvent être simulés par des systèmes dynamiques. La décidabilité d'un problème peut alors être reformulée comme le problème de l'atteignabilité (*reachability*) d'une certaine région de l'espace des phases, représentant l'état d'arrêt du système, à partir de régions représentant une possible configuration initiale. Au sein de la communauté informatique, il est fréquent d'entendre exprimer une conjecture informelle selon laquelle le problème de l'atteignabilité serait décidable pour les systèmes « robustes », « réalistes », ou « bruités ». Cette conjecture est si répandue qu'elle peut être présentée comme un résultat acquis par certains auteurs débattant de la validité de la thèse de Church-Turing, comme L. Fortnow ([100], 4) :

> One cannot distinguish between two analog processes that end with results closer than the inherent limits of measurement. So one can take any analog process and simulate it by a discrete process that breaks it into a finite number of transitions approximated to an appropriate finite precision that will compute a result within that limit.
>
> (...) So yes Virginia, the Earth is round, man has walked on the moon, Elvis is dead and everything computable is computable by a Turing machine.

Le caractère naturel de cette conjoncture est due à l'omniprésence d'une exigence de precision arbitraire dans les modèles d'hypercalcul analogiques, comme nous venons de le mentionner. Un tel argument intuitif n'est cependant pas évident à traduire sous une forme rigoureuse. E. Asarin a ainsi étudié la calculabilité pour des systèmes dynamiques décrits par des applications affines par morceaux. Il a pu démontrer que le problème de l'atteignabilité demeure indécidable pour de tels systèmes lorsqu'ils sont bruités

10. Pour plus de détails sur la critique de l'implémentabilité de ce modèle, voir ([62], 195-198).

par un bruit constant uniforme, ou par un bruit infinitésimal. Ce dernier résultat est d'ailleurs valable pour d'autres modèles, dont les Applications aux dérivées constantes par morceaux (*Piecewise Constant Derivative Systems)*, les Machines de Turing, et les Automates Linéaires Hybrides (*Linear Hybrid Automata*)[19]. Ce n'est qu'extrêmement récemment que Bournez et alii ont présenté une démonstration de l'équivalence entre robustesse aux perturbations infinitésimales et décidabilité du problème pour une classe extrêmement large de systèmes dynamiques [45]. Nous n'avons pas eu le loisir de consulter ce travail très récent en détail, mais nul doute qu'il permette d'appuyer sur une démonstration mathématique générale un scepticisme déjà robuste à l'égard des modèles d'hypercalcul analogiques.

6.3 Modèles relativistes

> *In the end, M-H spacetimes and the supertasks they underwrite may similarly prove to be recreationnal fictions for relativity theorists with nothing better to do. But to arrive at this latter position requires a resolution of some of the deepest foundations problems in classical general relativity [...]. It is this connection to real problems in physics that make them worthy of discussion.*

J. Earman and J. Norton, *Forever is a Day : Supertasks in Malament-Hogarth Spacetimes* [11]

Comme nous l'avons vu dans notre présentation des machines de Turing accélérantes, les relations entre temps de calcul et temps réel peuvent être plus complexes qu'une simple proportionnalité. Les modèles relativistes d'hypercalcul fournissent une nouvelle instance de ce problème fondamental.

Ces divers modèles sont tous fondés sur l'exploitation d'un trait paradoxal de la physique relativiste, popularisé par le célèbre paradoxe des jumeaux de Langevin, à savoir la relativisation de l'intervalle de temps écoulé entre deux événements au référentiel considéré, et à son état de mouvement. La stratégie des modèles relativistes consiste à exploiter cette relativisation de l'écoulement du temps pour exécuter un calcul infini en un temps infini dans un premier référentiel, qu'on nommera « référentiel du calculateur »et à communiquer son résultat à un second référentiel où s'est écoulé un temps fini, qu'on nommera « référentiel du programmeur ». Un observateur situé dans le référentiel du programmeur aurait donc accès au résultat d'un calcul infini en un temps réel fini.

Ces modèles sont souvent assimilés au modèle des machines de Turing accélérantes, puisqu'ils permettent d'accéder au résultat d'un temps de calcul

11. *Ibid.*, pp.40-41.

infini en un temps réel fini [87], [217]. Cette assimilation nous semble cependant dommageable, dans la mesure où elle masque les spécificités de chacun de ces modèles. D'un point de vue relativiste, les machines de Turing accélérantes effectuent leur calcul dans un référentiel fixé. Leurs performances calculatoires sont fondées sur une divergence du taux de traitement de l'information. Une telle propriété est absente des modèles relativistes : dans le référentiel du calculateur, l'exécution d'une infinité d'étapes prend un temps infini.

C'est d'ailleurs une propriété singulière des modèles relativistes, à savoir qu'ils ne proposent aucun schème original d'exécution du calcul. Dans la plupart des présentations, le calcul est exécuté dans le référentiel du calculateur par une machine de Turing. L'automatisation du calcul n'est pas même nécessaire à son exécution : on pourrait parfaitement envisager que le calcul soit exécuté par une civilisation de calculateurs armés de papier et de crayons, poursuivant l'exécution du calcul de génération en génération ([50], 787).

C'est la communication du résultat au référentiel du programmeur, où s'est écoulé un temps fini, qui permet d'implémenter un hypercalcul. Tant que le résultat n'a pas été communiqué au programmeur, tout ce que le modèle a permis est d'effectuer une recherche infinie en un temps infini. Comme nous allons le voir, la possibilité d'une telle communication sera au coeur des débats sur l'implémentabilité des modèles relativistes. Toute la puissance du modèle provient donc de la communication entre deux référentiels différents.

La singularité des modèles relativistes est si grande, qu'elle en vient à rendre problématique l'application de certaines catégories classificatoires. On peut ainsi s'interroger sur la pertinence d'appliquer la catégorie de « supertâche ». Dans un premier sens, la réponse à une telle interrogation est évidemment positive, puisque l'exécution d'une supertâche, comme l'exécution d'un calcul au titre des modèles relativistes, permet de connaître le résultat d'une recherche infinie en un temps fini. Dans un second sens, en revanche, la réponse est évidemment négative, puisque la notion de supertâche supposait que la recherche infinie soit exécutée en temps fini. Le modèle relativiste permet de distinguer le temps réel d'exécution du calcul du temps réel d'accès épistémique au résultat du calcul : distinction naturellement absente de la première compréhension des supertâches. Dans de telles conditions, le choix d'appliquer ou non la catégorie de supertâche est byzantine, et dépourvue de véritable contenu [12]. Ceci illustre encore une fois la difficulté qu'il peut y

12. Dans ([88], 232), J. Earman et J. Norton distinguent les *proper supertasks*, où un temps réel fini est expérimenté par l'agent exécutant le calcul, et les *bifurcated supertasks*, où un temps infini est expérimenté par l'agent calculateur, mais un observateur séparé a accès au résultat en un temps fini. Plutôt que d'ajouter un qualificatif au concept de « supertâches », on pourrait tout aussi bien employer deux termes différents, pour souligner la diversité de procédures qu'on pourrait avoir tendance à confondre sous une même appellation. Mais il s'agit là d'une simple affaire de convenance terminologique, et

avoir à appliquer les catégories du calcul effectif aux modèles physiquement définis.

Des problèmes analogues se posent pour l'application de la catégorie de « calcul effectif. »Un auteur comme J. Earman a ainsi pu remarquer : ([86], 120) [13] :

> The computational arrangements between γ_1 and γ_2 envisioned might also seem to bring into doubt Church's proposal that effective/mechanical computability is to be equated with Turing computability or recursiveness, for apparently γ_1 and γ_2 can in concert obtain a resolution to recursively unsolvable problems by means that certainly seem to merit the appelations of 'effective' and 'mechanical'. But putting the matter this way is a little unfair to Church since any account of effective/mechanical computability that implies that there are subsets of numbers which can be effectively/mechanically enumerated, but whose complements cannot be, will be subject to the one-upmanship of bifurcated supertasks. Perhaps the most illuminating way to state the moral to be drawn from bifurcated supertasks is that two levels of computation need to be distinguished : the first corresponding to what the slave computer γ_1 can do, the second to what γ_2 can infer by having causal access to all of γ_1's labors. Church's proposal is best construed as aimed at the first level and as asserting that Turing computability is an upper bound on what any physical instantiation of γ_1 can accomplish. Read in this way, there is nothing in present concerns to raise doubts about Church's proposal.

Là encore, la discussion des modèles relativistes introduit des distinctions inexistantes dans les discussions antérieures de l'effectivité. Il est donc difficile de désigner une intuition préthéorique sur la notion d'effectivité, ou même un trait explicite des modèles historiques du calcul effectif, qui exclut la possibilité du calcul relativiste. D'un point de vue historique, il serait malheureux d'affirmer que la thèse de Church-Turing avait « pour objet »le calcul effectué le long de la ligne d'Univers du référentiel du calculateur, tout comme il serait malheureux d'affirmer que la thèse de Church-Turing « n'excluait pas »de tels calculs.

Qu'on nous permette cependant de faire deux remarques. Comme nous l'avons vu au chapitre précédent, selon la conception épistémique du calcul, un calcul ne peut être considéré comme achevé que lorsque l'observateur fini dispose d'un accès épistémique au résultat. On souhaitera à présent ajouter que cet accès épistémique doit pouvoir se produire dans un intervalle de

non d'une question substantielle.

13. J. Earman et J. Norton désignent par γ_1 la ligne d'univers du référentiel du calculateur, et par γ_2 la ligne d'univers du référentiel du programmeur.

temps fini après le début de la procédure de calcul. Cette remarque a une valeur générale : elle s'applique tout aussi bien aux procédures effectives qu'aux autres procédures de calcul, et elle s'applique au modèle relativiste dans le référentiel du programmeur.

Au sujet des procédures de calcul effectif, on peut dire qu'elles sont soumises à une hypothèse spécifique, dont on n'osera dire qu'elle fait partie des intuitions préthéoriques sur la notion, mais dont on pourra au moins dire qu'elle ne va pas contre leur esprit : *si le nombre d'étapes élémentaires de calcul exécutées par le calculateur est infini, alors l'intervalle de temps réel précédant l'accès épistémique au résultat de calcul est infini.* Cette hypothèse est plus générale que l'impossibilité d'effectuer une infinité d'opérations en un temps fini, dont on a vu qu'elle est nécessaire pour rejeter le modèle des machines de Turing accélérantes. En se dégageant de toute perspective historique, on peut dire que cette hypothèse est implicite dans notre conception préthéorique du calcul effectif. Elle permet d'affirmer que la procédure utilisée dans les modèles de calcul relativistes n'est pas effective, car l'observateur attend un temps fini avant d'obtenir le résultat d'une recherche infinie exécutée dans le référentiel du calculateur.

6.3.1 Hypercalcul relativiste contrefactuel : les signaux supralumineux

Le premier modèle relativiste, dû à P. Németi et G. Székely, a pour particularité d'être basé sur une possibilité exclue par la physique contemporaine : le signal supralumineux. La théorie de la relativité restreinte, soumise à la causalité, exclut en effet la possibilité de propager une information à une vitesse supérieure à c. Dans ce sens, le modèle de P. Németi et G. Székely peut être qualifié de « modèle relativiste contrefactuel ». Il est relativiste au sens où il est physiquement défini dans le langage théorique de la relativité restreinte, et contrefactuel au sens où il viole explicitement un des résultats les plus importants de cette théorie.

Ce modèle ne permet donc guère de discuter de la valeur de vérité de la thèse de Church-Turing, puisqu'il est d'emblée basé sur une hypothèse tenue pour fausse. Mais il peut être utile pour discuter notre deuxième question, à savoir la signification empirique de cette thèse. C'est d'ailleurs le but affiché de P. Németi et G. Székely : ceux-ci cherchent à démontrer que la théorie de la relativité restreinte admet la formulation d'un modèle d'hypercalcul si et seulement si il existe des signaux supralumineux. S'il n'existe pas de tels signaux, la théorie de la relativité restreinte sera donc incompatible avec l'existence d'un modèle d'hypercalcul. Nous reviendrons en conclusion sur ce point essentiel.

Ce modèle exploite un trait fondamental de la relativité restreinte : s'il existait, un signal supralumineux pourrait se propager dans le passé causal de son émetteur.

Ce paradoxe de la théorie de la relativité restreinte permet de concevoir la procédure suivante. Soient un programmeur P et un ordinateur C, initialement placés dans un référentiel inertiel I. C entre en mouvement uniformément accéléré par rapport à I tout en énumérant un ensemble récursivement énumérable S. Il existe alors un événément O jouissant de la propriété suivante : tout événément E sur la ligne d'Univers de C est simultané à O dans le référentiel inertiel dont la ligne d'univers est tangente à C à E.

On fixe alors un événément M postérieur à O dans le référentiel I du programmeur. Admettons, pour fixer les idées, que le calculateur cherche à résoudre une question du type « $n \in S$ ». Si C découvre que n appartient à S, il envoie un signal supralumineux qui atteint P entre O et M. Si P ne reçoit aucun signal avant l'événement M, il sait qu'il peut en déduire que $n \notin S$. Une telle procédure permet de décider tout ensemble récursivement énumérable, et donc de résoudre toute question Σ_1^0 ou Π_1^0.

P. Németi et G. Székaly admettent que leur modèle rencontrerait des difficultés, même si l'existence de signaux supralumineux était un fait avéré. L'intervalle de temps séparant O et M est fixé, tandis que la distance séparant P et C ne cesse d'augmenter. Pour arriver à envoyer un signal atteignant P entre les événéments O et M, C doit envoyer son signal à une vitesse arbitrairement précise ([176], 536) :

> Of course our construction contains several engineering difficulties. For example, the larger the distance the more difficult to aim with a signal. Therefore, the computer has to calculate the speed of the FTL [Faster Than Light] signal more and more accurately to ensure that the signal arrives to the programmer between events O and M, (...). Thus the computer has to be able to aim with the FTL signal with arbitrary precision.

La qualification de l'exigence de précision arbitraire comme un simple « problème d'ingénierie »est problématique. On a vu au contraire que cette exigence représentait un problème fondamental pour de nombreux modèles d'hypercalcul. Sur cette base, on peut critiquer la caractérisation proposée par les auteurs de la thèse de Church-Turing physique pour les modèles définis au sein de la théorie de la relativité restreinte : la théorie de la relativité restreinte est compatible avec l'existence d'un modèle d'hypercalcul ssi les signaux supralumineux sont possibles [14]. Cette caractérisation néglige en effet l'exigence de précision fondamentale. L'existence d'un signal supralumineux ne suffit donc pas à justifier la possibilité d'un modèle d'hypercalcul. Il n'est donc pas évident, même au sein d'un modèle contrefactuel, d'isoler une hypothèse unique permettant de caractériser le respect ou l'irrespect de la thèse physique par le modèle considéré. Nous reviendrons plus tard sur cette difficulté des caractérisations de la thèse de Church-Turing physique (voir section 6.6).

14. *Ibid.*, pp.535-536.

6.3.2 Hypercalcul relativiste général : le modèle de Malament-Hogarth

Histoire et description intuitive du modèle

Contrairement au modèle relativiste restreint, le modèle relativiste général n'est pas un modèle contrefactuel : il ne part pas de l'exploration des possibilités offertes par la négation d'un résultat établi de la physique. Il part au contraire d'un trait remarquable propre à la relativité générale : non seulement l'intervalle de temps y est relatif au référentiel, mais cette relativité peut aller jusqu'à permettre l'expérience d'un intervalle de temps infini dans un référentiel, tandis qu'un autre référentiel expérimente un intervalle de temps fini. Dans le référentiel du calculateur, il est donc possible d'effectuer une recherche infinie en un temps infini, comme la recherche d'un contre-exemple à la conjecture de Goldbach, et d'en communiquer le résultat au bout d'un intervalle de temps fini dans le référentiel du programmeur. Plus précisément, le programmeur doit recevoir toute réponse positive au problème initial dans un intervalle fini fixé, et peut interpréter l'absence de réponse dans ce délai comme une réponse négative.

L'un des moyens envisagés pour implémenter un tel modèle est de faire traverser l'horizon d'événement d'un trou noir à un des référentiels, tandis que l'autre référentiel demeure à l'extérieur du trou noir. Pour cette raison, ce modèle d'hypercalcul est souvent nommé calcul avec des trous noirs (*black hole computation*). Il existe cependant des espaces-temps qui sont des implémentations possibles de ce modèle, sans être des trous noirs [15].

Les travaux d'I. Németi [174], D.B. Malament [16], I. Pitowsky [192], et M. Hogarth [138], [139] représentent les premières tentatives d'exploration du potentiel computationnel de ces espaces-temps. Dans le premier papier consacré à la critique de ces tentatives ([87], 24-28), J. Norton et J. Earman ont montré que l'approche d'I. Pitowsky souffrait de deux défauts particuliers. Tout d'abord, le modèle nécessitait que le programmeur connaisse une accélération constante non-bornée, dite de Born, ce qui entraînerait sa destruction par des forces gravitationnelles sans borne. Ensuite, le modèle souffre de l'inexistence d'un critère de terminaison du calcul lorsque la réponse est négative, et que le calculateur n'envoie pas donc pas de signal au programmeur ([86], 105-107 ; [87], 6).

Par conséquent, la suite de la discussion s'est centrée sur les espaces initialement proposés par D.B. Malament et M. Hogarth, et que J. Earman et J. Norton ont baptisé *espaces-temps de Malament-Hogarth*, appelation qui s'est imposée dans la littérature. Soit M une variété différentiable, g_{ab} une métrique lorentzienne définie sur tout M, on définit :

15. Voir ([14], 219), et les références mentionnées.

16. D.B. Malament n'a pas publié à la période des premiers travaux sur ces modèles, mais il est crédité par J. Earman dans ([86], 238) pour son influence par le biais de communications privées.

Définition 1. M, g_{ab} est un espace-temps de Malament-Hogarth ssi il existe une demi-courbe de type temps $\gamma_1 \subset M$ et un point $p \in M$ tels que $\int_{\gamma_1} d\tau = \infty$ et $\gamma_1 \subset I^-(p)$.

Par une extension naturelle, nous appellerons *modèle d'hypercalcul de Malament-Hogarth* le modèle de calcul fondé sur l'exploitation de ces mêmes espaces.

Les espaces de Malament-Hogarth permettent donc la conception d'une procédure bien définie, permettant de décider n'importe quel ensemble récursivement énumérable. Puisqu'il n'existe pas d'incohérence manifeste dans la définition de la procédure, l'intégralité de la discussion théorique s'est centrée sur la question de son implémentabilité. Nous allons maintenant examiner une à une les grandes objections auxquelles a été confronté le modèle d'hypercalcul de Malament-Hogarth.

L'existence des espaces de Malament-Hogarth

La première question est évidemment celle de l'existence même des espaces de Malament-Hogarth. Ces espaces satisfont une première exigence minimale, à savoir d'être solutions des équations d'Einstein décrivant la dynamique de la relativité générale. Comme nous l'avons vu plus haut, être solution des équations de la dynamique n'est en aucun cas une condition suffisante du caractère physiquement réaliste d'un modèle. L'usage consensuel en relativité générale veut que le tenseur énergie-tension obéissent à des conditions d'énergie particulières, dites faible, forte ou dominante (voir [131] pour une présentation précoce, [91] pour un état de l'art récent). Comme le remarquent J. Earman et J. Norton ([87], 14), aucun résultat théorique ne garantit que ce soit le cas pour tous les espaces de Malament-Hogarth. Il existe cependant des espaces de Malament-Hogarth satisfaisant de telles conditions. On peut citer en exemple le modèle de Reisser-Nordström pour un trou noir électriquement chargé (pour plus de détails, voir [87], 34, et références mentionnées).

Une condition plus forte, mais moins consensuelle, est imposée par les différentes formes d'une conjecture nommée « censure cosmique »(*cosmic censorship*) (voir [186], [187], pour des présentations historiques, et [189],[133], [134], pour des discussions récentes).

Ce principe, formulé par R. Penrose, vise à exclure les singularités « nues », c'est-à-dire les singularités qui ne sont pas dissimulées par l'horizon d'événément d'un trou noir, des solutions physiquement réalistes de la relativité générale. L'interprétation forte du principe interdit les espaces qui ne sont pas globalement hyperboliques, et exclut donc tout espace de Malament-Hogarth [17]. D'autres interprétations plus faibles laisseraient survivre certains espaces de Malament-Hogarth.

17. En relativité générale, on appelle *surface de Cauchy* une hypersurface de type espace rencontrant toutes les courbes causales inextensibles une et une seule fois. Un espace est dit

L'existence des espaces-temps de Malament-Hogarth est une condition nécessaire, mais non suffisante, pour l'implémentabilité du modèle de Malament-Hogarth. D'autres contraintes doivent être satisfaites pour que la structure de l'espace-temps, si elle existe réellement, puisse être exploitée pour réaliser un hypercalcul. On peut classer sommairement ces différentes contraintes de la manière suivante : les contraintes dynamiques pesant sur l'existence du programmeur et du calculateur, les contraintes pesant sur les ressources, et celles pesant sur la transmission du signal.

Les contraintes dynamiques

Comme le remarque J. Earman dans [86], il n'existe pas de liste de contraintes exhaustives permettant de caractériser une machine réaliste. On peut énumérer certaines contraintes dynamiques raisonnables pesant sur la procédure de Malament-Hogarth :

- L'accélération de l'observateur doit rester bornée, pour éviter que celui-ci soit écrasé par les forces gravitationnelles.
- L'accélération totale faite sur le trajet doit être supérieurement bornée, sous peine d'avoir besoin d'une infinité de carburant.
- Le système tombant dans l'espace de Malament-Hogarth ne doit pas subir d'effets de marée trop importants, menaçant de détruire le système. C'est notamment le cas pour les espaces-temps de Reissner-Nordström.

La conclusion de J. Earman et J. Norton est que si les espaces de Malament-Hogarth sont possibles et réalistes, alors le rôle du programmeur peut être joué par une ligne d'univers (ou un tube d'univers, pour prendre en compte l'extension spatiale de l'observateur) dynamiquement réaliste.

Un autre problème spécifique a été soulevé pour la première fois par J. Earman et J. Norton, et est devenu connu sous l'appellation de *décalage divergent vers le bleu*(*divergent blueshift*). Celui-ci fut initialement conçu comme une caractérisation des espaces-temps de Malament-Hogarth. Le problème peut être intuitivement présenté de la manière suivante. En observant le calculateur, le programmeur est censé recevoir l'infinité de photons émis par ce dernier pendant son temps propre infini pendant un intervalle de temps propre fini. La fréquence de réception des signaux doit donc diverger. Ceci exposerait le programmeur à recevoir un flux de photons d'énergie infinie, menaçant son existence même. La structure de l'espace-temps de Malament-Hogarth agit donc comme un amplificateur d'énergie arbitrairement puissant, menaçant la possibilité de réception du résultat de calcul ([87], 112-113) [18].

---

globalement hyperbolique s'il admet une surface de Cauchy. Pour reprendre l'interprétation intuitive de J. Norton [179] : un espace globalement hyperbolique est un espace qui n'admet pas de poches causalement isolées d'espace-temps. J. Norton et J. Earman ont montré que les espaces de Malament-Hogarth ne sont pas globalement hyperboliques dans [87].

18. J. Earman et J. Norton signalent également l'existence d'espaces de Malament-Hogarth mathématiquement bien définis qui ne sont pas soumis à cette condition, mais ils

Face à ce problème, plusieurs stratégies de contournement ont été proposées. La première fut d'ailleurs proposée par J. Earman et J. Norton eux-mêmes, et impliquait à la fois de refroidir les photons liés au bruit thermique et diminuer l'énergie des photons émetteurs pendant le trajet ([87], 114). Dans [175], I. Németi et G. David ont proposé d'autres stratégies de contournement. L'une d'entre elles consistait à faire rebondir le signal sur un autre émetteur qui voyage dans la direction opposée, et dont le décalage vers le rouge compense exactement le décalage vers le bleu. Une autre consistait à envoyer un autre récepteur intermédiaire se dirigeant vers l'horizon d'événement interne du trou noir pour apaiser le message par effet Dopler, ou simplement se fait détruire sous l'oeil du programmeur, ce qui représente un message pour ce dernier. L'objection du décalage divergent vers le bleu est donc plus une sérieuse difficulté, qu'un véritable argument dominateur visant à exclure en principe le modèle de Malament-Hogarth.

L'exigence en ressources

I. Pitowsky a soulevé un problème d'exigence en ressources qui serait propre au modèle de Malament-Hogarth. Dans [192], il formule l'argument suivant [19] :

> The real reason why Platonist computers are physically impossible *even in theory* has to do with computation space. According to general relativity the material universe is finite. Even if we use the state of every single elementary particle in the Universe, to code a digit of a natural number, we shall very soon run out of hardware.

La complétion du calcul dans le référentiel du calculateur supposerait des ressources en mémoire illimitées, ne serait-ce que pour encoder des entrées de taille de plus en plus importante. Mais, toujours selon l'argument d'I. Pitowsky, l'Univers matériel est fini, et n'offre donc qu'un espace borné pour l'effectuation d'un calcul. Bien qu'il dispose d'un temps infini, le calculateur sera bientôt arrêté par un défaut d'espace-mémoire.

Cet argument pose plusieurs problèmes. Comme le remarquaient déjà J. Earman et J. Norton, la relativité générale n'implique pas que l'Univers soit spatialement fini, ou qu'il ne contienne qu'une quantité finie de matière ([87], 20).Au contraire, les espaces de Reisser-Nordström que nous avons évoqué ci-dessus, sont spatialement infinis.

L'argument d'I. Pitowsky soulève un deuxième problème, qui à notre connaissance n'a pas été mentionné dans la littérature. Il suppose implicitement qu'une quantité illimitée d'information ne pourra être encodée dans une

ne satisfont alors plus les conditions désirables sur l'énergie (*idem*).

19. *Ibid.*, p.84. Par l'expression « ordinateurs platoniciens »(*Platonist computers*), I. Pitowsky se réfère à ce que nous avons appelé le modèle de Malament-Hogarth.

quantité bornée de matière-énergie. Il admet donc implicitement une forme de principe de densité finie de l'information, dont la vérité est elle-même au coeur des discussions sur la possibilité de l'hypercalcul (voir section 6.5.3). En l'absence d'une formulation rigoureuse de ce principe, l'argument d'I. Pitowsky ne peut lui-même être formulé précisément. Il n'est donc pas certain qu'on ait affaire là à un véritable argument de portée universelle, ou à un argument adressé spécifiquement à l'encontre de certaines formes d'encodages de l'information [20].

Divers problèmes

Avant d'aborder le dernier groupe d'objections au modèle de Malament-Hogarth, qui se montrera d'une importance cruciale pour la suite de notre propos, nous mentionnons brièvement d'autres difficultés soulevées dans la littérature, notamment dans [14], [175]. Nous les évoquons uniquement aux fins de complétude, en donnant le principe de leur possible contournement, renvoyant le lecteur aux références pour plus de détails :

- Le système peut-il véritablement perdurer pendant un temps infini au sein de l'espace-temps de Malament-Hogarth ? Peut-il notamment éviter de heurter la singularité présente dans cet espace ? Dans le cas où l'espace de Malament-Hogarth est un trou noir de taille suffisante, le calculateur peut vivre à jamais sans heurter la singularité, ce qui le détruirait. La singularité n'est plus ponctuelle, mais prend la forme d'un anneau, et il n'est plus inévitable que le calculateur heurte la singularité [175].
- G. Piccinini affirme que le modèle de Malament-Hogarth ne satisfait pas la contrainte d'utilisabilité, si jamais les espaces-temps sont trop éloignés pour qu'on y ait accès ([190], 757). Pour que cette objection soit bien formulée, il faudrait pouvoir formuler une objection de principe contre l'accessibilité de ces espaces-temps. Même si cela pouvait être réalisé, cette objection repose sur une confusion sur le sens de la contrainte d'utilisabilité. Celle-ci impose que le dispositif de calcul envisagé soit utilisable par un observateur fini : elle ne suppose pas que cet observateur fini soit membre de notre civilisation terrestre. Notre position dans l'Univers est une contingence physique, et l'existence d'une autre espèce capable d'utiliser des espaces de Malament-Hogarth pour effectuer un calcul en serait aussi une. Même si aucun espace de Malament-Hogarth ne nous était accessible pour des raisons

20. Ce problème demeure implicite dans la discussion de J. Norton et J. Earman, qui s'inquiètent que la quantité de matière-énergie utilisée soit tellement grande qu'elle en vienne à perturber la structure de l'espace-temps autour d'elle, et Andréka et alii, qui envisagent de répartir le travail de calcul et donc la mémoire de façon distribuée avec une flotte de calculateurs, évitant ainsi une concentration excessive de matière-énergie ([175], 136-137). Les hypothèses sur les relations entre densité de l'information et densité de la matière-énergie qui seraient nécessaires pour préciser la portée de ce problème ne sont jamais explicitées.

de principe, cela n'interdirait en rien qu'ils soient accessibles à d'autres observateurs finis, et c'est tout ce que requiert la contrainte d'utilisabilité.
- *Répétabilité de la procédure.* Voir ([190]), 756-757, ([175], 17).

Transmission et sémantique du signal

Un autre ensemble de difficultés est lié à la terminaison de la procédure. On doit distinguer deux cas, selon que la réponse au problème posé soit positive ou négative [21] :

- Dans le cas positif, le programmeur doit être capable de recevoir et interpréter, avec une fiabilité correcte, le signal envoyé par le calculateur.
- Dans le cas négatif, le programmeur doit être capable d'interpréter *l'absence* de signal après un intervalle de temps fini prédéterminé comme signe que la conjecture reçoit une réponse négative : il n'existe pas de contre-exemple.

Dans les deux cas, le programmeur doit être capable de distinguer un véritable signal d'un bruit émergeant de l'espace-temps observé. Selon J. Earman et J. Norton, il serait nécessaire de réaliser des mesures arbitrairement précises dans le cas positif, où le programmeur doit recevoir un signal indiquant l'existence d'un contre-exemple à la conjecture. À cause de l'absence d'hyperbolicité globale de l'espace, tout point p dans l'espace de Malament-Hogarth est sujet à des influences non-déterministes, quelle que soit la précision à laquelle sont fixées les conditions initiales (voir lemme 3 dans ([87], 37-38)). Rien dans les lois de la physique n'empêche qu'un faux signal n'émerge de la singularité, et fasse trompeusement croire en l'existence d'un contre-exemple. Le programmeur pourrait ignorer ces faux signals s'il était capable de distinguer tous les faux signaux provenant de la singularité des véritables signaux provenant du calculateur. Mais une telle tâche supposerait une capacité de discrimination d'une précision arbitraire.

Comme J. Norton et J. Earman le signalent eux-mêmes ([87], 39), une telle difficulté peut être contournée à l'aide d'un protocole de communication plus sophistiqué. Pour augmenter la fiabilité de la communication, le message peut contenir non seulement l'information de l'existence d'un contre-exemple, encodable dans un simple bit, mais également le contre-exemple découvert. À l'aide de ce protocole, le programmeur n'a plus à déterminer la source du signal, puisqu'il peut vérifier qu'il a bien affaire à un contre-exemple. Mais le problème représenté par l'exigence d'une précision de me-

21. Nous rappelons au lecteur que nous considérons ici le problème sous la forme de l'existence d'un contre-exemple à une conjecture donnée. En cas de présentation du problème comme la recherche de la valeur de vérité de la conjecture, la convention désignant une réponse comme positive ou négative serait bien sûr inversée.

sure arbitraire rejaillirait sous une autre forme (*idem*) [22] :

> Unfortunately, γ_2 may still have to make arbitrarily fine discriminations since the quadruple sent will be of arbitrarily great size (=number of bits) and must be compressed into a correspondingly small interval at γ_2.

Un autre encodage de l'information permettrait cependant de contourner l'argument de J. Earman et J. Norton. La plausibilité de ce dernier provient en effet de la réduction implicite des possibilités d'encodage à une simple alternative : soit la communication de l'existence d'un contre-exemple par un simple bit, soit la communication du contre-exemple entier, par un signal de taille arbitraire. Il est cependant possible d'encoder l'existence d'un contre-exemple non par un simple bit, mais par un codage long et redondant, afin d'augmenter la fiabilité de la communication. J. Earman et J. Norton pourront bien répondre que rien dans les lois de la physique ne prévient la possibilité que la singularité émette exactement, par une monstrueuse coïncidence, le signal que programmeur et calculateur avaient fixé à l'avance. Mais les signaux envoyés dans des communications bien plus ordinaires sont eux aussi soumis à des bruits aléatoires, sans que cela ruine la sécurité de toute communication. Le fait que l'existence d'un bruit aléatoire soit ici une question de principe, puisque l'espace considéré n'est plus déterministe, ne constitue pas un changement radical de situation pour la sécurité de la communication.

Un autre problème de précision arbitraire est cependant soulevé par G. Etesi et I. Németi ([92], 21-22). Nous en donnerons ici une présentation intuitive. Dans le référentiel du programmeur, l'intervalle de temps entre deux événements observés dans le référentiel du calculateur tend vers 0 tandis que le programmeur tend vers le point p de réception du signal. Si l'on considère que dans le référentiel du calculateur, l'envoi du signal positif prend un intervalle de temps fixé, ceci implique que le programmeur devra pouvoir détecter le signal pendant un intervalle de temps arbitrairement court. Dans ce sens, le modèle de Malament-Hogarth exige lui aussi une précision de mesure arbitraire [23]. Nous reviendrons plus loin sur ce point crucial (voir section 6.5).

6.4 Les modèles quantiques d'hypercalcul

Le calcul quantique est indiscutablement le modèle de calcul non-standard qui a été l'objet de la plus grande attention scientifique durant ces deux dernières décennies. L'essentiel de ce travail a été motivé, et est consacré, aux

22. J. Earman et J. Norton parlent dans cet exemple d'un quadruplet de nombres, parce qu'ils utilisent comme exemple récurrent la conjecture de Fermat.

23. Pour plus de discussion sur ce problème, on pourra consulter ([175],15-16), et [62], 188-190).

propriétés de complexité de ce modèle. Nous aurons donc amplement l'occasion d'introduire ce modèle plus en détail lorsque nous étudierons la complexité (voir section 8.3.2). De manière générale, on considère que le calcul quantique n'introduit pas de nouvelles propriétés de calculabilité.

Cet énoncé mérite cependant qualification. Le premier modèle historique du calcul quantique, les machines de Turing quantiques, et le modèle le plus courant, qu'on peut qualifier de standard, les circuits quantiques, sont démontrablement simulables par une machine de Turing. Il existe cependant un modèle de calcul quantique moins étudié, le calcul quantique adiabatique, dont les propriétés de calculabilité ont été l'objet de discussion plus complexes. Il existe notamment un algorithme quantique adiabatique, l'algorithme de Kieu, qui est l'un des modèles d'hypercalcul les plus connus. Nous allons donc à présent exposer ce modèle en détail.

L'algorithme de Kieu

L'algorithme de Kieu fut conçu pour résoudre le dixième problème de Hilbert. Au lieu de traiter directement de l'existence de solutions de l'équation diophantienne $D(x_1, x_2, ..., x_n) = 0$, il s'attaque à un problème équivalent, celui de la valeur du minimum global du polynôme carré $D(x_1, x_2, ..., x_n)^2$: l'équation diophantienne admet au moins une solution ssi la valeur du minimum global est zéro, et n'en admet aucune sinon. Cette traduction permet un encodage plus facile du problème dans le système considéré, comme on va le voir immédiatement.

L'approche de T. Kieu consiste alors à encoder la donnée de ce problème dans l'Hamiltonien d'un système aisément préparable, puis de faire lentement évoluer cet Hamiltonien en un Hamiltonien dont l'état d'énergie fondamentale encode la solution. Pour fixer les idées, considérons l'équation diophantienne suivante ([145], 3-4) :

$$(x+1)^3 + (y+1)^3 - (z+1)^3 + cxyz \tag{6.8}$$

avec x, y, z les inconnues et $c \in \mathbb{Z}$. Dans un espace de Fock, à l'aide des opérateurs de création $a_x^\dagger$,$a_y^\dagger$ et $a_z^\dagger$, on construit l'Hamiltonien suivant :

$$H_p = ((a_x^\dagger a_x + 1)^3 + (a_y^\dagger a_y + 1)^3 - (a_z^\dagger a_z + 1)^3 + c(a_x^\dagger a_x)(a_y^\dagger a_y)(a_z^\dagger a_z))^2 \tag{6.9}$$

Les opérateurs $Nj = a_j^\dagger a_j$ ont uniquement des valeurs propres entières positives, et $[N_j, N_i] = [N_j, H_p] = 0$: toutes ces observables peuvent donc être mesurées simultanément. L'état fondamental $|g\rangle$ de l'Hamiltonien vérifie les équations suivantes :

$$\begin{aligned} &N_j|g\rangle = n_j|g\rangle, \\ &H_p|g\rangle = ((n_x+1)^3 + (n_y+1)^3 - (n_z+1)^3 + cn_xn_yn_z)^2|g\rangle = E_g|g\rangle \end{aligned} \tag{6.10}$$

pour quelque vecteur (n_x, n_y, n_z).

Le problème de décision peut donc être résolu par une simple mesure de la valeur propre d'énergie associée à l'état fondamental E_g : l'équation diophantienne admet au moins une solution entière ssi $E_g = 0$, et n'en a aucune sinon.

L'algorithme permet en réalité de faire plus que résoudre le problème de décision initial. Si la solution entière de l'équation est unique, l'algorithme permet d'exhiber un témoin : la mesure des observables correspondant aux opérateurs N_j permettra de connaître les valeurs solutions des inconnues. Dans le cas où il existerait plusieurs solutions, l'état fondamental $|g\rangle$ sera une superposition linéaire d'états de la forme $|n_x\rangle \otimes |n_y\rangle \otimes |n_z\rangle$ avec (n_x, n_y, n_z) les valeurs solutions : une seule mesure ne permettra pas de déterminer l'ensemble des solutions.

Pour garantir l'existence de l'évolution désirée, T.D. Kieu fait emploi du théorème adiabatique[24] quantique. Intuitivement, ce théorème affirme que, s'il existe un intervalle significatif entre la valeur propre associée à un état propre et le reste du spectre, une transformation suffisamment lente permet de maintenir un système dans cet état propre[25].

Dans sa formulation rigoureuse, le théorème quantique adiabatique est un théorème limite : l'état propre désiré n'est atteint avec certitude que lorsque $t \longrightarrow \infty$. Cependant, le théorème permet de déduire qu'un système réaliste peut approximer le comportement adiabatique en un temps critique fini T : pour tout $t > T$, l'état du système sera arbitrairement proche de l'état cible. De manière équivalente, une mesure du système donnera pour résultat la valeur propre associée à l'état fondamental avec une probabilité arbitrairement élevée.

Ce temps critique dépend de deux facteurs. Le premier est l'intervalle entre la valeur propre associée à l'état propre cible et la valeur propre la plus proche dans le spectre du Hamiltonien, g_{min}. Le second est une échelle de temps intrinsèque au Hamiltonien, ϵ. Un système approxime l'évolution adiabatique si la condition suivante :

$$T >> \frac{\epsilon}{g_{min}^2} \tag{6.11}$$

est satisfaite. Comme il n'existe pas d'algorithme permettant de calculer cet intervalle, il n'existe donc pas non plus d'algorithme permettant de calculer ce temps critique. En ce sens, le théorème adiabatique quantique est non-constructif : il assure l'existence d'un temps fini au-delà duquel l'état désiré sera atteint avec une probabilité arbitraire, mais il ne permet pas de calculer cette valeur.

24. En théorie quantique, le terme « transformation adiabatique » ne désigne pas, comme en thermodynamique classique, une transformation se réalisant sans échange de chaleur avec l'environnement, mais l'équivalent quantique du concept de transformation quasi-statique.

25. Pour une présentation rigoureuse, voir ([146], 8-9).

Problèmes de correction

Ce dernier point constitue une difficulté essentielle pour l'algorithme de Kieu. En l'absence de tout critère permettant de savoir que l'état fondamental désiré a été atteint avec une probabilité supérieure à un seuil fixé, cet algorithme ne permet pas d'obtenir le résultat recherché. En termes intuitifs, il n'est guère utile de savoir que le système va atteindre *à un temps donné* un état encodant la solution du problème, si jamais il est impossible de savoir quand on doit mesurer cet état. Dans le cas d'une équation admettant une solution unique, il est encore possible de vérifier la solution trouvée par un calcul simple, en testant le contre-exemple exhibé par l'algorithme. Dans le cas de multiples solutions, ou d'une réponse négative, une telle vérification est impossible, et il devient nécessaire de disposer de garanties théoriques que la procédure ne soit pas victime de faux négatifs dûs à l'existence de minimaux locaux. T.D. Kieu a appelé ce problème « problème d'identification de l'état fondamental »(*ground-state identification problem*).

En l'absence d'un algorithme permettant de calculer la valeur de g_{min}, la correction de l'algorithme de Kieu dépend donc de l'existence d'une solution opérationnelle à ce problème. T.D. Kieu prétend avoir trouvé un critère statistique opérationnel permettant d'identifier l'état fondamental : pour tout $t \in T$, l'état fondamental serait le seul à avoir une probabilité d'occupation supérieure à $\frac{1}{2}$. Pour savoir si l'état fondamental est atteint, et terminer l'algorithme de Kieu, il suffit de faire évoluer un grand nombre de copies du système pendant un temps T arbitraire. Si aucun état n'est mesuré dans plus de la moitié des cas, c'est que l'état fondamental n'a pas encore été atteint. Il suffit alors de doubler le temps d'évolution T, et d'itérer jusqu'à obtenir le résultat désiré. La terminaison de l'algorithme est ainsi garantie, sans qu'il soit nécessaire de calculer le temps T nécessaire pour atteindre l'état fondamental.

Dans son papier original, T.D. Kieu ne fournit pas de démonstration de la correction de son critère, et se contente de donner des exemples. Plusieurs critiques ont tâché de montrer l'incorrection de l'algorithme de Kieu en montrant l'incorrection de son critère statistique [213], [128]. T.D. Kieu a ensuite développé dans plusieurs publications une défense détaillée de la correction de son algorithme [146], [147] [145]. Notre intérêt philosophique étant centré sur les questions d'implémentabilité, nous allons généreusement supposer la correction de l'algorithme de Kieu, et renvoyer le lecteur avide de plus d'informations aux références mentionnées.

Peut-on parler d'une supertâche ?

Comme dans le cas relativiste, il est difficile d'appliquer aux modèles quantiques des catégories qui furent conçues avant leur apparition. C'est particulièrement vrai pour les catégories de calcul parallèle, et celle de su-

pertâche.

Le modèle quantique est parfois qualifié de modèle parallèle, dans la mesure où il prendrait avantage de l'évaluation simultanée de la fonction en plusieurs arguments grâce au phénomène de superposition des états quantiques. Un tel emploi du concept de calcul parallèle est cependant discutable à plus d'un titre. Le premier est que l'emploi du terme mériterait au moins qualification, pour souligner une différence radicale entre parallélisme quantique et parallélisme classique. Le parallélisme quantique ne s'appuie pas sur des sous-systèmes, mais sur la superposition des états d'un même système [26]. Le deuxième relie la question de l'application du concept de « calcul parallèle »à la difficile question de l'interprétation réaliste de la fonction d'onde. Pour présenter brièvement cette dernière, elle consiste essentiellement en la position suivante : lorsqu'un état $|\phi\rangle$ est écrit comme une superposition d'états propres dans la base d'une observable donnée, $|\phi\rangle = \sum_i c_i|e_i\rangle$ [27], cette superposition ne doit pas être interprétée comme un simple instrument théorique de représentation des probabilités des résultats de mesure possibles, mais comme une représentation de l'existence réelle du système en plusieurs états à la fois. Dans le cas particulier du calcul quantique, l'interprétation réaliste mène naturellement à une vision parallèle du calcul quantique de la manière suivante : la supériorité de ce calcul proviendrait de sa capacité à faire évoluer l'état d'un état initial représentant une superposition des données du calcul, $|\phi_{initial}\rangle = \sum_x c_x|x\rangle$, à un état final représentant la superposition des évaluations de la fonction f désirée en ces divers arguments, $|\phi_{final}\rangle = \sum_x c'_x|f(x)\rangle$. En termes intuitifs, le pouvoir calculatoire du calcul quantique proviendrait de sa faculté à exécuter le calcul en parallèle sur plusieurs états superposés du même système.

Par voie de conséquence, l'on peut douter de l'exécution du calcul en parallèle par un système quantique, comme on peut douter de la réalité des différents états superposés qui décrivent théoriquement l'état du système. Dans ([78], 114, l'auteur souligne), D. Deutsch a tâché au contraire de montrer que l'interprétation parallèle du calcul quantique constituait un argument décisif en faveur de l'interprétation réaliste de la fonction d'onde. Il mit ainsi les opposants de l'interprétation réaliste au défi de comprendre le fonctionnement du calcul quantique, en posant la question « où se déroule le calcul ? » :

> In explaining the operation of quantum computers I have, where necessary, assumed Everett's ontology [28] . Of course the expla-

26. Ce point a été justement souligné par P. Cotogno ([71], 205).

27. Nous nous restreignons ici au cas discret, qui est suffisant pour discuter des problématiques liées au calcul quantique, même si l'emploi d'un formalisme continu est nécessaire pour parler de la fonction d'onde en toute généralité.

28. Pour le lecteur ignorant de la littérature en interprétation de la mécanique quantique, l'ontologie d'Everett est une forme d'interprétation réaliste de la fonction d'onde, où les différentes branches de la superposition sont interprétées comme des univers parallèles.

> nations could always be 'translated' into the conventional interpretation, but not without entirely losing their explanatory power. Suppose, for example, that a quantum computer was programmed (...). Each day it is given different data. The Everett interpretation explains well how the computer's behavior follows from its having delegated subtasks to copies of itself in other universes. On the days when the computer succeeds in performing two processor-days of computation, how would the conventional interpretations explain the presence of the correct answer. *Where was it computed?*

L'argument de D. Deutsch est fondé sur plusieurs hypothèses fortes. Il suppose naturellement que le calcul quantique est implémentable, et qu'il présentera un véritable avantage de complexité par rapport au calcul classique, deux points qui sont encore débattus à ce jour (voir section 8.3.2). Il suppose également que le pouvoir calculatoire supplémentaire du calcul quantique provient de la superposition des états quantiques. D'autres interprétations concurrentes ont depuis lors été proposées (pour une discussion, voir [49]), parmi lesquelles certaines soulignent notamment le rôle de l'intrication (voir par exemple [144]).

Mais même si on admet aux fins de la discussion toutes ces hypothèses, la manière même de poser la question en termes de *localisation* du calcul dans les branches de la superposition, suppose une vision réaliste de la fonction d'onde, plutôt qu'elle ne la supporte. La superposition se produit dans un espace de Hilbert, l'analogue quantique d'un espace des phases, et non dans l'espace réel. La question de D. Deutsch suppose de considérer comme un problème légitime la localisation du calcul représenté dans notre formalisme par l'évolution des différentes branches. Mais la question soulevée par l'interprétation réaliste de la fonction est précisément la nécessité de passer d'une superposition d'états dans l'espace théorique à une superposition d'états dans l'espace réel. Par le postulat de réduction du paquet d'ondes, lors de l'exécution d'une mesure projective de von Neumann (resp. de la lecture du résultat du calcul), l'observateur n'a d'accès opérationnel qu'à la valeur associée à un des états superposés (resp. au résultat de l'évaluation en un unique argument). D'un point de vue opérationnel, la superposition n'est donc pas sensible dans le résultat d'une mesure, mais dans la statistique d'un ensemble de mesures. La question centrale de l'interprétation de la fonction d'onde -doit-on admettre une interprétation réaliste de la fonction d'onde pour expliquer les statistiques des résultats de mesure ?- se reproduit simplement dans le cas du calcul quantique -doit-on admettre une interprétation réaliste de la fonction d'onde pour expliquer les statistiques des résultats lus, notamment la haute fréquence relative du résultat correct ? L'interprétation du calcul quantique comme un calcul parallèle ne fait que reproduire, au niveau calculatoire, les problématiques rencontrées dans l'interprétation

réaliste de la fonction d'onde.

Le flou qui affecte la catégorie de calcul parallèle affecte nécessairement l'emploi de la catégorie de supertâche. Il est naturel de parler de supertâche, dans la mesure où l'on considère que l'algorithme de Kieu exploite la dimensionnalité infinie de son système pour faire exécuter en parallèle une recherche infinie sur toutes les valeurs entières. Si l'on considère une telle présentation comme une mésinterprétation du formalisme quantique, il devient bien moins naturel de parler de supertâches. Là encore, les catégories conçues pour décrire le calcul effectif sont inopérantes lorsqu'elles sont appliquées à un modèle physiquement défini.

Critique de l'implémentabilité de la procédure

Dans [135], A. Hodges a affirmé que l'implémentation de l'algorithme de Kieu nécessiterait une précision infinie. Cette critique n'est pas à première vue évidente, car, contrairement à nombre de modèles analogues d'hypercalcul, le schème de Kieu ne nécessite pas l'exécution d'une mesure arbitrairement précise pour lire le résultat du calcul. Nul n'est besoin d'une mesure arbitrairement précise pour :

- distinguer l'état fondamental du système.
- estimer le temps d'évolution accompli.
- estimer la distribution de probabilité associée à l'état par les fréquences relatives.

Le spectre d'énergie du système étant discret, il n'est pas besoin d'une précision arbitraire pour distinguer l'état fondamental. L'exécution correcte de l'algorithme requiert uniquement que le temps d'évolution et la probabilité d'occupation d'un état donné aient dépassé un certain seuil, et non qu'ils aient adopté une valeur d'une précision arbitraire.

Ce n'est donc pas au niveau de la lecture du résultat, mais à celui de la préparation du système, que vont porter les arguments de Hodges ([135], 6-7). Avant de les exposer, il faut souligner avec T.D. Kieu que A. Hodges omet de distinguer une précision infinie d'une précision arbitraire [148]. Le premier argument d'Hodges reproche au schème de Kieu d'exiger une précision infinie, le second lui reproche une exigence de précision arbitraire :

1. Le processus de mesure nécessite l'implémentation de l'opérateur nombre $N = a^{\dagger}a$, dont l'expression est :

$$N = -L^2 \frac{\partial^2}{\partial x^2} + \frac{x^2}{L^2} \tag{6.12}$$

   avec $L \in \mathbb{R}$ la longueur caractéristique du système. L'application de cet opérateur requerrait l'encodage de L avec une précision infinie.

2. Cet argument a également été utilisé par Warren D. Smith ([213], 7-8), dont nous reprendrons ici la présentation. On considère le système

d'équations diophantiennes suivantes :

$$x^2 - 2y^2 = 0, \quad x = a^2 + b^2 + c^2 + d^2 + 1 \tag{6.13}$$

Ce système n'admet pas de solutions rationnelles. En revanche, si l'on substitue à 2 n'importe quel rationnel au carré, le système admet une infinité de solutions. On peut en particulier choisir de substituer à 2 n'importe quel rationnel au carré arbitrairement proche de 2. Pour résoudre cette instance du dixième problème à l'aide de l'algorithme de Kieu, il faut encoder le coefficient dans l'Hamiltonien avec une précision arbitraire.

À (1), T.D. Kieu a répondu que l'implémentation de son algorithme ne nécessitait aucunement d'encoder avec une précision infinie la longueur caractéristique, ou la fréquence, de l'oscillateur considéré, mais seulement de distinguer les quanta. C'est ainsi qu'il est possible de compter les photons, et de confirmer la nature corpusculaire de la lumière, même s'il est impossible de mesurer leur fréquence avec une précision arbitraire.

Tout en refusant le reproche de précision infinie, T.D. Kieu admet que son schème exige une précision arbitraire, comme le montre (2). Il défend que rien n'interdit d'obtenir cette précision arbitraire, et que la troisième loi de la thermodynamique, par analogie, nous permet d'approcher arbitrairement le zéro absolu, même si elle interdit de l'atteindre. De même aucun principe n'interdit d'implémenter un Hamiltonien quelconque, et l'énoncé d'un tel principe serait une innovation théorique majeure. Nous allons revenir sur ce problème dans la section suivante (voir section 6.5.2).

6.5 Précision arbitraire et densité finie de l'information

6.5.1 La précision de mesure bornée

De notre revue synthétique des modèles d'hypercalcul, nous pouvons tirer la conclusion suivante : *l'implémentation de tous les modèles d'hypercalcul physiquement définis exigerait une précision arbitraire.* Dans cette section, nous allons examiner sur quelles bases théoriques on peut rejeter tout modèle de calcul exigeant une telle précision. Mais avant de procéder, il nous faut encore préciser la signification de cette exigence de précision arbitraire.

Sous l'appellation d'*exigence de précision arbitraire*, trois difficultés doivent être distinguées en principe :

1. *Préparation idéale.* Pour pouvoir fonctionner comme modèle de calcul, le modèle devrait être réalisé de manière parfaite. Or un modèle physique contient habituellement des idéalisations qui ne peuvent être réalisées en laboratoire que de manière approchée. Dans le modèle

d'O. Bournez et M. Cosnard, la représentation des trajectoires par des droites parfaitement rectilignes, la section d'un plan avec un angle droit parfait, et la représentation d'une particule par un point sans dimension constituent des idéalisations impossibles à réaliser parfaitement, alors que la moindre approximation pourra entraîner des erreurs dans l'exécution du calcul. On résumera cette difficulté en disant qu'un tel modèle exige une *préparation idéale* du système physique l'implémentant.

2. *Validité théorique arbitraire.* Il s'agit là de la critique formulée par M. Davis, selon laquelle on ne peut raisonnablement espérer d'une théorie qu'elle soit valide à une précision arbitraire. Nous allons revenir ci-dessous sur cet argument.
3. *Accessibilité épistémique* ou *robustesse au bruit.* Quand bien même les deux difficultés précédentes pourraient être surmontées, nous n'aurons jamais opérationnellement accès à la quantité infinie d'information théoriquement contenue dans notre système. L'existence incontournable de bruit dans tout processus de mesure réaliste nous interdira toujours de réaliser des mesures à une précision arbitraire, comme il serait nécessaire pour extraire la valeur supposée être calculée par le système.

La condition de validité théorique arbitraire peut recevoir quatre interprétations différentes, dont les deux plus faibles ne faisaient pas partie de l'argument original de M. Davis :

- *Emploi d'une théorie au-delà de son domaine de validité connu.* Certains modèles d'hypercalcul, notamment les modèles de Beggs et alii [23], [22], [24] supposent qu'un ensemble d'hypothèses de la mécanique newtonienne sont valables à toute échelle de précision, afin d'utiliser la mesure d'une quantité classique comme un oracle. Outre que cela suppose d'accepter certaines idéalisations fortes de cette théorie, comme la représentation de certains systèmes par des points matériels, cela néglige l'échec historique de la physique classique à l'échelle atomique. En supposant qu'une théorie T est valide à une précision arbitraire, on fait donc une hypothèse contraire à notre savoir établi en physique théorique [29].
- *Limites des théories fondamentales courantes.* La deuxième interprétation part du constat que même nos théories physiques les plus fondamentales, à savoir la théorie quantique des champs et la relativité générale, ne sont pas censées être valides à une précision arbitraire. Ces deux théories ne peuvent être conciliées à l'échelle de Planck, qui est l'échelle à laquelle les effets de la gravitation seraient censées de-

29. Nous n'impliquons nullement qu'un des auteurs mentionnés serait victime d'une quelconque illusion sur ce point. Beggs et alii présentent explicitement leur travail comme une étude du pouvoir expressif autorisé par la physique classique.

venir sensibles à l'échelle quantique. La nécessité d'un dépassement de ces théories par une nouvelle théorie de la gravitation quantique est donc déjà admise, même si la formulation exacte de cette théorie est inconnue. Cette seconde interprétation est semblable à la première, en ce qu'elle nie la validité de ces théories à une échelle arbitraire. Elle en diffère cependant, en ce qu'on ignore la théorie prenant le relais au-delà du domaine de validité de ces dernières. Libre au physicien d'imaginer que cette théorie utilisera un formalisme continu et sera valide à une précision arbitraire, même si l'on verra que ce n'est certainement pas la direction choisie par tous les physiciens théoriciens (voir section 6.5.3). On remarquera que cet argument suffit pour à mettre en doute tout modèle d'hypercalcul basé sur une théorie physique admise et supposant une précision arbitraire. S'il n'exclut pas absolument la possibilité d'une violation de la thèse de Church-Turing physique par un modèle d'hypercalcul exigeant une précision arbitraire, cet argument en fait une possibilité purement spéculative, renvoyée aux calendes grecques d'une future révolution de la physique fondamentale.

- *Interprétation métathéorique.* La troisième interprétation est celle défendue par M. Davis. Celle-ci consiste à rejeter la possibilité même d'une théorie valable à une échelle de précision arbitraire, sur la base d'une induction métathéorique de nature historique : puisqu'aucune de nos théories physiques n'a été valide à plus de quelques décimales, il serait présomptueux d'attendre d'aucune théorie fondant un modèle de calcul à venir qu'elle soit valide à une précision arbitraire. L'argument de M. Davis est donc une forme de ce que les philosophes anglophones appellent le pessimisme inductif.
- *Interprétation théorique fondationnelle.* Cette dernière interprétation est basée sur un pari sur le devenir de la physique théorique fondamentale. Selon ce pari, non seulement l'avenir de la physique théorique ne nous révélera pas une théorie valide à un degré de précision arbitraire, mais elle nous révélera une théorie qui ruinera, par des arguments de principe, toute prétention à une précision arbitraire. Selon une version faible de cet argument, la théorie à venir pourra limiter la précision accessible sur certaines observables, comme les théories discrétisant l'espace et le temps, qui priveront de sens toute mesure de distance ou d'intervalle de temps au-delà d'une certaine échelle. Selon une version forte, les principes de la théorie excluront toute possibilité d'effectuer une quelconque mesure de précision arbitraire sur un système. Cette théorie défendra une version ou une autre d'un « principe de densité finie de l'information », selon lequel un système occupant un volume fini V ne peut contenir qu'une quantité finie d'information. Nous aurons l'occasion de revenir sur ce point lorsque nous discuterons du principe holographique (voir section 6.5.3). Pour le moment, nous nous contenterons de remarquer qu'une telle objection, même si elle

demeure fondée sur des spéculations théoriques, est bien mieux fondée en principe que l'interprétation métathéorique, qui s'appuie non sur de véritables principes, mais sur un bon sens théorique inspiré par l'histoire de la physique.

Avant d'examiner la possibilité d'un fondement théorique du rejet de la précision arbitraire, nous allons examiner de plus près la distinction entre préparation idéale et précision arbitraire, en présentant des modèles notionnels nécessitant la première sans nécessiter la seconde. Ceci nous mènera à interroger la capacité de la théorie quantique, sous sa forme actuelle, à placer suffisamment de limites sur les systèmes préparables en principe.

6.5.2 Principes, préparation et précision

L'intégralité des modèles que nous avons étudiés jusqu'à présent, lorsqu'ils définissent physiquement l'opération de lecture finale du résultat, encodent ce résultat dans la valeur d'une quantité mesurable. Mais les prédictions de la physique ne portent pas que sur des valeurs de quantités mesurables. Elles peuvent aussi être formulées en termes de probabilité. Ces probabilités, si elles ne sont pas directement observables, sont estimables par des fréquences relatives d'événements observables. Un tel changement d'encodage, comme nous allons le voir immédiatement, permet à première vue de contourner les objections liées à la robustesse au bruit. Nous allons donc à présent nous pencher sur les modèles de calcul probabilistes.

Pour comprendre en quoi les modèles probabilistes peuvent être un objet privilégié pour l'étude de l'hypercalcul, il faut distinguer plusieurs espèces de modèles probabilistes. Le modèle probabiliste le plus courant est la machine de Turing probabiliste. Celle-ci est constituée par une machine de Turing où la fonction de transition est remplacée par une relation de transition. Ce modèle n'est donc pas déterministe, au sens où la configuration $n+1$ n'est plus une fonction de la configuration n. Pour effectuer le choix entre les différentes options disponibles à chaque étape de calcul, la machine de Turing probabiliste interagit avec une source d'aléatoire, celle-ci pouvant être modélisée par un lancer de dé ou un générateur de nombre aléatoires. La valeur donnée par la source d'aléatoire permet de choisir entre les différentes options offertes par la relation de transition. La transition entre deux étapes de calcul ne dépend donc plus uniquement du contenu de la case lue, et de l'état du processeur, mais aussi du résultat donné par la source d'aléatoire.

Un calcul probabiliste peut donc s'achever sur différentes sorties, même s'il est initialisé sur la même entrée : il est donc nécessaire de redéfinir le calcul d'une fonction pour de tels modèles. Il existe deux définitions courantes. Selon la première, une Machine de Turing Probabiliste calcule une fonction f ssi, sur la donnée d'une entrée x, et pour toute mesure de précision $j \in \mathbb{N}$, elle produit comme sortie la valeur $y = f(x)$ avec une probabilité $p \geq 1-2^{-j}$. Selon une seconde définition alternative, une machine de Turing Probabiliste

calcule une fonction f ssi sur la donnée d'une entrée x, elle produit la valeur $y = f(x)$ avec une probabilité supérieure à $k = \frac{1}{2}$. La première définition exige que la machine puisse produire la valeur correcte avec une confiance arbitrairement grande ; la seconde avec une confiance franchissant un certain seuil. Les probabilités interviennent donc à deux niveaux au sein d'un modèle probabiliste : comme source de choix aléatoire entre les différentes options offertes par la relation de transition, et comme poids dans la détermination du résultat correct.

Il est aisé de montrer qu'un tel modèle est simulable par une machine de Turing déterministe à plusieurs bandes, et respecte donc la thèse de Church-Turing algorithmique. L'idée de base de la simulation est de simuler en parallèle les différents embranchements offerts par la relation de transition, tout en gardant trace de la probabilité d'obtention de chaque résultat.

La source d'aléatoire est typiquement présentée comme un lancer de pièce non-faussée. On peut montrer que l'emploi d'une pièce faussée, si le biais de la pièce est un réel calculable quelconque, ne modifie en rien l'ensemble des fonctions calculables au titre de ce modèle [30].

Dans un article récent [182], T. Ord et T. Kieu ont étudié le pouvoir expressif d'un machine de Turing probabiliste équipée d'une pièce avec un biais non-récursif. Dans ce cas de figure, il est aisé de démontrer que la machine de Turing probabiliste devient un modèle d'hypercalcul. On peut ainsi programmer une telle machine pour qu'elle effectue une série de lancers de la pièce faussée, et qu'elle note la fréquence relative des '1' parmi les résultats de lancer. Par la loi des grands nombres, cette suite de fréquences doit converger vers la probabilité d'obtenir 1, permettant ainsi l'évaluation d'un réel non-calculable.

T. Ord et T. Kieu démontrent également qu'une telle machine pourrait calculer l'expansion binaire de cette même probabilité. Si la probabilité en question encode un oracle X [31], la machine, en écrivant son expansion binaire, acquiert un pouvoir calculatoire équivalent à celui d'une o-machine munie d'un oracle décidant l'appartenance à X. Si X n'est pas récursif, la machine de Turing probabiliste est alors un modèle d'hypercalcul.

Le modèle probabiliste décrit par T. Ord et T. Kieu présente un intérêt particulier pour la discussion de la thèse de Church-Turing physique. Il s'agit bien évidemment d'un modèle notionnel : le protocole permettant de créer une pièce avec un biais non-récursif, ou les autres variantes envisagées de la source aléatoire, n'est à aucun moment défini dans les termes d'une théorie physique. Ce modèle a néanmoins l'avantage particulier qu'il ne requiert pas d'effectuer une mesure arbitrairement précise d'un tel modèle pour disposer d'une implémentation réaliste. Une implémentation d'un tel modèle exigerait

30. Voir [182] pour plus de détails sur ce point.

31. La probabilité p_X encode l'oracle décidant l'appartenance à un ensemble X ssi le n-ième digit de l'expansion binaire de p_X est 1 ssi $n \in X$ et O sinon.

uniquement qu'on puisse distinguer l'encodage d'un '0' de l'encodage d'un '1', opération que des décennies d'implémentation de machines numériques ont permis de réaliser de manière extrêmement fiable. Ce point est explicitement souligné par les auteurs ([182], 261) :

> Using randomness provides an alternative that does not run afoul of these limitations [of measurement precision]. It allows one to measure an underlying continuous quantity with a sequence of discrete measurements that do not individually become increasingly more accurate. It is the increasing total number of measurements that provides the accuracy, so no particular measurement needs to be more accurate than the quantum limits.

La mécanique quantique semble à première vue être un terrain propice où rechercher une implémentation d'une pièce avec un biais non-récursif. En théorie, elle permet en effet l'implémentation d'une pièce parfaite, par l'application d'une porte Hadamard à un qubit : $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. La mesure d'un qubit dans un tel état peut donner chacune des valeurs de bits avec une probabilité théorique $\frac{1}{2}$ parfaite. En outre, la théorie quantique permet aux amplitudes de probabilité d'adopter n'importe quelle valeur, et les probabilités induites par la prise du carré du module, peuvent être décrites par des réels arbitraires compris entre 0 et 1. D'un point de vue théorique, rien n'interdit donc à un qubit d'implémenter une pièce avec un biais arbitraire.

De prime abord, l'encodage de l'information non-calculable dans une fréquence relative permet donc de contourner les objections liées à la robustesse au bruit, et de poser des questions théoriques nouvelles, liées à la capacité à préparer un système avec une précision idéale. Comment préparer un système encodant un oracle non-récursif ? Intuitivement, il semble quelque peu miraculeux qu'un système physique en vienne à encoder non seulement une infinité d'information, mais une infinité d'information encodant les valeurs de la fonction d'arrêt, ou de quelque autre problème mathématique indécidable. Mais peut-on trouver dans la physique théorique actuelle une raison de principe pour rejeter une telle possibilité ?

Le même problème était déjà soulevé par T.D. Kieu à propos de son algorithme : rien n'interdit dans la théorie quantique de préparer un Hamiltonien arbitraire, y compris l'Hamiltonien encodant la solution du dixième problème de Hilbert. Ce problème a également été soulevé, dans une perspective plus générale, par M. Nielsen [177]. Il y définit l'observable d'arrêt h de la manière suivante :

$$h = \sum_{x=0}^{\infty} h(x)|x\rangle\langle x| \tag{6.14}$$

avec $\{|x\rangle\}$ une base orthonormale de l'espace des états, de dimension infinie dénombrable, de quelque système physique. L'observable d'arrêt est bien une observable au sens usuel de la théorie quantique : elle est un opérateur hermitien défini sur l'espace des états du système considéré. Par conséquent,

elle est en principe réalisable. Cependant, si tel est bien le cas, il devient trivial d'implémenter un modèle d'hypercalcul calculant la fonction. Il suffit de bâtir l'appareil de mesure correspondant à l'observable d'arrêt, de préparer le système dans l'état $|x\rangle$, et d'exécuter une mesure de l'observable sur le système. On obtiendra alors avec une probabilité 1 la valeur de la fonction d'arrêt pour l'argument x.

Un argument similaire peut être formulé pour les évolutions unitaires : il est possible de définir un opérateur unitaire, qui, s'il est réalisable, permettra d'envoyer un état $|x\rangle$ sur un état qui, mesuré dans la base $\{|x\rangle\}$, donnera pour résultat avec probabilité 1 une solution du problème de l'arrêt pour l'argument x [32]. Là encore, la théorie quantique permet en principe de violer la thèse de Church-Turing physique.

La possibilité d'implémenter un Hamiltonien ou une évolution unitaire arbitraire n'est certes pas un postulat de la mécanique quantique, ni un théorème démontrable à partir de ses postulats. Elle fait partie de ce qu'on peut appeler l'acception courante de la théorie, qui admet qu'en règle générale un Hamiltonien ou une évolution unitaire sont préparables. Il n'y a donc rien dans la théorie quantique en elle-même qui exclut de telles possibilités. D'après M. Nielsen, le physicien se trouve donc placé face au dilemme suivant : soit il accepte de modifier la théorie quantique pour interdire ces violations de la thèse de Church-Turing physique, soit il rejette cette même thèse.

Il est possible de voir une telle situation, comme le fait T.D. Kieu, comme un argument en faveur de l'implémentabilité de principe d'un modèle d'hypercalcul quantique. Mais il est aussi possible, et à notre sens préférable, de le voir comme un argument critique à l'égard de la théorie quantique, et du caractère relâché des contraintes qu'elle fait peser sur la préparation d'un système physique. Une telle critique n'est pas fondée sur une adhésion dogmatique préalable à la thèse de Church-Turing physique, mais sur la remarque suivante. L'autorisation implicitement accordée par notre théorie à des observables comme l'observable d'arrêt ne résout en aucune façon le problème de la construction d'un appareil de mesure capable de la mesurer. Notre théorie ne nous donne aucune indication sur une procédure permettant de préparer un tel appareil, et sa permission de principe ne mène donc à aucune prédiction testable. Le caractère indûment relâché des contraintes pesant sur la préparation d'une observable est rendue manifeste par l'existence d'opérateurs hermitiens qui, de l'avis général, ne représentent aucune observable qu'on puisse mesurer en laboratoire [33]. Plutôt que de prendre appui sur notre théorie quantique pour affirmer la possibilité d'implémenter un modèle d'hypercalcul, il est plus sage d'admettre que cette théorie n'est pas en l'état armée pour répondre à une telle question.

32. Voir ([177], 2), pour la définition exacte de cet opérateur unitaire.
33. Voir [21] pour une introduction à ces problématiques.

La précision de mesure arbitraire et la préparation idéale constituent donc bien deux problèmes distincts, auxquels nos théories physiques actuelles semblent incapables d'apporter une réponse principielle. Cependant, tous deux seraient résolus en faveur de la thèse physique si une forme du principe de densité finie de l'information venait à être admise. D'un point de vue intuitif, la précision de mesure arbitraire et la préparation idéale constituent deux problèmes duals l'un de l'autre. Le premier porte sur notre capacité à encoder de l'information dans un système ; le second sur notre capacité à l'en extraire. Un principe de densité finie de l'information interdit à la fois l'encodage et l'extraction d'une quantité arbitraire d'information dans et à partir d'un système donné. Il suffirait donc à rejeter tous les modèles d'hypercalcul physiquement définis connus à ce jour. Dans la section suivante, nous allons montrer qu'il existe dans la recherche théorique contemporaine un candidat précis au titre de principe de densité finie de l'information, et exposer les raisons qui plaident en faveur de son adoption.

6.5.3 Le principe holographique : un fondement théorique de la thèse de Church-Turing physique ?

Dans le paysage de la recherche théorique actuelle, le principe holographique est à notre connaissance le seul principe candidat au titre de principe de densité finie de l'information. Nous résumerons ici la vaste présentation synthétique de ce principe réalisée par R. Bousso [47], en l'assortissant de quelques commentaires, avant de l'appliquer aux considérations pertinentes pour notre travail.

Pour donner substance à l'idée, qu'un volume fini d'espace ne peut contenir qu'une quantité finie d'information, il faut tout d'abord trouver un corrélat théorique à la notion intuitive d'information, souvent employée de manière polysémique et peu rigoureuse. La suggestion la plus naturelle pour un physicien est de considérer le concept d'entropie.

Les premières bornes sur la quantité d'entropie dans un volume fini sont issues de la réflexion théorique sur les trous noirs, et des paradoxes suscités par l'application à ces systèmes du concept d'entropie. Ces premières bornes sont universelles, au sens elles ne dépendent pas de la composition de la matière. Mais elles sont cependant sujettes à des limitations particulières -notamment que l'action de la gravité soit suffisamment faible- sur lesquelles nous aurons l'occasion de revenir.

L'état stationnaire d'un trou noir est caractérisé par trois quantités : masse, moment angulaire, et charge. Lorsqu'un système massif s'effondre pour former un trou noir, celui-ci se stabilise ultimement dans un état stationnaire unique. Du point de vue d'un observateur extérieur, la formation d'un trou noir semble violer le second principe de la thermodynamique : peu importe l'état entropique du système initial, le trou noir final a une entropie nulle. L'espace des phases a été dramatiquement réduit, et des conditions

initiales très différentes mèneront à des résultats finaux identiques.

Une solution possible de ce paradoxe est suggéré par un théorème nommé *no-hair*. Stephen Hawking a démontré que l'aire de l'horizon d'événement d'un trou noir ne décroît jamais avec le temps :

$$dA \geq 0. \tag{6.15}$$

En outre, si deux trous noirs fusionnent, l'aire de l'horizon d'événément du nouveau trou noir excédera l'aire totale des deux trous noirs initiaux. L'aire d'un trou noir a ainsi un comportement analogue à l'entropie : elle est une fonction du temps monotone non-décroissante et additive. En outre, lorsqu'un objet sombre dans un trou noir, son entropie est perdue pour un observateur extérieur, mais l'aire du trou noir s'accroît : on peut ainsi concevoir l'accroissement de l'aire comme une compensation pour la perte d'entropie.

Sur la base de ce raisonnement, le physicien J. Bekenstein a introduit l'idée que l'entropie d'un trou noir est proportionnelle à son aire d'horizon $S_{BH} = \eta A$, avec η un nombre de l'ordre de l'unité.

Le coefficient de proportionnalité fut calculé par S. Hawking : $S_{BH} = \frac{A}{4}$ [34].

De plus, J. Bekenstein a proposé que le second principe de la thermodynamique ne vaille que pour la somme de l'entropie de la matière et de l'entropie du trou noir :

$$dS_{total} \geq 0. \tag{6.16}$$

Cette relecture du second principe de la thermodynamique est appelée Second Principe Généralisé (*Generalized Second Law*). Si le *no-hair theorem* affirme que pour un observateur extérieur, l'état classique du trou noir est unique, rien n'interdit que cet état unique soit compatible avec une multiplicité de microétats quantiques, dont la modélisation précise est encore objet de recherche.

L'application de ce Second Principe Généralisé aux trous noirs pose encore un autre problème. L'entropie d'un système tombant dans un trou noir est perdue pour l'observateur extérieur. L'aire du trou noir s'accroît certes, mais uniquement en fonction de la masse de l'objet considéré. S'il existait de la matière avec une entropie arbitrairement grande pour une masse et une taille donnée, on pourrait violer le Second Principe Généralisé. On peut alors choisir d'inverser le point de vue et d'exiger que le Second Principe Généralisé vaille pour tout processus.

Bekenstein a ainsi argumenté que pour tout système soumis à une interaction gravitationnelle faible dans un espace asymptotiquement plat (*weakly gravitating system in asymptotically flat space*), le Second Principe Généralisé impliquerait que

$$S_{matière} \leq 2\pi ER. \tag{6.17}$$

34. Les équations sont ici exprimées, sauf indication contraire, en unités de Planck, où $\hbar = G = c = k = \epsilon_0 = 1$. L'expression originale en unités SI est donc $S_{BH} = \frac{kAc^3}{4G\hbar}$.

avec E la masse-énergie totale du système, et R le rayon de la plus petite sphère entourant le système. La gravitation doit être suffisamment faible pour le système puisse être considéré au repos, et l'espace euclidien. Si on se limite à des systèmes en interaction gravitationnelle faible, la borne de Bekenstein n'a pas été réfutée empiriquement.

Des problèmes se posent immédiatement pour la généralisation d'un tel argument aux systèmes en interaction gravitationnelle forte. Il devient alors difficile de définir le rayon d'un système dans une géométrie fortement courbée. Cela fonctionnera à tout le moins pour un système à symétrie sphérique : R sera alors défini en termes de l'aire de surface.

Un trou noir de Scharzschild en quatre dimensions a $R = 2E$, et donc son entropie de Bekenstein est :

$$S = \frac{A}{4} = \pi R^2. \tag{6.18}$$

ce qui sature exactement la borne de Bekenstein. On peut donc affirmer, à la lumière de cet argument, qu'un trou noir est l'objet le plus entropique qu'on puisse mettre dans une surface sphérique.

Une autre approche, due à P. Susskind, consiste à formuler une borne d'entropie sphérique. Au lieu de considérer un processus de Geroch, où l'on ajoute de la matière à un trou noir, on considère la conversion d'un système en trou noir. Susskind a alors argumenté que le Second Principe Généralisé impliquerait que

$$S_{matière} \leq \frac{A}{4} \tag{6.19}$$

où A est une aire, définie de manière convenable, entourant le système. De nouveau, cette borne est valide pour tous les systèmes à symétrie sphérique, et pour les systèmes en interaction gravitationnelle faible, mais elle ne peut être définie pour les systèmes soumis à une interaction gravitationnelle forte et ne respectant pas la symétrie sphérique (*strongly gravitating system without spherical symmetry*). Le système doit être en outre stable pour une échelle de temps suffisante à la formation du trou noir. Enfin, la masse du système doit être inférieure à la masse M d'un trou noir de même aire de surface. Dans le cas contraire, le système ne serait pas gravitationnellement stable, et formerait déjà un trou noir du point de vue d'un observateur extérieur. La borne sphérique est en quelque sorte plus faible que la borne de Bekenstein, lorsqu'elles peuvent toutes deux être appliquées, mais la borne sphérique a l'avantage de pouvoir être généralisée en une forme covariante, et d'être ainsi plus apparentée au principe holographique.

Une deuxième série de travaux permet de définir une borne supérieure pour la mémoire encodable dans un système arbitraire. Nous allons comparer cette deuxième approche avec l'approche centrée sur la thermodynamique des trous noirs, pour mieux voir les convergences et divergences d'argumentations entre elles.

Là encore, on définit le nombre maximal de bits encodables dans un système par son entropie. Puisque l'on cherche à énoncer des bornes valides pour une description théorique ultime de la matière, on adopte une description en termes quantiques. Soit N le nombre de bits encodables dans un système, m la dimension de son espace des états $\mathcal{H}$, on a

$$N = \ln(m) = \ln(dim(\mathcal{H})) \tag{6.20}$$

Le nombre de degrés de liberté est donc défini à une constante multiplicative près, qui dépendra du choix d'encodage représenté par le choix d'une base du logarithme. Ainsi pour un système de 100 spins, chacun pouvant adopter deux états, on a $N = \ln(2^{100}) = 100 \ln 2$.

Cette quantité est couramment interprétée comme la quantité d'information nécessaire pour décrire complètement toute configuration physique possible du système, ou, de manière équivalente, comme la quantité d'information encodable dans ce même système.

Pour quantifier exactement cette information, il est nécessaire de choisir un modèle théorique fondamental qui précisera le nombre de degrés de liberté. Si l'on choisit par exemple une théorie quantique des champs, la quantité d'information n'est pas même finie. La théorie associe un oscillateur harmonique quantique à chaque point de l'espace. Pour décrire la physique d'un volume fini, on devrait donc prendre en compte une infinité d'oscillateurs harmoniques, dont l'espace de Hilbert est de dimension infinie. On a $N = \infty$.

Une telle estimation peut difficilement prétendre être réaliste, puisqu'elle exclut tout effet de la gravité. On peut par exemple supposer qu'aucune distance inférieure à la longueur de Planck, $l_P = 1.6 \times 10^{-33} cm$, n'est définie. On discrétise ainsi l'espace par un maillage de Planck (*Planck grid*) dont chaque case contient un oscillateur harmonique. Le spectre d'énergie de cet oscillateur est supérieurement bornée par l'énergie de Planck, $M_P = 1.3 \times 10^{19} GeV$, l'énergie la plus grande contenue dans un cube de Planck sans provoquer la formation d'un trou noir. Le spectre est également discrétisé et bornée inférieurement par des effets de volume fini. Pour un volume V en unités de Planck, il y aura donc V oscillateurs harmoniques à n états, $m\ n^V$. On obtient $N = ln(m) = V \ln(n) \geq V$, ce qui capture l'intuition que le nombre de degrés de liberté croit avec le volume.

Cette estimation est cependant beaucoup trop large pour un partisan des bornes en entropie que nous venons de présenter. D'après la borne sphérique de Susskind, l'entropie maximale contenue dans une sphère d'aire A, soit l'entropie d'un trou noir contenu dans cette sphère, S_{BH}, est proportionnelle à cette aire. Selon l'interprétation statistique de l'entropie, le nombre d'états quantiques compatibles avec une entropie S est e^S. Par conséquent, on a

$$N = \ln(e^S) = \ln(e^{A/4}) = \frac{A}{4}. \tag{6.21}$$

Cette estimation d'un nombre de degrés de liberté proportionnel à l'aire, et non plus au volume, est encore une fois justifiée par une prise en compte des effets gravitationnels. La limitation de la masse-énergie permise par cube de Planck est insuffisante pour garantir la stabilité gravitationnelle du système à de plus grandes échelles : la masse totale de la sphère pourrait encore excéder celle d'un trou noir de même dimension (pour plus de détails, voir ([47], 13)).

Du point de vue aérien autorisé par la discussion de la calculabilité, ces différences importent peu. La valeur exacte de la borne en information contenue dans un volume V, et sa relation aux autres grandeurs physiques, n'est pas tant essentielle que l'existence même d'une borne. On voit que celle-ci est imposée, contre le formalisme infinitaire de la théorie quantique des champs, par une prise en compte des effets gravitationnels, et plus précisément par l'exigence que notre système soit gravitationnellement stable.

Du point de vue de la signification empirique de la thèse de Church-Turing, le principe holographique s'analyse en deux composantes : un principe de densité finie de l'information, et une composante justifiant le qualificatif d'holographique, qui affirme que la quantité finie d'information contenue dans un volume peut être projetée sur une surface. Cette deuxième composante de la proposition, la plus paradoxale, est superflue du point de vue des discussions de la thèse physique. Nous reviendrons sur ce point en conclusion.

Reste à discuter la valeur de l'exigence de stabilité gravitationnelle. Pour le théoricien libéré des contingences pratiques, la formation d'un trou noir ne signifie pas nécessairement la perte de toute information. Le résultat ci-dessus implique qu'un système décrit par une théorie quantique des champs s'effondrera en un trou noir s'il est par trop excité. Il reste à montrer que les degrés de liberté supplémentaires décrits par la théorie quantique des champs doivent être supprimés, ou si l'on doit considérer qu'ils sont dissimulés à l'intérieur du trou noir. R. Bousso souligne très explicitement ce point :

> A conservative interpretation of this result is that the demand for gravitational stability merely imposes a practical limitation for the information content of a spatial region. If we are willing to pay the price of gravitational collapse, we can excite more than $A/4$ degrees of freedom-though we will have to jump into a black hole to verify that we have succeeded. With this interpretation, all the degrees of freedom of field theory should be retained. The region will be described by a quantum Hilbert space of dimension e^V.

L'argument majeur contre ce point de vue serait la violation de l'unitarité. L'évolution dynamique de la mécanique quantique conserve l'information : un état pur est envoyé unitairement sur un état pur. Considérons une région décrite par un espace de Hilbert de dimension e^V , qui s'effondre pour former un trou noir. À la fin de la formation du trou noir, la région est à

présent de dimension $e^{\frac{A}{4}}$. Le nombre de degrés de liberté a drastiquement chuté, rendant impossible la récupération de l'état initial à partir de l'état final. Si l'on veut préserver l'unitarité, il est donc nécessaire de supposer que l'espace de Hilbert initial était dès le départ de dimension $e^{\frac{A}{4}}$. C'est ce raisonnement qui a poussé G. t'Hooft et P. Susskin a adopté les premières présentations du principe holographique : *toute région dont la frontière est d'aire A peut être décrite par au plus A/4 degrés de liberté, ou* 1 *bit par aire de Planck.*

Le maintien de l'unitarité pour la physique des trous noirs est cependant loin de faire l'unanimité. La question se rattache à certains des problèmes majeurs de la physique contemporaine des trous noirs, comme la complémentarité des trous noirs (pour plus de détails, voir ([47], 14-15). En outre, les arguments justifiant la borne ne sont pas valides dans des espaces-temps généraux. En particulier, la borne d'entropie sphérique est dérivée sous des hypothèses fortes : structure asymptotique, stabilité gravitationnelle, et symétrie sphérique. R. Bousso a lui-même travaillé à une formulation covariante de la borne en entropie, et montré que cette formulation n'admet pas de contre-exemple évident. S'il est empiriquement défendable, le principe holographique n'a à l'heure actuelle pas pu être dérivé de principes plus fondamentaux.

Il est difficile d'isoler un ensemble de raisons théoriques qui soutiendraient ce principe. Néanmoins, d'un point de vue informel, tous les raisonnements menant à une borne sur l'information encodée dans un système s'appuie sur deux arguments essentiels : la préservation du Second Principe de la Thermodynamique pour un observateur extérieur à un trou noir, et la prise des comptes des effets induits par l'instabilité gravitationnelle. S'il est possible dans l'abstrait d'analyser le principe holographique en un principe de densité finie de l'information et un principe proprement holographique, il n'est guère possible de séparer dans sa justification physique une partie qui soutiendrait la première composante et une partie qui soutiendrait la seconde. C'est en effet pour préserver le Second Principe que J.D. Bekenstein a introduit l'idée que l'entropie d'un trou noir était proportionnelle à son aire : celle-ci était en effet la seule quantité accessible à l'observateur extérieur au comportement analogue à l'entropie.

Un tel principe, s'il devait être incorporé dans la future théorie de la gravitation quantique, pèserait fortement sur l'ensemble des choix théoriques possibles. Toute théorie basée sur une généralisation du formalisme infinitaire de la théorie quantique des champs serait d'emblée exclue, comme le dit J.D. Bekenstein ([25], 65) :

> [...] although the holographic way of thinking is not yet fully understood, it seems to be here to stay. And with it comes a realization that the fundamental belief, prevalent for 50 years, that field theory is the ultimate language of physics must give

> way. Fields, such as the electromagnetic field, vary continuously from point to point, and they thereby describe an infinity of degrees of freedom. Superstring theory also embraces an infinite number of degrees of freedom. Holography restricts the number of degrees of freedom that can be present inside a bounding surface to a finite number ; field theory with its infinity cannot be the final story.

Ce n'est pas tant le caractère continu de la théorie quantique des champs qui est ici problématique, que son caractère infinitaire : le principe holographique est tout aussi incompatible avec une théorie utilisant un ensemble infini dénombrable de degrés de liberté. Tout formalisme faisant emploi d'un nombre infini de degrés de liberté doit donc être compris comme une idéalisation, qui ne représente pas une véritable information encodable dans l'état du système. Le principe holographique permet donc de fonder en principe l'impossibilité d'encoder une infinité d'information dans des valeurs d'observable, et à ce titre prohibe tout modèle d'hypercalcul basé sur une telle stratégie. Le principe holographique permet donc de prohiber les modèles nécessitant une précision arbitraire de mesure d'une observable pour parvenir à la lecture du résultat, comme les modèles analogiques. Il permet également de prohiber les modèles nécessitant la préparation idéale d'un système en dimension infinie, comme le modèle de Kieu.

Le principe holographique suffit-il cependant à prohiber tout modèle d'hypercalcul ? La réponse a cette question cruciale se doit d'être nuancée par deux restrictions. Tout d'abord, comme nous l'avons vu au début de ce chapitre, la question n'est bien posée que pour les modèles physiquement définis. Nous avons notamment, durant notre commentaire du travail de T. Ord et T. Kieu, rencontré un modèle notionnel encodant l'information non-calculable non dans des valeurs d'observables, mais dans les probabilités de mesurer ces valeurs. Une borne sur l'entropie statistique du système ne s'applique nullement à une telle stratégie d'implémentation de l'hypercalcul. Mais en l'absence d'une bonne définition physique de ce modèle, nous pouvons réitérer la conclusion sceptique que nous avions articulée au sujet de l'observable de Nielsen : en l'absence de toute procédure opérationnelle pour préparer un système dans un tel état, l'absence de contrainte théorique doit par défaut être interprétée comme une faiblesse de la théorie, et non comme une autorisation de principe accordée à une telle stratégie.

Le principe holographique suffit-il donc à exclure tous les modèles d'hypercalcul physiquement définis ? La réponse doit être positive, à une exception près : le modèle de Malament-Hogarth. Celui-ci fait bien usage d'une mesure de précision arbitraire, mais il s'agit d'une précision arbitraire dans le temps de détection d'un signal. Une telle exigence en précision n'est nullement contrainte par une borne sur l'entropie statistique d'un système quantique. Le temps n'étant pas une observable, mais un paramètre en mécanique

quantique, une borne sur les degrés de liberté du système ne limite aucunement la précision atteignable sur ce paramètre mesurable. Un tel constat ne constitue aucunement une affirmation de la possibilité de principe d'implémenter un hypercalcul à l'aide du modèle de Malament-Hogarth. Tout d'abord, nous avons vu que ce modèle affrontait déjà une liste conséquente de difficultés théoriques, qu'il lui faudra surmonter avant même d'affronter le problème de son exigence de précision arbitraire en temps. Ensuite, et surtout, l'absence de contrainte théorique ne doit pas non plus dans ce cas être comprise comme une autorisation de principe. Une théorie ne peut donner de validation de principe à un modèle que dans le domaine où elle est supposée être valide. Sans prendre position dans des débats en physique théorique qui dépassent de loin le cadre de ce travail, il est sain d'assumer qu'aucune théorie actuellement à notre disposition ne peut prétendre décrire la détection d'un signal en un temps propre inférieur au temps de Planck. On retrouve ici l'interprétation de la condition de validité théorique arbitraire en fonction des limites des théories fondamentales courantes. L'absence d'interdiction de principe d'une telle précision de mesure dans nos théories fondamentales courantes ne doit donc pas être vue comme la validation d'une possibilité de principe, puisque seule une théorie de la gravitation quantique en devenir sera habilitée à fournir une telle validation théorique.

6.6 L'approche à la Gandy : tentative de caractérisation physique de la calculabilité

Nous allons à présent examiner une approche visant à caractériser la thèse de Church-Turing physique par un ensemble d'hypothèses empiriques nécessaires et suffisantes. Cette approche fut initiée par R. Gandy [108], reprise et simplifiée par W. Sieg [207] [208], [206], [205], et étendue au cas quantique par P. Arrighi et G. Dowek [18]. Nous la désignerons globalement sous le nom d'« approche de Gandy », même si nous verrons que cette approche n'est pas dépourvue de diversité interne, chaque auteur présentant des résultats techniques et une vision sensiblement différents.

A première vue, cet ensemble de travaux devrait permettre de déterminer avec précision la signification empirique de la thèse de Church-Turing physique, et les contraintes qu'elle ferait peser sur la théorie physique. Si l'on admet les hypothèses employées dans la caractérisation, on pourra la considérer comme une démonstration de la thèse de Church-Turing physique, au sens où l'on emploie ce terme en physique théorique : une dérivation mathématique à partir de postulats empiriques mathématisés. Mais même si l'on ne partage pas ce point de vue, et qu'on considère au contraire que cette thèse est fausse, la caractérisation conserverait un grand intérêt, parce qu'elle permettrait de détailler le coût théorique à payer pour violer la thèse de Church-Turing physique.

Le principal enjeu de notre discussion sera de savoir si on peut accepter l'approche de Gandy comme une authentique caractérisation physique. Comme nous le verrons, certains auteurs rejettent toute pertinence de cette approche pour la physique, et la considère comme un travail purement informatique. C'est donc sur le sens exact de l'approche de Gandy, plus que sur ses détails techniques, que nous allons nous pencher dans cette section. Nous allons donc tout d'abord présenter brièvement l'approche de Gandy et ses variantes, avant de discuter ses différentes critiques.

6.6.1 L'approche de Gandy : schème général et variantes

Dans [108], R. Gandy commence par opérer une distinction entre deux interprétations à la thèse de Church-Turing. La première est le théorème T :

Theorem T. *What can be calculated by an abstract human being working in a routine way is computable*[35].

la seconde est la thèse M :

Thesis M. *What can be calculated by a machine is computable.*

R. Gandy affirme également que l'argument de modélisation de Turing s'applique à un calculateur humain, et ne justifie pas à une extension de la thèse aux machines :

> Although some of Turing's arguments can be applied indifferently to men or machines, there are crucial steps in Turing's analysis where he appeals to the fact that the calculation is being carried out by a human being. One such appel is used to justify the assumption that the calculation proceeds as a sequence of elementary steps. A human being can only write one symbol at a time. But, if we abstract from practical limitations, we can conceive of a machine which prints an arbitrary number of symbols simultaneously. [...] Turing's arguments do not suffice, nor do I think he would have claimed that they suffice, to justify thesis M[36].

Au premier abord, la distinction opérée par R. Gandy, et les motivations qui la fondent, sont très similaires à celles justifiant notre distinction entre thèse de Church-Turing algorithmique et thèse de Church-Turing empirique. Cependant, R. Gandy restreint par la suite l'interprétation de la thèse M, pour la circonscrire aux seules machines discrètes et déterministes. La thèse M ne saurait donc être confondue avec la thèse de Church-Turing empirique : on peut au mieux la considérer comme une sous-forme de cette dernière.

35. *Abstract* signifie ici l'ignorance des contraintes pratiques de ressources.
36. [108], 124-125.

La deuxième particularité de l'article original de Gandy est l'insistance sur le parallélisme. Le parallélisme est présenté comme la propriété essentielle séparant le théorème T de la thèse M. Ceci peut sembler étrange, puisqu'une machine parallèle peut parfaitement être simulée par un modèle séquentiel, et que le parallélisme, s'il est usuellement vu comme une source d'efficacité, n'est pas conçu comme une source de pouvoir expressif supplémentaire : ce que N processeurs font en une étape peut être simulé par un processeur en un nombre d'étapes de l'ordre de N. En outre, si l'être humain armé de papier et d'un crayon calcule bien de manière séquentielle, cela n'est pas une propriété spécifique d'*Homo sapiens*, puisque nombre de machines sont également séquentielles.

Comme nous l'avons vu lors de notre étude de l'argument de modélisation (voir sections 1.3.3 et 2.2.1), les contraintes énoncées par Turing, même si elles sont initialement justifiées en termes de limitations sensorielles et cognitives de l'être humain, ne sont pas spécifiques au sujet humain. Il est naturel d'étudier leur possible généralisation en termes physiques, et c'est ce qu'envisage de faire R. Gandy :

> Turing's requirement that a computation step should depend only on a bounded portion of the record was motivated by the evident limitations of the human sensory apparatus ; as we mentioned already, memory limitations were viewed as the ultimate reason. This motivation is replaced here by an appeal to physical limitations : signals can only be conveyed with finite velocity ; the possibility of instantaneous action at a distance is rejected by contemporary physics.

R. Gandy énonce et justifie ensuite quatre principes, censés valoir pour toute machine discrète et déterministe. Les deux premiers principes n'ont rien d'empirique, et concernent la forme de la description théorique du modèle. Chaque état doit permettre une description finie. Cette description doit refléter la structure concrète de la machine, tout en étant suffisamment abstraite pour s'appliquer à différentes formes de machines [37].

R. Gandy choisit de représenter les états computationnels par des ensembles héréditairement finis [38] construits à partir d'un ensemble U d'atomes,

37.

> Our use of the term "discrete" presupposes that each state of the machine can be adequately described in finite terms. In order that we can apply any insights which we may have about mechanisms we want this description to reflect the actual, concrete, structure of the device in a given state. On the other hand, we want the form of description to be sufficiently abstract to apply uniformly to mechanical, electrical or merely notional devices.

Ibid., p.127.

38. La classe des ensembles héréditairement finis est la plus petite classe comprenant l'ensemble vide et close par passage à l'ensemble des parties. Elle capture l'intuition d'un ensemble fini dont les éléments sont eux mêmes des ensembles finis, et ainsi de suite.

qui représente intuitivement les composants élémentaires de la machine. Ce choix n'est pas motivé par des considérations physiques, et doit juste être considéré comme un choix possible parmi d'autres [39]. Les états computationnels doivent être compris comme des structures. L'unité d'analyse n'est donc pas un ensemble héréditairement fini donné x, mais l'ensemble des ensembles $\in$-isomorphes à x, appelé stéréotype de x. Lorsqu'on spécifie l'ensemble des états possibles d'un modèle de machine, on ne se référera donc pas à un simple ensemble mais à un ensemble structurel, c'est-à-dire clos par isomorphisme.

Cette vision structurelle des états computationnels est cependant tempérée par l'ambition de modéliser des machines concrètes. Les indices référant à une partie concrète de la machine, comme par exemple une adresse mémoire, doivent rester identiques tout au long de l'évolution du calcul [40].

On décide donc de considérer des isomorphismes qui soient l'identité sur un certain ensemble d'indices. On définit donc la notion de structures isomorphes sur un ensemble d'indices pour ce faire.

Lorsqu'on définit la fonction de transition, on impose qu'elle soit structurelle, c'est-à-dire qu'elle commute à l'isomorphisme restreint à l'identité sur les atomes. On ne requiert pas de commutation à l'isomorphisme [41], parce qu'on veut maintenir l'identité de certains éléments, et on ne demande pas non plus la pure et simple invariance par permutation, qui serait trop forte [42]. On aboutit finalement au premier principe suivant :

Principe I ***Forme de la description.*** Toute machine discrète déterministe M peut être décrite par la paire $\langle S_m, F\rangle$, avec S_m un ensemble struc-

---

39.

> We have chosen hereditarily finite sets; other forms of description might be equally acceptable.

Idem.

40.

> (...) it is natural to suppose that if a label refers to a particular element of a mechanism in the state described by x, then it will refer to the same element in the next state; most things preserve their identity as time passes. In general the next state will also contain some new elements; for example, when a Turing machine moves left from the leftmost square so far used a new square must be created. A transition function F which determines the description Fx of the next state must specify new labels for the new elements, but no physical significance attaches to that specification.

Ibid., p.128.

41. On remarquera la similitude et la différence de l'approche de Gandy avec celle de N. Dershowitz et Y. Gurevich dans [77], qui impose la commutation à l'isomorphisme des états computationnels.

42.

> It expresses precisely the requirement that F describes the transition between physical states with some persistent elements.

turel de descriptions d'états, et F une fonction structurelle de S dans S. Si $x \in S_m$ est l'état initial, alors $F(x)$, $F(F(x))$, ... décrivent les états suivants de M.

R. Gandy remarque ensuite que ce premier principe est trop permissif, et permettrait le calcul de tout prédicat arithmétique. Le deuxième principe va donc viser à restreindre la classe des machines de Gandy par une considération inspirée des machines réelles. R. Gandy part de l'intuition qu'une machine peut être conçue en termes hiérarchiques. Une unité arithmétique, ou une série de processeurs mis en parallèles sont soumises à une unité de contrôle ; certains types de données sont traitées comme des listes de listes [43]. Il est également naturel de penser que cette hiérarchie doit avoir un terme. En formalisant cette intuition en termes ensemblistes, on considère qu'un ensemble d'un rang donné a ses éléments comme subordonnés. Le deuxième principe limite donc le rang des états d'une machine donnée :

Principe II ***Limitation de la Hiérarchie.*** Le rang ensembliste des états est borné : il existe un $k \in \mathbb{N}$ tel que $S \subseteq HF^k$.

Par le troisième principe, on tâche de capturer l'intuition que toute machine peut être assemblée à partir de parties de taille bornée, et que ces parties sont ainsi indicées qu'il n'existe qu'une unique manière de les assembler :

> (...) any device can be assembled from parts of bounded size, and (...) these parts can be so labelled that there is a unique way to of putting them together. Model-construction kits aim, not always successfully, to satisfy this principle.

L'une des principales difficultés dans la formulation de ce principe est la nécessité de prendre en compte les possibles chevauchements entre différentes parties. La bande d'une machine de Turing peut être réassemblée à partir de l'ensemble de toutes les paires de cases consécutives avec le symbole écrit sur elles. Mais cet ensemble contiendrait de très nombreux recoupements (autant que d'éléments), qu'il faut pouvoir gérer de façon systématique [44].

La formulation précise de cette idée en termes d'ensembles héréditairement finis est également difficile. Que va-t-on considérer comme les parties de notre structure ? Intuitivement, on considère des listes de taille finie de parties finies, à partir desquels on pourra assembler de manière unique un état computationnel. On arrive ainsi au principe suivant :

Principe III ***Principe d'Unique Réassemblage.*** il existe une borne q et pour chaque ensemble $x \in S$ un ensemble Q inclus dans les parties finies

43. *Ibid.*, p.131.

44. La gestion des indices des différentes parties de la machine est l'une des grandes sources de difficulté technique dans le papier original de Gandy, qui est grandement simplifié dans la reformulation de W. Sieg [208], [205].

de $TC(x)$ à partir duquel x peut être réassemblé de manière unique tel que la taille de s est bornée par q pour tout $s \in Q$.

L'intervention de considérations empiriques dans la justification des principes n'intervient explicitement que pour le principe IV, que R. Gandy appelle « principe de causalité locale »(*principle of local causation*). Ce principe est présenté comme une généralisation de l'argument de Turing, selon lequel l'action du calculateur ne pouvait être fonction que d'une partie bornée de la mémoire. Cet argument, initialement justifié par les limites perceptives humaines, est généralisé sur des bases physiques. Gandy part en effet des deux hypothèses suivantes ([108], 126) :

> (...) there is a lower bound on the linear dimensions of every atomic part of the device and that there is an upper bound (the velocity of light) on the speed of propagation of changes.

La borne sur la vitesse de propagation de l'information implique que, dans un temps borné, chaque atome de la machine ne peut transmettre et recevoir de l'information qu'au sein d'un voisinage borné. Une transition entre deux états étant censée prendre un temps fini, la dynamique de notre machine va être déterminée par cette hypothèse. En outre, puisqu'il existe une borne inférieure sur la taille des atomes, le nombre d'atomes contenus dans ce voisinage borné est fini. Il s'ensuit que $F(x)$ est constitué de parties de x bornées, quoique pouvant se chevaucher.

La formulation précise de ce principe pose de nombreuses difficultés techniques sans pertinence pour notre travail. Nous nous contenterons donc de présenter le principe sous la forme semi-intuitive que lui donne initialement R. Gandy :

Principe IV Causalité locale. Les parties à partie desquelles $F(x)$ peut être réassemblée dépendent uniquement de parties bornées de x. L'état successeur, $F(x)$, d'une machine peut être réassemblé à partir de restrictions à des régions s et ces restrictions sont localement causées. Pour chaque région s de $F(x)$ il existe un voisinage causal $t \subsetneq Tc(x)$ de taille bornée tel que $F(x) \restriction s$ dépende uniquement de $x \restriction t$.

L'argument de Gandy peut ensuite être résumé de la manière suivante : tout machine discrète déterministe obéit aux quatre principes ci-dessus. Il démontre ensuite que toute machine obéissant à ces principes ne peut calculer que des fonctions récursives, donc toute machine discrète déterministe calcule des fonctions récursives.

Le travail de P. Arrighi et G. Dowek constitue à la fois une variante technique et une réinterprétation de l'approche de Gandy. Il propose en effet à la fois une réécriture des hypothèses de Gandy, et une extension de son approche à une version quantique de ces hypothèses. Cette extension est également inspirée par le problème soulevé par M. Nielsen, sur la manière

de modifier la théorie quantique pour en exclure toute possibilité d'y formuler un modèle d'hypercalcul. On considère tout d'abord un espace euclidien tridimensionnel, et on formule les hypothèses suivantes :

- *Homogénéité de l'espace.* Si τ est une translation, alors la région τA a le même ensemble d'états que A. La fonction associant l'état global du système au temps t à son état au temps $t+T$ commute à toutes les translations.
- *Homogénéité du temps.* La fonction associant à l'état global du système au temps t l'état global du système au temps $t+T$ est indépendante de t.
- *Densité bornée d'information.* Si A est une région finie de l'espace, l'espace des états $\Sigma(A)$ est un ensemble fini.
- *Vitesse bornée de propagation de l'information.* Il existe une constante T telle que pour toute région A et tout instant t, l'état de A au temps $t+T$, $\rho(A, t+T)$ dépend uniquement de $\rho(A', t)$, avec A' la région de rayon 1 autour de A.
- *Quiescence.* Pour chaque A, il existe un état canonique q_A nommé *état quiescent.* Si une région A est dans l'état quiescent, alors toute région B incluse dans A est dans l'état quiescent. À l'origine, tout l'espace à l'exception d'une région finie est dans l'état quiescent, et l'évolution globale préserve ce fait.

On fixe un système de coordonnées O, $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$, et on pave l'espace de cellules cubiques de la forme $[x, x+1) \times [y, y+1) \times [z, z+1)$ avec $x, y, z \in \mathbb{Z}$. On fixe également un ensemble de translations τ décrites par des vecteurs de la forme $x\mathbf{i}+y\mathbf{j}+z\mathbf{k}$ avec $x, y, z \in \mathbb{Z}$. Chaque cellule et chaque translation peut donc être désignée par un triplet d'entiers $\langle x, y, z \rangle$.

On peut ainsi définir les fonctions calculées par un système emplissant tout l'espace. L'état global du système peut être indicé à partir des indices des cellules, en nombre fini, qui ne sont pas dans l'état quiescent. La fonction calculée par le système est donc la fonction associant l'indice de l'état global au temps t_0 à l'indice de l'état global au temps t.

Comme on peut le voir immédiatement, les hypothèses dites de Gandy par P. Arrighi et G. Dowek, bien que manifestement portées par des intuitions similaires, diffèrent sensiblement de la version originale de ces hypothèses. P. Arrighi et G. Dowek abandonnent le format des ensembles héréditairement finis, qui était certes présenté comme un choix contingent par R. Gandy. Ils considèrent également un système de taille infinie, emplissant tout l'espace. Ceci mène à deux différences sensibles par rapport à l'approche de Gandy. La première est l'hypothèse de quiescence, dont l'utilité est simplement d'exclure que l'état initial du système encode une entrée non-calculable, en imposant qu'il ne puisse encoder d'entrée de taille infinie. La seconde est l'ajout des hypothèses d'homogénéité du temps et de l'espace. Celles-ci auraient été sans objet dans la formulation initiale de Gandy, qui ignorait temps et espace réels.

La filiation entre le travail de Gandy et celui de P. Arrighi et G. Dowek peut à son tour être résumée en trois points. Le premier est la restriction de l'attention aux systèmes déterministes à temps et à espace discrets. La seconde est l'analyse d'un modèle de calcul parallèle. La troisième, et la plus notable, est l'analyse du principe IV en deux hypothèses distinctes, la vitesse bornée de propagation de l'information et la densité bornée de l'information. L'idée d'une densité bornée de l'information peut être vue comme un éclaircissement des intuitions initiales de Gandy, puisque celui-ci parlait plutôt de taille minimale des composants de la machine. Ce qui compte véritablement pour un modèle de calcul n'est pas tant la taille minimale du composant, que sa capacité à charrier de l'information. La conjonction des deux hypothèses produit un résultat identique au principe IV : l'état de chaque cellule ne dépend que d'un voisinage causal de taille finie, qui ne peut contenir qu'une quantité finie d'information. La fonction associant l'indice de cette cellule au moment t à son indice au moment $t+1$ sera donc calculable, parce que de domaine fini.

L'originalité du travail de P. Arrighi et G. Dowek est d'étendre l'approche de Gandy au cas quantique, en considérant un pavage de l'espace par un automate cellulaire quantique [45]. L'approche de Gandy est alors étendue au-delà de ses ambitions premières, puisqu'un automate cellulaire quantique est une machine discrète probabiliste, et non déterministe. Mais l'intérêt du travail est précisément de généraliser les intuitions de Gandy sur la densité et la vitesse bornée de l'information au cas quantique, en surmontant plusieurs difficultés de reformulation, dues notamment à la prise en compte de l'intrication (voir [18], 1136-1139).

6.6.2 Critiques de l'approche

L'approche de Gandy a été la cible de nombreuses critiques, qu'on peut classer en deux catégories. Le premier groupe de critiques va jusqu'à nier toute pertinence empirique à cette approche : celle-ci ne saurait être présentée comme une caractérisation de la thèse de Church-Turing physique. Un second groupe, plus modéré, argumente plutôt que l'approche de Gandy, sans permettre de prouver un résultat physique fondamental, a néanmoins une pertinence empirique.

Dans [77], N. Dershowitz et Y. Gurevich présentent l'article de R. Gandy comme une caractérisation de la thèse algorithmique, sans commenter plus en détails l'apparente différence d'intentions entre leur travail et celui de Gandy [46].

45. On pourra aussi remarquer que, tout comme le travail de W. Sieg, le travail de P. Arrighi et G. Dowek dans le cas classique est considérablement plus simple que l'article original de R. Gandy.

46. Ils peuvent ainsi écrire ([77], 340, nous soulignons) :

> Unlike all previous formalizations of effectiveness, the postulates proposed

Dans ([125], 8), Y. Gurevich nie explicitement le caractère empirique de l'approche de R. Gandy, et la réinterprète comme une tentative de caractérisation des algorithmes parallèles :

> It seems obvious that Gandy abstracts from material and views discrete deterministic machines as algorithms, abstract algorithms. So Gandy's claim can be restated thus : parallel algorithms satisfy the axioms.

Y. Gurevich juge ensuite sévèrement les axiomes de Gandy dans cette perspective [47](voir aussi [126], 8-9).

Enfin, dans [190], G. Piccinini affirme que les principes de Gandy ne sont que partiellement justifiés empiriquement, et que cela réduit fortement leur intérêt pour la discussion de la thèse de Church-Turing physique :

> Some of [Gandy's and Sieg's] assumptions-such as a lower bound on the size of atomic components and an upper bound on signal propagation-are empirically well motivated. But other assumptions-such as discreteness of states and discreteness of dynamics-are not empirically justified. Thus, there remains the empirical question of whether there are physical systems that violate Gandy and Sieg's assumptions and yet are computationally more powerful than Turing machines. This is the real question of interest here, and it is not settled by Gandy and Sieg's work. This is probably why Gandy and Sieg's work has had relatively little impact on discussions of physical computability.

La critique de G.Piccinini n'est pas bien formulée en l'état. La discrétisation des états et de la dynamique, comme nous l'avons vu plus haut, n'a pas à être justifiée empiriquement : elle est simplement le fruit d'une restriction explicite et pleinement assumée de la portée du travail. La position de G. Piccinini, si on cherche à la réinterpréter avec générosité, serait donc que de telles restrictions empêchent l'approche de Gandy d'avoir un intérêt véritable dans une discussion générale de la thèse de Church-Turing physique.

> here apply to transition systems with arbitrary structures as states. Our abstract states can hold one-dimensional tapes (as in Turing's original work [107]), two-dimensional arrays (as extended in [93]), bounded-degree graphs (as in Kolmogorov machines [56, 57]), or *bounded rank hereditarily finite sets (as in Gandy machines* [35, 102]).

47. Cette réinterprétation avait déjà été défendue par J. Byrnes et W. Sieg, quoique de manière plus empathique, dans ([208], 151) :

> It seems to us that Gandy's analysis is conceptually convincing and provides a sharp mathematical form of the informal assumption(s) "underlying almost all models of concurrent systems."Indeed, the central informal idea is just this : operations on states are composed of local operations that modify 'parts' of bounded size, and the resulting configurations are then uniquely assembled into the next state of the system.

Nous reviendrons sur ce point dans notre discussion du deuxième groupe de critiques.

Comme on peut le voir à la lecture des passages cités, l'indécision autour du statut épistémologique de l'approche de Gandy est telle, que les critiques demeurent à l'état inchoatif : ceux qui s'opposent à cette approche tiennent pour évident qu'elle ne saurait constituer une caractérisation de la thèse physique. Sans s'aventurer dans l'herméneutique des positions de chaque auteur, on peut subodorer que cette hostilité spontanée est due en large partie au caractère notionnel du modèle : ni l'état, ni l'évolution de la machine ne sont décrits dans les termes d'une théorie physique. Même lorsqu'une hypothèse est manifestement inspirée par une théorie physique, le modèle de Gandy n'adopte pas le formalisme de cette théorie. Ainsi l'idée d'une vitesse bornée de propagation de l'information est explicitement inspirée par la relativité restreinte, mais il n'est fait aucune mention dans le principe IV d'espace de Minkovski ou de cône de lumière. Les hypothèses de Gandy, bien qu'étant d'inspiration physique, ne sont pas physiquement définies.

Cela ne justifie cependant pas un rejet total de la portée empirique du modèle. Une hypothèse n'a pas besoin d'être physiquement définie pour être de nature empirique. Les hypothèses de densité finie et de vitesse bornée de propagation de l'information sont indiscutablement empiriques, et elles jouent un rôle actif dans la détermination du pouvoir expressif du modèle. L'approche de Gandy ne mérite donc ni l'excès d'honneur de se voir considérée comme une caractérisation de la thèse physique - fût-elle restreinte au cas discret et déterministe - ni l'excès d'indignité de se voir privée de toute pertinence empirique. Nous allons voir à la fois les limites et les mérites heuristiques de cette approche en examinant le deuxième type de critiques.

Dans ([70], 14-19), J. Copeland et O. Shagrir argumentent que le modèle de Malament-Hogarth est un modèle discret et déterministe qui permet néanmoins de réaliser un hypercalcul. Lesquels des principes de Gandy seraient donc violés par ce modèle ?

Pour répondre à cette dernière question, les deux auteurs distinguent entre déterminisme physique et déterminisme informatique[48]. Dans ([108], 126), R. Gandy emploie tout d'abord ce terme en son sens physique : l'évolution de l'état du système en fonction du temps est univoquement déterminé une fois donnée une description complète de l'état initial. En ce sens, le modèle de Malament-Hogarth est déterministe. En revanche, lorsqu'il formule le principe I, R. Gandy emploie le terme dans son sens proprement informatique. Celui-ci est strictement plus fort que le sens physique, puisqu'il suppose une discrétisation des états du système : l'état n de la machine est fonction de l'état $n-1$. En ce deuxième sens, le modèle de Malament-Hogarth n'est pas déterministe. Notons T_P la machine placée dans le référentiel du

48. Dans ([202], 98-99), I. Pitowsky et O. Shagrir ont formulé une critique des principes de Gandy fondée sur une confusion de ces deux notions.

programmeur, et T_C la machine placée dans le référentiel du calculateur. Dans le cas où T_P ne reçoit pas de signal de T_C et s'arrête, son état d'arrêt devrait, pour que le modèle soit déterministe au sens informatique, dépendre du dernier état de T_P. Mais T_P étant par hypothèse une machine de Turing qui ne s'arrête jamais, il n'existe pas de tel dernier état. L'état de T_P est cependant bien déterministe au sens physique, puisqu'il dépend de la limite des états précédents de T_P et T_C.

Quelle conclusion tirée de ce résultat ? Un partisan de l'approche de Gandy pourrait y voir une illustration de la nécessité de ses principes. On a en effet là un exemple de modèle de calcul permettant de violer la thèse de Church-Turing physique, et qui ne respecte pas au moins un des principes de Gandy. J. Copeland et O. Shagrir choisissent au contraire de l'interpréter comme un signe de l'inadéquation de l'approche de Gandy à la discussion de la thèse physique. Le modèle de Malament-Hogarth est en effet un modèle déterministe au sens physique, et un modèle discret, si on le considère composé fondamentalement de deux machines de Turing en communication. En outre, le modèle respecte l'esprit du Principe IV, sinon sa lettre. Il ne nécessite pas de composants élémentaires arbitrairement petits ou de violation de l'hypothèse de densité finie. En tant que modèle relativiste, il ne permet pas une communication d'information à une vitesse arbitraire. Ce ne sont pas les hypothèses empiriques justifiant les principes de Gandy qui sont violées, mais la forme de description particulière adoptée par Gandy, d'inspiration résolument informatique. Le modèle de Malament-Hogarth viole donc la lettre des principes de Gandy tout en en respectant l'esprit. Loin de confirmer la nécessité de ces principes, il en montre l'incapacité à saisir les enjeux empiriques pour lesquels ils ont été conçus (*ibid.*, 18-19) :

> The availability of robust examples of (ideal) physical machinery that fails to conform to Gandy's definition is a telling indication that the logico-mathematical notion is just *different* from the notion of 'mechanical' present in physical (and psycho-physical) discussion.

On pourrait reprocher à J. Copeland et O. Shagrir d'aller quelque peu vite en besogne, en affirmant que le modèle de Malament-Hogarth est discret. Comme nous l'avons déjà souligné (voir section 6.3), la propriété cruciale du modèle est la distinction entre le temps de calcul discret et le temps d'accès épistémique au résultat du calcul. Pour pouvoir décrire le modèle, il faut non seulement pouvoir décrire deux machines de Turing discrètes et la communication discrète entre elles, mais aussi préciser le référentiel dans lequel elles sont placées et l'écoulement du temps réel dans ces référentiels, ces deux derniers traits nécessitant le formalisme continu de la relativité générale. En outre, on pourrait également reprocher à J. Copeland et O. Shagrir d'ignorer les problèmes soulevés par la nécessité d'une mesure de précision arbitraire en temps dans le modèle de Malament-Hogarth (voir

section 6.3.2).

Mais ces dernières remarques renforcent la conclusion de J. Copeland et O. Shagrir, plutôt qu'elles ne la minent. Le modèle de Gandy ne permet pas de poser la question de l'exécution d'un nombre infini d'étapes de calcul avec un temps d'accès épistémique réel fini, parce que son formalisme ne permet pas d'exprimer une telle distinction. Le problème n'est donc pas tant que le modèle de Malament-Hogarth violerait les principes de Gandy, que les principes de Gandy ne permettent pas d'exprimer un jugement sur le modèle de Malament-Hogarth. La même remarque vaut évidemment pour les problèmes de précision de mesure, qui ne peuvent être posés dans le formalisme de Gandy.

Le sort du modèle de P. Arrighi et G. Dowek doit ici être distinct. L'espace réel y est mentionné, puisque ce modèle est justement fondé sur l'hypothèse d'un pavage de l'espace réel par un automate cellulaire, et le temps réel y est au moins mentionné dans l'hypothèse d'invariance. Mais le modèle utilisé de l'espace réel est euclidien. Or la géométrie de l'espace de Malament-Hogarth n'est pas euclidienne, et la relativité générale, dans le cas d'une forte interaction gravitationnelle, peut remettre en cause la possibilité même de définir un pavage de l'espace, comme celui employé par P. Arrighi et G. Dowek.

Malgré ses limitations intrinsèques, il serait injuste de dénier toute pertinence pour la thèse de Church-Turing physique à l'approche de Gandy. La généralisation des principes de Gandy au cas quantique montre à tout le moins la valeur heuristique de l'approche, et ce d'autant plus qu'elle impose de démontrer plusieurs propriétés non-triviales. P. Arrighi et G. Dowek ne disent pas autre chose ([18], 1144) :

> Because these hypotheses are physically motivated, this is a step along Nielsen's programme of a computable Quantum theory. Further work could be done along this direction by introducing a notion of 'reasonable measurement' [27], or investigating the continuous-time picture as started by [31, 32]. Prior to that however this work raises deeper questions : Is the bounded density of information really compatible with modern physics ? For instance, can we really divide up space into pieces under this condition, without then breaking further symmetries such as isotropy ?

Au vu de la difficulté qu'il y a à formuler une démonstration générale de la thèse de Church-Turing physique, l'intérêt d'une telle caractérisation, fût-il heuristique, ne saurait donc être négligé. À notre connaissance, le travail de P. Arrighi et G. Dowek est la première mention d'une borne sur la vitesse de propagation de l'information comme hypothèse pertinente pour la calculabilité, avant même le travail de I. Németi et G. Székely. La démonstration de la nécessité de l'hypothèse est certes sujette à caution. On a vu dans le modèle d'I. Németi et G. Székely que les signaux supralumineux ne

permettaient l'implémentation d'un modèle d'hypercalcul qu'au prix d'une exigence de précision arbitraire. Le caractère notionnel du modèle d'Arrighi-Dowek permet donc une autonomisation des hypothèses qui n'existe pas dans le modèle physiquement défini. Ceci illustre à la fois le mérite heuristique de l'approche, tout en soulignant la prudence de rigueur pour toute extrapolation de résultats obtenus avec des modèles notionnels.

Conclusion

Une revue des modèles d'hypercalcul embrasse l'ensemble des tentatives existantes pour concevoir un modèle d'hypercalcul : elle ne peut prétendre embrasser toutes les tentatives possibles. Dans l'état actuel de formalisation de la thèse, il n'est pas possible de formuler une démonstration d'impossibilité de principe, qui montrerait que tout modèle d'hypercalcul est voué à l'échec. Notamment, il convient de rappeler que l'impossibilité d'un modèle d'hypercalcul à la fois non-relativiste et sans exigence de précision infinie n'a pas été démontrée ([188], 231-238). Il n'existe pas même à notre connaissance d'argument heuristique contre la possibilité d'un tel modèle. Cependant, l'ensemble des tentatives existantes exhibent déjà suffisamment de régularité pour qu'il soit possible de tirer des enseignements généraux, et de prendre position sur la plausibilité d'un modèle d'hypercalcul.

Dans l'état actuel de la recherche, tous les modèles physiquement définis, en sus d'affronter des difficultés propres à chacun, connaissent un problème commun, à savoir une exigence de précision arbitraire. D'un point de vue opérationnel, cette difficulté suffit à condamner toute tentative d'implémentation de ces modèles.

D'un point de vue théorique, nous venons de voir que le principe holographique suffisait à proscrire tout modèle d'hypercalcul basé sur l'encodage d'une information non-calculable dans des valeurs d'observable. Un tel résultat suffit à exclure la possibilité théorique de l'intégralité des modèles physiquement définis, à l'exception du modèle de Malament-Hogarth. Lorsque ce dernier exige une précision de mesure arbitraire, il s'agit d'une précision arbitraire sur le temps de détection d'un signal, qui n'est pas contrainte par le principe holographique. Mais outre que ce modèle affronte de nombreuses difficultés de principe qui lui sont propres, il serait erroné de conclure de l'absence d'interdiction de principe d'une telle précision de mesure dans nos théories actuelles à une possibilité de principe aux échelles concernées. Sur la base de nos connaissances théoriques actuelles, aucun modèle d'hypercalcul n'est implémentable, pas même en principe.

Il est donc justifié d'adopter une attitude sceptique à l'égard de l'implémentabilité d'un modèle d'hypercalcul. Si l'on ne peut à proprement parler montrer que la thèse de Church-Turing physique est vraie, on doit au moins conjecturer qu'elle l'est, et placer la charge de la preuve sur le partisan de la possibilité d'un modèle d'hypercalcul.

En l'absence d'une démonstration de la thèse de Church-Turing physique, on peut néanmoins s'interroger sur le contenu empirique de cette proposition. Quelles contraintes la thèse de Church-Turing fait-elle peser sur notre théorie physique, si on accepte qu'elle est valide ?

Pour répondre à cette question, il importe de se demander en premier lieu si la thèse de Church-Turing physique doit être comprise comme un nouveau principe théorique. Le terme de « principe »est employé de manière informelle en physique théorique, et il peut revêtir diverses significations. Pour le seul bien de ce travail, nous distinguerons un sens faible et un sens fort. On peut ainsi appeler « principe »au sens faible un résultat significatif qui peut se démontrer sur la base des postulats de la théorie, comme le *principe d'incertitude de Heisenberg* ou le *principe de superposition*. Par « principe »au sens fort, j'entendrai ici une proposition qui exerce une nouvelle contrainte sur notre théorie physique. Un principe ne doit donc pas être démontrable à partir des postulats de théories admises ou de programmes de recherche en cours. Il ne doit pas non plus faire double emploi avec un autre principe théorique. Par exemple, un principe au sens fort doit permettre d'exclure un modèle qu'on ne peut exclure sur la base d'un principe déjà établi.

Comme nous venons de le voir, le modèle de Malament-Hogarth rencontre de nombreuses difficultés, liées à des problèmes fondamentaux d'interprétation de la théorie de la relativité générale. Il n'est donc point besoin d'instaurer un principe de Church-Turing physique pour exclure un tel modèle. De la même manière, dans les autres modèles physiquement définis, il suffit d'invoquer l'exigence de précision arbitraire pour les écarter. Le principe holographique, s'il venait à être accepté, pourrait être la base théorique fondamentale du rejet de tout modèle exigeant une précision arbitraire, que ce soit dans la mesure des valeurs d'observable encodant le résultat, ou dans la préparation du système en dimension infinie. Il serait donc redondant de formuler un quelconque principe de Church-Turing pour exclure tout modèle de ce type. A ce titre, l'état présent de la recherche sur les modèles d'hypercalcul ne permet pas d'imposer l'adoption d'un nouveau principe comme la thèse de Church-Turing physique.

Il est possible d'utiliser la thèse de Church-Turing physique comme un principe heuristique, de la manière suivante. Un physicien théoricien peut considérer qu'un modèle théorique autorisant l'implémentation d'un modèle d'hypercalcul doit être irréaliste, et conjecturer qu'un tel modèle doit violer un principe établi. Mais un tel emploi de la thèse de Church-Turing physique n'en fait pas le principe théorique justifiant qu'on considère un modèle comme irréaliste, mais plutôt un signe heuristique, menant à la recherche des véritables causes physiques de ce caractère irréaliste. Parler d'un « principe de Church-Turing »pour de tels emplois heuristiques de la proposition risque donc d'induire en erreur sur sa véritable fonction théorique. Nous en avons vu un exemple lorsque nous avons exposé le dilemme de M. Nielsen : soit tout opérateur hermitien ne représente pas une observable mesurable, soit la

thèse de Church-Turing physique est fausse. Résoudre ce dilemme en faveur de la thèse de Church-Turing physique n'implique pas d'énoncer cette dernière comme un nouveau principe. Le principe holographique suffirait pour exclure les observables envisagées par M. Nielsen, puisqu'elles vivent dans un espace de dimension infinie.

Nos dernières considérations n'interdisent nullement à un physicien théoricien de travailler à la formulation d'un principe de Church-Turing physique, et d'en explorer les conséquences. Elles interdisent uniquement de motiver l'introduction d'un tel principe par la seule volonté d'exclure l'implémentation d'un modèle existant d'hypercalcul. Celle-ci étant déjà justifiée par d'autres considérations théoriques, l'introduction d'un nouveau principe serait à cet égard redondant. Mais rien n'interdit d'employer un tel principe à d'autres fins, et notre propos n'est pas de nous livrer à des spéculations sur l'avenir de la physique théorique.

Il nous faut enfin conclure sur l'interprétation empirique de la calculabilité. D'un point de vue intuitif, le débat sur les modèles d'hypercalcul exhibe amplement la dépendance de la possibilité d'un modèle d'hypercalcul aux hypothèses empiriques sous-jacentes à l'implémentabilité du modèle. Ce débat fournit donc bien l'argument positif en faveur de l'interprétation empirique de la calculabilité que nous attendions (voir chapitre 5, Conclusion). Les arguments échangés au cours du débat sur les modèles d'hypercalcul, s'ils permettent uniquement de conjecturer la valeur de vérité de la thèse physique, permettent de conclure avec certitude sur son statut épistémologique. Même si aucun modèle ne permet le calcul d'une fonction qu'un calcul papier-crayon ne pourrait calculer, les considérations justifiant le respect de cette limite sont indiscutablement empiriques.

On pourrait argumenter qu'un tel résultat ne valait pas tant de peine, et que l'interprétation empirique était en quelque sorte triviale, à l'aide de l'argument suivant. *Les o-machines, nommées aujourd'hui machines de Turing à oracles, furent conçues par A. Turing dans son travail de thèse [222] sous la direction d'A. Church, peu de temps après celles des machines de Turing. L'idée de machines violant la thèse de Church-Turing est donc pratiquement aussi ancienne que cette thèse même. L'existence de telles machines n'implique pas de contradiction. Si l'existence d'un tel objet n'implique pas de contradiction, alors l'expérience seule permet de décider de sa possibilité. L'interprétation empirique des limites de la calculabilité est donc trivialement vraie, et aurait pu être défendue en même temps que la thèse de Church-Turing elle-même.* En critiquant cette trivialisation de l'interprétation empirique de la calculabilité, nous allons pouvoir, en guise de conclusion, préciser l'apport de notre étude à cette question.

Plusieurs raisons se conjuguent pour montrer l'insuffisance de cette trivialisation. La première est que l'existence d'un modèle d'hypercalcul, en sus de devoir passer une épreuve de consistance logique, doit également passer un test de faisabilité méthodologique. Pour que l'existence d'un modèle d'hy-

percalcul soit une question empirique, l'implémentabilité de ce modèle par un système donné doit pouvoir être établie par un protocole expérimental tenant compte de nos limitations opérationnelles. L'argument de vérification, et l'argument sceptique de Davis concluaient tous deux, quelque peu hâtivement, qu'il ne pouvait exister de tel protocole expérimental. Nous avons vu l'insuffisance de l'argument de vérification, en dénonçant la confusion entre indécidabilité d'un problème et impossibilité de résoudre aucune de ses instances. Nous avons également critiqué l'argumentation de M. Davis, en montrant qu'elle supposait indûment qu'il ne peut exister de modèles discrets de l'hypercalcul, et que l'impossibilité de tels modèles ne serait pas une question empirique. Dans l'état actuel de la littérature, il n'existe pas de raison de supposer qu'un argument méthodologique prohibe toute implémentation d'un modèle d'hypercalcul.

Pour décider de la possibilité d'un modèle d'hypercalcul, il semble donc nécessaire d'examiner les différents modèles existants dans la littérature, et d'expliciter les hypothèses aux fondements de la discussion de leur implémentabilité. À ce stade, il pourrait sembler que l'interprétation empirique de la calculabilité est déjà établie par l'existence même du débat sur les modèles d'hypercalcul, et la nature des arguments échangés. Le débat sur les modèles d'hypercalcul permet en effet de formuler l'argument contrefactuel suivant : *si tel état de chose s'était révélé différent de ce qu'il est avéré être, il aurait été possible d'implémenter un modèle d'hypercalcul.* Cet argument ferait plus que révéler le statut empirique de la thèse de Church-Turing physique, et par là des limites ultimes de la calculabilité. Il permettrait d'exhiber les hypothèses desquelles dépend le modèle d'hypercalcul étudié, et par là, d'identifier les hypothèses les plus générales dont dépend la possibilité de l'hypercalcul. Cet argument est implicite dans la citation suivante de C. Calude et B. Pavlov ([51], 124-125) :

> The main task of [the authors of hypercomputational models] is not to describe their methods, but to argue/prove that they do not contradict any *known* physical law. If a method would be shown to not be "theoretically implementable," then the result would still be interesting as that would show a new type of computational limit, *physical*, not logical.

Cet argument contrefactuel doit cependant être manié avec précaution. Employé de manière aveugle, il ne suffit pas à établir l'interprétation empirique de la calculabilité. Quand bien même on considérerait cette interprétation comme établie, il ne permet pas d'isoler la signification empirique de la thèse physique. Pour mieux comprendre ce dernier point, considérons cette remarque de J. Norton ([179], *Geometric morals, and a few more*) :

> A standard notion in the theory of computation is that it is impossible to complete a computation that requires infinitely many steps. It is easy to think of this as a logical requirement. It is

> not. It is a physical requirement. That has been made clear by the identification of spacetimes in which the infinite lifetime of one entity can be fully contained within the past light cone of another (...). The key element is that we need access to a particular sort of spacetime, which shows that it is a matter of physical contingency that we have no such combinations.

Si le débat sur le modèle de Malament-Hogarth montre que l'existence d'un modèle d'hypercalcul n'est pas qu'une question logique, et dépend de propositions empiriques contingentes, encore faut-il être prudent dans le choix de ces propositions. On pourrait ainsi formuler l'argument contrefactuel suivant : *si l'exécution d'une infinité d'étapes en un temps fini était possible, alors il serait possible de violer la thèse physique.* Nous avons montré au contraire qu'un tel argument contrefactuel ne peut appuyer l'interprétation empirique de la calculabilité. Tant qu'un modèle d'hypercalcul implémentant une telle supertâche n'a pas été physiquement défini, il n'est pas possible de déterminer si l'implémentabilité d'un tel modèle dépend ou non de considérations empiriques. Si son implémentation suppose une divergence d'une quantité observable, il serait possible d'exclure un tel modèle, mais non d'affirmer qu'il est impossible pour une raison proprement empirique. L'existence de modèles physiquement définis, ne souffrant pas de problèmes de divergence systématique, est donc nécessaire pour établir par argument contrefactuel l'interprétation empirique de la calculabilité.

L'emploi de l'argument contrefactuel pour établir la signification empirique de la thèse de Church-Turing doit également être employé avec précaution, lorsqu'il s'agit de déterminer la signification empirique d'un principe computationnel. Ainsi, il serait incorrect de formuler, comme nous y inviterait une lecture naïve de la remarque de J. Norton, l'argument contrefactuel suivant : *si les espaces de Malament-Hogarth existent réellement, il est possible d'implémenter un modèle d'hypercalcul.* L'existence de tels espaces, comme nous l'avons vu dans notre revue de ces questions, ne suffit pas à établir la possibilité du modèle de Malament-Hogarth. Il se pourrait que les espaces de Malament-Hogart existent, et qu'il soit strictement impossible de les exploiter pour réaliser un modèle d'hypercalcul. De nombreuses autres conditions doivent être satisfaites pour que le modèle de Malament-Hogarth puisse être implémenté et, comme le remarque J. Norton, les conditions garantissant l'accès à cet espace sont particulièrement importantes, et objets de débat.

En toute rigueur, l'antécédent de l'argument contrefactuel devrait consister en une conjonction d'hypothèses empiriques indépendantes suffisantes pour garantir l'implémentabilité de principe du modèle considéré. Pour qu'un tel raisonnement soit possible, il est impératif que toutes les hypothèses figurant dans l'antécédent soit d'authentiques propositions empiriques, et non des propositions méthologiques déguisées. Si l'antécédent décrit un modèle

impliquant une divergence de quantité observable, alors l'argument contrefactuel échouera non seulement à valider l'interprétation empirique de la calculabilité, mais aussi à montrer la dépendance spécifique de l'hypercalcul à telle ou telle hypothèse empirique. Dans le cas de prédictions de divergences de quantités observables, le modèle d'hypercalcul est simplement dépourvu de véritable signification empirique.

L'existence de telles anomalies est parfois une maladie que quelques ajustements théoriques peuvent guérir. Tant que ce traitement n'a pas été administré, il est néanmoins impossible de prétendre avoir montré la dépendance de la thèse de Church-Turing physique à des hypothèses empiriques données, ou plus précisément à une conjonction d'hypothèses empiriques donnée.

C'est donc l'ensemble des conditions garantissant l'implémentabilité d'un modèle physiquement défini qui doivent être prises comme antécédent de l'argument contrefactuel, pour formuler un argument correct. Seule la description complète du processus de calcul permet d'affirmer qu'on a bien affaire à une question empirique, et de s'interroger sur sa signification empirique précise. Toute discussion de la signification de la calculabilité devra tenir compte du fait de la pratique qu'un modèle physiquement défini du calcul, comme tout modèle de physique théorique, est une conjonction complexe de nombreuses hypothèses empiriques.

L'articulation de modèles physiquement définis d'hypercalcul est donc nécessaire, non seulement pour formuler un argument positif en faveur de l'interprétation empirique de la calculabilité, mais aussi pour préciser la signification empirique de la thèse de Church-Turing empirique.

Quatrième partie

Physique et complexité

.

Introduction de la partie

Nous allons à présent abandonner la théorie de la calculabilité, pour nous tourner vers la théorie de la complexité computationnelle. Au lieu de nous interroger sur la calculabilité des problèmes, nous allons à présent nous interroger sur leur « difficulté », et tâcher de déterminer quels problèmes doivent être considérés comme « faciles »ou « difficiles ». Par ces notions de « facilité »ou de « difficulté », nous désignerons ici, en abandonnant les guillemets de précaution, la préconception intuitive que la, ou plutôt les théories de la complexité essayent de quantifier par des mesures de complexité [49].

Durant les deux dernières décennies, l'examen des modèles non-standard a là aussi mené à considérer de nouvelles possibilités, et a placé les considérations d'implémentabilité au coeur des débats scientifiques. À bien des égards, la théorie de la complexité a été le théâtre d'une activité scientifique bien plus intense sur ces questions que la théorie de la calculabilité. Tandis que les modèles d'hypercalcul rencontrent fréquemment un fort scepticisme, et ne font à notre connaissance l'objet d'aucun programme concret d'expérimentation, un modèle non-standard de la complexité, le calcul quantique, a été l'objet d'investigations théoriques et expérimentales des plus intenses, au point de devenir une sous-discipline à part entière, mêlant informaticiens et physiciens : la théorie du calcul et de l'information quantique. Aucun autre modèle non-standard ne peut se targuer d'avoir été l'objet d'efforts scientifiques aussi considérables, et de figurer parmi les enjeux majeurs de la théorie du calcul et de la physique de ces vingt dernières années.

Comme le lecteur peut aisément le soupçonner, certaines des problématiques et positions que nous allons étudier sont analogues à celles rencontrées en calculabilité ; d'autres seront inédites. Nous veillerons à préciser à tout moment dans quelle mesure l'analyse de la complexité reconduit des problématiques rencontrées en théorie de la calculabilité, et dans quelle mesure elle en introduit de nouvelles.

À un niveau conceptuel, la théorie de la complexité computationnelle se démarque immédiatement de la théorie de la calculabilité, en ceci que ces premières définitions sont plus discutées, et parfois mêmes critiquées, que celles de la théorie de la calculabilité. En particulier, la notion de difficulté d'un problème se montre ardue à définir de manière à la fois rigoureuse et naturelle. Nous allons donc commencer par une introduction aux premiers éléments de la théorie de la complexité, et aux questionnements qu'ils ont pu susciter avant les débats soulevés par les modèles non-standard. Nous verrons ensuite le changement de point de vue introduit par ces modèles, avec l'étude de l'interprétation empirique de la complexité. Nous présenterons les enjeux

49. L'un des articles fondateurs de la théorie de la complexité moderne, écrit par A. Cobham, s'intitulait de fait *The intrinsic computational difficulty of functions* [61].

spécifiques de cette interprétation, les débats qu'elle a pu susciter, et ses conséquences pour la physique et la théorie de la complexité.

Chapitre 7

La difficulté des problèmes : problèmes de définition

> *It would be unfortunate for any rigid criterion to inhibit the practical developments of algorithms which are either not known or known not to conform nicely to the criterion. Many of the best algorithmic ideas known today would suffer by such theoretical pedantry.*

Jack Edmonds, *Paths, trees, and flowers*, 1965 [1]

Introduction : l'objet de la théorie de la complexité computationnelle

Il n'est pas évident par soi de parler de la, ou des difficultés d'un problème. Lorsqu'on évalue la complexité d'un problème, quelque soit la ressource envisagée, on étudie en réalité les propriétés de complexité d'un algorithme résolvant ce problème. Pour qu'on puisse attribuer la complexité à un problème, il faut disposer d'un résultat d'optimalité, permettant d'attribuer au problème les propriétés de complexité de l'algorithme le plus efficace.

Avant le début des travaux sur les modèles non-standard, on peut dire que la théorie de la complexité était fondée sur trois hypothèses :

- *Indépendance au hardware.* Les concepts et résultats de la théorie de la complexité sont indépendants des détails de conception du hardware.
- *Indépendance au modèle de calcul.* La plupart des résultats de théorie de la complexité, en particulier ceux portant sur la distinction entre calcul faisable et calcul infaisable, sont indépendants du modèle de calcul considéré.
- *Pertinence pour l'implémentation.* Malgré son abstraction théorique, les résultats donnés par la théorie de la complexité sont pertinents

1. [89], 451.

pour estimer la consommation de ressources exigée par l'exécution d'un programme sur une machine concrète.

Pour comprendre l'importance de ces hypothèses, comparons deux visions de l'objet de la théorie de la complexité, que l'on peut toutes deux trouver dans les manuels introductifs. La théorie de la complexité, selon un premier point de vue, serait une science mathématique ayant pour objet la mesure de la difficulté qu'il y a à résoudre un problème de calcul (*computational problem*). Selon une deuxième vision, tout aussi courante, la théorie de la complexité a pour objet l'estimation des ressources nécessaires à une machine donnée pour exécuter un programme résolvant un problème. Elle permet de répondre concrètement, du moins par un ordre de grandeur, à des questions d'ingénierie telle que « ai-je assez de mémoire sur mon ordinateur pour effectuer ce calcul ? », ou « combien de temps prendra l'exécution de ce calcul ? » [2].

Au premier regard, il semblerait qu'il y ait là deux visions incompatibles d'une même théorie. Cependant, une telle opposition n'est pas exprimée dans la rédaction des manuels, et elle ne donne pas lieu à des oppositions philosophiques au sein de la communauté des théoriciens de la complexité. La raison en est qu'une telle opposition est par trop schématique, au regard des hypothèses que nous venons de formuler. La théorie de la complexité n'est pas une branche particulière de l'ingénierie, dans la mesure où elle n'étudie pas des modèles particuliers du hardware d'une machine : c'est le sens de l'indépendance au hardware. Lorsqu'il cherche à déterminer le temps et l'espace pris par un calcul, le théoricien de la complexité n'étudie pas les limites d'un processeur, ou les capacités d'un support concret de mémoire. Il étudie dans l'abstrait le nombre d'étapes pris par un tel calcul, et le nombre d'adresses mémoire nécessaires, au sein d'un modèle abstrait du calcul, sans que ses résultats soient relatifs à ce modèle particulier : c'est le sens de l'indépendance au modèle de calcul. Ces contraintes pèseront ensuite sur toute machine qui implémentera un tel calcul. C'est en ce sens, et en ce sens uniquement, que la théorie de la complexité est pertinente pour déterminer les performances de véritables machines dans le temps et dans l'espace : c'est le sens de la pertinence pour l'implémentation.

Sous ces hypothèses, il semble naturel de définir l'objet de la théorie de

2. On trouve ainsi dans le manuel d'O. Goldreich [115] la citation suivante :

> Complexity Theory is a central field of the theoretical foundations of computer science. It is concerned with the study of *intrinsic complexity of computational tasks*. That is, a typical complexity theoretic study refers to the computational ressources required to solve a computational task (or a class of such tasks), rather than referring to a specific algorithm or algorithmic schemata.

On trouve la seconde présentation dans le manuel de C. Papadimitriou ([183], v) :

> Computational complexity is the area of computer science that contemplates the reason why some problems are so hard to solve by computers.

la complexité comme l'étude de la complexité des problèmes mathématiques, dont les résultats s'imposent à toute forme d'implémentation du calcul. Selon cette vision, le calcul est un processus abstrait, objet d'une théorie mathématique *a priori*. Les résultats de cette théorie déterminent des contraintes pertinentes pour toute implémentation de ce processus : le temps d'exécution et l'espace mémoire sont ainsi déterminés, à un certain ordre de grandeur près, par les étapes logiques décrites par l'algorithme [3].

Pour comprendre cette vision mathématique, en examiner les justifications, mais aussi en souligner les difficultés, nous allons tout d'abord présenter la définition de la difficulté par des fonctions de la taille des entrées, et l'hypothèse d'indépendance au hardware. Nous explorerons ensuite l'hypothèse d'indépendance au modèle de calcul, les propositions qui la justifient et celles qui en découlent.

7.1 Définition d'une fonction de complexité, et indépendance au hardware

La théorie de la complexité computationnelle choisit de modéliser la difficulté d'un problème par une fonction de la taille des instances de ce problème. Cette définition est à la fois naturelle et surprenante d'un point de vue intuitif. Elle est naturelle parce qu'elle permet d'étudier comment la difficulté du calcul croit avec la taille des instances, ce qui correspond à l'intuition élémentaire que des instances plus grandes sont plus complexes à résoudre que des instances plus courtes. Ainsi, il semble en règle générale plus simple de réaliser une addition à deux chiffres que de réaliser une addition à vingt-neuf chiffres. Néanmoins, considérer que la difficulté d'un problème n'est que la fonction de la taille des instances est en soi contre-intuitif. Ainsi, il semble intuitivement beaucoup plus facile de savoir si 22222222222 est un nombre premier, plutôt que de répondre à la même question pour 68718952447. Ces deux instances du problème TEST DE PRIMALITÉ ont pourtant la même taille en représentation décimale. Définir la complexité d'un problème, par opposition à la complexité de ses instances, suppose cependant d'adopter un

3. Cette vision mathématique de la théorie de la complexité est bien illustrée par la citation suivante de S. Arora et B. Barak, qui met en équivalence démonstrations d'impossibilité en mathématiques et démonstrations d'optimalité en complexité ([16], xxii) :

> How can we ever prove [the optimality of a given algorithm]? There are infinitely many possible algorithms! So we have to *mathematically* prove that each one of them is least efficient that the known algorithm. This may be possible because computation is a mathematically precise notion. In fact, this kind of result (if proved) would fit into a long tradition of *impossibility results* in mathematics, such as the independence of Euclid's parallel postulate from the basic axioms of geometry, or the impossibility of trisecting an arbitrary angle using a compass and straightedge. Such results count among the most interesting, fruitful, and surprising results in mathematics.

tel point de vue d'ensemble, qui perd en finesse ce qu'il gagne en généralité. En définissant la difficulté d'un problème comme une simple fonction de la taille de ces instances, la théorie de la complexité se permet d'ignorer les variations provoquées par telle ou telle symétrie particulière d'une instance, et de parler de complexité du problème en toute généralité. Lorsque deux instances de même taille ont de fait une complexité différente, la théorie de la complexité choisit de prendre la complexité de l'instance la plus difficile comme la valeur de la fonction de complexité pour cette taille d'argument : elle est donc une théorie du pire cas (*worst-case complexity*).

Du moins la théorie de la complexité du pire cas est le référent usuel de l'expression « théorie de la complexité ». Il existe une théorie alternative majeure, la théorie de la complexité du cas moyen. Celle-ci part du constat que bien des problèmes considérés comme difficiles au titre de la complexité du pire cas admettent de nombreuses instances faciles à résoudre. Dans certains cas, l'immense majorité des instances rencontrées dans la pratique sont faciles à résoudre, malgré la difficulté supposée du problème. Il existe ainsi des programmes permettant de résoudre en un temps polynômial un grand nombre d'instances d'un problème NP-complet [4]comme 3-SAT. L'informaticien L. Levin a défendu sur cette base la nécessité de définir une complexité de cas moyen, permettant de théoriser rigoureusement l'intuition de problèmes difficiles mais « faciles en moyenne »[159]. La définition rigoureuse de cette notion de complexité est quelque peu subtile, et nous laisserons le lecteur avide de plus de détails se reporter à la littérature (voir notamment [141], [236]).

Doit-on considérer comme difficile un problème dont la plupart des instances sont faciles ? De manière duale, doit-on considérer comme facile un problème dont il existe des instances très difficiles ? Ces question n'appellent pas de réponse rigoureuse, et permettent juste de prendre conscience de la subtilité d'une définition de la « difficulté d'un problème »à partir de la difficulté de ses instances. Entre la théorie de la complexité du pire cas, et celle de la complexité du cas moyen, il n'y a aucune raison de choisir la « véritable »théorie de la complexité. Toutes deux constituent des approches légitimes de l'étude de la difficulté des problèmes computationnels, qui partent de définitions différentes de cette difficulté. Il n'y a pas de raison *a priori* qu'une notion intuitive reçoive une définition scientifique unique.

Notre travail se concentrera uniquement sur la théorie du pire cas, pour deux raisons simples. La première est que tous les travaux reliant complexité et physique traitent de complexité du pire cas. Nous nous référerons donc

4. Soit C une classe de complexité. Un problème est C-difficile ssi un algorithme pour ce problème peut résoudre tout problème dans C. Un problème est C-complet ssi il est C-difficile et dans C. Pour montrer que la classe C est incluse dans la classe D, il suffit de montrer qu'un problème C-complet est dans D. Si l'on accepte la conjecture que la classe P est strictement incluse dans NP, il ne peut exister de solution efficace à un problème NP-complet.

dorénavant, par l'expression « complexité du problème », à la complexité du pire cas. Lorsque nous parlerons désormais de *la* difficulté du problème, le lecteur devra prendre l'article défini *cum grano salis*. La seconde est notre problématisation philosophique, centrée sur les limites fondamentales du calcul, qui renforce notre intérêt pour les performances optimales dans le cas le plus difficile.

Une fois admis le cadre conceptuel de la complexité du pire cas, il n'existe en réalité pas une mais des fonctions de complexité pour chaque problème. On peut mesurer la difficulté d'un problème en fonction de la taille de l'instance, par le plus grand nombre d'étapes de calcul nécessaire pour arriver au résultat, ou encore par le plus grand nombre d'adresses mémoire qu'on devra utiliser pour accomplir la tâche. On parlera respectivement alors de « complexité en temps »et de « complexité en espace. »La quantité mesurée par une fonction donnée de complexité est appelée une « ressource. »

Le temps et l'espace ne sont pas les seules ressources envisageables pour étudier la difficulté d'un calcul. À ce stade de notre analyse, on voit déjà comment la terminologie suggère la consommation de ressources physiques par le calculateur. Dans cette perspective, définir ainsi le temps de calcul comme le simple nombre d'étapes exécutées peut sembler problématique, dans la mesure où le temps pris par le calcul dépend également du taux d'exécution d'une étape élémentaire par unité de temps réel. Des considérations similaires valent évidemment pour l'espace. Même si nous aurons l'occasion de revenir sur cette question (voir section 8.2), on peut articuler succinctement les raisons pour lesquelles on a pu de prime abord choisir de telles définitions des ressources, sans penser perdre toute pertinence pour l'implémentation.

Tout d'abord, afin de se constituer comme théorie, la théorie de la complexité doit s'abstraire de détails technologiques tels que le progrès de rapidité des puces. Ensuite, et surtout, le nombre d'étapes exigées par l'algorithme peut être si écrasant qu'il se révèle plus pertinent pour l'implémentation que tout progrès technologique envisageable (voir section 6.1).

Pour garantir la portée théorique des résultats de complexité, il est bon de s'abstraire d'autres détails contingents, comme le choix d'une représentation particulière des données. Considérons pour fixer les idées la définition du temps de calcul. Il est intuitif de penser que le nombre exact d'étapes de calcul va dépendre du choix de la représentation des données. Pour reprendre un exemple dû à Arora et Barak [16], l'algorithme scolaire pour additionner deux entiers prendra environ trois fois plus d'étapes si l'on choisit une représentation binaire des entiers, plutôt qu'une représentation décimale. En outre, il existe un théorème d'accélération linéaire (*linear speed-up theorem*), qui stipule que si un problème est soluble par un algorithme d'une complexité donnée, alors il existe un algorithme permettant de résoudre ce même problème avec une accélération linéaire arbitraire [5]. Ce résultat joue également

5. Pour tout problème, pour tout $\epsilon > 0$, s'il existe une machine de Turing M résolvant

sur la représentation des données : il est basé sur la possibilité de simuler l'action de la première machine sur une suite des symboles de son alphabet, par une machine agissant sur une concaténation de cette suite en un unique symbole d'un nouvel alphabet enrichi.

Ces différentes remarques n'affectent cependant pas véritablement le projet de mesurer la complexité en temps d'un problème par le nombre d'étapes de calcul. Au lieu de compter le nombre exact d'étapes de calcul, qui est sensible à un choix de représentation des données, la théorie de la complexité détermine un ordre de grandeur du nombre d'étapes de calcul, défini à un facteur linéaire près. Cette conception a en outre l'avantage de ne pas donner une vision déformée de la complexité du problème, si les instances de petite taille sont artificiellement difficiles.

Cette négligence des facteurs linéaires ne résout pas tous les problèmes liés à la représentation des données. Si l'on veut être complet, il faut rappeler l'existence de problèmes admettant un algorithme *pseudopolynômial*, soit un algorithme polynômial en l'entier codant l'instance, et le plus grand entier apparaissant dans la définition de l'instance, plutôt qu'en la taille de la représentation de cette instance. L'algorithme est qualifié de pseudopolynômial parce qu'il est en réalité exponentiel en la taille de la représentation, sauf si cette représentation est unaire : la représentation unaire est en effet la seule représentation qui ne soit pas logarithmique en la taille de l'instance. Ceci mène à une propriété remarquable, à savoir l'existence de problèmes NP-complets *faibles*. Ceux-ci sont des problèmes NP-complets admettant un algorithme pseudopolynômial : il s'agit donc de problèmes supposés difficiles, mais qui admettent un algorithme polynomial en représentation unaire. Dans ces cas, un simple changement de représentation permet donc une accélération exponentielle de la vitesse de l'algorithme, et non une simple accélération linéaire. Par contraste, on appelle problèmes NP-complets *forts*, ou problèmes NP-complets *unaires*, les problèmes NP-complets qui demeurent difficiles même en représentation unaire. La plupart des problèmes NP-complets sont NP-complets forts, et cette remarque n'empêche pas qu'en règle générale la complexité soit indépendante du choix de représentation.

En outre, les fonctions de complexité sont définies de manière asymptotique dans le cadre de la notation O. La fonction de complexité d'un problème $g(n)$ est $O(f(n))$ ssi il existe une constante c et un entier n_0 tels que $\forall n > n_0$ $f(n) \leq cg(n)$. La fonction $g(n)$ est donc supérieurement bornée par la fonction $f(n)$ à un facteur linéaire près, pour toute valeur sauf un nombre fini d'entre elles. Comme le signalait déjà J. Edmonds dans [89], cette définition peut être quelque peu grossière dans la pratique. La taille des entrées examinées en pratique peut en effet être bornée sévèrement, tandis que le n_0 de la définition peut déjà être astronomiquement grand. Nous allons revenir sur ces problèmes dans la section suivante.

---

ce problème en temps $f(n)$, il existe une machine M' en temps $\epsilon f(n) + n + 2$.

7.2 L'indépendance au modèle de calcul

7.2.1 La thèse de Church-Turing étendue

Le deuxième obstacle rencontré dans la définition de la difficulté d'un problème est celui de la relativisation des résultats à un modèle de calcul. Pour être formulé rigoureusement, un algorithme, et les mesures de sa complexité, doivent être formalisés dans un modèle de calcul particulier. C'est uniquement à l'aide d'une formalisation rigoureuse des algorithmes qu'on peut évaluer leurs propriétés de complexité. La terminologie même de la théorie de la complexité porte la trace de cette relativisation, puisque de nombreuses classes de complexité, notamment les fameuses classes P et NP, sont définies relativement à un modèle de calcul particulier, à savoir les machines de Turing. Cette relativisation est souvent ignorée dans les manuels de théorie de la complexité, et l'on parle bien souvent de la classe P comme la « la classe des problèmes résolubles en temps polynômial », sans préciser « sur une machine de Turing déterministe » [6]. Comment définir une notion de complexité d'un problème qui soit invariante par changement de modèle de calcul ? En réalité, cette invariance a un statut particulier en théorie de la complexité, puisqu'elle fait l'objet d'une thèse, la thèse de Church-Turing étendue.

L'article de P. van Emde Boas publié en 1990 [230] constitue, à notre connaissance, l'une des plus anciennes présentations systématiques de la thèse de Church-Turing étendue. Il y exprime tout d'abord une vision de la théorie de la complexité en parfait accord avec les trois hypothèses que nous avons formulées (*ibid.*, 3, nous soulignons) :

> "(...) there are many notions of computationnal complexity, depending on the machine model one likes to adopt, and there seems to be no uniform notions of *time* or *space* complexity."
> "The conventional notions of time and space complexity within theoretical computer science are based on the implementation of algorithms on abstract machines, called *machine models*. The study of complexity measures in real computers is not pursued in theory, because it would make the results dependent on existing technology and hardware peculiarities rather than on insight and mathematical experience. Rather, *through suitable machine models one attempts to provide a reasonable approximation of what one might expect if a real computer were used for one's computations.*"

Les relations de simulation entre modèles permettent d'isoler, parmi les

6. En toute rigueur, il faut parler de la classe des problèmes de décision solubles en temps polynômial. Mais comme de nombreux types de problèmes peuvent s'encoder comme des problèmes de décision, cette limitation technique n'est usuellement pas tenue pour significative.

résultats de complexité, ce qui est relatif au modèle de calcul et ce qui peut être qualifié d'invariant, ou, pour reprendre un anglicisme devenu courant, de robuste. Dans la simulation,

> The size of the overhead factors tells us to what extent assertions about complexity are machine-based and to what extent they are machine-independent.

Dans l'expérience historique des théoriciens de la complexité, tous les modèles de calcul qu'on peut qualifier de « raisonnables »sont simulables par une machine de Turing avec un ralentissement au plus polynômial. La croyance que cet état de fait historique est voué à perdurer constitue une hypothèse fondamentale de la théorie de la complexité, appelée « thèse de Church-Turing étendue »(*Extended Church-Turing Thesis*)[7] :

Thèse de Church-Turing étendue. *Tout modèle de calcul raisonnable est simulable par une machine de Turing avec un ralentissement au plus polynômial.*

On peut ajouter à cette thèse une condition similaire sur l'espace, et obtenir la thèse d'invariance de P. van Emde Boas : les machines raisonnables peuvent se simuler avec au plus un ralentissement polynômial en temps et constant en espace[8].

La thèse de Church-Turing étendue, tout comme la thèse de Church-Turing, est en l'état une hypothèse informelle. Elle n'est pas une proposition mathématique, et ne peut donc ni être démontrée, ni être prise en axiome. La raison de cet état de faits épistémologique nous est familière depuis le début de notre étude sur la calculabilité : la notion de « modèle de calcul »n'est pas définie rigoureusement, et il n'est donc pas possible de raisonner mathématiquement sur l'ensemble des modèles de calculs possibles. Il est seulement possible de montrer les relations de simulation existant entre les modèles connus.

En outre, certaines formulations de la thèse de Church-Turing étendue font une référence spécifique au modèle des machines de Turing probabilistes (voir, par exemple [13], [178]). Cette référence est due à un trait de l'état de l'art en théorie de la complexité : il n'est pas démontré qu'une machine de Turing probabiliste, si elle n'a assurément aucun pouvoir expressif supplémentaire par rapport à une machine déterministe, ne disposerait pas d'un

7. La thèse de Church-Turing étendue est connue sous de nombreuses appelations, variant essentiellement par le qualificatif employé : forte (*strong*), quantitative (*quantitative*), efficace (*effective*). Nous avons choisi le qualificatif « étendue »pour la simple raison qu'il s'est trouvé être le plus fréquemment employé dans la littérature que nous avons pu consulter.

8. P. van Emde Boas souligne que cette thèse admet une interprétation orthodoxe (une unique simulation doit avoir les deux propriétés) et une interprétation libérale (une simulation pour chaque propriété). Presque toutes les simulations satisfont l'interprétation orthodoxe, et il est intéressant d'étudier des classes définies à la fois par des bornes temporelles et spatiales : deux arguments en faveur de l'interprétation orthodoxe.

pouvoir supplémentaire en complexité. Plus précisément, on définit la classe BPP, la classe des problèmes solubles par une machine de Turing probabiliste avec une erreur bornée, comme équivalent de la classe P pour une machine probabiliste[9]. S'il est trivial que $P \subseteq BPP$, il n'a pas été démontré que $BPP \subseteq P$. En ce sens, la thèse de Church-Turing étendue est assise sur une base moins solide que la thèse de Church-Turing, puisque deux modèles majeurs n'ont pas été démontrés équivalents. Nous aurons encore l'occasion de revenir sur la comparaison entre thèse de Church-Turing et thèse de Church-Turing étendue dans le chapitre suivant.

La thèse de Church-Turing étendue joue un rôle fondationnel en théorie de la complexité, dans la mesure où elle permet d'affirmer que tous les résultats portant sur la distinction polynômial-suprapolynômial sont indépendants du choix de modèle. Elle permet ainsi de raisonner au sein d'un modèle de calcul particulier, le plus souvent la machine de Turing, et de généraliser tous les résultats obtenus sur le caractère polynômial d'un algorithme à tous les autres modèles raisonnables. C'est en ce sens qu'elle permet donc de faire de la théorie de la complexité une véritable théorie unifiée, et non une simple collection d'études de modèles disparates sans relation entre eux. Ensuite, elle joue également un rôle dans l'assimilation des distinctions faisable-infaisable et polynômial-suprapolynômial, l'un des points les plus controversés de la théorie de la complexité.

7.2.2 La thèse de Cobham-Edmonds

Au sein d'une théorie de la complexité, il peut sembler désirable de distinguer entre les problèmes « faciles », ou « faisables », et les problèmes « difficiles », ou « infaisables ». En 1964, A. Cobham a proposé une identification entre problèmes faciles et problèmes solubles en temps polynômial [61]. Cette identification, tout en faisant l'objet de nombreuses critiques sur lesquelles nous reviendrons, s'est depuis lors établi dans l'usage, et est parfois nommée « thèse de Cobham-Edmonds », en l'honneur d'A. Cobham et J. Edmonds, qui a proposé une identification similaire en 1965 dans [89][10].

9. La classe *Bounded-Error Probabilistic Polynomial-time*, ou BPP, est définie comme la classe des langages L telle que, si M est une machine de Turing non-déterministe fonctionnant en temps polynômial, on ait :
- si $w \in L$ alors au $\frac{2}{3}$ des chemins sont acceptants.
- si $w \notin L$ alors au plus $\frac{1}{3}$ des chemins sont acceptants.

10. La classe des problèmes faciles étant identifié à la classe des problèmes solubles en temps polynômial, la classe des problèmes difficiles est souvent identifiée à la classe des problèmes solubles en temps exponentiel. Cet usage, qui prend probablement racine dans l'opposition de J. Edmonds entre temps « algébrique »et temps « exponentiel », est cependant trompeur, puisqu'il existe des classes intermédiaires, comme la classe sous-exponentielle. Nous préférerons donc parler de problèmes suprapolynomiaux pour désigner la classe des problèmes identifiés comme difficiles.

L'identification du polynômial et du facile est problématique. Un algorithme de complexité polynômial doué d'un exposant important n'est pas en pratique exécutable. Un gigabyte de données élevé à la puissance douze donne déjà un nombre de pas supérieur au nombre de baryons dans l'Univers observable [11]. De même, le caractère asymptotique de la complexité du pire cas rend problématique l'identification du suprapolynômial et du difficile. Si un algorithme fonctionne en $n^{\frac{1}{5}loglog10}$ opérations, il est en pratique plus utile qu'un algorithme en n^2, parce qu'il faut attendre des entrées de taille très grande (de l'ordre de $e^{e^{10}}$) pour que la seconde fonction vaille plus que la première [12]. Le même argument peut être employé pour les constantes ignorées par la notation O, puisque celles-ci peuvent être très grandes [13].

La légitimité de telles critiques peut cependant être discutée. Elle dépend fortement du cahier des charges imposé à la théorie de la complexité, plus précisément des aspects de la pratique qu'ils lui incombent de modéliser. La notion même d'algorithme, en tant que procédure applicable uniformément à toute entrée, contient une saine dose d'idéalisation : dans la pratique, on ne peut guère faire tourner un calculateur sur des entrées de taille arbitraire. La théorie de la complexité, si elle veut accomplir son objectif de définition d'une difficulté intrinsèque des problèmes par le biais d'une démonstration d'optimalité d'un algorithme, peut légitimement s'abstraire de bornes sur la taille des entrées encodables, surtout lorsque celles-ci sont vouées à évoluer avec les progrès de la technologie. Lorsque la taille considérée atteint des magnitudes cosmologiques, le problème est certes plus aigu. Mais pour préserver la pertinence pour l'implémentation, il n'est pas nécessaire que la théorie de la complexité modélise tous les aspects de la pratique liés à la consommation des ressources, mais plutôt qu'elle soit capable de formuler des prédictions concrètes sur certains aspects de la pratique. Comme toute théorie, la théorie de la complexité dispose d'un droit à l'idéalisation. Ce droit à l'idéalisation est justifié à la fois par le maintien d'un contact avec la pratique, et par l'aide que cette idéalisation apporte à la construction d'une théorie élégante et explicative, comme l'exprime bien la remarque suivante de C. Papadimitriou :

> It should not come as a surprise that our choice of polynomial algorithm as the mathematical concept that is supposed to capture the informal notion of "practically efficient computation" is open to criticism from all sides. Any attempt, in any field of mathematics, to capture an intuitive, real-life notion (for example, that

11. Ce nombre est couramment estimé être de l'ordre de 10^{80}.

12. La complexité du premier algorithme n'est pas qu'une simple vue de l'esprit, puisque le meilleur test de primalité déterministe est en $O(n^{cloglogn})$ (voir références dans [122]).

13. Il vaut sans doute la peine d'être souligné que cette difficulté n'était nullement ignorée par les premiers défenseurs de la thèse : J. Edmonds lui-même remarquait la difficulté posée à sa définition du faisable par les bornes pratiques sur la taille de l'entrée (voir [89], 451).

> of "smooth function" in real analysis) by a mathematical concept (such as C_∞) is bound to include some undesirable specimens, while excluding others that arguably could be embraced. Ultimately, our argument for our choice must be this : *Adopting polynomial worst-case performance as our criterion of efficiency results in an elegant and useful theory that says something meaningful about practical computation, and would be impossible without this simplification*[14].

En outre, la thèse de Cobham-Edmonds bénéficie d'arguments en sa faveur :

- *Propriétés de clôture.* La classe des fonctions calculables en temps polynômial jouit de propriétés de clôture désirables, notamment celles de clôture par somme, produit et composition. La somme, le produit et la composition de deux problèmes faciles semblent intuitivement devoir donner un problème facile. Une telle propriété ne serait pas satisfaite par d'autres définitions, par exemple celle identifiant calcul faisable et calcul en temps linéaire.
- *Exposants raisonnables en pratique.* Notre expérience en complexité montre que lorsqu'un programme possède un algorithme à temps polynômial avec un exposant élevé, il est en règle générale possible d'abaisser son exposant jusqu'à ce qu'il devienne plus « raisonnable », typiquement de l'ordre d'une puissance 2 ou 3.
- *Invariance par changement de modèle.* La thèse de Church-Turing étendue garantit que cette distinction entre « facile »et « difficile »est robuste, comme il semble désirable qu'elle le soit.

La thèse de Cobham-Edmonds peut être l'objet d'autres critiques, qui n'ont pas trait à la taille des données. Dans [122], Y. Gurevich souligne qu'il est regrettable de définir la faisabilité d'un calcul par référence à une seule ressource, en l'occurence le temps. Un algorithme peut avoir des exigences

14. ([183], 7, l'auteur souligne). Ce point est également exprimé par S. Aaronson dans un billet que nous citerons autant pour sa verve que pour sa justesse [4] :

> Yes, I'm aware that a polynomial-time algorithm can be impractical because of huge constant factors or a whopping exponent. I'm aware that an NP-complete problem can be easy on those instances that arise in practice. Even if I have a debilitating brain injury, so that I no longer remember my own name or how to count to 10, I like to think that I'll *still* be aware of these facts. To me, dismissing complexity theory because of its love affair with worst-case, asymptotic analysis is like dismissing physics because of its love affair with frictionless surfaces, point particles, elastic collisions, and ideal springs and resistors. In both cases, people make the simplifying assumptions not because they're under any illusions that the world really is that way, but rather because their goal is *understanding*. And in both cases, the theory itself gives you the tools to complicate your model -to put legs and hooves on your spherical cow- until you get reasonably-accurate predictions for the practical problems that interest you.

raisonnables en une ressource, et déraisonnables en une autre : la faisabilité d'un problème peut donc dépendre d'une combinaison de ressources. On n'a donc pas prouvé qu'un problème est faisable parce qu'il est faisable en temps. Nous nous permettrons d'ajouter que la même remarque s'étend à la notion d'« algorithme optimal » : l'optimalité d'un algorithme doit être relativisée à la quantité démontrée optimisée. Pour montrer qu'un algorithme est véritablement optimal, il faudrait donc démontrer qu'il est optimal *en toutes les ressources pertinentes*, ce qui n'est pas sans poser quelques problèmes de définition. Outre qu'il faut arriver à formuler une liste exhaustive de ressources pertinentes, l'algorithme optimal en toutes les ressources pertinentes n'est pas nécessairement l'algorithme qui optimise toutes les ressources simultanément, mais celui qui se situe dans l'intersection des classes de problèmes faisables en une ressource donnée (voir section 8.3.1.).

En outre, la notion de faisabilité peut être modifiée en fonction de la notion de difficulté du problème étudiée. Un même problème peut ainsi prendre un temps polynômial en complexité du cas moyen, tandis qu'il n'existe pas d'algorithme efficace en complexité du pire cas. Sur ce point, nous nous pouvons que reproduire la position pluraliste que nous avons déjà formulée plus haut : différentes notions de difficulté mènent à différentes notions de faisabilité, qui n'ont pas de raison *a priori* de converger.

7.2.3 Note sur les axiomes de Blum, et la théorie de la complexité computationnelle indépendante du modèle

La relativisation des résultats de complexité à un modèle, et son dépassement par l'énoncé d'une thèse informelle comme la thèse de Church-Turing étendue, peut sembler superficiellement contradictoire avec l'existence de ce qu'on appelle la théorie de la complexité computationnelle indépendante du modèle (*model-independent complexity theory*). Dans cette brève note, que le lecteur peut sauter sans grand dommage en première lecture, nous expliquerons brièvement pourquoi il n'en est rien.

La théorie de la complexité indépendante du modèle est fondée sur l'énoncé des axiomes de Blum [35]. Ces axiomes visent à donner une définition indépendante du modèle d'une mesure de complexité admissible. Soit (φ, ϕ), φ un codage de Gödel des fonctions récursives partielles $P^{(1)}$, et $\phi : \mathbb{N} \mapsto P^{(1)}$. On note φ_i la i-ème fonction calculable sous le codage de Gödel φ, et ϕ_i pour la fonction calculable partielle ϕ_i. On a :

1. $Dom(\varphi_i) = Dom(\phi_i)$.
2. L'ensemble $\{(i, x, t) \in \mathbb{N}^3 | \phi_i(x) = t\}$ est récursif.

De manière alternative, et quelque peu plus intuitive, on peut également les formuler de la manière suivante. Soit M une machine de Turing et m son code de Gödel, ϕ une mesure de complexité, on a :

1. $\phi(m, x)$ est fini ssi $M(x)$ s'arrête.

2. Il existe une machine de Turing décidant, sur la donnée (m, x, r) si $\phi(m, x) = r$.

Ces axiomes garantissent respectivement :

1. que la mesure de la ressource n'est définie que sur les entrées sur lesquelles la fonction est définie.
2. savoir si une valeur donnée est bien la valeur de la mesure correspondant à une entrée donnée pour un programme donné est un problème décidable.

Les axiomes de Blum interviennent notamment dans la démonstration d'un théorème fondamental, valide pour toute mesure de Blum. Ce théorème, dit *speed-up theorem*, est également du à M. Blum. Il affirme qu'il existe un langage décidable L tel que, pour toute mesure de complexité de Blum $\phi(n)$, si L est de complexité $\phi(n)$, L est de complexité $\ln(\phi(n))$. La fonction logarithme neperien peut être remplacée par n'importe quelle fonction calculable de croissance arbitrairement lente. Le théorème démontre donc qu'il n'existe des langages dont la complexité en une ressource quelconque peut être arbitrairement abaissée. D'un point de vue intuitif, on peut dire que de tels langages n'ont pas de complexité optimale. Il n'est donc pas possible de parler de la complexité intrinsèque de tout problème.

La portée de ce résultat, si intéressant soit-il, ne doit pas être exagérée. Le *speed-up theorem* n'empêche pas ainsi de penser que la plupart des problèmes « naturels »ont une complexité optimale. Sans cette croyance, la théorie de la complexité s'effondrerait, puisqu'il serait vain de distinguer entre diverses classes de complexité.

En outre, les seules mesures de complexité de Blum sont le temps et l'espace. La définition ne nous fait pas découvrir de nouvelles ressources intéressantes, comme on pourrait le désirer avec une définition abstraite.

Comme le remarque l'informaticien E. Blakey [30], les axiomes de Blum sont à la fois naturels et extrêmement faibles. Si leur caractère nécessaire est difficilement discutable, leur caractère suffisant l'est bien davantage. E. Blakey a ainsi récemment défendu la nécessité d'enrichir les conditions de Blum pour l'étude de la complexité dans les modèles non-standard. Nous reviendrons sur ce point lorsque nous aborderons les problèmes liés à la définition des ressources (voir section 8.2).

Conclusion

Avant de voir comment la conception de la théorie de la complexité comme théorie mathématique de la difficulté des problèmes a pu être contestée, nous allons en guise de conclusion résumer brièvement ce qui fait sa singularité.

L'unité de la théorie de la complexité est fondée sur une hypothèse informelle, la thèse de Church-Turing étendue. Sans elle, la théorie de la com-

plexité ne serait qu'une collection d'étude de modèles de calcul disparates, sans possibilité de généraliser ses résultats de l'un à l'autre. En outre, le caractère mathématique de la théorie de la complexité est lui-même fondée sur cette hypothèse informelle. La notion de fonction de complexité n'admet pas de définition mathématique absolue dans le langage mathématique usuel : elle est une expression générique désignant une famille de fonctions définies au sein d'une famille de modèles, dont l'équivalence est garantie par des relations de simulation. La thèse de Church-Turing étendue permet ensuite d'étendre cette relation de simulation entre modèles existants aux modèles à venir, et de considérer les résultats de complexité comme révélateurs de propriétés intrinsèques du problème. La théorie de la complexité n'est donc pas une théorie mathématique au sens usuel, dont les objets peuvent être définies au sein de la théorie des ensembles, mais au sens plus lâche où elle permet une étude *a priori* des propriétés de la complexité. Toute démonstration réalisée au sein du modèle des machines de Turing se généralise aux autres modèles, sans qu'une expérience empirique soit nécessaire pour valider cette généralisation.

Toutes ces considérations s'appliquent tout aussi bien à la thèse de Church-Turing originale qu'à sa version étendue. Nous nous pencherons un peu plus loin sur les questions propres à la théorie de la complexité (voir section 8.2).

Chapitre 8

Problèmes autour de l'indépendance au modèle de calcul : l'interprétation empirique de la complexité

Introduction

Le problème évident posé par la thèse de Church-Turing étendue, telle qu'elle est formulée par P. van Emde Boas, est l'emploi du qualificatif « raisonnable ». Celui-ci suggère qu'il existe déjà des modèles de calcul violant la thèse de Church-Turing étendue, qu'on ne peut exclure qu'en argumentant qu'ils ne sont pas « raisonnables », ou « réalistes ». Le problème est qu'un tel emploi intuitif des termes permet des ajustements *ad hoc* lorsqu'on rencontre un contre-exemple, et rend ainsi la notion de contre-exemple vague et imprécise [1].

Selon P. van Emde Boas, la thèse n'est pas vraiment une proposition susceptible de vérité, mais une directive pour classer des modèles. On peut ainsi remplacer le concept de 'modèle raisonnable' par celui de 'classe de machine' (*machine class*) [2]. Lorsque l'on dit que tous les modèles de calcul raisonnables sont simulables efficacement par une machine de Turing, il faut comprendre cette proposition comme une définition implicite de la notion de modèle raisonnable. Selon cette interprétation, la thèse de Church-Turing étendue n'a de fonction que classificatoire, et n'a pas à être définie et discutée rigoureusement.

Cette solution de repli est cependant insatisfaisante à bien des égards. Tout d'abord, elle accorde un privilège arbitraire aux machines de Turing parmi tous les modèles de la complexité. Ensuite, et surtout, elle fait beau jeu

1. [230],6.
2. Voir [230], et les références mentionnées.

du rôle joué par la thèse de Church-Turing étendue dans notre compréhension des fondements de la théorie de la complexité. Une position comme celle de P. van Emde Boas entraînerait l'abandon de toute ambition de formuler des résultats robustes pour tout modèle. Elle rendrait plus problématique encore l'assimilation du faisable au polynômial, en renonçant à la robustesse de la distinction entre polynômial et suprapolynômial. P. van Emde Boas ne prend pas position sur ces questions, qui sont pourtant des enjeux cruciaux de la thèse de Church-Turing étendue.

À la veille de l'émergence du calcul quantique, l'état de l'art de l'analyse de la thèse étendue ne semble guère satisfaisant. Ni la solution de repli de van Emde Boas, ni l'absence complète d'analyse ne sont désirables. Il se préfigure déjà, en filigrane du papier de P. van Emde Boas, la mention de critères d'implémentabilité. Celui-ci écrit en effet ([230], 14), en commentant quelques modèles du calcul parallèle aux performances apparemment exceptionnelles :

> It seems that the marvelous speed-ups obtained by the parallel models [...] require severe violations of the basic laws of nature [...]. Stated differently : if physical constraints are taken into account, all gains of parallelism seem to be lost [...].

Tout comme dans le cas de la thèse physique, ce sont ces critères d'implémentabilité qui vont jouer un rôle central dans les interprétations récentes de la thèse étendue.

8.1 L'interprétation empirique de la complexité

Durant ces deux dernières décennies, plusieurs auteurs, dont P. Shor [204], E. Bernstein, U. Vazirani [28] et D. Aharonov [13], ont défendu une interprétation alternative de la thèse de Church-Turing étendue. Dans [204], P. Shor réagit à l'article de P. van Emde Boas, et propose une compréhension empirique de la notion de modèle raisonnable (*ibid.*, pp.2-3) :

> The generally accepted compromise between coarseness and precision distinguishes efficiently and inefficiently computable functions by whether the length of the computation scales polynomially or superpolynomially with the input size. The class of problems which can be solved by algorithms having a number of steps polynomial in the input size is known as P. For this classification to make sense, we need it to be machine-independent. That is, we need to know that whether a function is computable in polynomial time is independent of the kind of computing device used. This corresponds to the following quantitative version of Church's thesis, which Vergis et al. [1986] have called the "Strong Church's Thesis" and which makes up half of the "Invariance Thesis" of van Emde Boas [1990] (Quantitative Church's

> thesis). Any physical computing device can be simulated by a Turing machine in a number of steps polynomial in the resources used by the computing device. (...)
>
> Because of the remarkable effectiveness of our mathematical models of computation, computer scientists have tended to forget that computation is dependent on the laws of physics. This can be seen in the statement of the quantitative Church's thesis in van Emde Boas [1990], where the word "physical"in the above phrasing is replaced with the word "reasonable."It is difficult to imagine any definition of "reasonable"in this context which does not mean "physically realizable," i.e., that this computing machine could actually be built and would work.

P. Shor propose donc de résoudre le problème posé par le flou artistique entourant la notion de « modèle raisonnable », en lui substituant la notion de « modèle physiquement réalisable. »Cette substitution appelle plusieurs commentaires. Tout d'abord, tout comme le cas de la calculabilité, il est naturel de considérer que l'implémentabilité est une condition nécessaire d'acceptation d'un modèle. D'autre part, on peut se demander s'il s'agit bien là d'une condition suffisante. Comme nous venons de le voir, la machine de Turing unaire est un modèle implémentable, mais elle viole la thèse de Church-Turing étendue pour des raisons liées à la seule représentation des données. Dans le cas de la complexité, l'assimilation pure et simple de la notion de « modèle raisonnable »avec celle de « modèle implémentable »est donc discutable dans le détail. Mais puisque ces subtilités ne joueront pas de rôle dans nos discussions, nous continuerons à employer les qualificatifs de « raisonnable », « réaliste »ou « implémentable »comme des synonymes.

En tout état de cause, une telle interprétation de la notion de « modèle raisonnable »bouleverse radicalement le statut épistémologique de la thèse de Church-Turing étendue. L'impossibilité de démontrer rigoureusement cette thèse viendrait alors du fait qu'il n'y a pas de démonstration à faire, mais une hypothèse empirique à accepter. La thèse de Church-Turing étendue n'est pas une thèse informelle, ou une proposition mathématique susceptible d'être formalisée, puis démontrée ou prise comme axiome, mais une hypothèse empirique. Cette hypothèse empirique, en tant que telle, ne peut être mathématiquement démontrée.

En gardant cette nouvelle vision à l'esprit, il est plus aisé de comprendre pourquoi la thèse de Church-Turing étendue ne doit pas être considérée comme une définition implicite, comme le défendait P. van Emde Boas. Si l'on pense que cette thèse devrait être une proposition mathématiquement bien définie, un malaise peut naître de l'absence de définition rigoureuse de la notion de « modèle raisonnable. »A première vue, il pourrait donc sembler pertinent de considérer cet énoncé comme une définition implicite de la notion. Mais on deviendrait alors incapable de discuter des mérites de modèles

de calcul originaux, une conséquence fort peu désirable.

Un point de vue empirique nous permet de sortir de cette impasse. Nombre de concepts fondamentaux de la physique ne sont jamais définis avec une parfaite rigueur mathématique. Qu'on songe par exemple au principe de la réduction du paquet d'ondes en mécanique quantique : lorsqu'on effectue une mesure, la fonction d'onde du système mesuré se projette sur un état propre de l'observable mesurée. Un tel principe ne précise pas ce qui est censé compter comme une mesure, et cela a précisément été l'objet de discussions au sein des interprétations de la mécanique quantique (voir par exemple [129]). Cela n'implique pas que le principe soit une définition implicite de ce qu'est une mesure. En mathématiques, on ne peut rigoureusement parler de contre-exemple à une proposition qui n'est pas explicitement définie. Ce n'est pas le cas en sciences empiriques : les physiciens s'accordent pour penser qu'ils reconnaîtraient un contre-exemple à un principe fondamental, même si ces principes ne sont jamais définis avec une parfaite rigueur, et laissent toujours place à interprétation. En ce sens, les principes fondamentaux, s'ils ne peuvent être démontrés, peuvent être falsifiés, et doivent être interprétés comme d'authentiques propositions, et non comme de simples définitions implicites.

De manière analogue, nous n'arriverons pas, et ne devrions pas essayer, d'atteindre une parfaite rigueur mathématique dans la définition de ce qu'est un modèle raisonnable de la complexité. Cela ne devrait pas nous retenir de discuter cette problématique, et ne nous empêchera pas de reconnaitre un contre-exemple à la thèse de Church-Turing étendue, si jamais nous en rencontrions un. Ainsi, cette réinterprétation empirique de la thèse de Church-Turing étendue illumine son statut présent, et la compréhension que nous devons en avoir. C'est une proposition empirique, qui ne peut être démontrée, mais qui peut être falsifiée par un contre-exemple, et corroborée inductivement par le rejet de contre-exemples supposés, tout comme nos principes physiques fondamentaux sont corroborés par des années de résistance à des expériences parfois en apparence rétives.

Une fois que la thèse de Church étendue a été réinterprétée comme une hypothèse empirique, les arguments présentés en sa faveur peuvent également être réinterprétés (*ibid.*, p.3) :

> Computer scientists have become convinced of the truth of the quantitative Church's thesis through the failure of all proposed counter-examples. Most of these proposed counter-examples have been based on the laws of classical mechanics ; however, the universe is in reality quantum mechanical. Quantum mechanical objects often behave quite differently from how our intuition, based on classical mechanics, tells us they should. It thus seems plausible that the natural computing power of classical mechanics corresponds to Turing machines,(...) while the natural compu-

ting power of quantum mechanics might be greater.

Notre compréhension de la thèse de Church-Turing étendue aurait donc été influencée, de manière subreptice, par des présupposés physiques. Notre incapacité à trouver un contre-exemple à la thèse de Church-Turing étendue proviendrait non pas d'une quelconque validité de cette thèse, mais de sources d'inspiration trop étroites, parce que trop influencées par des intuitions « classiques ». Un modèle de calcul quantique plus original, prenant parti des propriétés exotiques de la physique quantique, permettrait de rompre le charme exercé par la thèse de Church-Turing étendue[3].

Dans notre terminologie, l'analyse de P. Shor peut être qualifiée d' interprétation empirique de la complexité computationnelle : « la définition et l'évalution robustes d'un coût en complexité d'une tâche computationnelle au sein d'un modèle dépendent des hypothèses empiriques assurant l'implémentabilité du modèle. »On peut formuler la même interprétation de manière plus succincte en disant que « le caractère raisonnable d'un modèle de la complexité dépend d'hypothèses empiriques. »Dans une telle perspective, les limites du pouvoir calculatoire d'un modèle raisonnable reposent sur des hypothèses empiriques. C'est précisément l'idée que nous avions capturée sous l'appellation d'interprétation empirique de la complexité.

Comme on vient de le voir, la thèse de Church-Turing étendue joue un rôle fonctionnel essentiel en théorie de la complexité, à la fois comme fondement de l'hypothèse d'indépendance au modèle de calcul, et comme argument en faveur de l'assimilation du faisable au polynômial. Si la thèse de Church-Turing étendue est un principe empirique, c'est donc à une révision de notre compréhension du sens de la théorie de la complexité que nous sommes invités.

8.2 *Apples to oranges* : enjeux spécifiques de l'interprétation empirique de la complexité

Comme son nom l'indique, la thèse de Church-Turing étendue semble être une simple extension de la thèse de Church-Turing : en sus de calculer l'intégralité des fonctions calculables, le modèle de Turing permet de calculer de manière efficace l'intégralité des fonctions efficacement calculables. En outre, nos premières réflexions sur les rapports entre théorie de la complexité et implémentabilité peuvent sembler un simple décalque des problématiques rencontrées lors de notre analyse de la calculabilité. L'interprétation empirique de la complexité serait un simple prolongement de l'interprétation empirique de la calculabilité. Un regard plus attentif va cependant nous révéler que la théorie de la complexité soulève des enjeux spécifiques, et que

3. Ce n'est d'ailleurs pas un hasard si les partisans de cette interprétation que nous avons mentionnés, D. Aharonov, E. Bernstein, P. Shor, et U. Vazirani, ont tous contribué à l'étude du calcul quantique.

les analogies de formulation cachent parfois d'importantes différences entre les deux thèses.

Nous avons déjà affirmé, lors de notre étude de la calculabilité (voir section 1.3.3), que l'implémentabilité était une condition nécessaire du caractère raisonnable d'un modèle. Nous pouvons à présent préciser cette condition, en affirmant que l'implémentabilité est une condition nécessaire non seulement pour déterminer l'ensemble des fonctions calculables au titre d'un modèle, mais aussi pour quantifier les ressources consommées au titre de ce modèle. Les deux affirmations ne sont pas redondantes. L'opération de β-réduction du λ-calcul est manifestement implémentable par un calcul papier-crayon, puisqu'il s'agit d'une opération de substitution uniforme au sein d'un mot fini. En revanche, compter une étape de β-réduction comme une mesure du nombre d'étapes de calcul est manifestement une définition irréaliste de la complexité en temps, puisque cela imposerait de considérer comme équivalentes une substitution à une occurrence de la variable à un ensemble de substitutions à un nombre arbitraire d'occurrences de la variable. On peut être un modèle raisonnable de la calculabilité sans permettre en l'état une définition raisonnable du temps de calcul.

Tout comme la thèse de Church-Turing algorithmique, la thèse de Church-Turing étendue ne doit pas être comprise comme une définition implicite de la notion de « modèle raisonnable ». Il est néanmoins difficile d'attribuer à la thèse étendue une fonction similaire à celle de la thèse algorithmique, à savoir celle de reconstruction rationnelle d'un concept intuitif. On ne peut en effet affirmer que la thèse étendue vise à procurer une reconstruction rationnelle d'une notion comme celle de « calcul efficace par une procédure effective ». Pour adopter une telle position, en sus d'ignorer l'alternative entre théorie du pire cas et théorie du cas moyen, il faudrait considérer comme allant de soi l'équation du polynômial et de l'efficace. Or, nous avons vu que cette mise en équation faisait l'objet d'une proposition distincte, à savoir la thèse de Cobham-Edmonds, qui s'appuie elle-même sur la validité de la thèse étendue. La conjonction de la thèse de Cobham-Edmonds et de la thèse étendue peut être considérée comme une reconstruction rationnelle de la notion de « calcul efficace par une procédure effective », mais pas la thèse étendue considérée isolément. C'est cette complexité supplémentaire de la reconstruction rationnelle des notions intuitives qui fait que les premières notions de la théorie de la complexité sont bien plus discutées que celles de la calculabilité.

Parce qu'il ne suffit pas d'être un modèle de la calculabilité pour être un modèle de la complexité, la thèse algorithmique et la thèse étendue se distinguent également par le sens de leur mention du modèle de Turing. La thèse de Church-Turing algorithmique se réfère au modèle de Turing en tant qu'exemple de modèle universel. Elle peut être reformulée avec n'importe quel autre modèle équivalent, comme les fonctions récursives ou le λ-calcul. La situation est tout autre dans le cas de la thèse de Church-Turing étendue. Celle-ci se réfère au modèle de Turing en tant que modèle particulier, au

sein duquel a été historiquement défini le temps et l'espace de calcul discrets d'une procédure effective. La thèse de Church-Turing étendue affirme donc que les mesures de complexité définies à l'aide du modèle de Turing sont raisonnables, et qu'un autre modèle raisonnable de la complexité sera donc simulable pas à pas par une machine de Turing, à un facteur polynômial près.

Cette dernière affirmation appelle un commentaire immédiat, qui va nous amener à la différence majeure entre thèse simple et thèse étendue. On peut être tenté de préciser que les mesures définies au sein du modèle de Turing sont raisonnables *pour une procédure effective.* La définition d'une mesure de complexité raisonnable dans des modèles non-effectifs pourrait relever de considérations totalement différentes, et faire donc l'objet d'une thèse différente. Il serait donc désirable de distinguer une thèse étendue algorithmique et une thèse étendue physique, comme on l'a fait pour la calculabilité.

À ce titre, la littérature rapprochant calcul et physique montre une dissymétrie de traitement significative entre calculabilité et complexité. Dans les discussions sur la calculabilité, la distinction entre thèse physique et thèse algorithmique, sous diverses appelations, est devenue très répandue. À notre connaissance, une distinction analogue n'a jamais été défendue au sujet de la thèse étendue. La formulation en termes de « modèle raisonnable », choisi par exemple par P. Shor, cache sous le même chapeau les modèles de calcul par procédure effective, et les modèles de calcul non-effectifs.

Nous allons voir qu'une telle omission n'est pas qu'une contingence historique, et qu'il est problématique de formuler une distinction équivalente pour la théorie de la complexité. Pour mieux le voir, examinons une formulation naïve de cette distinction :

Thèse de Church-Turing étendue (algorithmique). *toute fonction calculable en temps polynômial par une procédure effective est calculable en temps polynômial par une machine de Turing.*

ainsi que

Thèse de Church-Turing étendue (empirique). *toute fonction calculable par une machine en temps polynômial est calculable en temps polynômial par une machine de Turing.*

L'incongruité manifeste de la formulation algorithmique est due à l'un des points que nous venons de souligner, à savoir que la thèse étendue ne peut être conçue en elle-même comme une reconstruction rationnelle d'une notion intuitive. La notion de « fonction calculable en temps polynômial par une procédure effective »ne correspond guère à une notion préexistante à la théorie de la complexité computationnelle, dont il s'agirait par l'intermédiaire de la thèse de déterminer l'extension mathématique.

Mais surtout, l'examen plus précis des contraintes pesant sur une définition raisonnable du temps montre que ces deux thèses ne sont pas autonomes.

Pour mieux comprendre ce dernier point, nous allons à présent examiner les relations complexes entre ressources physiques consommées par une machine et quantification de ces ressources au sein du modèle de calcul.

Comme nous l'avons vu précédemment (voir section 7.2), le modèle de Turing définit le temps de calcul par le nombre d'étapes élémentaires effectuées. Cette définition est rendue naturelle par l'hypothèse implicite nommée *taux de traitement borné de l'information* : il existe une borne supérieure au nombre d'opérations élémentaires effectuables par unité de temps. Grâce à cette hypothèse, l'intervalle de temps réel d'exécution du calcul dépend avant tout du nombre d'étapes effectuées d'un point de vue fondamental, et non simplement en vertu d'un certain état d'avancement de notre technologie. L'hypothèse d'un taux de traitement borné de l'information est donc le fondement théorique de la notation O en complexité asymptotique. Si le temps d'exécution d'une étape élémentaire de calcul est inférieurement bornée pour des motifs fondamentaux, la complexité d'une tâche peut être définie dans l'ignorance du temps d'exécution réel d'une étape élémentaire. Celui-ci, décrit par une constante, sera négligeable asymptotiquement face au nombre d'étapes défini par la fonction de complexité en temps. Cette dépendance primordiale du temps réel d'exécution au temps discret de la fonction de complexité permet de relier une différence d'ordre de grandeur dans la fonction de complexité en temps de deux algorithmes à une différence d'ordre de grandeur dans les intervalles de temps réels pris par toute machine exécutant ces algorithmes pour parvenir au résultat [4]. Par exemple, si deux algorithmes a et a' sont respectivement de complexité en temps polynômiale et exponentielle, cela signifie que toute machine exécutant ces algorithmes consommera un intervalle de temps réel supérieur en exécutant l'algorithme exponentiel. Par le biais de l'hypothèse de taux de traitement borné, les résultats de complexité en temps deviennent ainsi des prédictions sur le comportement de tout système physique exécutant un algorithme donné, et sur l'intervalle de temps réel pris par son évolution.

Ces dernières remarques, si elle sont formulées pour le temps de calcul, peuvent aisément se généraliser à toute définition discrète d'une ressource. Pour l'espace mémoire, la quantité observable correspondante serait intuitivement le volume réel, à condition qu'un principe de densité minimale de l'information soit admis comme principe fondamental de la physique [5]. Là

4. Cette formulation doit cependant être prise *cum grano salis*. De par la définition asymptotique des fonctions de complexité, le nombre d'étapes élémentaires exigées par un algorithme peut ne devenir supérieur à celui d'un algorithme de complexité inférieure qu'à partir d'un certain N, qui peut être astronomiquement grand. Mais comme nous l'avons vu dans notre discussion des idéalisations de la complexité asymptotique (voir section 7.2.2), on accorde crédit à la théorie de la complexité asymptotique tant que l'on pense que ce type de situation demeure une exception, et que la théorie nous permet malgré tout de prédire en général le comportement relatif concret des algorithmes.

5. Si le principe holographique était admis, la quantité d'espace consommé ne serait plus mesurée par un volume mais par une surface, ce qui illustre encore une fois la dépen-

encore, une fois admis l'existence d'une borne inférieure fondamentale sur la densité d'encodage, l'ordre de grandeur de l'espace mémoire nécessaire a un calcul est avant tout décrit par le nombre de cases mémoire exigées. De manière générale, si on nomme « ressource théorique »la définition d'une ressource au sein d'un modèle, et « ressource observable »la quantité mesurable modélisée par la ressource théorique, on peut parler d'*hypothèse de proportionnalité des ressources observables aux ressources théoriques.*

Ce type d'hypothèses, qui relient la quantification d'une ressource au sein du modèle et consommation de son pendant observable lors de l'exécution du calcul par une machine, est nécessaire à la pertinence pour l'implémentation du modèle. Mais elles constituent des hypothèses empiriques indépendantes de la thèse de Church-Turing étendue. La plausibilité de la thèse de Church-Turing étendue est avant tout basée sur l'existence de relations de simulation efficace entre modèles que l'on sait par ailleurs être implémentables. Mais l'hypothèse de proportionnalité est nécessaire pour que ce type de démonstrations mathématiques aient une quelconque pertinence pour l'implémentation, et ne prouve pas juste l'équivalence d'un modèle irréaliste à un autre modèle irréaliste.

Nos dernières remarques s'appliquent évidemment aux modèles « classiques »de la complexité, qui sont inspirés de la pratique des procédures effectives. Mais elles valent également pour tout modèle de calcul à temps et espace discrets. Cette généralisation n'est pas sans importance conceptuelle, puisque les modèles à temps et espace discrets peuvent être fondés sur une conception du calcul très différente de celle des procédures effectives. Nous avions déjà évoqué ce point lors de notre évocation de la définition d'une étape de calcul élémentaire par N. Margolus et L. Levitin (voir section 6.1). Pour comprendre la portée de l'hypothèse d'un taux de traitement borné, nous allons maintenant comparer la conception d'une étape élémentaire de calcul dans un modèle du calcul effectif, les circuits booléens, et dans un modèle non-effectif, les circuits quantiques.

Le modèle des circuits quantiques, comme son nom l'indique, se veut un analogue théorique du modèle des circuits booléens pour le calcul quantique. Dans le cas booléen ou classique, entrées et sorties sont encodées par des suites de n bits $\{0,1\}$. L'exécution du calcul est réalisée par des opérations élémentaires sur des ensembles de bits, nommées *portes logiques.* Cette appelation est justifiée par l'analogie dénotationnelle entre l'action de certaines de ces portes sur les bits et les tables de vérité de certains connecteurs du calcul propositionnel classique. Ainsi l'action d'une porte $BIT-FLIP$, $0 \longrightarrow 1$ et $1 \longrightarrow 0$ est identique à la table de vérité de la négation du calcul propositionnel classique : on parlera ainsi de porte NOT. Les circuits sont usuellement construits à partir d'un *ensemble universel de portes logiques*, soit un ensemble fini de portes logiques permettant d'effectuer n'importe

dance de notre conception de la complexité à des hypothèses empiriques implicites.

quelle opération sur une suite de bits.

La définition de la calculabilité et de la complexité d'un problème doit être adaptée pour les circuits, puisqu'un circuit donné ne peut traiter que des entrées d'une taille n fixée, et non des entrées de taille arbitraire. On définit donc calculabilité et complexité par rapport à une *famille uniforme de circuits*. Une famille uniforme de circuits est une suite de circuits, $C_1, C_2, ..., C_n$ avec C_n le circuit prenant des entrées de taille n, telle qu'il existe une machine de Turing permettant d'engendrer la description de tout circuit C_n sur la donnée de n.

La *profondeur* d'un circuit est définie comme le plus long chemin, quantifié en nombre de portes logiques, menant d'une porte d'entrée à une porte de sortie. La taille d'un *circuit* est le nombre de portes qu'il comprend. On peut mesurer la complexité d'un circuit par deux fonctions de la taille des entrées, une fonction de complexité en profondeur et une fonction de complexité en taille. Pour donner un exemple de classe de complexité qui nous servira par la suite, BPP est la classe des problèmes de décision ou langages L qui peuvent être résolus par une famille uniforme de circuits booléens de taille polynômiale en la taille des données, qui accepte une chaîne de caractères avec une probabilité $3/4$ au moins si la chaîne est dans L, et les rejette avec une probabilité au moins $3/4$ s'ils ne sont pas dans le langage.

Comme on peut le voir aisément, le modèle des circuits booléens est un modèle de calcul effectif. La description de chaque circuit peut être engendrée par une procédure effective, et chaque opération élémentaire sur les bits est effectuable par un calcul papier-crayon.

Dans le modèle des circuits quantiques, chaque bit est encodé dans un vecteur de base d'une observable donnée d'un système vivant dans un Hilbert de dimension 2, $\{|0\rangle, |1\rangle\}$, nommée *base computationnelle*. Chaque système quantique porteur d'une information binaire est nommé *qubit*. Les entrées et sorties du calcul sont donc encodées par une suite de qubits, qui vit dans le produit tensoriel des espaces des états de chacun des qubits : une entrée ou une sortie de taille n est donc encodée dans le vecteur $|\phi_1, \phi_2, ..., \phi_n\rangle$, où chaque $|\phi_i\rangle$ appartient à la base computationnelle. L'exécution du calcul doit être assurée par l'évolution dynamique des qubits. Par le second postulat de la mécanique quantique, celle-ci est représentée par une transformation unitaire sur l'espace des états du qubit. La lecture du résultat final du calcul doit être assurée par l'exécution d'une mesure projective dans la base computationnelle. L'exécution de la transformation unitaire est décomposée en une suite de transformations unitaires élémentaires, ou *portes quantiques*, choisies parmi un ensemble fini fixé.

La conception de l'étape élémentaire de calcul sépare le modèle des circuits quantiques de tout modèle d'une procédure effective. L'action d'une transformation unitaire sur un qubit n'est pas nécessairement reproductible par un calcul papier-crayon. Ainsi l'action d'une porte Hadamard H sur les

vecteurs de la base computationnelle :

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}; H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{8.1}$$

n'est pas reproductible par un calcul papier-crayon. Il sera impossible à une feuille de papier d'être en superposition d'états où elle porte un 1 et d'un état où elle porte un 0. Le calcul papier-crayon permet de produire une simulation entrées-sorties, puisqu'il est possible de calculer le résultat de la transformation unitaire par un calcul matriciel : c'est d'ailleurs ce qui garantit le respect de la thèse de Church-Turing physique par le modèle des circuits quantiques. Mais cette opération matricielle ne pourra plus compter comme une opération élémentaire, comme c'est le cas pour l'exécution de la porte Hadamard. C'est justement cette différence dans la conception des étapes élémentaires de calcul qui fait du calcul quantique un candidat possible à un pouvoir calculatoire exceptionnel.

Dans le modèle des circuits quantiques, le temps de calcul est défini comme la profondeur du circuit nécessaire à la résolution du problème, en fonction de la taille de l'entrée. D'un point de vue théorique, cette définition naît d'une analogie avec la théorie des circuits classiques. D'un point de vue physique, définir le temps de calcul comme une suite de transformations unitaires suivie d'une mesure projective pose apparemment problème, comme le remarque avec ironie J. Bub ([49], 125-134) :

> Since any sequence of unitary transformations is equivalent to a single unitary transformation, and a unitary transformation followed by a measurement in a certain basis is equivalent to simply performing a measurement in a different basis, any quantum computation can always be reduced to just one step : a measurement in a particular basis !

En interprétant nos définitions de manière naïve, on pourrait donc dire que tout circuit quantique calcule la solution du problème en une unique étape. Mais le jeu d'équivalences théoriques servant de base à cette mésinterprétation est dépourvu de sens computationnel. Comme nous venons de le voir, l'utilisation de notre système quantique comme calculateur dépend du choix d'un encodage des bits $\{0, 1\}$ dans la base d'une observable donnée, dite *base computationnelle*. En changeant de base de mesure finale, on perd donc trace de l'encodage du résultat dans une base donnée. Pour retrouver trace de cet encodage, l'observateur fini sera obligé de calculer par une procédure effective le changement de base effectué, ce qui peut représenter un calcul très complexe. Le calcul effectué par le circuit quantique n'est donc simplifié artificiellement qu'au prix d'un report de la charge de calcul sur le dos de l'utilisateur. D'un point de vue intuitif, la définition permettant d'effectuer tout calcul quantique en une étape n'est pas pertinente pour l'implémentation, parce qu'elle donne une estimation radicalement faussée du nombre de

manipulations de l'information nécessaires pour accéder au résultat, et donc du temps réel de l'exécution du calcul.

Pour obtenir une définition pertinente, il importe donc de définir une suite d'opérations préservant l'encodage des données. On définit un *ensemble universel de portes quantiques* de manière analogue au cas classique : un ensemble fini de portes quantiques est universel ssi toute transformation unitaire peut être approximée à une précision arbitraire par un circuit quantique ne comprenant que ces portes. La convention usuelle du modèle des circuits quantiques impose donc :

- Que les transformations unitaires élémentaires soient choisies parmi un ensemble fini formant un *ensemble universel de portes quantiques.*
- Que toutes les mesures soient effectuées dans la base computationnelle.

Les circuits classiques et les circuits quantiques sont tous deux des modèles de calcul en temps discret. Mais la conception d'une étape élémentaire de calcul quantique n'est plus une manipulation de bits simulable par un calcul papier-crayon, mais une transformation unitaire d'un qubit ou une mesure projective obéissant aux contraintes que nous venons d'énoncer. L'hypothèse d'un taux de traitement borné de l'information, si elle doit être formulée pour cette notion quantique d'étape de calcul, est soumise à des contraintes différentes de celles énoncées pour le calcul classique. Il ne peut ainsi être question de définir une étape de calcul comme une rotation de $\frac{\pi}{2}$ d'un état, comme proposé par N. Margolus, L. Levitin et S. Lloyd (voir section 6.1). Parmi les portes quantiques pouvant figurer dans un ensemble universel, on peut ainsi trouver la porte Hadamard, qui correspond à une rotation de $\frac{\pi}{4}$ des vecteurs de la base computationnelle. Les arguments valables pour une forme de l'hypothèse de traitement borné ne valent donc pas automatiquement pour l'autre forme. Dans le cas classique, l'hypothèse assigne des bornes à l'intervalle de temps réel d'exécution de toute transformation sur des bits. Dans le cas quantique, une hypothèse de traitement borné de l'information assignerait des bornes à l'intervalle de temps réel d'exécution de toute transformation unitaire appartenant à un ensemble universel.

À notre connaissance, il n'existe pas à l'heure actuelle de définition physique de la notion d'étape élémentaire de calcul qui permette d'embrasser tous les modèles de calcul à temps discret. L'hypothèse d'un taux de traitement de l'information bornée, si elle s'impose comme un fondement physique naturel de tous les modèles de calcul à temps discret, n'admet pas à l'heure actuelle de formulation rigoureuse en termes physiques.

L'hypothèse d'un taux de traitement borné de l'information, si informelle soit-elle, s'impose à tout modèle à temps discret raisonnable, qu'il soit effectif ou non. C'est la raison pour laquelle il n'est pas pertinent de distinguer une forme algorithmique d'une forme physique de la thèse de Church-Turing étendue. L'hypothèse du taux de traitement borné de l'information s'impose aussi bien aux modèles discrets effectifs qu'aux modèles discrets non-effectifs du calcul : elle est la garantie fondamentale de la proportionnalité des res-

sources théoriques aux ressources observables, et donc de la pertinence pour l'implémentation. Une fois admise cette hypothèse, la simulation d'un modèle par l'autre est l'affaire d'une étude purement mathématique. Il n'est donc point besoin d'énoncer deux thèses autonomes sur les modèles du calcul effectif et du calcul non-effectif discret.

Pour les modèles à temps et espace discrets, qu'ils soient modèles du calcul effectif ou non, la nécessité d'une hypothèse empirique comme le taux borné de traitement de l'information est rendue manifeste par le contraste entre, d'une part, la modélisation discrète du temps et de l'espace au sein du modèle de calcul, et, d'autre part, la modélisation continue des intervalles de temps réel écoulés et de l'espace occupé dans le laboratoire.

Dans le cas des modèles analogiques à temps et espace continus, la relation entre représentation théorique du temps et intervalle de temps réel peut sembler bien plus directe. Le temps continu du calcul peut être une variable du modèle, et l'information est encodée dans des valeurs continues qui réfèrent moralement à des quantités observables. Mais la notion d'étape discrète de calcul n'ayant plus de sens, il est nécessaire de redéfinir le concept de fonction de complexité pour ces modèles. Bien que les machines analogiques soient historiquement plus anciennes que les machines numériques modernes, l'étude de la définition des concepts de base de la complexité, comme la mesure de la taille de l'entrée et la définition du temps de calcul comme une fonction de cette taille, est extrêmement récente pour de tels modèles(voir [53]).

En sus de raisons historiques contingentes, ceci peut être expliqué par le caractère problématique, pour les modèles à temps et espace continus, de la définition même de la complexité comme fonction de la taille des entrées. La taille des entrées ne peut en effet plus être mesurée par le nombre de symboles nécessaires à leur encodage, puisque ces entrées ne sont plus encodées de manière discrète. Avant même de songer à une comparaison avec les modèles de complexité discrets, la définition des mesures de complexité en calcul analogique pose donc de nombreux problèmes conceptuels, qui ont fait l'objet de travaux scientifiques extrêmement récents. La formulation d'une thèse de Church-Turing étendue pour les modèles analogiques a ainsi fait l'objet de travaux d'O. Bournez, qui a défini et démontré une relation de simulation efficace entre G.P.A.C. et machines de Turing [46]. Nous ne pourrons donc que demander l'indulgence du lecteur, si nous ne pouvons, dans le cadre de cette thèse, formuler un commentaire philosophique satisfaisant de ces travaux scientifiques en cours.

Enfin, non seulement la définition des ressources, mais la liste même des ressources pertinentes, dépend de considérations empiriques. Lorsqu'on évoque les ressources computationnelles, on songe spontanément au temps et à l'espace. L'intuition préthéorique selon laquelle ces deux ressources constituent la liste exhaustive des ressources pertinentes pour le calcul provient de notre expérience du calcul papier-crayon. Mais rien ne garantit qu'un mo-

dèle non-standard de calcul, pour être justement compris, n'exige de définir d'autres ressources computationnelles. Dans [31], E. Blakey a montré l'existence d'un algorithme de factorisation dans un modèle analogique original basé sur un schème d'interférence électromagnétique. Ce modèle aurait la propriété remarquable de factoriser tout nombre entier en temps constant, mais avec une exigence de précision en la préparation du système et en la mesure du résultat exponentielle en n. L'intérêt de ce modèle théorique n'est pas de constater qu'il ne constitue pas un candidat réaliste à la factorisation efficace de nombres entiers, mais que ce caractère irréaliste ne peut être compris qu'en quantifiant une autre ressource que le temps. E. Blakey en tire la conclusion que dans un tel modèle analogique, la précision de la mesure doit être elle aussi considérée comme une ressource. Il n'est pas sensé de considérer qu'on a défini un algorithme efficace lorsqu'un algorithme est efficace en une ressource, mais possède une exigence impossible à satisfaire en une autre. Pour traiter ce problème, E. Blakey a proposé d'introduire la notion de prédominance (*dominance*) d'une ressource pour désigner une ressource dont la mesure de complexité pour un algorithme devient asymptotiquement très grande par rapport aux autres ressources, et les rend ainsi négligeables dans l'évaluation des coûts d'un algorithme [6]. Ce concept pourrait se montrer utile dans l'étude des calculs ayant des coûts significatifs en plusieurs ressources, ainsi que pour la comparaison des coûts induits par différents modèles de calcul.

Au-delà des analogies de surface avec la calculabilité, l'examen plus attentif de la complexité computationnelle révèle de nombreux enjeux spécifiques. L'hypothèse de pertinence pour l'implémentation impose que les mesures de complexité définies par la théorie modélise la consommation de ressources observables par le calculateur. Pour que cette exigence soit satisfaite, et qu'ainsi les mesures de complexité produites au sein de modèles d'inspirations très diverses puissent être comparées dans un cadre commun, une hypothèse empirique doit être satisfaite, que nous avons appelée *hypothèse de proportionnalité des ressources théoriques aux ressources observables.* Il n'est donc pas pertinent de séparer une forme algorithmique d'une forme physique de la thèse de Church-Turing étendue, car toutes deux reposent sur la même hypothèse empirique. Si cette hypothèse était fausse, les résultats obtenus au sein d'un modèle du calcul effectif comme la machine de Turing, et d'un modèle physiquement défini comme les circuits quantiques, perdraient tous deux leur pertinence pour l'implémentation. Dans ces circonstances, il serait erroné de présenter ces deux thèses comme des propositions autonomes.

Si l'on ajoute à cette dernière remarque celles d'Ed Blakey que nous venons de commenter, on peut donc dire que non seulement la définition raisonnable, mais l'établissement d'une liste exhaustive des coûts computationnels et la comparaison entre elles des ressources supposent, sinon de satisfaire à

6. Voir ([31], 26-27) pour plus de détails sur cette notion.

des hypothèses empiriques explicites, du moins de partir d'un examen du processus empirique implémentant le calcul. La théorie de la complexité ne peut donc paraître comme une science mathématique pure qu'au prix de la négligence de l'authentique travail de modélisation qui garantit sa pertinence.

8.3 Modèles de calcul non-standard et interprétation empirique de la complexité

À l'heure actuelle, il existe cependant des modèles de calcul non-standard pour lesquels il n'existe pas de simulation efficace démontrée avec une machine de Turing. Certains de ces modèles sont quasi-unanimement considérés comme irréalistes ; d'autres, bien que réalistes, ne possèdent pas, selon toute vraisemblance, de propriétés exceptionnelles de complexité ; pour d'autres enfin, notamment le calcul quantique, la recherche est en cours. Dans tous les cas de figure, autant que le caractère réaliste ou irréaliste du modèle, ce sont les raisons mises en avant pour justifier ce jugement qui font l'objet de notre intérêt.

Un modèle de calcul non-standard peut parfaitement être qualifié d'irréaliste, sans qu'aucune considération empirique sophistiquée n'entre en jeu. Ainsi A. Shamir, l'inventeur du protocole de cryptographie R.S.A. dont la sûreté est fondée sur la difficulté de la factorisation, a démontré que dans un modèle non-standard la factorisation prenait seulement $0(\ln n)$ étapes [203]. Ces performances extraordinaires au niveau théorique sont aisément expliquées par la définition abstraite du coût unitaire en temps, qui considère toutes les opérations arithmétiques (addition, soustraction, multiplication et division entières) comme des opérations élémentaires. A. Shamir ne souffrait d'aucune illusion à ce sujet, et qualifiait l'intérêt de son modèle de purement « théorique » [7]. Aucune connaissance de physique théorique n'entre en jeu pour comprendre le caractère irréaliste d'un tel modèle, seulement une pratique élémentaire du calcul papier-crayon. Pour discuter l'interprétation empirique de la thèse de Church-Turing étendue, il faut nous concentrer sur les modèles de calcul où des arguments empiriques plus sophistiqués jouent, ou semblent jouer un rôle important dans la qualification du modèle comme réaliste ou irréaliste, et préciser en quel sens exactement ces arguments peuvent être qualifiés d'empiriques.

7. Le qualificatif de « théorique »doit bien être compris ici par opposition à « réaliste », comme le montre la citation suivante (*ibid.*, p. 28) :

> We develop an algorithm which finds a non-trivial factor of a composite number n in $O(logn)$ arithmetic steps, and we conjecture that it is optimal. This result does not imply that natural numbers can be factored in polynomal time, since our measure of complexity ignores the size of the numbers involved. The algorithm presented in this paper is thus mainly of theoretical interest (...).

L'interprétation empirique de la complexité peut être appuyée sur l'argument contrefactuel suivant : « si tel état de chose avait été différent de ce qu'il s'est avéré être, alors il aurait été possible d'implémenter un modèle au pouvoir calculatoire extraordinaire. »Par l'expression « modèle au pouvoir calculatoire exceptionnel », nous signifierons le plus souvent un modèle violant la thèse de Church-Turing étendue, mais on pourra trouver des modèles proposant d'autres performances encore, violant les conjectures les plus solides de la théorie de la complexité, comme les modèles permettant la solution efficace d'un problème NP-complet.

Tout argument de forme contrefactuelle joue en faveur de l'interprétation empirique de la complexité, en tant qu'il exhibe une dépendance des résultats de complexité à des considérations empiriques. Comme nous allons le voir immédiatement, les « états de chose »mentionnés varient grandement en fonction du modèle étudié, donnant à chaque fois une forme spécifique à notre argument contrefactuel générique.

On retrouve évidemment ici l'argumentation contrefactuelle en faveur de l'interprétation empirique que nous avons déjà rencontrée lors de notre examen de la calculabilité (voir chapitre 6, Conclusion). Encore une fois, cet argument permet de discuter le caractère empirique de la thèse de Church-Turing étendue, indépendamment de sa vérité, en exhibant la dépendance de nos croyances en complexité envers certaines hypothèses empiriques. Nous allons voir à présent comment cet argument se décline dans le cas de la complexité, en passant en revue divers modèles non-standard, et en suivant une gradation du moins au plus plausible. Comme dans le cas de la calculabilité, et pour les mêmes raisons, nous nous restreindrons aux modèles physiquement définis. Une excellente revue de grand nombre de ces modèles ayant déjà été réalisée par S. Aaronson dans [3], je vais me concentrer sur leur apport à la discussion sur le statut empirique de la théorie de la complexité, en examinant quelques exemples significatifs.

8.3.1 Pouvoir calculatoire exceptionnel : quelques exemples de modèles non-standard

Certains modèles sont basés sur une variante hétérodoxe d'une théorie admise. S. Lloyd et D.S. Abrams ont ainsi montré que les variantes non-linéaires de la mécanique quantique permettraient le calcul de tout problème NP-complet ou #P [8] en temps polynômial [8]. D'autres s'appuient sur des résultats théoriques généralement rejetés comme dépourvues de sens physique, comme l'existence de courbes fermés de type temps *(Closed Timelike Curves, ou* C.T.C.) dans la relativité générale. S. Aaronson et D. Watrous ont ainsi montré qu'en présence d'une courbe fermée de type temps, la classe des problèmes solubles efficacement par un ordinateur quantique, BPQ_{CTC},

8. La classe #P contient les problèmes d'énumération des solutions d'un problème dans P. Elle est conjecturée être une classe extrêmement vaste, contenant strictement NP.

est égale à la classe des problèmes solubles efficacement par une machine de Turing, P_{CTC}, et à $PSPACE$: $P_{CTC} = BPQ_{CTC} = PSPACE$. De tels modèles ne sont pas nécessairement proposés par des partisans d'une physique hétérodoxe, mais peuvent être au contraire explicités par des sceptiques, qui considèrent que la possibilité de telles performances computationnelles ne fait qu'augmenter l'implausibilité de tels modèles [9]. Ainsi, D. Abrams et S. Lloyd défendent l'intérêt de leur modèle en utilisant explicitement un argument contrefactuel ([8], 3992) :

> In particular, we show that it is possible to exploit nonlinear time evolution so that the classes of problems NP and #P (including oracle problems) may be solved in polynomial time. An experimental question - that is, the exact linearity of quantum mechanics - could thereby determine the answer to what may have previously appeared to be a purely mathematical one. This Letter therefore establishes a new link between physical law and the theoretical power of computing machines.

Un autre modèle non-standard, plus éloigné de la physique théorique, a fait parler de lui pour ses performances supposées en complexité. Il s'agit du calcul avec des brins d'ADN, initié par L. M. Adleman [12]. Le problème considéré par Adleman est le problème NP-complet du circuit hamiltonien : étant donné un graphe orienté à n sommets, et deux sommets distingués de ce graphe, s_{in} et s_{out} : existe-t-il un circuit entre ces deux sommets passant par chaque arc du graphe ?

Pour implémenter un algorithme de force brute sur un brin d'ADN, on utilise une propriété remarquable de ces brins, à savoir la complémentation de Watson-Crick. Le lecteur saura probablement que la protéine d'ADN est composée de quatre nucléotides de base : l'adénine A, la thymine T, la cytosine C et la guanine G. Selon le principe de complémentation, la base d'un nucléotide peut interagir avec la base d'un autre nucléotide pour former une liaison, mais les paires pouvant être appariées sont réduites à deux possibilités : {A, T}, {C, G}. On encode chaque sommet i par une séquence d'ADN O_i. L'arc $i \to j$ est encodé par la chaîne complémentaire de la chaîne encodant a et b. On remarque que cet encodage maintient l'information d'orientation de l'arc. Il devient alors possible de trouver la solution du problème de circuit hamiltonien en employant l'algorithme suivant [10] :

9. On peut ainsi lire sous la plume de D.S. Abrams et S. Lloyd ([8], 3995) :

> (...)we would like to note that we believe that quantum mechanics is in all likelihood exactly linear, and that the above conclusions might be viewed most profitably as further evidence that this is indeed the case.

10. Pour plus de détails, notamment sur les manipulations expérimentales permettant l'implémentation de chaque étape, voir [185]. Ces manipulations biochimiques ne sont ici que mentionnées, à l'usage du lecteur déjà familier de ces problématiques. Leur exposé détaillé aurait pris une place démesurée pour les fins du présent travail.

- Engendrer de manière aléatoire l'ensemble des circuits possibles à travers le graphe orienté. **On mélange un ensemble de séquences d'ADN représentant tous les sommets et tous les arcs, qui vont se combiner en des doubles séquences encodant des chemins aléatoires.**
- Sélectionner les chemins commençant par s_{in} et finissant par s_{out}. **On amplifie le résultat de l'étape 1 par PCR (*Polymerase Chain Reaction*), en utilisant les séquences de base O_0 et le complément de 0_{n-1}. On n'amplifie ainsi que les chemins désirés.**
- Sélectionner les chemins de longueur n. **On sélectionne les brins de la longueur désiré, on amplifie à nouveau ce produit par PCR et on le purifie avec un gel.**
- Sélectionner les chemins passant par l'intégralité des n sommets au moins une fois. **On sépare les doubles brins produits par l'étape 3.**
- S'il demeure des chemins, terminer et répondre *Oui*. Sinon, répondre *Non*. **On utilise l'attraction entre base complémentaire pour sélectionner les brins contenant le codage de tous les arcs. Puis on détecte les molécules d'ADN restantes.**

Adleman a implémenté un tel algorithme pour $n = 7$. L'implémentation sur une seule instance, qui plus est de taille réduite, ne permet pas de se prononcer sur l'existence d'un pouvoir calculatoire exceptionnel du modèle. Dans [130], J. Hartmanis a souligné que l'implémentabilité d'un tel modèle serait sujette à d'insurmontables difficultés, dues à la complexité en espace de l'algorithme. Si on définit cette dernière comme le nombre de molécules nécessaire à l'exécution de l'algorithme, la première étape de l'algorithme suppose d'engendrer l'intégralité de l'espace de recherche. Or un problème NP-complet comme le circuit hamiltonien possède un espace de recherche gigantesque. Ainsi, pour un graphe orienté à n sommets, le total des arcs est de $log(n)^n$. Si chaque sommet est encodé par une séquence d'ADN de longueur constante c, on aura un total de $cnlog(n)^n$ molécules. Avec 70 sommets, et en considérant le poids moyen d'une molécule, la masse totale des molécules nécessaire à l'exécution de l'algorithme serait de l'ordre de 10^{25} kg. A titre de comparaison, la masse de la Terre est de l'ordre de $6.10^{24} kg$. Avec des tailles d'instance encore plus grandes, de l'ordre des plus grandes instances solubles sur un ordinateur conventionnel, on arrive à un total de 10^{70} molécules, le nombre de baryons dans l'Univers observable étant estimé être de l'ordre de 10^{80}. L'inefficacité de l'algorithme en termes d'espace condamne toute velléité d'exploiter une éventuelle efficacité en temps. On retrouve ici la remarque de Y. Gurevich : la faisabilité à la Cobham-Edmonds est une conception insuffisante de la faisabilité, tant qu'elle ne prend pas en compte toutes les ressources pertinentes.

Une autre classe de modèles part d'un constat simple. La physique théorique affirme que tout système non-forcé tend à rejoindre un état d'équilibre

minimisant l'énergie du système. On peut donc voir un système physique comme une machine calculant un problème d'optimisation. Si l'on prend cette analogie au sérieux, on peut envisager de calculer une instance d'un problème d'optimisation en l'encodant dans l'état d'un système et en utilisant son évolution naturelle vers l'équilibre pour en obtenir la solution. Même s'il existe des cas où le retour à l'équilibre peut prendre un temps géologique, un choix judicieux de système pourrait nous permettre de calculer en un temps très court des instances de problèmes difficiles. Il ne s'agit pas là d'une simple vue de l'esprit, puisque il est ainsi possible d'encoder le problème 3-SAT dans un verre de spins avec interaction à plusieurs spins (*spin glass with multi-spin interaction*), et d'appliquer ainsi des approches de mécanique statistique à la résolution d'instances d'un problème NP-complet [11].

Le problème essentiel posé par une telle idée est que rien dans la physique théorique ne garantit que, si le dispositif fonctionne convenablement pour certaines instances, il fonctionnera de manière satisfaisante pour toutes les instances du problème. Le système peut rester figé à un minimum local au lieu d'atteindre un minimum global, ou peut connaître des temps de relaxation arbitrairement longs.

Comme on le voit à la lecture de cet argument, la proposition initiale de résoudre un problème d'optimisation par encodage dans l'état d'un système tendant vers l'équilibre ne va à l'encontre d'aucun principe physique établi. Aucun principe de physique théorique, à notre connaissance, n'affirme qu'il doit exister pour tout système encodant un problème des circonstances où le retour à l'équilibre doit prendre un temps arbitrairement long. Le reproche fait serait plutôt que pour garantir son bon fonctionnement en toute généralité, le modèle de calcul proposé devrait prouver un nouveau résultat physique, à savoir l'existence de systèmes jouissant d'une propriété de « retour à l'équilibre efficace »en toute circonstance.

En renversant la perspective théorique, le physicien théoricien peut alors utilisé la théorie de la complexité comme principe heuristique. Si jamais un modèle théorique autorise une évolution dynamique encodant un problème NP-complet, alors cette évolution dynamique doit admettre un grand nombre de minimum locaux, ou des temps de relaxation arbitrairement longs [12].

Dans le domaine du calcul quantique, Fahrii et alii ont montré la possibilité d'utiliser le calcul quantique adiabatique pour résoudre certaines instances de problème NP-complet en temps polynômial [94]. Un tel argument ne suffit cependant pas à prouver un pouvoir calculatoire exceptionnel, puisque un problème NP-complet peut parfaitement admettre nombre d'instances faciles. Van Dam et alii ont montré qu'il était improbable qu'on puisse appliquer une telle méthode à toutes les instances d'un problème NP-complet

11. Pour plus de détails sur ces idées, voir [76].

12. Pour plus de détails sur cette idée, et d'autres exemples d'encodage de problèmes NP-complets dans des systèmes physiques, voir [76].

[229]. Sous certaines hypothèses, le modèle de calcul adiabatique nécessiterait la mesure de différences spectrales (*spectral gaps*) arbitrairement petites. On peut donc retrouver le problème de la précision arbitraire au niveau de la complexité, même si l'on voit bien que cet argument est loin d'être aussi récurrent que dans le cas de la calculabilité.

Qu'apporte l'étude des modèles de calcul non-standard à notre discussion de l'interprétation empirique de la complexité ? Pour les modèles où les ressources computationnelles ne sont pas encore rigoureusement définies, il est difficile de se prononcer sur le caractère empirique ou non des considérations menant à leur acceptation ou rejet. Si ces modèles font l'objet d'un développement plus poussé, il serait intéressant de voir si des considérations empiriques interviennent dans la définition même des ressources computationnelles. Pour les autres modèles, il semble que leur rejet, ou à tout le moins le scepticisme exprimé à leur égard, soient motivés par des considérations empiriques. L'existence de temps de relaxation arbitrairement longs, ou de minimums locaux ; l'absence de précision arbitraire dans les résultats de mesure ; le poids d'un nucléide : toutes ces considérations peuvent bien être qualifiées d'empiriques. Cette brève revue de quelques modèles non-standard de la complexité permet déjà de souligner l'absence d'unicité des considérations menant à la disqualification des modèles aux performances exceptionnelles. Il n'existe pas de problème récurrent, comme l'exigence d'une précision arbitraire en calculabilité, qui appellerait comme réponse unique en un principe physique, comme le principe holographique. Ceci permet de souligner *a posteriori* le caractère remarquable de cette récurrence d'un même problème dans des modèles d'hypercalcul fort divers.

Comme nous l'avons vu en étudiant l'interprétation empirique de la calculabilité, un argument contrefactuel n'est parfaitement convaincant que s'il prend en antécédent la possibilité d'un modèle de calcul physiquement défini. Les deux premiers modèles que nous avons examinés étaient des modèles physiquement définis, mais bâtis sur des propositions théoriques considérées avec scepticisme par leurs propres auteurs. Nous allons finir cette revue des modèles de calcul non-standard doués d'un pouvoir calculatoire exceptionnel par un modèle à la fois physiquement défini et conçu comme un candidat authentique à la violation de la thèse étendue, à savoir le calcul quantique.

8.3.2 Un modèle privilégié : le calcul quantique

Le calcul quantique est une proposition parmi d'autres de modèles de calcul violant la thèse de Church-Turing étendue. Son caractère réaliste fait encore débat à ce jour dans les communautés scientifiques concernées. À ce titre, il a un statut identique aux modèles que nous venons d'étudier.

Néanmoins, ce modèle de calcul non-standard fait l'objet d'une attention scientifique sans commune mesure avec tous les autres modèles précédemment cités. Il est l'un des sujets d'étude d'une sous-discipline entière, la

théorie de l'information et du calcul quantique, et des équipes entières de recherche travaillent à la construction d'un éventuel ordinateur quantique, et à l'étude des propriétés théoriques de ce modèle. En tant que tel, le modèle du calcul quantique mérite également un traitement privilégié de notre part.

D'un point de vue historique, c'est la publication par P. Shor d'un algorithme quantique de factorisation efficace qui fit exploser l'attention accordée aux modèles de calcul quantique. Cette nouvelle attention semble justifiée par au moins trois raisons. La première est qu'en l'état actuel de nos connaissances, il n'existe aucun algorithme efficace de factorisation dans un modèle classique du calcul. Si l'on conjecture qu'il n'en existe pas, alors l'algorithme de Shor constitue bien une violation de la thèse de Church-Turing effective. La seconde est que le modèle de calcul utilisé n'est appuyé ni sur une physique contrefactuelle, ni sur des modèles au statut imprécis, mais au contraire sur l'une parmi les mieux confirmées de nos théories de la nature. La troisième est qu'on a pu démontrer que d'autres algorithmes quantiques, comme ceux de Deutsch-Jozsa ou Grover, permettent de réaliser des performances calculatoires inatteignables par une machine de Turing. L'algorithme de Grover permet ainsi de trouver un élément dans une liste sans structure en temps $O(\sqrt{n})$, alors que l'algorithme démontré optimal sur une machine de Turing prend un temps $O(n)$. La supériorité du calcul quantique sur le calcul classique a donc déjà été démontré, mais dans un sens qui ne remet pas en cause la thèse de Church-Turing étendue, puisqu'on parle ici d'une accélération quadratique. Ces trois raisons ont conféré à l'algorithme de Shor à la fois un intérêt et une plausibilité sans commune mesure avec les autres modèles de calcul au pouvoir exceptionnel.

Il faut immédiatement souligner que le caractère réaliste du modèle quantique est une question ouverte. Les prototypes d'ordinateur quantique bâtis à ce jour ne permettent de manipuler que des données de petite taille, les projets les plus ambitieux annonçant des ordinateurs d'une mémoire de 512 qubits [55]. Avec des instances de si petite taille, il est impossible de démontrer expérimentalement que l'ordinateur quantique tient bien ses promesses en matière de complexité. La question centrale du calcul quantique est donc la suivante : est-il possible de construire un ordinateur quantique de grande taille (*large scale quantum computer*), qui permettra de jouir des avantages supposés de ce modèle en matière de complexité ? Pour reprendre la terminologie que nous avons introduite plus haut (voir section 5.2), la question est ouverte de savoir si le calcul quantique est un modèle trivial en mémoire.

Le scepticisme à l'égard du calcul quantique consiste à affirmer qu'une telle chose est impossible.

L'interprétation empirique et le scepticisme à l'égard du calcul quantique

Pour discuter de la pertinence du débat autour du calcul quantique pour l'interprétation empirique de la complexité, ce n'est pas tant encore une fois l'existence d'arguments *pro* et *contra* qui nous intéresse, que la nature de ces arguments. Je distinguerai donc deux formes de scepticisme à l'égard du calcul quantique. Le premier, ou scepticisme physique, est fondé sur des arguments issus de la physique théorique.

Ces arguments sceptiques peuvent être basés sur une critique explicite de la mécanique quantique. Si le calcul quantique ne pourra fournir de pouvoir calculatoire supplémentaire, c'est simplement parce que la physique quantique sous-jacente à ce modèle se révélera fausse à l'échelle de précision requise pour son implémentation. Comme le souligne à juste titre S. Aaronson ([2], 122), il faut encore montrer que les défauts prétendus de la théorie quantique réduiront le pouvoir calculatoire du modèle. Comme nous l'avons vu ci-dessus avec le travail de D.S. Abrams et S. Lloyd, une interprétation hétérodoxe de la théorie quantique pourrait mener à plus, et non à moins, de pouvoir calculatoire que la théorie telle qu'elle est présentement acceptée.

La position de G. t'Hooft doit être considérée comme une forme de scepticisme physique ([140],12-13). Au niveau de précision requis pour l'implémentation d'un ordinateur quantique, une théorie déterministe se substituera à la théorie quantique. Cette théorie annulera tout avantage comparatif que l'ordinateur quantique aurait été censé avoir par rapport à un ordinateur classique :

> Quantum mechanics, as we know it, leads to many more phenomena that are at odds with classical determininistic descriptions. An example of this is the so-called quantum computer [...]. Using quantum mechanics, a device can be built that can handle information in a way no classical machine will ever be able to reproduce, such as the determination of the prime factors of very large numbers in an amount of time not much more than what is needed to do multiplications and other basic arithmetic with these large numbers. If our theory is right, it should be possible to mimick such a device using a classical theory. This gives us a falsifiable prediction :
>
> *It will never be possible to construct a 'quantum computer' that can factor a large number faster, and within a smaller region of space, than a classical machine would do, if the latter could be built out of parts at least as large and as slow as the Planckian dimensions.*
>
> [...] Quantum computers are known to suffer from problems such as 'decoherence'. Often, it is claimed that decoherence is nothing but an annoying technical problem. In our view, it will be one

> of the essential obstacles that will forever stand in the way of constructing super powerful quantum computers, and unless mathematicians find deterministic algorithms that are much faster than the existing ones, factorization of numbers with millions of digits will not be possible ever.

De manière très claire, le scepticisme à l'égard de l'ordinateur quantique est ici fondée sur des conjectures portant sur les principes régissant la physique à l'échelle pertinente, et mène à des prédictions testables.

A. Hagar formule également plusieurs objections de principe contre la possibilité du calcul quantique [127]. La croyance en la possibilité du calcul quantique est largement dépendante de la démonstration de l'existence de codes de correction d'erreur quantique. Ces codes de correction protègent le calcul quantique contre l'objection d'irréalisme adressée aux modèles ignorants les problèmes de bruit. A. Hagar formule trois objections distinctes à l'égard de la possibilité du calcul quantique, qui toutes tendent à critiquer la démonstration de principe de l'existence d'un calcul quantique protégé contre le bruit. L'une des objections d'A. Hagar peut être succintement résumée de la manière suivante. L'une des approches pour protéger l'état quantique encodant l'information contre le bruit consiste à le faire évoluer au sein d'un sous-espace invulnérable à la décohérence (*decoherence-free subspace*). A. Hagar formule à l'encontre d'une telle approche la conjecture sceptique suivante (*ibid.*, p.27) :

> Decoherence-free subspaces supporting quantum states that allow computational superiority become exponentially rare as the system's size increase.

Cette conjecture serait analogue à l'impossibilité d'un comportement réversible fréquent des macrosystèmes en thermodynamique. Elle peut être reformulée en termes computationnels (*idem*, p.30) :

> The physical computational complexity of keeping a computationnally superior quantum state in its noise-resilient (decoherence-free) subspace increases exponentially with the size of the input and affects the putative computational superiority such a state might afford.

A. Hagar admet qu'une telle reformulation en termes computationnels est purement heuristique, dans la mesure où la complexité computationnelle physique est une science en devenir.

Ces dernières remarques d'A. Hagar sont étrangement proches de celle d'un partisan du calcul quantique, à savoir S. Aaronson. Ce dernier a ainsi affirmé qu'un scepticisme bien fondé à l'égard du calcul quantique devait être capable de répondre à la question suivante : *quelles propriétés séparent les états quantiques que l'on peut créer, et qui rendent compte des résultats d'expérience connus, et les états de Shor, ou états quantiques nécessaires à*

l'implémentation de l'algorithme de Shor ? ([2], 5-6). Une théorie alternative n'a pas nécessairement à séparer de manière stricte états possibles et états impossibles. S. Aaronson envisage ainsi la possibilité de quantifier la difficulté de préparation d'un état quantique. Le scepticisme à l'égard du calcul quantique pourrait ainsi triompher en montrant que la difficulté de préparer un état de Shor croit exponentiellement avec la dimension de l'espace des états. Le sceptique du calcul quantique et son partisan peuvent donc se rejoindre dans le désir de voir apparaître une théorie de la complexité de certaines tâches physiques.

Cette forme de rencontre théorique a été utilisée par les partisans du calcul quantique comme une motivation supplémentaire pour poursuivre leur programme de recherche, sous la forme d'un argument « gagnant-gagnant ». Tenter de construire un ordinateur quantique serait une entreprise du plus grand intérêt, parce qu'elle aboutirait soit à montrer la faisabilité du calcul quantique, soit à montrer que la mécanique quantique est fausse, ou à tout le moins que notre compréhension de cette théorie n'était pas aussi aboutie qu'on pouvait le penser. Selon ces dernières considérations, la possibilité d'une solution efficace au problème de factorisation ne dépend pas de notre ingénuité en algorithmique, ou de nos connaissances en logique, mais de faits non-triviaux sur le monde physique, comme la possibilité de protéger contre la décohérence des systèmes multiparticules, la précision avec laquelle on peut contrôler une particule donnée, ou la difficulté de préparation d'un état quantique donné. Même si un ordinateur quantique de taille raisonnable se révèle impossible à bâtir, l'interprétation empirique des fondements de la théorie de la complexité en sortira confirmée. La réponse à une question de complexité -existe-t-il une solution efficace au problème de factorisation ?- aura dépendu en dernier recours de la résolution d'une question empirique ouverte -est-il possible de construire un ordinateur quantique capable de traiter des données de grande taille ? Que la réponse soit positive ou négative ne modifie en rien la dépendance de la théorie de la complexité envers des hypothèses empiriques.

Ce type de scepticisme à l'égard du calcul quantique est donc parfaitement compatible avec l'interprétation empirique de la complexité. L'argument « gagnant-gagnant »est donc une forme particulièrement forte de l'argument contrefactuel. La force particulière de cet argument provient de la nécessité, pour réfuter la possibilité du calcul quantique, de récolter de nouvelles informations sur le monde physique, par opposition à l'utilisation de principes physiques déjà bien établis, ou à l'emploi d'arguments sceptiques à l'égard de modèles déjà abondamment critiqués. En d'autres termes, le questionnement autour de l'implémentabilité du calcul quantique montre la dépendance de la thèse étendue à des questions empiriques ouvertes, et non à des questions empiriques déjà abordées dans le cadre d'autres débats. En acceptant que l'implémentabilité d'un ordinateur quantique de grande taille est une question expérimentale, on a déjà adopté l'interprétation empirique

de la complexité.

Pour le bien de notre discussion de cette interprétation, il est donc préférable de se concentrer sur une seconde forme de scepticisme. Celui-ci affirme que la croyance en la possibilité d'un pouvoir calculatoire exceptionnel du calcul quantique est fondée sur une mésinterprétation du formalisme de la théorie physique employée, qui mènerait à des pseudoprédictions dépourvues de sens empirique. Par analogie avec la position défendue par M. Davis, nous qualifierons cette position de *scepticisme méthodologique*. Une telle perspective est parfaitement compatible avec le rejet de l'interprétation empirique. C'est donc sur une argumentation en faveur du scepticisme méthodologique, due à l'informaticien L. Levin, que nous allons à présent nous pencher.

Le scepticisme méthodologique à l'égard du calcul quantique : les arguments de L. Levin

Certaines voix sceptiques ont pu ainsi défendre que le pouvoir calculatoire supposé des modèles quantiques serait une illusion fondée sur une mauvaise estimation du coût unitaire d'une opération. Ainsi, O. Goldreich a critiqué la convention usuelle dans le modèle des circuits quantiques, qui consiste à définir une transformation unitaire comme une opération élémentaire, affirmant qu'une telle définition menait à une estimation erronée des coûts computationnels. O. Goldreich n'a cependant pas développé cet argument, et je vais m'intéresser à un argument plus explicite en faveur du scepticisme méthodologique, formulé par L. Levin [13].

L. Levin compare le calcul quantique au modèle d'O. Shagrir, que nous avons déjà mentionné plus haut : celui-ci ne permettait des performances

13. L. Levin emploie, dans son argumentation contre la possibilité d'un ordinateur quantique, des arguments de nature diverse. Certains sont clairement de nature physique, comme le scepticisme portant sur la possibilité d'isoler suffisamment un ordinateur quantique de son environnement. Il va même jusqu'à admettre que déterminer les exactes raisons de l'impossibilité du calcul quantique serait riche d'enseignements pour les fondements de la physique ([160], section 2) :

> It is worth noting that the reasons why QC must fail are by no means clear ; they merit thorough investigation. The answer may bring much greater benefits in the understanding of basic physical concepts than any benefits factoring devices could promise. The present attitude is analogous to, say, Maxwell selling the Daemon of his famous thought experiment as a path to cheaper electricity from heat. If he did, much of insights of today's thermodynamics might be lost or delayed.

Face à la diversité des arguments employés, il est difficile de classifier la position d'ensemble de L. Levin comme relevant d'un scepticisme méthodologique ou d'un scepticisme physique. Il n'est pas non plus évident de concilier la citation précédente avec d'autres passages de son argumentation, où il semble explicitement attaquer l'argument « gagnant-gagnant », et affirmer que rien de nouveau pour la physique ne pourra émerger des débats sur le calcul quantique. En me désintéressant de ces questions doxographiques, je vais me pencher uniquement sur la part de l'argumentation de L. Levin qui relève du scepticisme méthodologique.

exceptionnelles qu'au prix d'une estimation complètement irréaliste du coût unitaire. Il est clair que si le calcul quantique n'offrait aucune performance supplémentaire en termes d'efficacité pour de telles raisons, cela ne nous apprendrait rien de fondamentalement nouveau sur la physique.

La première, et la principale objection de Levin est que le calcul quantique serait fondé sur une interprétation formaliste poussée à l'absurde des équations de la mécanique quantique. Pour que le calcul quantique soit possible, il faudrait que ces équations soient valides jusqu'à une précision totalement irréaliste pour toute théorie physique connue :

> The major problem is the requirement that basic quantum equations hold to multi-hundredth if not millionth decimal positions where the significant digits of the relevant quantum amplitudes reside. We have never seen a physical law valid to over a dozen decimals. Typically, every few new decimal places require major rethinking of most basic concepts. Are quantum amplitudes still complex numbers to such accuracies or do they become quaternions, colored graphs, or sick-humored gremlins ? I suspect physicists would doubt even the laws of arithmetic pushed that far. In fact, we know that the most basic laws cannot all be correct to hundreds of decimals : this is where they stop being consistent with each other !

On remarquera que dans ce bref passage, L. Levin emploie un argument basé sur une exigence de précision arbitraire, à la fois dans son interprétation métathéorique à la M. Davis ("We have never seen a physical law valid to over a dozen decimals."), et son interprétation fondée sur les limites des théories fondamentales courantes ("[...]we know that the most basic laws... consistent with each other.").

Tout en admettant l'une des prémisses de l'argument de Levin -à savoir qu'il est irréaliste de supposer qu'une loi physique puisse demeurer valide à une précision arbitraire- le partisan du calcul quantique pourrait aisément répondre que savoir exactement à quel niveau de précision la mécanique quantique cesse d'être valide est une question du plus haut intérêt pour la physique théorique. C'est précisément la base de l'argument « gagnant-gagnant ». La réponse de L. Levin consiste alors à affirmer que la construction d'un ordinateur quantique n'implique pas de tester les prédictions de la théorie quantique à des niveaux de précision inexplorés, mais de les tester à des niveaux *inexplorables* :

> Consider, however, this scenario. With few q-bits, QC is eventually made to work. The progress stops, though, long before QC factoring starts competing with pencils. The QC people then demand some noble prize for the correction to the Quantum Mechanics. But the committee wants more specifics than simply a nonworking machine, so something like observing the state of the

> QC is needed. Then they find the Universe too small for observing individual states of the needed dimensions and accuracy. (Raising sufficient funds to compete with pencil factoring may justify a Nobel Prize in Economics.)
>
> Let us make some calculations. In cryptography, the length n of the integers to factor may be a thousand bits (and could easily be millions.) By $*n$, I will mean a reasonable power of n. A 2^{*n}-dimensional space H has $2^{exp(*n)}$ nearly orthogonal vectors. Take a generic v in H. The minimal size of a machine which can recognize or generate v (approximately) is $K = exp(*n)$ — far larger than our Universe. This comes from a cardinality argument : 2^{*K} machines of K atoms. Let us call such v "megastates."
>
> There is a big difference between untested and untestable regimes. Claims about individual megastates are untestable. I can imagine a feasible way to separate any two QC states from each other. Yet, as this calculation shows, no machine can separate a generic QC state from the set of all states more distant from it than QC tolerates. So, what thought experiments can probe the QC to be in the state described with the accuracy needed ? I would allow to use the resources of the entire Universe, but not more !

Avant de discuter plus précisément l'argument de L. Levin, il faut rappeler un fait élémentaire sur la précision des prédictions physiques. Lorsqu'on rappelle, comme le fait L. Levin, que gagner quelques décimales en précision a historiquement exigé des modifications fondamentales de notre théorie, et qu'on ne connaît pas de théorie formulant des prédictions à plus de quelques décimales, on songe usuellement aux prédictions formulées sur des valeurs d'observables. Dans le cas de l'algorithme de Shor, c'est la précision dans la valeur d'amplitudes de probabilités -et, donc, d'un point de vue opérationnel, de statistiques- qui est l'objet du débat. Or, comme le fait remarquer S. Aaronson ([2], 3),

> (...) even classically, one can flip a coin a thousand times to produce probabilities of order 2^{1000}. Should one dismiss such probabilities as unphysical ?

En outre, l'attribution d'amplitudes de probabilité extrêmement petites est une composante nécessaire du formalisme quantique, dès lors qu'on considère des systèmes comprenant un grand nombre de particules. Pour reprendre l'exemple de S. Aaronson ([2], 6), lorsque 10000 photons sont polarisés à un angle de $\frac{\pi}{4}$, leur état est décrit par le vecteur de l'espace produit tensoriel $2^{-5000}(|0\rangle + |1\rangle)^{\otimes 10000}$. Sous sa forme actuelle, l'argument de L. Levin pourrait mener à un rejet du formalisme de la mécanique quantique comme une description satisfaisante d'un amas de particules, conséquence qu'on pourrait difficilement qualifiée de triviale.

Une autre objection de L. Levin serait que la description d'un « mégaétat »(*megastate*) nécessaire à la construction d'un ordinateur quantique est pratiquement impossible. Même en supposant qu'on encode chaque bit de sa description dans l'état d'un atome, la description exigerait plus d'atomes que l'Univers n'en contient. Il ne s'agit cependant pas là de l'unique quantité employée en physique qui ne puisse ainsi être écrite explicitement. Comme le remarque R. Penrose ([188], 444-446), la quantité d'entropie de l'Univers est elle aussi un nombre gigantesque, qu'on ne peut espérer écrire même en encodant chaque bit dans une particule de l'Univers connu. Ceci ne permet point de conclure que le calcul de l'entropie globale de l'Univers est dépourvu de sens physique. Encore une fois, l'argument ne permet pas de conclure sous sa forme présente.

Enfin et surtout, le principal problème de cet argument est qu'il porte sur les ressources nécessaires pour identifier un état générique. L'état de Shor n'est pas un état générique. C'est un état où l'une des branches de la superposition -celle encodant la réponse correcte au problème- a une probabilité d'être détectée bien plus élevée que toutes les autres branches. L'algorithme de Shor peut parfaitement s'accommoder de petites erreurs sur tous les états encodant des réponses incorrectes, tant que l'état encodant la bonne réponse dépasse une certaine probabilité seuil de détection. La signification à accorder aux très petites amplitudes dans le formalisme quantique n'est donc pas un problème en soi pour l'algorithme de Shor, tant que ces très petites amplitudes n'affectent pas la capacité de l'algorithme à produire une probabilité suffisante pour l'état encodant la solution. L. Levin ne fournit aucune raison pour laquelle ce devrait être le cas.

Les arguments sceptiques de L. Levin ne sont donc pas concluants, et ne fournissent pas de raisons de croire que l'existence d'un ordinateur quantique de grande taille soit autre chose qu'une question empirique. Nous allons à présent examiner les conséquences de l'interprétation empirique de la complexité sur notre compréhension de la théorie de la complexité comme discipline.

8.4 Conséquences de l'interprétation empirique

8.4.1 Une autre vision de la complexité

Pour articuler ces conséquences, il faut tout d'abord distinguer la question de l'interprétation empirique de la complexité de celle de la vérité de la thèse de Church-Turing étendue. L'ensemble des arguments contrefactuels permet d'exhiber la dépendance de la thèse de Church-Turing étendue à des considérations empiriques, quand bien même cette dernière sortirait confirmée de ces différentes épreuves. On pourrait donc, dans tous les cas de figure, parler d'interprétation empirique de la théorie de la complexité, au sens où certaines de ses hypothèses les plus fondamentales, comme l'indépendance du

caractère polynômial d'un problème au modèle de calcul, seraient appuyées ou renversées sur la base de réflexions empiriques.

Dans une telle perspective, on ne peut plus démontrer qu'un problème, compris comme une entité mathématique abstraite, possède telle ou telle propriété de complexité. Comme on l'a vu précédemment, c'est l'existence d'algorithmes optimaux, ainsi que la thèse de Church-Turing étendue, qui permettent d'attribuer une complexité à un problème, et non pas seulement à un algorithme. Mais si la thèse de Church-Turing est en réalité une hypothèse empirique, alors l'attribution de la complexité à un problème n'est pas un pur résultat mathématique, démontrable par des méthodes purement *a priori*, mais résulte d'un processus de confirmation empirique.

Dans l'éventualité où la thèse de Church-Turing étendue serait fausse, il ne serait plus possible de considérer les propriétés de complexité comme des propriétés intrinsèques des problèmes. Cela est particulièrement flagrant dans le cas où un ordinateur quantique serait possible. Il existerait alors un problème -FACTORISATION- dont la complexité serait suprapolynômiale sur une machine de Turing, et polynômiale sur une machine quantique. Bien sûr, on pourrait toujours parler de « complexité d'un problème », pour parler de la complexité de l'algorithme connu le plus efficace, mais celle-ci serait une propriété extrinsèque, dépendant du modèle de calcul considéré.

On peut toujours attribuer à un algorithme les propriétés de complexité, mais la notion même d'algorithme doit être vue d'un regard neuf. L'algorithme de Shor peut être simulé par une machine de Turing, puisqu'il peut être exprimé dans le langage linéaire-algébrique de la mécanique quantique. Cette simulation serait cependant inefficace, et entraînera un ralentissement exponentiel de l'exécution du calcul. Les propriétés de complexité de l'algorithme ne sont donc pas préservées par un changement d'implémentation. Pour avoir les propriétés désirées de complexité, l'algorithme de Shor doit être implémenté sur une machine quantique, faute de quoi il perd tout avantage comparatif. Là encore, on ne peut plus concevoir l'algorithme comme une pure entité mathématique abstraite, dont les propriétés seraient démontrables *a priori*. L'algorithme est une modélisation abstraite d'un processus empirique de traitement de l'information, dont l'implémentation doit satisfaire certaines conditions empiriques spécifiques pour préserver les propriétés de complexité désirées. Si l'on veut continuer à considérer la complexité comme une propriété des algorithmes, un algorithme ne peut plus être conçu comme une méthode mathématique abstraite.

Si l'on admet une telle vision, c'est même la distinction entre hardware et software qui doit être révisée. Un algorithme comme l'algorithme de Shor n'est plus portable, au sens où on pourrait l'implémenter sur n'importe quelle machine tout en préservant ses propriétés théoriques. La conception même de l'algorithme, et ses propriétés de complexité, dépendent d'un certain modèle du hardware, et de ses propriétés théoriques. Plus précisément, comme on va le voir ci-dessous, il existe une dépendance à la physique régissant le

hardware, plutôt qu'aux détails technologiques du hardware.

Lorsque l'on considère la théorie de la complexité comme l'étude de la difficulté intrinsèque de problèmes mathématiques, un algorithme décrit un ensemble d'étapes logiques par lequel doit passer le calcul de la solution. Si l'algorithme est formalisé au sein d'un modèle de calcul raisonnable, ces étapes logiques s'imposent à toute machine implémentant un tel calcul : c'est en ce sens que les résultats de la théorie de la complexité sont pertinents pour l'implémentation. Dans le cas du calcul quantique, la conception même de l'algorithme ne peut être vue comme un pur processus logicomathématique, sans aucune relation avec une conception de l'implémentation. L'application d'une étape de l'algorithme de Shor n'est pas une opération logique, qu'un être humain pourrait simuler étape par étape avec un crayon et du papier, tout en préservant ces propriétés de complexité.

Comme on l'a vu plus haut (voir section 7.2.1), la thèse de Church-Turing étendue garantit la coexistence pacifique de deux présentations de la théorie de la complexité computationnelle : une pure discipline mathématique, et une étude des performances réelles de machines. Si la thèse de Church-Turing étendue est fausse, on pourrait être tenté de considérer la théorie de la complexité computationnelle comme une forme d'ingénierie, étudiant les propriétés de divers modèles de calcul sans unité systématique. Nous allons voir qu'une telle conclusion n'est pas impliquée par notre analyse. Pour mieux comprendre comment la théorie de la complexité computationnelle maintiendrait son unité en tant que discipline scientifique, il faut regarder de plus près l'hypothèse d'indépendance au hardware.

8.4.2 L'unité de la théorie de la complexité computationnelle

Tout d'abord, nous appellerons « dépendance à la physique »l'hypothèse que les propriétés de complexité d'un modèle de calcul donné dépendent des hypothèses de physique théorique sous-jacente à ce modèle. La dépendance à la physique doit être bien distincte de la dépendance au hardware comprise comme une dépendance à une ingénierie particulière. Ainsi les propriétés de complexité d'un algorithme quantique dépendent des propriétés quantiques du système utilisé pour son implémentation, mais elles ne dépendent pas du détail technologique d'une stratégie particulière d'implémentation. Du point de vue de la théorie de la complexité, il n'importe guère qu'un ordinateur quantique soit fait d'ions piégés ou de photons, pour citer deux stratégies d'implémentation réellement explorées. La théorie de la complexité est donc toujours indépendante du hardware compris comme le détail technologique des stratégies d'implémentation, mais dépend de la physique théorique sous-jacente à toutes les stratégies d'implémentation.

Dans un deuxième temps, la dépendance à la physique doit également être distincte de la dépendance au modèle. Ces deux dernières décennies ont vu la conception de nombreux modèles de calcul quantiques : circuits

quantiques [80], [243], automates cellulaires quantiques [237], *measurement-based quantum computation*[199], machines de Turing quantiques [79], calcul quantique topologique [149], [180], calcul quantique adiabatique [94], ainsi que divers modèles d'ordre supérieur (λ-calcul quantique [201], [201], [17], [234], [200], langages catégoriques de l'école d'Oxford, [10], [9]). Cette liste est bien entendue ouverte, tout comme est ouverte la liste des modèles de calcul « classiques ». L'étude des relations de simulation entre ces modèles posent des problèmes exactement similaires à ceux posés aux modèles précédents. La classe BQP est démontrablement robuste pour nombre de ces modèles (voir par exemple [80],[104]) mais certains auteurs ont pu prétendre qu'un autre modèle, le calcul quantique adiabatique, était encore plus puissant que les autres modèles quantiques, permettant le calcul d'un problème NP-complet en temps polynômial [94]. Ce modèle a cependant été critiqué selon l'argument usuel, pour son manque de réalisme.

On peut donc formuler pour les modèles de calcul quantiques une nouvelle thèse de Church-Turing étendue : *tous les modèles de calcul quantiques raisonnables s'intersimulent avec un ralentissement au plus polynômial.* La thèse de Church-Turing étendue survit donc sous une forme relative : elle ne porte plus désormais sur tous les modèles de calcul concevables, mais sur une classe de modèles de calcul partageant un ensemble d'hypothèses physiques communes [14]. La théorie de la complexité, dans une telle perspective, ne doit plus être comprise comme une simple forme d'ingénierie, mais pourrait être vue comme une branche particulière de la physique mathématique, étudiant les possibilités computationnelles offertes par telle ou telle théorie physique. Il n'est cependant pas nécessaire d'assimiler la théorie de la complexité à une discipline déjà existante, et on peut la considérer comme une discipline empirique en soi.

Cette vision demeure cependant problématique, et doit être interprétée plus comme un programme que comme un résultat. Depuis deux décennies, physiciens et informaticiens tâchent de déterminer quel trait théorique précis de la mécanique quantique causerait l'accélération exponentielle qu'un ordinateur quantique serait censé fournir, et la question demeure controver-

14. M. Freedman, A. Kitaev, M. J. Larsen et Z. Wang ne disent pas autre chose dans ([103], 31) :

> Quantum computation is a catch-all for several models of computation based on a theoretical ability to manufacture, manipulate and measure quantum states.(...) The class of functions, BQP (functions computable with bounded error, given quantum resources, in polynomial time), has been defined in three distinct but equivalent ways : via quantum Turing machines [2], quantum circuits [3], [6], and modular functors [7], [8]. The last is the subject of this article. We may now propose a "thesis" in the spirit of Alonzo Church : all "reasonable" computational models which add the resources of quantum mechanics (or quantum field theory) to classical computation yield (efficiently) inter-simulable classes : there is one quantum theory of computation.

sée (voir, par exemple, [49]). De manière duale, il est difficile d'expliquer ce que « classique »signifie dans « modèle de calcul classique ». Comme nous l'avons vu durant notre examen des modèles d'hypercalcul (voir section 6.1), les modèles inspirés par le calcul papier-crayon, telle que la machine de Turing, ne sont pas physiquement définis. L'entrée n'est pas encodée dans l'état d'un système classique, la dynamique du système n'est pas rendue explicite, et aucune mention n'est faite de la mesure. Des modèles notionnels comme les modèles historiques du calcul par procédure effective, ne sont classiques qu'au sens vague où ils ne sont pas quantiques : il n'y est pas fait mention de propriétés spécifiquement quantiques comme la superposition des états et l'intrication. Même si l'on souscrit à l'idée que la théorie de la complexité computationnelle dépend de la physique, articuler cette idée s'est révélé difficile, et doit donc être compris plus comme un programme de recherche que comme un résultat bien défini.

Si elle se trouvait corroborée, l'idée de dépendance à la physique permettrait à la fois de différencier les modèles selon les théories physiques sous-jacentes, et de garantir l'unité systématique de la théorie de la complexité. Même si la physique théorique est composée de plusieurs théories dont l'unification est problématique, ces théories partagent un petit nombre de principes fondamentaux, comme les symétries fondamentales ou le second principe de la thermodynamique. De manière analogue, on peut envisager que la théorie de la complexité, bien que connaissant sa propre diversité interne, soit unifiée par un ou des principes fondamentaux, auxquels tous les modèles de calcul se conformeraient. Même si la thèse de Church-Turing étendue est démontrée fausse, et ne peut donc plus jouer un tel rôle, un principe computationnel plus faible, garantissant la robustesse d'autres concepts et d'autres résultats de la théorie de la complexité, pourrait venir se substituer à elle. Pour mieux comprendre cette idée, il faut distinguer entre les différents modèles de calcul, et leurs performances supposées au niveau de la complexité.

Si l'algorithme de Shor est bel et bien implémentable, la classe des problèmes de difficulté polynômiale n'est pas invariante par changement de modèle. Ceci n'implique pas que la distinction entre classes de complexité polynômiale et suprapolynômiale ne soit pas robuste, ni même que la distinction entre les classes P et NP ne le soit pas. Selon la conjecture courante, *FACTORISATION* n'est pas un problème NP-complet. Il appartiendrait à la classe de problèmes NPI (NP-Intermediate), la classe des problèmes dans NP qui ne sont ni NP-complets, ni calculables en temps polynômial sur une machine de Turing. Selon d'autres modèles que nous avons étudié, il serait possible de calculer tous les problèmes NP en temps polynomial, voire de rendre faisable des classes de complexité plus grandes encore. Nous avons vu que ces modèles sont bien moins plausibles que le calcul quantique, mais l'échec de tentatives n'implique pas l'échec d'une idée. Doit-on s'attendre à ce qu'une étude plus profonde des moyens mis à notre disposition par la

physique théorique permette de résoudre un problème NP-complet en temps polynômial, voir un problème plus dur encore ? Ou doit-on au contraire considérer cela comme improbable ? Le théoricien de la complexité S. Aaronson a formulé une argumentation visant à défendre la robustesse physique de la distinction entre les classes P et NP. Cette distinction acquerrait ainsi le statut d'un véritable principe physique, le *No SuperSearch Principle*.

Une fois l'interprétation empirique acceptée, la théorie de la complexité toute entière peut ainsi se voir réévaluée d'un regard nouveau. Qu'est-ce qui, au sein de nos résultats et conjectures de complexité présents, doit être considéré dépendant de théories physiques particulières, et qu'est-ce qui peut-être considéré comme « physiquement robuste » ?

Une fois admise l'interprétation empirique de la complexité, une véritable relation à double sens peut ainsi s'instaurer entre la complexité et la physique. L'interprétation empirique de la complexité ne doit pas être vue simplement comme une invitation à briser en mille morceaux les acquis de la théorie de la complexité, mais aussi comme la possibilité de formuler une rétroaction de la théorie de la complexité vers la physique théorique. Comme on va le voir à présent, le *No SuperSearch Principle* de Scott Aaronson est précisément une tentative pour formuler cette rétroaction, et substituer ainsi à la thèse de Church-Turing étendue, présumée fausse, un nouveau principe fondamental pour la complexité.

8.4.3 Scott Aaronson et la robustesse de la distinction P-NP : le *No SuperSearch Principle*

Dans les écrits de S. Aaronson, on peut distinguer deux arguments en faveur du *No Supersearch Principle*.

Le premier consiste en un recours à une forme d'argument théorique inductif. Il existe de nombreux modèles de calcul censés permettre le calcul d'un problème NP-complet en temps polynômial, voire même d'une classe de problèmes plus vaste encore. Cependant, tous ces modèles se sont révélés irréalistes. Tous ces échecs, en s'accumulant, finissent par former une forme d'évidence inductive en faveur de l'impossibilité de principe de tels modèles. S. Aaronson défend explicitement que cette résistance à de prétendus contre-exemples est analogue à celle de principes fondamentaux de la physique, qui sortent renforcés des tentatives faites pour les réfuter [15]. Cet argument, s'il

15.

> Many of the deepest principles in physics are impossibility statements : for example, no superluminal signalling and no perpetual motion machines. What intrigues me is that there is a two-way relationship between these principles and proposed counterexamples to them. On the one hand, every time a proposed counterexample fails, it increases our confidence that the principles are really correct, especially if the counterexamples almost work but not quite. (Think of Maxwell's Demon, or of the subtle distinction between quantum nonlocality and superluminal communication.) On the other hand,

est bien sûr pas pourvu d'une certaine valeur, n'est cependant pas absolument contraignant. Nous disposions exactement du même type d'argument en faveur de la thèse de Church-Turing étendue, avant que n'adviennent les difficultés posées par le calcul quantique. En outre, la théorie de la complexité computationnelle étant une discipline jeune, et la conscience de ces problématiques étant encore plus récente, on ne peut dire que le *No Super-search Principle* a résisté aux assauts du temps comme les plus vénérables de nos principes physiques.

Le second argument peut être qualifié d'argument contrefactuel théorique. Par là on entend que si une machine calculant efficacement des problèmes NP-complets existait, il s'en suivrait des conséquences théoriques titanesques, et fort peu plausibles. S. Aaronson insiste sur le fait qu'une telle machine ne constituerait pas simplement une grande avancée pratique, mais constituerait bel et bien une gigantesque révolution conceptuelle, d'une magnitude totalement différente de celle que représenterait un ordinateur quantique.

Le problème P-NP peut en effet être présenté comme une version finitaire de l'*Entscheidungsproblem*. La raison essentielle en est que, pour tout système axiomatique usuel en mathématiques A, l'appartenance à l'ensemble $\{(\phi, 1^n) : \phi$ possède une preuve formelle de longueur $\leq n$ dans le système A.$\}$ est dans NP.

S'il n'existe pas d'algorithme permettant de décider toute conjecture mathématique ϕ, il existe un algorithme permettant de décider s'il existe une preuve de longueur inférieure à une taille donnée. Si cet algorithme était de complexité polynômiale, il deviendrait possible de résoudre pratiquement toute conjecture mathématique, en se limitant à l'examen de preuves de longueur raisonnable, par exemple une longueur inférieure au nombre de particules dans l'Univers. On atteindrait alors une forme d'automatisation de la créativité mathématique, pour reprendre l'expression de S. Aaronson (*automated mathematical creativity*), malgré que l'*Entscheidungsproblem* ait une solution négative.

On pourra remarquer qu'un argument du même type pourrait évidemment être formulé pour la théorie de la calculabilité, puisque l'*Entscheidungsproblem* est un problème de calculabilité. L'une des fonctions de l'argument de S. Aaronson est précisément de faire remarquer que les limites de la complexité, tout comme celle de la calculabilité, sont des problèmes conceptuels fondamentaux, qui ne doivent pas être traitées avec légèreté comme de « simples »problèmes d'ingénierie ([3], 18-19).

Sur la base de ces arguments, S. Aaronson propose que les physiciens

> as we become more confident of the principles, we also become more willing to use them to constrain the search for new physical theories. Sometimes this can lead to breakthroughs : for example, Bekenstein discovered black hole entropy just by taking seriously the impossibility of entropy decrease.

([3], 17 (pdf)).

admettent un nouveau principe physique :

No SuperSearch Principle. *Il n'existe aucun moyen physique permettant de résoudre un problème NP-complet en temps polynômial.*

Après avoir présenté les arguments en sa faveur, nous allons discuter le sens et la formulation précise de ce principe.

Remarquons tout d'abord que l'énoncé d'un tel principe ne se substitue en aucun cas à une éventuelle démonstration de la conjecture P ?=NP, mais il constitue une réinterprétation du sens d'une telle démonstration. Celle-ci serait adossée à une prémisse physique, le *NoSuperSearch Principle*, qui garantira que ce résultat sera significatif pour tout modèle de calcul physiquement réalisable. La démonstration mathématique que P est strictement inclus dans NP, qui ne peut être valable que pour le modèle des machines de Turing, pourrait être généralisée, par le biais du *NoSuperSearch Principle*, en une forme de théorème de physique mathématique générale valable pour tout modèle de calcul. Ce principe serait en droit révisable, mais en pratique aussi bien établi que les principes les plus fondamentaux de la physique.

Si l'algorithme de Shor ouvre la possibilité que le caractère polynomial d'un problème donné soit relatif à la physique sous-jacente au modèle de calcul, le *No SuperSearch Principle* interdit un effondrement complet de la distinction entre problèmes polynomiaux et problèmes suprapolynomiaux. Si aucun problème NP-complet ne peut être résolu en temps polynomial, alors *a fortiori* les classes plus grandes incluant NP, comme EXP ou PSPACE, ne sauraient voir tous leurs problèmes résolus en temps polynomial. La distinction entre problèmes polynomiaux et problèmes suprapolynomiaux, si elle n'est pas entièrement robuste, survit donc sous une forme plus faible.

Pourquoi alors choisir précisément la classe NP ? Pourquoi ne pas prendre une classe plus petite, ou plus grande ? La réponse intuitive est que la classe NP est la plus petite classe pour laquelle on puisse formuler un argument contrefactuel théorique[5]. Cet argument permet de souligner le profond fossé conceptuel entre un algorithme efficace pour un problème NPI comme FACTORING, et un algorithme efficace pour résoudre un problème NP-complet comme 3-SAT.

S. Aaronson justifie également la formulation de ce principe par son pouvoir explicatif. Accepter le No SuperSearch Principle permettrait de rejeter certains modèles théoriques, parce qu'ils permettent la solution d'un problème NP-complet en temps polynomial. On pourrait ainsi rejeter les variantes non-linéaires de la mécanique quantique, les *Closed Time-Like Curves*, ou expliquer la présence d'*eigenvalue gaps* arbitrairement petits dans les évolutions quantiques adiabatiques, sur la base du *No SuperSearch Principle*. Un autre argument formulé par S. Aaronson en faveur du choix de la classe NP est précisément le plus grand nombre de modèles physiques sur lesquels elle fait peser des contraintes. Par contre, si on formulait le principe à l'aide de la classe EXP, on ne disposerait d'aucun argument à l'encontre

des *Closed Time-Like Curves*, les variantes non-linéaires de la mécanique quantique, tout en continuant à permettre en théorie l'automatisation de la créativité mathématique.

Au stade actuel de la recherche, il n'est pas évident d'attribuer un pouvoir explicatif au *No SuperSearch Principle*. Sans vouloir rentrer dans l'épineux débat sur la nature de l'explication en physique théorique, il nous semble nécessaire de restreindre l'attribution d'un pouvoir explicatif propre au principe au sens fort que nous avons énoncé plus haut (voir conclusion du chapitre 6). Parce que seuls les principes au sens fort sont censés exercer une contrainte propre sur l'activité de théorisation, ils sont les seuls à pouvoir se voir attribuer une puissance explicative.

Trois difficultés majeures s'opposent à l'admission du *No Supersearch Principle* comme principe au sens fort. La première est l'absence de mathématisation de la proposition. Tant que la proposition n'est pas mathématisée, il n'est pas possible d'examiner avec rigueur les relations logiques entre l'*explanans* supposé et son *explanandum*, et donc de dire ce qui explique quoi. La principale difficulté d'une telle mathématisation serait de définir physiquement la classe de problèmes NP, et la notion de problème complet. Sans une telle définition, il serait étrange de placer au fondement de notre physique un concept - la NP-complétude - qui a été formulé à la suite d'une variation théorique sur un modèle du calcul papier-crayon, sans aucune inspiration physique évidente. Pourquoi un modèle abstrait du calcul papier-crayon serait-il le lieu d'une révélation fondamentale pour la physique ? La signification empirique de la NP-complétude reste à extraire.

La seconde difficulté tient au mode même d'explication employée par S. Aaronson. Celui-ci est le plus souvent de nature contrefactuelle : l'argument montre que si telle propriété physique était vérifiée, il s'en suivrait des conséquences jugées peu plausibles en termes de pouvoir calculatoire. L'explication physique ne peut se contenter d'être purement contrefactuelle, et d'arguer l'impossibilité d'un modèle physique sur la base des conséquences catastrophiques que cela aurait pour d'autres disciplines. D'abord parce que cela ferait dépendre la physique de conjectures ouvertes en théorie de la complexité, et que l'argument n'a de valeur que dans la mesure où l'on croit à la plausibilité d'une solution donnée pour ces conjectures ; ensuite, parce qu'on ne peut qualifier d'explication physique un argument qui n'est pas formulé de bout en bout en termes physiques, et fait appel à des conséquences exprimées en termes extrinsèques à la discipline. Si les arguments contrefactuels peuvent se montrer suffisants pour établir le statut épistémique des principes computationnels, ils ne le sont pas pour déterminer la valeur de vérité de ces principes.

La troisième, et la plus importante, est que le *No Supersearch Principle* ne fait pas à l'heure actuelle peser de contraintes autonomes sur la théorie physique. Les variantes non-linéaires de la mécanique quantique ou les *Closed Timelike Curves* ont déjà été l'objet de nombreuses critiques de physiciens

sans aucun rapport avec des problématiques computationnels[16]. Il n'existe pas à notre connaissance d'emploi d'un tel principe qui permettrait d'exclure un modèle qui ne soit pas critiquable, et déjà critiqué, sur d'autres bases. Un authentique postulat ne sera accepté que s'il permet de formuler de nouvelles prédictions, et non uniquement de formuler des arguments contrefactuels à l'égard de modèles déjà abondamment critiqués pour d'autres raisons.

Pour toutes ces raisons, la possibilité que le *No Supersearch Principle* soit un principe au sens fort doit être considérée avec prudence. Il est préférable de le considérer comme un potentiel principe au sens fort. Pour l'appuyer, le physicien théoricien devra avant tout travailler à définir physiquement la notion de NP-complétude, afin d'en extraire la signification physique. L'interaction entre physique et complexité ne peut prendre la forme d'une imposition des catégories computationnelles à la physique, mais plutôt d'une interprétation à nouveau frais de ces catégories dans le langage de la physique, comme l'exprime bien A. Hagar ([127], 3) :

> (...) the idea that abstract mathematical concepts such as *complexity* and *(in)tractability* may not only be translated into physics, but also rewritten by physics, militates against the autonomous character of some of the theoretical entities of computer science -the so called "computational kinds".

Conclusion

Comme dans le cas de la thèse de Church-Turing simple, l'examen de la thèse de Church-Turing étendue et des modèles de calcul non-standard a révélé l'importance d'hypothèses empiriques dans la compréhension de la complexité computationnelle. La définition, l'établissement d'une liste exhaustive des ressources pertinentes et la comparaison des modèles de complexité raisonnables nécessitent l'examen d'hypothèses empiriques diverses. Encore une fois, les arguments sceptiques visant à établir la vérité d'une thèse fondamentale de la théorie sur des considérations purement méthodologiques se sont révélés insuffisants, et l'interprétation empirique de la théorie en est sortie justifiée.

En revanche, le paysage théorique offert par la théorie de la complexité computationnelle est bien plus morcelé que pour la théorie de la calculabilité. Outre les difficultés bien connues portant sur le caractère naturel des premières définitions de la théorie, notre travail a montré l'existence d'autres difficultés liées à l'interprétation empirique de la complexité.

La première de ces difficultés est liée à la valeur de vérité de la thèse de Church-Turing étendue. Celle-ci est couramment menacée par la possibilité d'une implémentation du calcul quantique. Si l'on reste fidèle à l'interprétation empirique, seules les tentatives expérimentales d'implémentation d'un

16. Voir par exemple [113] pour la mécanique quantique non-linéaire, et [191], pour les *Closed Timelike Curves*.

ordinateur quantique de grande taille peuvent décider de la vérité de la thèse de Church-Turing étendue, et il ne nous reste plus qu'à attendre le résultat de ces recherches.

Comme nous venons de le voir, la fausseté de la thèse de Church-Turing étendue n'impliquerait pas une dissolution de la théorie de la complexité computationnelle en une multitude de modèles sans lien systématique. Même si la thèse de Church-Turing étendue est fausse, la distinction entre polynômial et suprapolynômial peut demeurer robuste. Cette robustesse pourrait être la base de la formulation d'un nouveau principe physique. En examinant la proposition de S. Aaronson, le *No SuperSearch Principle*, nous avons également vu que déterminer la formulation exacte d'un tel principe, et le pouvoir explicatif qui peut lui être accordée, n'est pas sans difficulté. Les arguments de S. Aaronson ont le mérite de souligner l'importance des enjeux théoriques de cette question, et de montrer que la classe NP est la plus petite classe dont l'égalisation à P aurait des conséquences titanesques.

Ceci nous amène à la deuxième difficulté, qui est de préciser la signification empirique de la complexité. Dans le cas de la calculabilité, on pouvait à tout le mois conjecturer la vérité d'une hypothèse -la thèse de Church-Turing physique- et l'existence d'un principe théorique précis interdisant l'implémentation de la quasi-intégralité des modèles d'hypercalcul : le principe holographique. Dans le cas de la complexité, la formulation même de l'hypothèse en question fait débat, vu l'incertitude entourant la thèse de Church-Turing étendue, et une revue raisonnée des modèles au pouvoir calculatoire exceptionnel ne permet pas même de formuler une conjecture sur la base d'hypothèses récurrentes.

Conclusion

L'essentiel de notre travail a été consacré aux trois questions suivantes :

1. l'interprétation empirique des limites du calcul ;
2. la position exacte de ces limites, c'est-à-dire la valeur de vérité de la thèse de Church-Turing et de la thèse de Church-Turing étendue ;
3. la signification empirique des principes computationnels énonçant ces limites du calcul.

Nous avons défendu, tout au long de notre analyse, la justesse de l'interprétation empirique. Un algorithme doit fondamentalement être compris comme un processus empirique de manipulation de l'information, qui permet à un observateur fini d'obtenir la solution d'un problème donné avec une consommation de ressources finie. Les limites ultimes de ces processus, en termes de calculabilité comme de complexité, doivent être comprises comme des questions empiriques.

Dans le cas de la calculabilité, nous avons tout d'abord insisté, avec nombre d'auteurs contemporains, sur la nécessaire distinction entre deux interprétations autonomes de la thèse de Church-Turing, la forme algorithmique et la forme empirique. La première a pour objet la reconstruction rationnelle de la notion intuitive de « fonction calculable par une procédure effective », et en tant que telle, sa valeur de vérité ne dépend pas de considérations empiriques. Nous avons défendu le statut empirique de la seconde forme, contre des objections tendant à en faire une simple proposition méthodologique. Nous avons également montré comment on pouvait, à partir des débats sur l'implémentabilité des modèles d'hypercalcul, articuler un argument contrefactuel en faveur de l'interprétation empirique de la calculabilité. Nous avons précisé les conditions d'emploi légitime de cet argument, en différenciant modèles notionnels et modèles physiquement définis, et en montrant que l'antécédent de l'argument devait porter sur la description d'un modèle physiquement défini.

Notre travail sur l'interprétation empirique de la complexité a donné nombre de résultats analogues. Là encore nous avons rencontré une position sceptique refusant l'interprétation empirique, et nous l'avons trouvée insuffisante. Là encore, le débat sur l'implémentabilité des modèles non-standard sert, par argument contrefactuel, de justification à l'interprétation empirique.

Le cas de la complexité présente cependant des différences notables avec celui de la calculabilité. Il n'est pas pertinent de distinguer une forme algorithmique et une forme empirique de la thèse de Church-Turing étendue. Tout d'abord, la thèse de Church-Turing ne peut être comprise comme la reconstruction rationnelle d'une notion intuitive de « fonction efficacement calculable par une procédure effective. »La thèse de Cobham-Edmonds constitue une proposition supplémentaire, nécessaire pour reconstruire une telle notion par celle de « fonction calculable en temps polynômial sur une machine de Turing déterministe. »Ensuite, les mesures de complexité doivent être comprises comme des estimations des consommations de ressources observables pendant l'exécution du calcul. Pour que cette fin théorique soit atteinte, une hypothèse empirique doit être faite : c'est ce que nous avons appelé l'hypothèse de proportionnalité des ressources observables aux ressources théoriques. Cette hypothèse doit être valide, que l'on considère un modèle standard du calcul effectif ou un modèle non-standard, pour que la théorie de la complexité jouisse d'une véritable pertinence pour l'implémentation. La distinction entre forme algorithmique et forme empirique n'aurait donc pas d'emploi, puisque celles-ci reposent toutes deux sur une hypothèse empirique commune. Pour montrer que l'interprétation empirique de la complexité est juste, non seulement il est possible d'employer un argument contrefactuel comme dans le cas de la thèse physique, mais il suffit de réaliser que la théorie de la complexité computationnelle, si elle veut accomplir ses fins affichées, doit être comprise comme un travail de modélisation des ressources consommées par des processus empiriques.

L'interprétation empirique des limites du calcul amène à réviser en profondeur notre manière d'envisager la signification des concepts premiers de l'informatique. Comme nous l'avons déjà remarqué (voir chapitre 5), l'informatique a ceci de singulier que nombre de ses concepts premiers -algorithme, processus, modèle de calcul- n'ont pas de définition explicite. Si l'interprétation empirique des limites du calcul est juste, il n'y a pas à attendre une telle définition explicite, et à considérer son absence comme un signe du manque de maturité théorique de l'informatique. Ces concepts sont des concepts empiriques, et en tant que tels ils prennent leur signification, non d'une définition explicite, mais de leur emploi fécond dans de nombreux contextes particuliers. En guise d'analogie, on peut remarquer qu'il n'existe pas de définition explicite du concept fondamental d'énergie en physique, et sa signification scientifique doit être saisie avant tout par ses multiples applications, qui constituent une liste ouverte : énergie mécanique, énergie électrique, énergie chimique... De la même manière, il n'y a pas à attendre une définition explicite rigoureuse des concepts d'« algorithme »ou de « modèle de calcul », mais des listes ouvertes d'applications.

Ces dernières considérations n'impliquent pas que toute rigueur doive être abandonnée dans le traitement de ces concepts. Tout d'abord, l'absence de définition explicite absolue n'empêche pas la mathématisation. Ainsi, les

différentes formes de l'énergie reçoivent une définition mathématique rigoureuse, soumise à des principes généraux comme le principe de conservation, au sein des différents modèles considérés. De même, on peut rigoureusement définir un algorithme par une famille uniforme de circuits booléens, ou une famille uniforme de circuits quantiques, sans pouvoir donner de définition formelle de la notion d'algorithme, qui soit à la fois rigoureuse et indépendante de tout modèle. Ensuite, le philosophe peut articuler une liste de contraintes informelles, afin d'éviter des usages impropres qui seraient source de malentendus et de faux débats. C'est ce que nous avons fait en adoptant une version amendée de la conception épistémique du calcul, défendue par G. Piccinini, en montrant comment elle résistait aux objections, et permettait la bonne position des différentes questions mêlant calcul et physique.

D'une telle position sur la signification des concepts premiers de l'informatique, on ne doit pas déduire trop rapidement une condamnation des tentatives de caractérisation des thèses algorithmique ou physique, telles qu'on peut respectivement les trouver dans les travaux de Y. Gurevich et son école, et dans les approches à la Gandy. Il suffit d'éviter de les interpréter comme des « démonstrations », qui prétendraient clore définitivement la question de la définition des algorithmes, ou des calculs effectuables par une machine. Outre leur intérêt technique, ces caractérisations ont un véritable intérêt heuristique, puisqu'elles gagnent en précision technique ce qu'elles perdent en généralité conceptuelle.

La deuxième question majeure était celle de la valeur de vérité des principes computationnels comme la thèse physique, la thèse étendue ou le *No Supersearch Principle*. Là encore, il importe de bien distinguer le cas de la calculabilité de celui de la complexité.

Dans l'état actuel de la recherche, il n'existe aucun modèle d'hypercalcul dont l'implémentabilité ne se heurte, non seulement à de graves difficultés opérationnelles, mais aussi à des objections de principe majeures. Si l'on réfléchit par simple induction sur les tentatives existantes, on peut donc conclure qu'aucun modèle d'hypercalcul n'est en principe implémentable, et que la thèse de Church-Turing empirique est respectée.

Le cas de la complexité est sensiblement différent. Le sort du principe analogue à la thèse de Church-Turing, la thèse étendue, est actuellement suspendu au succès de l'implémentation d'un ordinateur quantique, qui doit être comprise comme une véritable possibilité de principe. Si la thèse étendue venait à être prise en défaut, le *No SuperSearch Principle* pourrait s'imposer comme un principe de substitution. En tout état de cause, les limites exactes auxquelles serait soumis tout modèle raisonnable de la complexité sont à l'heure actuelle incertaines.

Le troisième problème majeur de notre travail était l'articulation de la signification empirique des limites du calcul. Il s'agissait pour nous de déterminer quelles contraintes des principes computationnels pouvaient faire peser sur la théorie physique. Le sens donné à ces contraintes varie selon

qu'on considère les principes computationnels comme des principes au sens fort ou des principes au sens faible. Si on les considère comme des principes au sens faible, on cherchera les hypothèses de la physique théorique, qu'elles soient établies ou encore en discussion, sur lesquelles ces principes seraient susceptibles d'être fondés. Si on les considère comme des principes au sens fort, ce sont ces principes computationnels qui serviront de fondements aux prochains efforts de théorisation physique. Il existerait alors une véritable rétroaction de la compréhension physique des fondements du calcul sur la physique elle-même. Il ne s'agissait pas seulement de voir comment la physique pourrait fournir clés en main les fondements du calcul, mais aussi comment les limites fondamentales du calcul pouvaient devenir des limites fondamentales pour la physique.

Il n'existe pas en l'état actuel de la recherche de démonstration de la thèse de Church-Turing physique à partir de principes physiques fondamentaux, ni même d'argument heuristique en ce sens. Notre démarche a là aussi consisté en une induction sur les différents modèles d'hypercalcul, visant à déterminer l'existence éventuelle d'hypothèses récurrentes dans les débats portant sur leur implémentabilité. D'un point de vue méthodologique, nous avons montré que seule l'étude des modèles physiquement définis, en sus de permettre une argumentation de principe en faveur de l'interprétation empirique des limites du calcul, permettaient de dégager la signification empirique des principes computationnels. Les modèles physiquement définis permettent en effet la traduction des concepts informatiques en modèles physiques précis, dont les caractéristiques peuvent ensuite être discutées dans les termes de la physique théorique.

Nous avons donc limité notre induction à des modèles d'hypercalcul physiquement définis, et montré que tous ces modèles partagent une hypothèse commune : l'exigence de précision arbitraire. Cette hypothèse suffit à condamner tout programme d'implémentation de ces modèles d'un point de vue opérationnel. D'un point de vue théorique, l'exigence d'une précision arbitraire dans la mesure et la préparation d'une observable est incompatible avec un principe théorique au coeur des débats actuels en physique théorique, le principe holographique. Nous avons vu que la précision arbitraire en temps de détection d'un signal, exigée par le modèle de Malament-Hogarth, n'était pas prohibé par ce principe. S'il n'existe rien comme une démonstration de l'impossibilité de l'hypercalcul, il existe à tout le moins un principe putatif incompatible avec la quasi-intégralité des modèles physiquement définis, l'unique exception affrontant des objections de principe qui lui sont propres. Un tel résultat, outre qu'il justifie un scepticisme robuste à l'égard de la possibilité de l'hypercalcul, permet d'identifier une hypothèse empirique précise au fondement de l'impossibilité de nombre de modèles.

Le principe holographique peut donc être vu comme une composante de la signification empirique de la thèse physique. Il ne saurait être vu, en l'état actuel de la recherche, comme l'intégralité de cette signification.

Outre l'exception représentée par le modèle de Malament-Hogarth, il faudrait tout d'abord pouvoir préciser les raisons physiques interdisant l'existence d'un modèle d'hypercalcul non-relativiste et discret, pour prétendre capturer l'ensemble des contraintes que ce principe computationnel ferait peser sur la physique théorique. Il faudrait ensuite, et surtout, réussir à mathématiser la thèse physique au sein des diverses théories existantes. C'est uniquement à ce stade de la recherche qu'il sera possible d'obtenir une formulation précise de la signification empirique de ce principe, et qu'on pourra décider de la pertinence de le considérer comme un principe au sens fort.

En l'absence de tels résultats, l'état présent de la recherche sur les modèles d'hypercalcul ne justifie aucunement l'introduction d'un principe de Church-Turing parmi les principes fondamentaux de la physique. L'intégralité des modèles physiquement définis peuvent parfaitement être exclus sur la base de principes qui, s'il sont toujours en discussion, constituent des propositions théoriques bien formulées et disposant d'arguments indépendants en leur faveur. L'emploi d'un principe de Church-Turing serait donc redondant pour exclure les modèles d'hypercalcul. En revanche, ce n'est pas un moindre mérite de la discussion de la thèse physique que de mettre en valeur un nouvel enjeu de la discussion du principe holographique.

Pour la théorie de la complexité, la situation est encore une fois différente. Cette asymétrie prend sa source dans l'incertitude entourant la valeur de vérité de la thèse étendue, que nous venons d'évoquer. Même si l'on admet la possibilité de formuler un principe computationnel à partir de la théorie de la complexité, force est de reconnaitre qu'il existe une plus grande incertitude quant au principe computationnel à choisir que dans le cas de la calculabilité, où la thèse physique est le seul candidat naturel. En outre, si l'on admet l'interprétation empirique de la complexité, on doit également admettre que le sort de la thèse étendue dépend de la résolution de questions expérimentales ouvertes. Dans de telles circonstances, il serait scientifiquement prématuré de prétendre connaître avec certitude la signification empirique de cette thèse, et les raisons de son éventuelle violation, comme nous l'ont montré les incertitudes entourant l'origine du pouvoir calculatoire supposé du calcul quantique.

En outre, une recherche inductive sur les modèles de calcul non-standard au pouvoir calculatoire exceptionnel ne permet pas la mise en évidence d'une hypothèse récurrente, comme dans le cas de la calculabilité. Néanmoins, que l'on considère la thèse étendue ou le *No SuperSearch Principle*, on arrive à une conclusion similaire à celle atteinte dans le cas de la calculabilité. Aucun de ces principes ne constitue une contrainte nouvelle pour la physique théorique, et les modèles permettant leur violation sont critiquables sur la base des connaissances théoriques existantes. On peut ainsi défendre la plausibilité du *No SuperSearch Principle* en tant que proposition théorique sur la base de nos connaissances physiques établies, sans en faire un principe au sens fort. Si ce principe computationnel acquiert une forte plausibilité

intuitive par l'argument contrefactuel employé par S. Aaronson, cela ne doit pas nous faire oublier son absence de mathématisation. Celle-ci interdit, ici comme dans le cas de la calculabilité, de le considérer pour l'heure comme un authentique principe physique.

Toutes nos dernières remarques ne sont qu'un commentaire de l'état actuel de la recherche sur ces principes computationnels. Elles ne doivent pas être lues comme une condamnation de l'emploi de principes computationnels en physique théorique. La mathématisation rigoureuse d'un principe physique, et l'extraction de ses conséquences pour la physique théorique -ce que nous avons appelé la signification empirique d'un principe- représentent un travail long et difficile. Celui-ci peut aisément prendre des années. À ce titre, il faut souligner que certaines des propositions que nous avons examinées, comme par exemple le *No SuperSearch Principle*, n'ont que quelques années, et qu'aucune n'a plus de trente ans[17].

Pour le philosophe, l'un des enjeux majeurs de cette recherche à venir est le devenir des concepts venus de l'informatique, qu'on peut qualifier de concepts computationnels. Le principe holographique permet d'exclure les modèles d'hypercalcul sans employer de concepts computationnels. Aucune mention n'y est faite de calcul, mais uniquement de l'information encodable dans un système. Le principe holographique ne peut intervenir dans la discussion de la thèse physique que par l'exigence de précision arbitraire, qui crée le lien entre considérations computationnelles et considérations physiques. Les tentatives existantes pour donner une définition physique à un concept computationnel comme l'étape élémentaire de calcul ne nous ont pas paru décisives. Il reste à voir si les concepts computationnels ont vocation à se voir définir physiquement, ou si les principes computationnels ne seront validés que par le recours à des concepts extrinsèques à la théorie du calcul.

Pour finir, nous allons brièvement évoquer nos perspectives de travail. Comme nous l'avons expliqué dans notre introduction, cette thèse n'est qu'une petite partie d'un plus vaste programme de recherche. Mais celui-ci constitue une perspective de très long terme, et, comme nous l'avons signalé à plusieurs reprises, le travail présenté dans cette thèse est susceptible d'être éclairé par des résultats scientifiques plus ou moins imminents, et pourrait être approfondi en certains points. Nous allons donc nous en tenir aux perspectives de court terme, qui sont le plus à même de prolonger de manière féconde l'étude ici commencée.

La première est un approfondissement philosophique des travaux récents en complexité et calculabilité analogique. Cette branche longtemps négligée de l'informatique a connu un regain d'activité récent, si récent que nous nous sommes trouvés dans l'impossibilité d'accorder à certaines publications l'at-

17. La première interprétation d'un principe computationnel comme un principe physique au sens fort fut proposée en 1985 par D. Deutsch, dans son article fondateur du calcul quantique [78].

tention scrupuleuse qu'elles auraient méritée. Les travaux d'O. Bournez, E. Hainry et D.S. Graça sur la thèse de Church-Turing et la thèse de Church-Turing étendue pour les modèles analogiques sont notamment d'un grand intérêt. Une telle étude serait d'autant plus désirable que la calculabilité et la complexité sur les réels ont été malheureusement négligées par les philosophes. Autant la machine de Turing a été l'objet d'une profonde appropriation thématique par la communauté philosophique, bien au-delà des spécialistes de la logique et de la philosophie des sciences, autant le calcul quantique a pu récemment éveillé l'attention des philosophes des sciences, autant le calcul sur les réels, et en particulier le calcul analogique, n'a à notre connaissance fait l'objet d'aucune analyse philosophique approfondie. Depuis la première page de ce travail, nous plaidons qu'il devrait en être autrement. Même si ces modèles ne montrent en dernière analyse aucun pouvoir expressif et aucun pouvoir calculatoire extraordinaires, la possibilité d'effectuer les mêmes calculs par des moyens profondément différents devrait en soi mériter l'attention des philosophes.

Le deuxième sujet, partiellement connexe, est la définition de la complexité dans les modèles non-standard, la comparaison des mesures entre différentes familles de modèles, et l'étude de la complexité en plusieurs ressources. Nous n'avons pu qu'effleurer la profondeur du sujet, qui fait l'objet de travaux scientifiques très récents, comme ceux d'E. Blakey. L'hypothèse de proportionnalité des ressources théoriques aux ressources observables que nous avons énoncée doit avant tout être vue comme une base heuristique pour de prochaines études.

Tout au long de cette thèse, nous avons été frappés de voir comment la réflexion conceptuelle du philosophe, jusque dans une apparente innocence, pouvait rencontrer les préoccupations de la science en train de se faire. Nous ne pouvons qu'espérer que cette rencontre se poursuive dans des travaux à venir.

Bibliographie

[1] Scott Aaronson. Forthcoming. Why Philosophers Should Care about Computational Complexity. *Computability : Gödel, Turing, Church, and beyond.*

[2] Scott Aaronson. Multinear Formulas and Skepticism of Quantum Computing. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 118–127, Chicago, IL, USA, 2004. Laszlo Babai.

[3] Scott Aaronson. NP-complete Problems and Physical Reality. *SIGACT News*, March 2005. quant-ph/0502072.

[4] Scott Aaronson. Anti-Complexitism, November 2010.

[5] Scott Aaronson. RE : a question about the NoSuperSearch Principle, June 2012.

[6] Scott Aaronson. The Toaster-Enhanced Turing Machine, August 2012.

[7] Scott Aaronson. RE : questions about Zeno Computer, August 2013.

[8] D.S. Abrams and Seth Lloyd. Nonlinear quantum mechanics implies polynomial-time solution for NP-complete and #P problems. *Physical Review Letters*, 81 :3992–3995, 1998. quantum-ph/9801041.

[9] Samson Abramsky and Bob Coecke. A categorical semantics of quantum protocols. In *Logic in Computer Science, 2004. Proceedings of the 19th Annual IEEE Symposium*, pages 415–425, 2004.

[10] Samson Abramsky and Ross Duncan. A Categorical Quantum Logic. *Mathematical Structures in Computer Science*, 16(3) :469–489, 2006.

[11] Wilhelm Ackermann and David Hilbert. *Grundzüge der theoretischen Logik.* Springer Verlag, Berlin, 1928.

[12] Leonard M. Adleman. Molecular computation of solutions to combinatorial problems. *SCIENCE*, pages 1021–1023, 1994.

[13] Dorit Aharonov and Umesh Vazirani. Is Quantum Mechanics Falsifiable ? A computational perspective on the foundations of Quantum Mechanics. *ArXiv preprint arXiv :1206.3686*, 2012.

[14] Hajnal Andréka, István Németi, and Péter Németi. General Relativistic Hypercomputing and Foundation of Mathematics. *Computing*, 8(3) :499–516, 2009.

[15] Konstantine Arkoudas. Computation, Hypercomputation, and Physical Science. *Journal of Applied Logic*, 6(4) :461–475, 2008.

[16] Sanjeev Arora and Barak Boaz. *Computational Complexity. A Modern Approach*. Cambridge University Press, 2009.

[17] Pablo Arrighi and Gilles Dowek. Linear-algebraic λ-calculus : higher-order, encodings, and confluence. In *Rewriting Techniques and Applications*, pages 17–31, 2008.

[18] Pablo Arrighi and Gilles Dowek. The Physical Church-Turing Thesis and the Principles of Quantum Theory. *International Journal of Foundations of Computer Science*, 23(05) :1131–1145, 2012.

[19] Eugene Asarin. Chaos and Undecidability. Draft. 1995.

[20] Jonathan D. Barrow. *Pourquoi le monde est-il mathématique ?* Odile Jacob, Paris, 1996.

[21] Stephen D. Bartlett, Terry Rudolph, and Robert W. Spekkens. Reference frames, superselection rules, and quantum information. *Reviews of Modern Physics*, 79(2) :555–609, April 2007.

[22] Edwin Beggs, José Félix Costa, Bruno Loff, and John V. Tucker. Computational complexity with experiments as oracles. *Proceedings of the Royal Society A : Mathematical, Physical and Engineering Science*, 464(2098) :2777–2801, 2008.

[23] Edwin Beggs, José Félix Costa, Bruno Loff, and John V. Tucker. On the complexity of measurement in classical physics. In *Theory and Applications of Models of Computation*, pages 20–30. Springer, 2008.

[24] Edwin J. Beggs and John V. Tucker. Experimental computation of real numbers by Newtonian machines. *Proceedings of the Royal Society A : Mathematical, Physical and Engineering Science*, 463(2082) :1541–1561, 2007.

[25] Jacob D. Bekenstein. Information in the holographic universe. *Scientific American*, 289(2) :58–65, 2003.

[26] Paul Benacerraf and Hilary Putnam. *Philosophy of Mathematics : Selected Readings*. Cambridge University Press, 1983.

[27] Charles H. Bennett. Undecidable dynamics. *Nature*, 346 :606–607, 1990.

[28] Ethan Bernstein and Umesh Vazirani. Quantum Complexity Theory. *SIAM Journal on Computing*, 26(5) :1411–1473, 1997.

[29] Garrett Birkhoff and John Von Neumann. The Logic of Quantum Mechanics. *The Annals of Mathematics*, 37(4) :823–843, 1936.

[30] Ed Blakey. Beyond Blum : What Is a Resource ? *IJUC*, 6(3-4) :223–238, 2010.

[31] Ed Blakey. Computational Complexity in Non-Turing Models of Computation : The What, the Why and the How. *Electronic Notes in Theoretical Computer Science*, 270(1) :17–28, February 2011.

[32] Andreas Blass and Yuri Gurevich. Algorithms : A Quest for Absolute Definitions. *Bulletin of the European Association for Theoretical Computer Science*, 2003.

[33] Lenore Blum, Felipe Cucker, Mike Shub, and Steve Smale. Complexity and Real computation : A Manifesto. *International Journal of Bifurcation and Chaos*, 6(01) :3–26, 1996.

[34] Lenore Blum, Mike Shub, and Steve Smale. On a Theory of Computation and Complexity over the Real Numbers : NP-Completeness, Recursive Functions and Universal Machines. *The Collected Papers of Stephen Smale*, 3 :1293, 2000.

[35] Manuel Blum. A Machine-Independent Theory of the Complexity of Recursive Functions. *Journal of the ACM (JACM)*, 14(2) :322–336, 1967.

[36] Egon Boerger and James K. Huggins. Abstract state machines 1988-1998 : Commented ASM bibliography. *arXiv preprint cs/9811014*, 1998.

[37] Paul Boghossian and Christopher Peacocke. Introduction. In *New Essays on the A Priori*, pages 1–10. Oxford University Press, 2000.

[38] Paul Artin Boghossian. Analyticity Reconsidered. *Noûs*, 30(3) :360–391, 1996.

[39] Udi Boker and Nachum Dershowitz. The Church-Turing Thesis over Arbitrary Domains. In *Pillars of computer science*, pages 199–229, 2008.

[40] Udi Boker and Nachum Dershowitz. Three Paths to Effectiveness. In *Fields of logic and computation*, pages 135–146. Springer, 2010.

[41] George Boolos and Richard Jeffrey. *Computability and logic*. Cambridge University Press, Cambridge, Great Britain, 1974.

[42] Olivier Bournez, Manuel L. Campagnolo, Daniel S. Graça, and Emmanuel Hainry. The General Purpose Analog Computer and Computable Analysis are Two Equivalent Paradigms of Analog Computation. In *Theory and Applications of Models of Computation*, pages 631–643. Springer, 2006.

[43] Olivier Bournez and Michel Cosnard. On the computational power of dynamical systems and hybrid systems. *Theoretical Computer Science*, 168(2) :417–459, 1996.

[44] Olivier Bournez and Gilles Dowek. Preface. *Natural Computing*, 11(1) :1–1, August 2011.

[45] Olivier Bournez, Daniel S. Graça, and Emmanuel Hainry. Computation with Perturbed Dynamical Systems. *Journal of Computer and System Sciences*, pages 714–724, 2013.

[46] Olivier Bournez, Daniel S. Graça, and Amaury Pouly. Turing Machines Can Be Efficiently Simulated by the General Purpose Analog Computer. In *Proceedings TAMC 2013*, pages 169–180, Hong Kong, China, May 2013. Springer.

[47] Raphael Bousso. The Holographic Principle. *Reviews of Modern Physics*, 74(3) :825, 2002.

[48] Vasco Brattka, Peter Hertling, and Klaus Weihrauch. A Tutorial on Computable Analysis. In *New Computational Paradigms*, pages 425–491. Springer, 2008.

[49] Jeffrey Bub. Quantum computation : Where does the speed-up come from. *Philosophy of Quantum Information and Entanglement. Cambridge University Press, Cambridge*, pages 231–246, 2010.

[50] Tim Button. SAD Computers and Two Versions of the Church–Turing Thesis. *The British Journal for the Philosophy of Science*, 60(4) :765–792, 2009.

[51] Cristian S. Calude and Boris Pavlov. Coins, Quantum Measurements, and Turing's Barrier. *Quantum Information Processing*, 1(1-2) :107–127, 2002.

[52] Cristian S. Calude and Karl Svozil. Quantum Randomness and Value Indefiniteness. *Advanced Science Letters*, 1 :165–168, 2008.

[53] Manuel L. Campagnolo. *Computational complexity of real valued recursive functions and analog circuits.* PhD thesis, IST, Universidade Técnica de Lisboa, 2001.

[54] Manuel Lameiras Campagnolo, Cristopher Moore, and José Félix Costa. An analog characterization of the Grzegorczyk hierarchy. *Journal of Complexity*, 18(4) :977–1000, 2002.

[55] Charles Choi. Google and NASA Launch Quantum Computing AI Lab. *MIT Technology Review*, May 2013.

[56] Alonzo Church. An Unsolvable Problem of Elementary Number Theory. *American Journal of Mathematics*, 58(2) :345–363, 1936.

[57] Alonzo Church. Review : AM Turing, On Computable Numbers, with an Application to the Entscheidungsproblem. *Journal of Symbolic Logic*, 2(1) :42–43, 1937.

[58] Alonzo Church. Review of Post [1936]. *The Journal of Symbolic Logic*, 2(43), 1937.

[59] Alonzo Church. On the Concept of a Random Sequence. *Bulletin of the American Mathematical Society*, 46(2) :130–135, 1940.

[60] Carol E Cleland. The Concept of Computability. *Theoretical Computer Science*, 317(1) :209–225, 2004.

[61] Alan Cobham. The intrinsic computational difficulty of functions. In *Proceedings of the International Conference on Logic, Methodology, and Philosophy of Science*, pages 24–30, 1965.

[62] Paul Cockshott, Lewis M. Mackenzie, and Gregory Michaelson. *Computation and its Limits*. Oxford University Press, 2012.

[63] S. Barry Cooper. *Computability theory*. Chapman & Hall, Boca Raton London New York Washington, D.C., 2004.

[64] S. Barry Cooper, Benedikt Löwe, and Andrea Sorbi. *New computational paradigms : Changing Conceptions of What is Computable*. Springer, 2008.

[65] Jack Copeland. Accelerating Turing Machines. *Minds and Machines*, 12(2) :281–300, 2002.

[66] Jack Copeland. Hypercomputation. *Minds and Machines*, 12(4) :461–502, 2002.

[67] Jack Copeland. Computation. In *The Blackwell Guide to the Philosophy of Computing and Information*, pages 3–17. Blackwell, Oxford and Cambridge, Mass., 2003.

[68] Jack Copeland. *The Essential Turing*. Oxford University Press, 2004.

[69] Jack Copeland and Diane Proudfoot. Alan Turing's forgotten ideas in Computer Science. *Scientific American*, pages 99–103, 1999.

[70] Jack Copeland and Oron Shagrir. Physical computation : How general are Gandy's principles for mechanisms ? *Minds and Machines*, 17(2) :217–231, 2007.

[71] Paolo Cotogno. Hypercomputation and the Physical Church-Turing Thesis. *The British Journal for the Philosophy of Science*, 54(2) :181–223, 2003.

[72] Martin Davis. Why Gödel didn't have Church's thesis. *Information and Control*, 54(1/2) :3–24, 1982.

[73] Martin Davis. The Myth of Hypercomputation. In *Alan Turing : Life and Legacy of a Great Thinker*. Springer, 2004.

[74] Martin Davis. The Church-Turing Thesis : Consensus and Opposition. *Logical Approaches to Computational Barriers*, pages 125–132, 2006.

[75] Martin Davis. Why There is No Such Discipline as Hypercomputation. *Applied Mathematics and Computation*, 178(1) :4–7, 2006.

[76] Frederik Denef and Michael R. Douglas. Computational Complexity of the Landscape : Part I. *Annals of physics*, 322(5) :1096–1142, 2007.

[77] Nachum Dershowitz and Yuri Gurevich. A Natural Axiomatization of Computability and Proof of Church's Thesis. *Bulletin of Symbolic Logic*, 14(3) :299–350, 2008.

[78] David Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818) :97–117, 1985.

[79] David Deutsch. Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818) :97–117, 1985.

[80] David Deutsch. Quantum Computational Networks. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 425(1868) :73–90, 1989.

[81] David Deutsch. *The Fabric of Reality*. Penguin, London, 1998.

[82] David Deutsch, Artur Ekert, and Rossella Lupacchini. Machines, Logic and Quantum Physics. *Bulletin of Symbolic Logic*, pages 265–283, 2000.

[83] Robert Devaney. *An Introduction to Chaotic Dynamical Systems, 2nd Edition*. Westview Pr, March 2003.

[84] Gilles Dowek. *Les métamorphoses du calcul : une étonnante histoire de mathématiques*, volume 21. Le Pommier, 2007.

[85] John Earman. *A Primer on Determinism*, volume 32. Springer, 1986.

[86] John Earman. *Bangs, Crunches, Wimpers, Shrieks*. Oxford : Oxford University Press, 1995.

[87] John Earman and John Norton. Forever is a day : Supertasks in Pitowsky and Malament-Hogarth spacetimes. *Philosophy of Science*, pages 22–42, 1993.

[88] John Earman and John Norton. Infinite Pains : the Trouble with Supertasks. *Benacerraf and His Critics*, 11 :271, 1996.

[89] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17(3) :449–467, 1965.

[90] Bertil Ekdahl. Interactive computing does not supersede Church's thesis. *Proc. Computer Science*, page 261–265, 1999.

[91] George F. R. Ellis, Roy Maartens, and Malcolm A. H. MacCallum. *Relativistic Cosmology*. Cambridge University Press, March 2012.

[92] Gábor Etesi and István Németi. Non-Turing computations via Malament-Hogarth space-times. *International Journal of Theoretical Physics*, 41(2) :341–370, 2002.

[93] William Bragg Ewald, editor. *From Kant to Hilbert. A Source Book in the Foundations of Mathematics.*, volume 2. Oxford University Press, Oxford, 1996.

[94] Edward Fahri, Jeffrey Goldstone, and Samuel Gutmann. A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science*, 292 :472–476, 2001. quant-ph/0104129.

[95] Marie Farge. Numerical Experimentation : A Third Way to Study Nature. In *Frontiers of Computational Science*, pages 15–28. Springer Verlag, 2007.

[96] Dror G. Feitelson. *Optical Computing*. The MIT Press, 1988.

[97] Richard P. Feynman. *The Character of Physical Law*. Penguin Books Limited, September 2007.

[98] Richard P. Feynmann. *Leçons sur l'informatique*. SCIENCES. Odile Jacob, 2006.

[99] Hartry Field. Apriority as an Evaluative Notion. In *New Essays on the A Priori*, pages 117–149. Oxford University Press, 2000.

[100] Lance Fortnow. The Enduring Legacy of the Turing Machine. *The Computer Journal*, 55(7) :830–831, 2012.

[101] Florent Franchette. La thèse de l'hyper-calcul : enjeux et problèmes philosophiques. *Philosophia Scientiae*, 16(3) :17–38, 2012.

[102] Edward Fredkin and Tommaso Toffoli. Conservative Logic. *International Journal of Theoretical Physics*, 21(3-4) :219–253, April 1982.

[103] Michael Freedman, Alexei Kitaev, Michael Larsen, and Zhenghan Wang. Topological Quantum Computation. *Bulletin of the American Mathematical Society*, 40(1) :31–38, 2003.

[104] Michael H. Freedman, Alexei Kitaev, and Zhenghan Wang. Simulation of Topological Field Theories by Quantum Computers. *Communications in Mathematical Physics*, 227(3) :587–603, June 2002.

[105] Michael Friedman. Geometry, Convention, and the Relativized A Priori : Reichenbach, Schlick, and Carnap. *Logic, language, and the structure of scientific theories, ed. Wesley Salmon, and Gereon Wolters*, pages 21–34, 1994.

[106] Michael Friedman. Transcendental philosophy and a priori knowledge : A neo-Kantian perspective. *New Essays on the A Priori, Clarendon Press, Oxford*, pages 367–383, 2000.

[107] Michael Friedman. *Dynamics of Reason*. CSLI Publications Stanford, 2001.

[108] Robin Gandy. Church's thesis and Principles for Mechanisms. *Studies in Logic and the Foundations of Mathematics*, 101 :123–148, 1980.

[109] Robert Geroch. *Perspectives in Computation*. University of Chicago Press, 2009.

[110] Robert Geroch and James B. Hartle. Computability and physical theories. *Foundations of Physics*, 16(6) :533–550, 1986.

[111] Jean-Yves Girard. Du pourquoi au comment : la théorie de la démonstration de 1950 à nos jours. *Development of Mathematics*, 2000 :515–546, 1950.

[112] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types, volume 7 of Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, 1989.

[113] N. Gisin. Weinberg's non-linear quantum mechanics and supraluminal communications. *Physics Letters A*, 143(1-2) :1–2, January 1990.

[114] Dina Goldin and Peter Wegner. The interactive nature of computing : Refuting the strong Church-Turing thesis. *Minds and Machines*, 18(1) :17–38, 2008.

[115] Oded Goldreich. *Computational Complexity*. Cambridge University Press, 2008.

[116] Naveen Sundar Govindarajulu and Selmer Bringsjord. The Myth of 'The Myth of Hypercomputation'. *Parallel Processing Letters*, 22(03) :1240012, September 2012.

[117] Serge Grigorieff. A remaining issue after Turing's work : formalize the notion of algorithm, May 2012.

[118] Andrzej Grzegorczyk. Computable functionals. *Fund. Math*, 42(19553) :168–202, 1955.

[119] Andrzej Grzegorczyk. On the definitions of computable real continuous functions. *Fund. Math*, 44 :61–71, 1957.

[120] Yukio-Pegio Gunji, Yuta Nishiyama, and Andrew Adamatzky. Robust Soldier Crab Ball Gate. *ArXiv e-prints*, 1204 :1749, April 2012.

[121] Yuri Gurevich. A new thesis. In *Abstracts, American Mathematical Society*, volume 6, page 317, 1985.

[122] Yuri Gurevich. Feasible functions. *London Mathematical Society Newsletter*, 206 :6–7, 1993.

[123] Yuri Gurevich. Sequential abstract-state machines capture sequential algorithms. *ACM Transactions on Computational Logic (TOCL)*, 1(1) :77–111, 2000.

[124] Yuri Gurevich. Abstract state machines : An overview of the project. In *Foundations of Information and Knowledge Systems*, pages 6–13. Springer, 2004.

[125] Yuri Gurevich. What is an algorithm ? Technical report, Microsoft Research, July 2011.

[126] Yuri Gurevich. Foundational Analyses of Computation. In *How the World Computes. Lecture Notes in Computer Science*, volume 7318, pages 264–275, Cambridge, UK, 2012. Springer Verlag.

[127] Amit Hagar. *The Complexity of Noise : A Philosophical Outlook on Quantum Error Correction.* Morgan & Claypool Publishers, January 2011.

[128] Amit Hagar and Alex Korolev. Quantum Hypercomputation-Hype or Computation ? *Philosophy of Science*, 74(3) :347–363, 2007.

[129] James B. Hartle. Excess Baggage. *arXiv preprint gr-qc/0508001*, 2005.

[130] Juris Hartmanis. On the Weight of Computations. *Bulletin of the EATCS*, 55, 1995.

[131] Stephen W. Hawking. *The Large Scale Structure of Space-Time.* Cambridge University Press, 1973.

[132] David Hilbert. Mathematische Probleme. *Nachrichten von der Königlichen Gesellschaft der Wissenschaften zu Göttingen, Math.-Phys. Klasse*, pages 253–297, 1900.

[133] Shahar Hod. Weak cosmic censorship : as strong as ever. *Physical Review Letters*, 100(12) :121101, 2008.

[134] Shahar Hod. Cosmic censorship : Formation of a shielding horizon around a fragile horizon. *Physical Review D*, 87(2) :024037, 2013.

[135] Andrew Hodges. Can quantum computing solve classically unsolvable problems ? *arXiv preprint quant-ph/0512248*, 2005.

[136] Andrew Hodges. Did Church and Turing have a Thesis about Machines ? *Church's Thesis after*, 70 :242–252, 2006.

[137] Douglas Hofstadter. *Gödel Escher Bach. Les Brins d'une Guirlande Eternelle.* Dunod, Paris, 2000. Version française de Jacqueline Henry et Robert French.

[138] Mark Hogarth. Does general relativity allow an observer to view an eternity in a finite time ? *Foundations of Physics Letters*, 5(2) :173–181, 1992.

[139] Mark Hogarth. Non-Turing Computers and Non-Turing Computability. In *PSA : Proceedings of the Biennial Meeting of the Philosophy of Science Association*, pages 126–138, 1994.

[140] Gerard 't Hooft. Quantum gravity as a dissipative deterministic system. *Classical and Quantum Gravity*, 16(10) :3263, 1999.

[141] Russell Impagliazzo. A Personal View of Average-Case Complexity. In *Structure in Complexity Theory Conference, 1995., Proceedings of Tenth Annual IEEE*, pages 134–147, 1995.

[142] Jean-Baptiste Joinet. Sur le temps logique. In *Logique, dynamique et cognition*, Logique, language, sciences, philosophie, pages 31–49. Publications de la Sorbonne, Paris, September 2007.

[143] Jean-Baptiste Joinet and Samuel Tronçon. Ouvrir la logique au monde. *Philosophie et mathématique de l'interaction, Actes de l'École*, 2009.

[144] Richard Jozsa and Noah Linden. On the role of entanglement in quantum-computational speed-up. *Proceedings of the Royal Society of London. Series A : Mathematical, Physical and Engineering Sciences*, 459(2036) :2011–2032, 2003.

[145] Tien D. Kieu. Quantum adiabatic algorithm for Hilbert's tenth problem : I. The algorithm. *arXiv preprint quant-ph/0310052*, 2003.

[146] Tien D. Kieu. Hypercomputability of Quantum Adiabatic Processes : Fact versus Prejudices. *arXiv :quant-ph/0504101*, April 2005.

[147] Tien D. Kieu. On the identification of the ground state based on occupation probabilities : An investigation of Smith's apparent counterexamples. *arXiv :quant-ph/0602145*, February 2006.

[148] Tien D. Kieu. Reply to Andrew Hodges. *arXiv :quant-ph/0602214*, February 2006.

[149] A. Yu Kitaev. Fault-Tolerant Quantum Computation by Anyons. *arXiv preprint quant-ph/9707021*, 1997.

[150] Philip Kitcher. *The Nature of Mathematical Knowledge*. Cambridge University Press, 1984.

[151] Philip Kitcher. A Priori Knowledge Revisited. In *New Essays on the A Priori*, pages 65–91. Oxford University Press, 2000.

[152] Donald E. Knuth. *Art of Computer Programming, Volume 1 : Fundamental Algorithms*. Addison-Wesley Professional, third edition, July 1997.

[153] Ker-I. Ko. *Complexity theory of real functions*. Birkhauser Boston Inc., 1991.

[154] Jean-Louis Krivine. Fonctions, programmes et démonstrations. *Gaz. Math. Soc. Math. Fr*, 60 :63–f73, 1994.

[155] Jean-Louis Krivine. Mathématiques des programmes et programme des mathématiques. *Turbulence*, 1(1) :94–100, 1994.

[156] Boris A. Kushner. Markov's constructive analysis ; a participant's view. *Theoretical Computer Science*, 219(1) :267–285, 1999.

[157] Boris Abramovich Kushner. *Lectures on constructive mathematical analysis*, volume 60 of *Translation of Mathematical Monographs*. American Mathematical Society, 1984.

[158] Daniel Lacombe. Extension de la notion de fonction récursive aux fonctions d'une ou plusieurs variables réelles III. *CR Acad. Sci. Paris*, 241 :151–153, 1955.

[159] Leonid Levin. Average Case Complete Problems. *SIAM Journal on Computing*, 15(1) :285–286, February 1986.

[160] Leonid Levin. The Tale of One-Way Functions. *Problems of Information Transmission.*, 39(1) :92–103, 2003.

[161] Ignazio Licata. Beyond Turing : Hypercomputation and Quantum Morphogenesis. *Asia Pacific Mathematics Newsletter*, 2(3), July 2012.

[162] Seth Lloyd. Ultimate Physical Limits to Computation. *arXiv :quant-ph/9908043*, August 1999.

[163] Seth Lloyd. Computational capacity of the universe. *arXiv :quant-ph/0110141*, October 2001. Phys.Rev.Lett.88 :237901,2002.

[164] Leopold Löwenheim. Über Möglichkeiten im Relativkalkül. *Mathematische Annalen*, 76(4) :447–470, 1915.

[165] Penelope Maddy. Naturalism and the A Priori. *New Essays on the A Priori*, pages 92–116, 2000.

[166] Norman Margolus and Lev B. Levitin. The maximum speed of dynamical evolution. *Physica D : Nonlinear Phenomena*, 120(1) :188–195, 1998.

[167] Jean Marguin. *Histoire des instruments et machines à calculer : trois siècles de mécanique pensante, 1642-1942*. Hermann, 1994.

[168] Yuri V. Matiyasevich. Enumerable sets are diophantine. *Doklady Akademii Nauk SSSR*, 191(2) :279–282, 1970.

[169] Richard Montague. *Towards a General Theory of Computability*. Springer, 1962.

[170] Cristopher Moore. Unpredictability and undecidability in dynamical systems. *Physical Review Letters*, 64(20) :2354–2357, 1990.

[171] Cristopher Moore. Recursion theory on the reals and continuous-time computation. *Theoretical Computer Science*, 162(1) :23–44, 1996.

[172] Jerzy Mycka. μ-recursion and infinite limits. *Theoretical Computer Science*, 302(1–3) :123–133, June 2003.

[173] Jerzy Mycka and José Félix Costa. Real recursive functions and their hierarchy. *Journal of Complexity*, 20(6) :835–857, 2004.

[174] Istvan Németi. *On logic, relativity, and the limitations of human knowledge. Iowa State University, Department of Mathematics*. PhD thesis, Ph. D. course during the academic year 1987/88, 1987.

[175] István Németi and Gyula Dávid. Relativistic computers and the Turing barrier. *Applied Mathematics and Computation*, 178(1) :118–142, 2006.

[176] Péter Németi and Gergely Székely. Existence of faster than light signals implies hypercomputation already in special relativity. In *How the World Computes*, pages 528–538. Springer, 2012.

[177] Michael A. Nielsen. Computable Functions, Quantum Measurements, and Quantum Dynamics. *Physical Review Letters*, 79(15) :2915, 1997.

[178] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge series on information and the natural sciences. Cambridge University Press, 2000.

[179] John Norton. Einstein for Everyone. Online Course.

[180] R. Walter Ogburn and John Preskill. Topological Quantum Computation. In *Quantum Computing and Quantum Communications*, pages 341–356. Springer, 1999.

[181] Toby Ord. The Many Forms of Hypercomputation. *Applied Mathematics and Computation*, 178(1) :143–153, 2006.

[182] Toby Ord and Tien D. Kieu. Using Biased Coins as Oracles. *arXiv preprint cs/0401019*, 2004.

[183] Christos H. Papadimitriou. *Computational Complexity*. Addison Wesley Longman, 1995.

[184] Michel Parigot. On the representation of data in lambda-calculus. In Egon Börger, Hans Kleine Büning, and Michael M. Richter, editors, *CSL '89*, number 440 in Lecture Notes in Computer Science, pages 309–321. Springer Berlin Heidelberg, January 1990.

[185] Gheorghe Paun, Grzegorz Rozenberg, and Arto Salomaa. *DNA Computing : New Computing Paradigms*. Springer, January 1998.

[186] Roger Penrose. Gravitational Collapse : The Role of General Relativity. Technical report, Birkbeck Coll., London, 1969.

[187] Roger Penrose. Gravitational Collapse. *Gravitational Radiation and Gravitational Collapse*, 64 :82–91, 1974.

[188] Roger Penrose. *The Emperor's New Mind Concerning Computers, Minds, and the Laws of Physics*. Oxford University Press, Oxford, 1989.

[189] Roger Penrose. The Question of Cosmic Censorship. *Journal of Astrophysics and Astronomy*, 20(3-4) :233–248, 1999.

[190] Gualtiero Piccinini. The Physical Church-Turing Thesis : Modest or Bold ? *The British Journal for the Philosophy of Science*, 62(4) :733–769, 2011.

[191] J. L. Pienaar, T. C. Ralph, and C. R. Myers. Open Timelike Curves Violate Heisenberg's Uncertainty Principle. *Physical Review Letters*, 110(6) :060501, February 2013.

[192] Itamar Pitowsky. The Physical Church Thesis and Physical Computational Complexity. *Iyyun*, 39 :81–99, 1990.

[193] Emil L. Post. Finite combinatory processes-formulation 1. *The Journal of Symbolic Logic*, 1(3) :103–105, 1936.

[194] Marian B. Pour-El and J. Ian Richards. *Computability in analysis and physics*. Springer Verlag, 1989.

[195] Michael Prasse and Peter Rittgen. Why Church's Thesis Still Holds. Some Notes on Peter Wegner's Tracts on Interaction and Computability. *The Computer Journal*, 41(6) :357–362, January 1998.

[196] Hilary Putnam. The Analytic and the Synthetic. In *Philosophical Papers. Volume 2. Mind, Language, and Reality*, pages 33–69. Cambridge University Press, 1975.

[197] Hilary Putnam. The Logic of Quantum Mechanics. *Boston Studies in the Philosophy of Science*, 5, 1975.

[198] Willard Van Orman. Quine. Two Dogmas of Empiricism. *From a Logical Point of View*, pages 2–46, 1963.

[199] Robert Raussendorf, Daniel E. Browne, and Hans J. Briegel. Measurement-based quantum computation on cluster states. *Physical Review A*, 68(2) :022312, August 2003.

[200] Peter Selinger. A brief survey of quantum programming languages. In *Functional and Logic Programming*, pages 1–6. Springer, 2004.

[201] Peter Selinger and Benoit Valiron. A lambda calculus for quantum computation with classical control. *Mathematical Structures in Computer Science*, 16(3) :527–552, 2006.

[202] Oron Shagrir and Itamar Pitowsky. Physical Hypercomputation and the Church–Turing Thesis. *Minds and Machines*, 13(1) :87–101, 2003.

[203] Adi Shamir. Factoring Numbers in O(log n) Arithmetic Steps. *Inf. Process. Lett.*, 8(1) :28–31, 1979.

[204] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Computing*, pages 1484–1509, 1997.

[205] Wilfried Sieg. Calculations by Man & Machine : Mathematical Presentation. In *Proceedings of the Cracow International Congress of Logic, Methodology and Philosophy of Science*, pages 245–260. Kluwer Academic Publishers, 2002.

[206] Wilfried Sieg. *Reflections on the Foundations of Mathematics*, chapter Calculations by Man and Machine : Conceptual Analysis, pages 396–415. 2002.

[207] Wilfried Sieg. Church without dogma : Axioms for computability. *New computational paradigms*, pages 139–152, 2008.

[208] Wilfried Sieg and John Byrnes. An abstract model for parallel computations : Gandy's thesis. *The Monist*, 82(1) :150–164, 1999.

[209] Hava T Siegelmann. Computation beyond the Turing limit. *Science*, 268(5210) :545–548, 1995.

[210] Hava T. Siegelmann. *Neural Networks and Analog Computation : Beyond the Turing Limit*, volume 20. Birkhäuser Boston, 1998.

[211] Daniel Silva Graça and José Félix Costa. Analog Computers and Recursive Functions over the Reals. *Journal of Complexity*, 19(5) :644–664, 2003.

[212] Peter Smith. *An Introduction to Gödel's Theorems*. Cambridge University Press, 2007.

[213] Warren D. Smith. Three counterexamples refuting Kieu's plan for "quantum adiabatic hypercomputation" ; and some uncomputable quantum mechanical tasks. *Applied Mathematics and Computation*, 178(1) :184–193, July 2006.

[214] Elliott Sober. Quine's Two Dogmas. In *Proceedings of the Aristotlean Society*, volume 74, pages 237–280, 2000.

[215] Matthew C. Spencer, Etienne B. Roesch, Slawomir J. Nasuto, Thomas Tanay, and J. Mark Bishop. Abstract Plateforms of Computation. In *Proceedings of the AISB 2013 Convention*, April 2013. À paraître.

[216] Michael Stannett. The Case for Hypercomputation. *Applied Mathematics and Computation*, 178(1) :8–24, 2006.

[217] Mike Stannett. Hypercomputational Models. In *Alan Turing : Life and legacy of a great thinker*, pages 135–157. Springer, 2004.

[218] Charles A. Stewart. *On the formulae-as-types correspondence for classical logic*. PhD thesis, University of Oxford, 2000.

[219] Larry Stockmeyer and Albert R. Meyer. Cosmological lower bound on the circuit complexity of a small problem in logic. *Journal of the ACM (JACM)*, 49(6) :753–784, 2002.

[220] Alan Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42 :230–265, 1936.

[221] Alan Turing. Computability and λ-definability. *The Journal of Symbolic Logic*, 2(4) :153–163, 1937.

[222] Alan Turing. Systems of logic based on ordinals. *Proceedings of the London Mathematical Society*, 2(1) :161–228, 1939.

[223] Alan Turing. Computing machinery and intelligence. *Mind*, 59(236) :433–460, 1950.

[224] Alan Turing. Programmers' handbook for Manchester electronic computer, 1950.

[225] Alan Turing. Can digital computers think ? *The Essential Turing*, pages 482–486, 2004.

[226] Alan Turing. Intelligent machinery (1948). *The Essential Turing*, pages 395–432, 2004.

[227] Alan Turing. Intelligent machinery (1948). *Jack Copeland*, page 395, 2004.

[228] Alan Turing. Lecture on the Automatic Computing Engine (1947). *The Essential Turing*, pages 362–394, 2004.

[229] Wim Van Dam, Michele Mosca, and Umesh Vazirani. How powerful is adiabatic quantum computation ? In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium*, pages 279–287, 2001.

[230] Peter van Emde Boas. Machines Models and Simulation. In *Handbook of Theoretical Computer Science*, volume A. Algorithms and complexity. The MIT Press, 1990.

[231] Jan Van Leeuwen and Jirí Wiedermann. Breaking the Turing barrier : the case of the Internet. *Techn. Report, Inst. of Computer Science, Academy of Sciences of the Czech Rep., Prague*, 2000.

[232] Jan van Leeuwen and Jirí Wiedermann. On algorithms and interaction. In *Mathematical Foundations of Computer Science 2000*, pages 99–113. Springer, 2000.

[233] Jan van Leeuwen and Jirí Wiedermann. Beyond the Turing limit : evolving interactive systems. In *SOFSEM 2001 : Theory and Practice of Informatics*, pages 90–109, 2001.

[234] André van Tonder. A lambda calculus for quantum computation. *SIAM Journal on Computing*, 33(5) :1109–1135, 2004.

[235] Philip Wadler. Church's Coincidences. Keynote SICSA PhD Conference, 2012.

[236] Jie Wang. Average-Case Complexity Forum.

[237] John Watrous. On one-dimensional quantum cellular automata. In *Foundations of Computer Science, 1995. Proceedings, 36th Annual Symposium*, pages 528–537, 1995.

[238] Peter Wegner. Why Interaction is More Powerful Than Algorithms. *Communications of the ACM*, May 1997.

[239] Peter Wegner and Dina Goldin. *Interaction, Computability, and Church's Thesis.* Draft, 1999.

[240] Peter Wegner and Dina Golding. The Church-Turing Thesis : Breaking the Myth. In *LNCS 3526*, pages 152–168, Amsterdam, 2005. Springer.

[241] Klaus Weihrauch. *Computable Analysis : an Introduction.* Springer, 2000.

[242] Hermann Weyl. *Philosophy of Mathematics and Natural Science. Revised and augmented English edition based on a translation by Olaf Helmer.* Princeton University Press, Princeton, NJ, 1949.

[243] Andrew Chi-Chih Yao. Quantum Circuit Complexity. In *Proceedings of the Thirty-Fourth IEEE Symposium on Foundations of Computer Science*, pages 352–361, Palo Alto, California, 1993.

[244] Andrew Chi-Chih Yao. Classical physics and the Church-Turing Thesis. *Journal of the ACM (JACM)*, 50(1) :100–105, 2003.