

UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

ÉCOLE DOCTORALE SCIENCES TECHNOLOGIE SANTE (547)

# THÈSE

Pour obtenir le grade de

**DOCTEUR DE L'UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE**

***Discipline : MATHÉMATIQUES***

Présentée et soutenue publiquement par

**Olivia MIRAUCOURT**

Le 3 novembre 2016

---

## **Génération de modèles vasculaires cérébraux : Segmentation de vaisseaux et simulation d'écoulements sanguins**

---

Thèse dirigée par **Stéphanie SALMON**

### **JURY**

Mme. Maïtine BERGOUNIOUX,	, Professeur,	à L'Université d'Orléans,	, <b>Président</b>
M. Frédéric HECHT,	, Professeur,	à l'Université Paris 6 Pierre et Marie Curie,	, <b>Rapporteur</b>
M. Gabriel PEYRE,	, Directeur de Recherche,	à l'Université Paris 9,	, <b>Rapporteur</b>
M. Nicolas PASSAT,	, Professeur,	à l'Université Reims Champagne-Ardenne,	, <b>Examineur</b>
Mme. Stéphanie SALMON,	, Professeur,	à l'Université Reims Champagne-Ardenne,	, <b>Examineur</b>
Mme. Marcela SZOPOS,	, Maître de Conférences ,	à l'Université de Strasbourg,	, <b>Examineur</b>
M. Hugues TALBOT,	, Professeur,	ESIEE Noisy Le Grand,	, <b>Examineur</b>
Mme. Valérie WOLFF,	, Maître de Conférences HDR,	à l'Université de Strasbourg,	, <b>Examineur</b>





*À ma marraine.*



# Remerciements

Je suis très heureuse d'adresser mes premiers remerciements aux trois professeurs qui m'ont encadré durant ces quatre années, Stéphanie Salmon pour la partie simulation, Hugues Talbot pour la partie segmentation, et Nicolas Passat qui a permis de faire le lien. Je les remercie pour leurs vastes connaissances qu'ils m'ont fait bénéficier, mais aussi pour leurs encouragements, leur aide et leur sympathie. Ils forment une belle équipe, et j'espère qu'ils poursuivront leur collaboration en encadrant de nouveaux doctorants.

Un jour, Hugues m'a dit qu'il y avait trois grandes étapes avant de maîtriser un problème : comprendre tout d'abord ce problème, être capable ensuite de le reformuler sur papier, et enfin de le réexpliquer à l'oral. C'est cette dernière étape qui m'a intéressé davantage, et c'est pourquoi je me destine au métier d'enseignante. J'ai eu la chance d'avoir des encadrants qui m'ont soutenu pour ce projet et je leur suis très reconnaissante.

Je tiens aussi à remercier Frédéric Hecht et Gabriel Peyré d'avoir tous deux accepté de rapporter mon mémoire de thèse. Merci à Frédéric Hecht d'organiser les journées FreeFem++ qui m'ont permis de débloquent plusieurs codes. Merci à Gabriel Peyré de m'avoir donné l'idée de créer le site "The Numerical Tour of Fluid Dynamics", fournissant nos codes de simulation à la communauté CFD, grâce à son site "A Numerical Tour of Signal Processing". Merci également à Marcela Szopos, Maïtine Bergounioux et Valérie Wolff qui me font l'honneur de participer au jury. Un merci particulier à Marcela qui m'a aidé à de nombreuses reprises.

Je remercie également tous les membres du Laboratoire de Mathématiques de Reims pour leur accueil. Merci à mes collègues d'enseignement Michel Percy, Lisette Jager, Odile Fleury-Barka et Laurent Amour qui m'ont guidé lors de mon contrat doctoral élargi et de mon ATER. Un grand merci à Loïc, Rupert, Philippe, Alain, Lisette, Laurent et Thierry pour leur soutien et les bons moments passés ensemble. Merci profondément à Christelle pour toute sa bienveillance en mon égard, et Julie pour sa bonne humeur.

Merci à mes collègues doctorants (anciens ou actuels) Elena, Emna, Océane, Anthony, Mathieu, Manuel, David, Alexandre, Grégory et Mohamed pour tous les moments de détente et l'ambiance de travail très agréable. Un grand merci à Mathieu, Anthony, David et Océane pour m'avoir aidé à mettre en place le séminaire Jeunes Chercheurs du Laboratoire, qui nous a permis de rencontrer des doctorants d'autres horizons. Un merci particulier à Anthony qui me supporte depuis le début de cette thèse et qui a un don pour remonter le moral des troupes. Merci également aux jeunes chercheurs du projet Vivabrain, avec lesquels il a été très plaisant de travailler, Simon, Odyssée, Ranine, Anna, Alice, Alexandre F. et Alexandre A.

Je tiens à remercier chaleureusement mes amis les plus proches, qui se sont intéressés à l'avancement de ma thèse (ou alors ont très bien su faire semblant), Stéphanie, Etienne, Arnaud, Armelle, Kevin, Elodie et Angélique. Enfin, j'adresse un grand merci à toute ma famille qui a toujours été à mes côtés lorsque j'en ai eu besoin, en particulier merci à mes parents Chantal et Daniel, ma soeur Delphine et mon frère Bryan. Merci à tous ceux qui ont participé de près comme de loin à ce travail, et que j'ai omis involontairement de citer.

Finalement, il me reste à remercier mon compagnon Nicolas pour sa patience, son réconfort et son optimisme qui me donnent la force de me surpasser. Sans lui, la concrétisation de cette thèse aurait été impossible et il me tarde maintenant de commencer une nouvelle page à trois.

## Résumé

Ce travail a pour objectif de générer des modèles vasculaires et de simuler des écoulements sanguins réalistes à l'intérieur de ces modèles. La première étape consiste à segmenter/reconstruire le volume 3D du réseau vasculaire. Une fois de tels volumes vasculaires segmentés et maillés, il est alors possible de simuler des écoulements sanguins à l'intérieur de ceux-ci. Pour la segmentation, nous utilisons une approche variationnelle. Nous proposons un premier modèle qui inclut un a priori de tubularité dans les modèles de débruitage ROF et TV-L1. Néanmoins, bien que ces modèles permettent de réhausser les vaisseaux, ils ne permettent pas de les segmenter. C'est pourquoi nous proposons un deuxième modèle amélioré qui inclut à la fois un a priori de tubularité et de direction dans le modèle de segmentation de Chan-Vese. Les résultats sont présentés sur des images synthétiques 2D, ainsi que sur des images rétiniennes. En ce qui concerne la simulation, nous nous intéressons d'abord au réseau veineux cérébral, encore peu étudié. Les équations de la dynamique des fluides qui régissent les écoulements sanguins dans notre géométrie sont alors les équations de Navier-Stokes. Pour résoudre ces équations, la méthode classique des caractéristiques est comparée avec un schéma d'ordre plus élevé. Ces deux schémas sont validés sur des solutions analytiques avant d'être appliqués aux cas réalistes du réseau veineux cérébral premièrement, puis du polygone artériel de Willis.

**Mots clés :** segmentation de vaisseaux, modèles variationnels, mesures de tubularité, simulation d'écoulements sanguins, équations de Navier-Stokes, réseau vasculaire cérébral.

## Abstract

The aim of this work is to generate vascular models and simulate blood flows inside these models. A first step consists of segmenting/reconstructing the 3D volume of the vascular network. Once such volumes are segmented and meshed, it is then possible to simulate blood flows. For segmentation purposes, we use a variational approach. We first propose a model that embeds a vesselness prior in the denoising models ROF and TV-L1. Although these models can enhance vessels, they are not designed for segmentation. Then, we propose a second, improved model that includes both vesselness and direction priors in the Chan-Vese segmentation model. The results are presented on 2D synthetic images, as well as retinal images. In the second part, devoted to simulation, we first focus on the cerebral venous network, that has not been intensively studied. The equations governing blood flows inside our geometry are the Navier-Stokes equations. For their resolution, the classical method of characteristics is compared with a high-order scheme. Both schemes are validated on analytical solutions before their application on the realistic cases of the cerebral venous network, and the arterial polygone of Willis.

**Keywords:** vessel segmentation, variational models, vesselness, blood flow simulation, Navier-Stokes equations, cerebral vascular network.



# Table des matières

Liste des figures	xiii
Liste des tableaux	xvii
Liste des acronymes	xviii
Liste des symboles	xix
Introduction	1
<b>I Segmentation d'images angiographiques</b>	<b>5</b>
<b>1 État de l'art sur les méthodes variationnelles</b>	<b>7</b>
1.1 Introduction	8
1.2 Modèles de débruitage	11
1.2.1 Modèle de Tikhonov	11
1.2.2 Modèle ROF	13
1.2.3 Modèle TV-L1	14
1.3 Modèles de segmentation	15
1.3.1 Modèles de contours actifs	15
1.3.2 Modèles d'intensité	18
<b>2 Vue d'ensemble sur les algorithmes d'optimisation</b>	<b>23</b>
2.1 Introduction	25
2.2 Algorithmes de programmation linéaire	28
2.2.1 Méthode du simplexe	28
2.2.2 Méthode de points intérieurs	29
2.3 Algorithmes à directions de descente	30
2.3.1 Recherche linéaire	30
2.3.2 Algorithme du gradient	32
2.3.3 Algorithme du gradient conjugué	32
2.3.4 Algorithme de Newton	34
2.3.5 Algorithme de quasi-Newton	34
2.4 Algorithmes de minimisation sous contraintes	36
2.4.1 Algorithme du gradient projeté	36
2.4.2 Algorithme de pénalisation	37
2.4.3 Méthode d'Uzawa	38
2.5 Algorithmes d'optimisation convexe	39
2.5.1 Notion de sous-différentielle	40
2.5.2 Opérateur proximal : définition et exemples	41

2.5.3	Opérateur proximal de la variation totale . . . . .	42
2.5.4	Algorithme explicite-implicite (FB : Forward-Backward) . . . . .	46
2.5.5	Algorithme de Douglas-Rachford (DR) . . . . .	48
2.5.6	Algorithme Proximal Parallèle (PPXA) . . . . .	51
2.5.7	Algorithme Primal-Dual . . . . .	51
2.5.8	Application au débruitage d'images . . . . .	53
<b>3</b>	<b>Filtrage et segmentation des vaisseaux</b>	<b>57</b>
3.1	Introduction . . . . .	59
3.2	Tour d'horizon des méthodes de filtrage et segmentation des vaisseaux . . .	61
3.2.1	Croissance de régions . . . . .	61
3.2.2	Segmentation basée sur des modèles . . . . .	62
3.2.3	Recherche de chemins . . . . .	63
3.2.4	Le suivi de vaisseaux . . . . .	63
3.2.5	Morphologie mathématique . . . . .	64
3.2.6	Approches statistiques . . . . .	64
3.2.7	Filtrage bilatéral . . . . .	64
3.2.8	Filtrage non-local . . . . .	65
3.3	Filtrage basé sur les dérivées . . . . .	66
3.3.1	L'opérateur gradient . . . . .	66
3.3.2	L'opérateur hessien . . . . .	67
3.3.3	Mesures de tubularité (vesselness) . . . . .	70
3.3.4	Tenseur de structures . . . . .	71
<b>4</b>	<b>Inclusion de la tubularité dans le modèle variationnel</b>	<b>73</b>
4.1	Introduction . . . . .	74
4.2	Inclusion de la tubularité dans les modèles de débruitage ROF et TV-L1 . . .	74
4.2.1	Méthodologie . . . . .	74
4.2.2	Algorithme primal-dual . . . . .	75
4.2.3	Résultats . . . . .	77
4.3	Inclusion de la tubularité et des directions dans le modèle de segmentation de Chan-Vese . . . . .	81
4.3.1	Inclusion de la tubularité . . . . .	81
4.3.2	Inclusion de la tubularité et des directions . . . . .	86
<b>5</b>	<b>Conclusion et perspectives</b>	<b>95</b>
5.1	Bilan . . . . .	96
5.1.1	Inclusion de la tubularité dans les modèles de débruitage ROF et TV-L1	96
5.1.2	Inclusion de la tubularité et des directions dans les modèles de seg- mentation de Chan-Vese . . . . .	96
5.2	Perspectives . . . . .	97
5.2.1	Modèle de Chan avec les paramètres $c_1$ et $c_2$ spatialement variants .	97
5.2.2	Gradient directionnel à voisinage étendu . . . . .	98
5.2.3	Segmentation 3D . . . . .	99
<b>II</b>	<b>Simulation d'écoulements sanguins</b>	<b>101</b>
<b>6</b>	<b>Modélisation pour la simulation d'écoulements sanguins cérébraux</b>	<b>103</b>
6.1	Introduction . . . . .	104
6.2	Modélisation de la géométrie des vaisseaux . . . . .	107

6.3	Modèle anatomique des réseaux veineux et artériel . . . . .	108
6.3.1	Réseau veineux cérébral . . . . .	108
6.3.2	Réseau artériel cérébral . . . . .	109
6.4	Modèle biomécanique . . . . .	112
6.4.1	Hypothèses de travail . . . . .	112
6.4.2	Analyse dimensionnelle . . . . .	113
6.4.3	Conditions initiales et limites . . . . .	114
6.4.4	Modèle final . . . . .	116
<b>7</b>	<b>Méthodes de résolution des équations de Stokes et Navier-Stokes</b>	<b>117</b>
7.1	Introduction . . . . .	118
7.2	Résolution du problème de Stokes . . . . .	119
7.2.1	Formulation variationnelle . . . . .	119
7.2.2	Discrétisation . . . . .	120
7.3	Résolution du problème de Navier-Stokes . . . . .	122
7.3.1	Méthode des caractéristiques . . . . .	122
7.3.2	Méthode d'Uzawa . . . . .	124
7.3.3	Schéma découplé d'ordre élevé . . . . .	126
<b>8</b>	<b>Validation numérique sur des solutions analytiques</b>	<b>131</b>
8.1	Introduction . . . . .	132
8.2	Problème de Stokes . . . . .	132
8.2.1	K-test d'Ethier-Steinman . . . . .	133
8.2.2	Cas-test de Bercovier-Engelman . . . . .	135
8.2.3	Cas-test de Minev-Ethier . . . . .	136
8.2.4	Courbes de convergence . . . . .	137
8.3	Problème de Navier-Stokes . . . . .	139
8.3.1	Cas-test d'Ethier-Steinman . . . . .	139
8.3.2	Écoulement de Poiseuille . . . . .	143
<b>9</b>	<b>Application à des géométries réalistes</b>	<b>149</b>
9.1	Introduction . . . . .	150
9.2	Réseau veineux cérébral . . . . .	150
9.2.1	Problème de Stokes . . . . .	150
9.2.2	Problème de Navier-Stokes . . . . .	151
9.3	Polygone de Willis . . . . .	157
<b>10</b>	<b>Conclusion et perspectives</b>	<b>161</b>
10.1	Bilan . . . . .	162
10.1.1	Modélisation . . . . .	162
10.1.2	Algorithmes . . . . .	163
10.2	Perspectives . . . . .	164
10.2.1	Modélisation . . . . .	164
10.2.2	Algorithmes . . . . .	164
10.2.3	Validation . . . . .	164
10.2.4	Liaison entre segmentation et simulation . . . . .	164
10.2.5	Application à des cas cliniques . . . . .	164

<b>III Annexes</b>	<b>I</b>
<b>A Transformée de Legendre-Fenchel</b>	<b>III</b>
A.1 Transformée de Legendre vs transformée de Legendre-Fenchel . . . . .	III
A.2 Théorème de Fenchel-Moreau . . . . .	IV
<b>B Comparaison FreeFem++ et Feel++</b>	<b>VII</b>
<b>C Liste des publications et communications issues des travaux de thèse</b>	<b>XI</b>
C.1 Publications . . . . .	XI
C.1.1 Articles de journal . . . . .	XI
C.1.2 Articles dans des actes de conférences avec comité de lecture . . . . .	XI
C.2 Communications . . . . .	XII
C.2.1 Posters . . . . .	XII
C.2.2 Communications orales . . . . .	XII
C.3 Autres . . . . .	XII
<b>Bibliographie</b>	<b>XIII</b>

# Liste des figures

1	Exemples de maladies cérébrovasculaires . . . . .	2
2	Contexte de la thèse : projet ANR VIVABRAIN . . . . .	3
1.1	Principe de Fermat . . . . .	8
1.2	Courbe brachistochrone . . . . .	9
1.3	Comparaison de différents modèles variationnels pour le débruitage . . . . .	12
1.4	Comparaison de la régularisation quadratique et de la variation totale . . . . .	13
1.5	Modèle de contour actif snake . . . . .	16
1.6	Courbes de niveau . . . . .	17
1.7	Changement topologique à l'aide de la formulation en courbes de niveau . . . . .	17
1.8	Fidélité du modèle de Chan-Vese . . . . .	19
1.9	Comparaison de différents modèles d'intensité . . . . .	20
2.1	Exemple de minimum d'un problème différentiable, non-convexe et sans contrainte . . . . .	25
2.2	Fonction convexe (interprétation graphique) . . . . .	28
2.3	Méthode du simplexe . . . . .	29
2.4	Méthode de points intérieurs . . . . .	29
2.5	Fonction de Rosenbrock . . . . .	30
2.6	Algorithme du gradient . . . . .	32
2.7	Algorithme du gradient conjugué . . . . .	34
2.8	Algorithme de Newton . . . . .	35
2.9	Algorithme de quasi-Newton . . . . .	36
2.10	Fonction semi-continue inférieurement . . . . .	40
2.11	Fonction convexe (définition) . . . . .	40
2.12	Exemples de sous-différentielle . . . . .	41
2.13	Projection sur un ensemble convexe . . . . .	41
2.14	Seuillage doux . . . . .	43
2.15	Forward-Backward versus Douglas-Rachford . . . . .	50
2.16	Comparaison des différents algorithmes proximaux pour le débruitage de l'image test de Lena . . . . .	54
2.17	Convergence des différents algorithmes proximaux . . . . .	55
3.1	Segmentation des vaisseaux sanguins de la rétine . . . . .	59
3.2	Segmentation des neurites du cerveau antérieur d'un rat . . . . .	60
3.3	Segmentation des fibres du muscle plantaire chez l'humain . . . . .	60
3.4	Croissance de régions pour la segmentation vasculaire . . . . .	62
3.5	Traversée d'un vaisseau par une séquence de superellipsoïdes . . . . .	62
3.6	Suivi de vaisseaux . . . . .	63
3.7	Débruitage par moyennes non-locales . . . . .	65
3.8	Gradient d'un contour . . . . .	67

3.9	Comparaison des différents filtres de détection de contours . . . . .	67
3.10	Comparaison du filtre hessien et du filtre hessien lissé . . . . .	68
3.11	Ellipsoïdes représentant la structure de l'image . . . . .	69
3.12	Vesselness de Frangi . . . . .	71
4.1	Modèles ROF et TV-L1 pondérés par la vesselness appliqués à une image synthétique . . . . .	79
4.2	Modèles ROF et TV-L1 pondérés par la vesselness appliqués à une image rétinienne . . . . .	80
4.3	Comparaison des algorithmes de TV, GP et FGP pour le débruitage d'une image synthétique 2D . . . . .	83
4.4	Convergence des algorithmes de TV, GP et FGP . . . . .	84
4.5	Modèles Chan classique et Chan pondéré par la vesselness appliqués à une image synthétique 2D . . . . .	84
4.6	Modèles Chan classique et Chan pondéré par la vesselness appliqués à une image rétinienne (image 1 de la base DRIVE) . . . . .	85
4.7	Modèles Chan classique et Chan pondéré par la vesselness appliqués à une image rétinienne (image 3 de la base DRIVE) . . . . .	85
4.8	Principe de régularisation dans le cas de la TV classique, la TV pondérée par la vesselness et la TV directionnelle . . . . .	86
4.9	Famille de vecteurs $(\nu_1, \nu_2, \nu_3, \nu_4)$ sur laquelle le gradient directionnel est projeté . . . . .	87
4.10	Comparaison des gradients classique et directionnel sur une image synthétique 2D . . . . .	87
4.11	Décomposition du vecteur direction sur chaque vecteur de la famille $(\nu_1, \nu_2, \nu_3, \nu_4)$ . . . . .	88
4.12	Comparaison des directions extraites de la vesselness de Frangi et du filtre RORPO. . . . .	90
4.13	Comparaison des modèles Chan classique, Chan pondéré par la vesselness et Chan incluant les directions . . . . .	91
4.14	Modèles Chan classique, Chan incluant les directions extraites de la vesselness de Frangi et extraites de RORPO appliqués à une image rétinienne (image 1 de la base DRIVE) . . . . .	92
4.15	Modèles Chan classique, Chan incluant les directions extraites de la vesselness de Frangi et extraites de RORPO appliqués à une image rétinienne (image 3 de la base DRIVE) . . . . .	93
5.1	Premières expérimentations pour les paramètres $c_1$ et $c_2$ spatialement variants (image 1 de la base DRIVE) . . . . .	98
5.2	Familles de vecteurs pour différentes tailles de voisinage . . . . .	99
5.3	Segmentation 3D : Image synthétique 3D et orientations possible en 3D . . . . .	99
6.1	Schéma de la pose d'un stent. . . . .	104
6.2	Schéma de l'embolisation d'un anévrisme à l'aide de coils . . . . .	105
6.3	Comparaison de modèles 1D et 3D pour la simulation d'écoulements sanguins . . . . .	106
6.4	Modélisation de la géométrie des vaisseaux . . . . .	108
6.5	Réseau veineux cérébral . . . . .	109
6.6	Schéma des artères du cerveau . . . . .	110
6.7	Polygone de Willis . . . . .	110

7.1	Structure de l'avion B-747 et maillage produit par la MEF . . . . .	118
7.2	Exemple de triangulation. . . . .	121
7.3	Exemples d'éléments finis triangulaires . . . . .	122
8.1	Labellisation du cube pour la validation numérique . . . . .	132
8.2	Pression sur le cube unité pour le cas-test d'Ethier-Steinman . . . . .	134
8.3	Vitesse sur une section du cube unité pour le cas-test d'Ethier-Steinman . . . . .	134
8.4	Pression sur le cube unité pour le cas-test de Bercovier-Engelman . . . . .	136
8.5	Vitesse sur une section du cube unité pour le cas-test de Bercovier-Engelman . . . . .	136
8.6	Pression sur le cube unité pour le cas-test de Minev-Ethier . . . . .	137
8.7	Vitesse sur une section du cube unité pour le cas-test de Minev-Ethier . . . . .	138
8.8	Courbes de convergence pour le problème de Stokes . . . . .	138
8.9	Pression sur une section du cube unité pour le cas-test d'Ethier-Steinman pour la méthode des caractéristiques . . . . .	140
8.10	Vitesse sur une section du cube unité pour le cas-test d'Ethier-Steinman pour la méthode des caractéristiques . . . . .	140
8.11	Courbes de convergence pour le problème de Navier-Stokes (méthode des caractéristiques) . . . . .	141
8.12	Pression sur une section du cube unité pour le cas-test d'Ethier-Steinman pour la méthode des caractéristiques . . . . .	142
8.13	Vitesse sur une section du cube unité pour le cas-test d'Ethier-Steinman pour le schéma découplé . . . . .	142
8.14	Courbes de convergence pour le problème de Navier-Stokes (schéma dé- couplé) . . . . .	143
8.15	Description de l'écoulement de Poiseuille dans un cylindre . . . . .	144
8.16	Résultats de simulation pour l'écoulement de Poiseuille dans un tube . . . . .	147
9.1	Résultats de simulation pour les équations de Stokes sur le réseau veineux cérébral (vitesse) . . . . .	150
9.2	Résultats de simulation pour les équations de Stokes sur le réseau veineux cérébral (pression) . . . . .	151
9.3	Comparaison des 5 maillages du réseau veineux cérébral . . . . .	152
9.4	Résultats de simulation pour les équations de Navier-Stokes sur le réseau veineux cérébral (pression) . . . . .	153
9.5	Comparaison entre les mesures IRM de débits données dans la littérature et les mesures calculées dans notre modèle . . . . .	154
9.6	Résultats de simulation pour les équations de Navier-Stokes sur le réseau veineux cérébral (vitesse) . . . . .	156
9.7	Maillage du polygone de Willis . . . . .	157
9.8	Résultats de simulation pour les équations de Navier-Stokes sur le polygone de Willis . . . . .	158
9.9	Profils de vitesse imposés sur les 3 entrées du polygone de Willis . . . . .	158
9.10	Débits calculés sur les entrées et sorties du polygone de Willis . . . . .	159
9.11	Gradient de pression calculé sur le polygone de Willis . . . . .	159
10.1	Exemple d'un maillage de polygone de Willis d'un cas clinique . . . . .	165
A.1	Transformée de Legendre . . . . .	III
A.2	Fonctions non adaptées à la transformée de Legendre . . . . .	IV
A.3	Transformée de Legendre-Fenchel . . . . .	IV

A.4	Théorème de Hahn-Banach . . . . .	V
A.5	Application du théorème de Hahn-Banach . . . . .	V
B.1	Fantôme physique représentant une double bifurcation. . . . .	VII
B.2	Deux manières d'imposer des conditions aux limites de Dirichlet . . . . .	VIII

# Liste des tableaux

2.1	Exemples de pénalisation extérieure . . . . .	38
2.2	Exemples de pénalisation intérieure . . . . .	38
2.3	Temps de calcul pour débruiter l'image test de Lena . . . . .	54
2.4	Avantages et inconvénients des différents algorithmes proximaux . . . . .	55
3.1	Différentes formes d'objets dans une image 2D et 3D . . . . .	69
4.1	Comparaison des indicateurs pour les modèles ROF et TV-L1 pondérés par la vesselness . . . . .	79
4.2	Comparaison des indicateurs pour les modèles Chan classique et Chan pondéré par la vesselness . . . . .	84
4.3	Comparaison des indicateurs pour les modèles Chan classique, Chan pondéré par la vesselness et Chan incluant les directions . . . . .	90
4.4	Comparaison des indicateurs pour les modèles Chan incluant les directions extraites de la vesselness de Frangi et Chan incluant les directions extraites de RORPO . . . . .	91
6.1	Géométrie des veines : valeur approximative ( <i>mm</i> ) du rayon . . . . .	111
6.2	Géométrie des artères : valeur approximative ( <i>mm</i> ) du rayon . . . . .	112
6.3	Valeurs des paramètres mécaniques dans le modèle . . . . .	115
6.4	Valeurs des paramètres adimensionnés. . . . .	115
8.1	Erreurs $L^2$ et $H^1$ pour la solution exacte d'Ethier-Steinman . . . . .	135
8.2	Erreurs $L^2$ et $H^1$ pour la solution exacte de Bercovier-Engelman . . . . .	135
8.3	Erreurs $L^2$ et $H^1$ pour la solution exacte de Minev-Ethier . . . . .	138
8.4	Erreurs en espace pour la méthode des caractéristiques (solution exacte d'Ethier-Steinman) . . . . .	140
8.5	Erreurs cumulées en temps pour la méthode des caractéristiques (solution exacte d'Ethier-Steinman) . . . . .	141
8.6	Erreurs en espace pour le schéma découplé (solution exacte d'Ethier-Steinman) . . . . .	143
8.7	Erreurs cumulées en temps pour le schéma découplé (solution exacte d'Ethier-Steinman) . . . . .	143
8.8	Erreurs pour le problème de l'écoulement de Poiseuille dans un tube . . . . .	147
9.1	Caractéristiques des 5 maillages du réseau veineux cérébral . . . . .	151
9.2	Débits calculés aux entrées et sorties du réseau veineux cérébral . . . . .	154
9.3	Caractéristiques du maillage du polygone de Willis . . . . .	157
B.1	Comparaison des débits dans les veines cérébrales, calculés par FreeFem++ et Feel++ . . . . .	IX

# Liste des acronymes

**ARM** : Angiographie par Résonance Magnétique.

**ATC** : Angiographie par Tomographie Calculée.

**BE** : Bercovier-Engelmann.

**CFD** : Computational Fluid Dynamics (mécanique des fluides numérique, en français).

**DR** : Douglas-Rachford.

**EDO** : Équation aux Dérivées Ordinaires.

**EDP** : Équation aux Dérivées Partielles.

**ES** : Ethier-Steinman.

**FB** : Forward-Backward.

**FGP** : Fast Gradient Projection.

**FISTA** : Fast Iterative Shrinkage-Thresholding Algorithm.

**GP** : Gradient Projection.

**ISTA** : Iterative Shrinkage-Thresholding Algorithm.

**ME** : Minev-Ethier.

**MEF** : Méthode des Éléments Finis.

**MIP** : Maximum Intensity Projection (projection du maximum d'intensité, en français).

**MV** : Maximum de Vraisemblance.

**PPXA** : Parallel Proximal Algorithm.

**ROF** : Rudin, Osher et Fatemi.

**TV** : Total Variation (variation totale, en français).

**TV-L1** : Régularisation TV et fidélité de norme L1.

**TwIST** : Two-step Iterative Shrinkage/Thresholding.

# Liste des symboles

$\Omega$  : Domaine de l'image (en segmentation) ou du fluide (en simulation).

**aff** : Enveloppe affine d'un ensemble convexe.

**dom** : Domaine d'une fonction.

**Int** : Intérieur d'un ensemble.

**Per** : Périmètre d'un ensemble.

**prox** : Opérateur proximal d'une fonction.

**ri** : Intérieur relatif d'un ensemble convexe.

**sign** : Signe d'une variable.

$\Gamma^0(\mathbb{R}^n)$  : Ensemble des fonctions convexes, propres et semi-continues inférieurement.

$\partial f$  : Sous-différentielle d'une fonction.

$f^*$  : Fonction conjuguée ou transformée de Legendre-Fenchel.

$P_K$  : Opérateur de projection sur le convexe  $K$ .

$1_K$  : Fonction indicatrice du convexe  $K$ .

$\nabla$  : Opérateur gradient.

$\nabla_D$  : Opérateur gradient directionnel.

**div** : Opérateur de divergence.

$\Delta$  : Opérateur laplacien.

$\langle \cdot, \cdot \rangle$  : Produit scalaire.

$\| \cdot \|$  : Norme euclidienne.



# Introduction

« Il est une bonne chose de lire des livres de citations, car les citations lorsqu'elles sont gravées dans la mémoire vous donnent de bonnes pensées. »

---

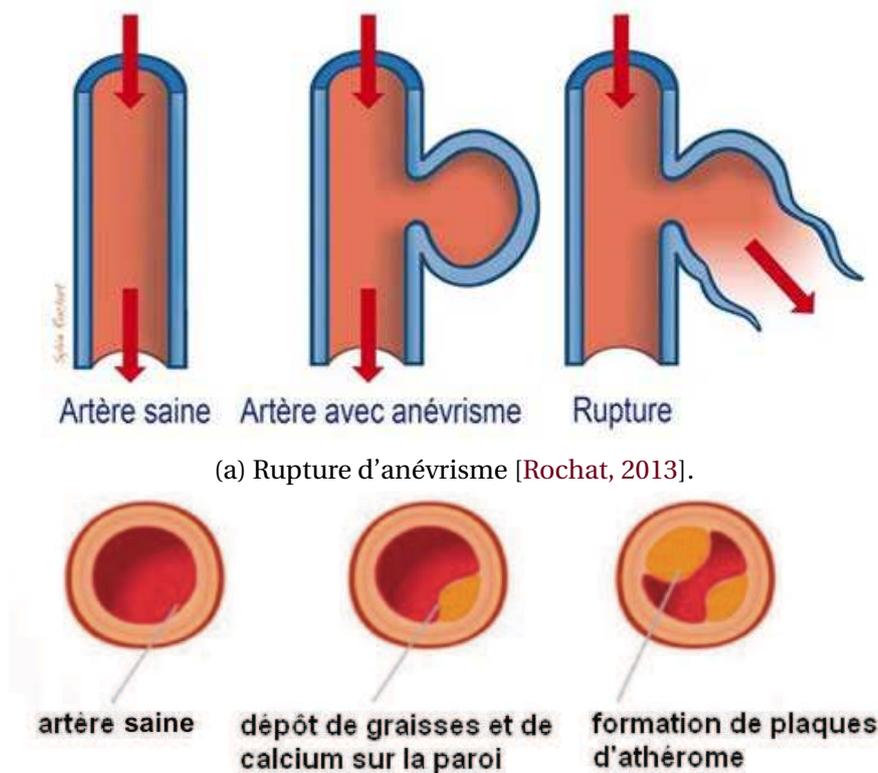
Winston Churchill

Cette thèse a été menée au sein du Laboratoire de Mathématiques de Reims entre octobre 2012 et août 2016. Elle a été co-dirigée par Stéphanie Salmon, Professeur au Laboratoire de Mathématiques de Reims, et Hugues Talbot, Professeur au Laboratoire d'Informatique Gaspard-Monge à Champs-sur-Marne. Ces travaux ont été également encadrés par Nicolas Passat, Professeur au Centre de Recherche en STIC de l'Université de Reims Champagne-Ardenne. Cette thèse a été financée par la Région Champagne-Ardenne et est indépendamment liée au projet ANR VIVABRAIN.

## Problématiques et contexte

Aujourd'hui, les maladies vasculaires sont la seconde cause de mortalité dans les pays industrialisés, la première étant les maladies cardiovasculaires. Notre objectif est de mettre à la disposition des médecins des outils de simulation de l'écoulement sanguin dans les vaisseaux. Le développement des méthodes numériques et l'augmentation de la puissance des ordinateurs a permis la mise en place de techniques de simulation numérique. Par analogie avec les expressions *in vivo* ("en vie") et *in vitro* ("en éprouvette"), le terme *in silico* a été introduit pour qualifier les méthodes numériques mises en œuvre pour simuler un phénomène biologique. Ce terme fait référence au silicium, matériau principal retrouvé dans les puces électroniques des ordinateurs. L'étape préalable à la simulation est la modélisation, qui consiste à représenter le phénomène ou modèle sous formes d'équations mathématiques. Le milieu étudié est ensuite numérisé à l'aide d'objets géométriques et des algorithmes sont alors développés pour simuler le modèle dans ce milieu. L'avantage de ces expérimentations *in silico* est qu'elles sont non-invasives, reproductibles en grand nombre, permettent de parcourir des espaces de paramètres et de faire évoluer le modèle dans le temps. Dans le cas de la simulation d'écoulements sanguins, l'objectif à long terme est d'aider à comprendre l'influence des facteurs hémodynamiques dans le développement des maladies vasculaires. Par exemple, on cherche à savoir comment l'écoulement sanguin influe sur la rupture d'anévrisme cérébral ou la formation de plaques d'athérome dans les artères. La rupture d'anévrisme est une dilatation associée à un affaiblissement anormal de la paroi d'une artère irriguant le cerveau. Sous la pression artérielle, un ballon de sang se forme au niveau de l'anévrisme et une rupture de l'anévrisme entraîne alors une hémorragie cérébrale qui peut être

fatale (voir figure 1a). La formation de plaques d'athérome consiste, quant à elle, en une infiltration des couches internes de la paroi artérielle par un dépôt de graisses, ou de calcium (voir figure 1b). L'accumulation de ce dépôt finit par obstruer le vaisseau : c'est ce qu'on appelle l'*athérosclérose*.



(a) Rupture d'anévrisme [Rochat, 2013].

(b) Formation des plaques d'athérome [Anonyme, 2013].

FIGURE 1 – Exemples de maladies cérébrovasculaires.

Ce travail s'inscrit, par ailleurs, dans le cadre du projet interdisciplinaire VIVABRAIN<sup>1</sup> [Passat *et al.*, 2016], financé par l'Agence Nationale de la Recherche, qui a pour objectif final de simuler des images angiographiques cérébrales virtuelles. La première étape consiste à reconstruire le volume 3D du réseau vasculaire à partir des images obtenues par angiographie par résonance magnétique (ARM), et donc segmenter les vaisseaux. Une fois obtenus, ces volumes vasculaires « bruts » sont structurés à l'aide d'éléments tétraédriques pour obtenir un maillage 3D. À partir de ces maillages, il est alors possible de mettre en place des processus de simulation d'écoulements sanguins. Finalement, la simulation d'images virtuelles permet de valider cette dernière étape en comparant les images obtenues avec les images acquises initialement. Cette thèse se positionne à l'interface entre les domaines de l'informatique et des mathématiques appliquées (voir encadré en pointillé sur la figure 2). L'étape intermédiaire de génération de maillages entre segmentation et simulation ne fait pas l'objet de cette thèse puisque des codes « maison » ont déjà été développés dans le cadre du projet. Le manuscrit est donc divisé en deux parties : une première partie sur la segmentation d'images angiographiques et une seconde partie sur la simulation d'écoulements sanguins.

1. <http://vivabrain.fr>

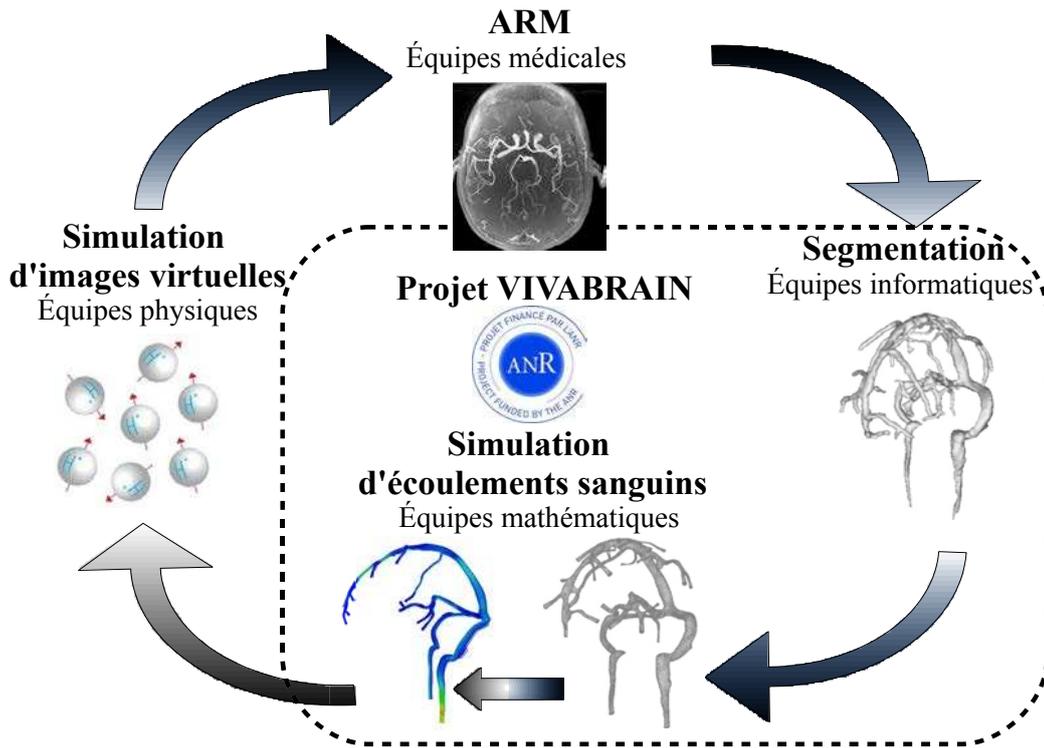


FIGURE 2 – Contexte de la thèse : projet ANR VIVABRAIN.

## Organisation du manuscrit

Pour la segmentation, nous utilisons une approche variationnelle dont le principe est de trouver l'image  $x$  qui est la plus proche de l'image originale  $f$  mais avec des variations bornées, excluant ainsi le bruit. Le problème revient alors à minimiser une énergie composée de deux termes : la régularisation et l'attache aux données. Un état de l'art relatif aux modèles variationnels est proposé dans le chapitre 1, suivi d'une vue d'ensemble des algorithmes d'optimisation dans le chapitre 2, permettant de minimiser une énergie. L'accent est mis plus particulièrement sur les algorithmes proximaux, adaptés aux problèmes variationnels. Un bref tour d'horizon sur les méthodes de segmentation et de filtrage des vaisseaux est ensuite présenté dans le chapitre 3. Nous nous focalisons sur les méthodes de filtrage basées sur les dérivées et notamment les mesures de tubularité, c'est-à-dire les fonctions qui indiquent si l'on se trouve dans un vaisseau. Enfin, le chapitre 4 présente nos contributions apportées à la segmentation d'images angiographiques. Nous proposons un premier modèle amélioré qui inclut un a priori de tubularité dans les modèles variationnels de débruitage ROF et TV-L1. Néanmoins, bien que ces modèles permettent de réhausser les vaisseaux dans l'image, une étape de seuillage est nécessaire pour obtenir une segmentation. C'est pourquoi nous proposons un deuxième modèle qui inclut un a priori de tubularité dans le modèle variationnel de segmentation de Chan-Vese. Les images obtenues sont alors binaires mais des déconnexions sont visibles au niveau des vaisseaux. Nous proposons finalement un dernier modèle qui inclut à la fois un a priori de tubularité et de directions pour pallier ce défaut. Les résultats sont fournis sur des images synthétiques 2D, ainsi que sur des images rétiniennes de la base DRIVE qui permettent une étude quantitative puisqu'elles ont été segmentées manuellement. Le chapitre 5 présente les conclusions et perspectives de cette partie "segmentation".

En ce qui concerne la simulation, nous nous sommes intéressés d'abord au réseau veineux cérébral, encore peu étudié, et également au réseau artériel avec principalement le polygone de Willis. En premier lieu, le choix du modèle est discuté dans le chapitre 6. Le sang est considéré comme un fluide newtonien, incompressible et les interactions fluide-structure sont négligées. Les équations de la dynamique des fluides qui régissent les écoulements sanguins dans notre géométrie sont alors les équations de Navier-Stokes. Nous travaillons avec des conditions aux limites mêlées avec une condition de Dirichlet non homogène en entrée pour la vitesse, une condition de Dirichlet homogène sur la paroi et une condition de Neuman en sortie pour la pression (homogène pour une sortie libre et non homogène pour simuler la traction due à l'appel du cœur à la fin du cycle cardiaque, par exemple). Pour résoudre ces équations, la méthode classique des caractéristiques est comparée avec un schéma d'ordre plus élevé proposé par Z. Sheng et al. [Sheng *et al.*, 2013] dans le chapitre 7. Ces deux schémas sont validés sur des solutions analytiques, avec conditions limites mêlées de type Dirichlet + Neumann, dans le chapitre 8. Une fois validés, ces schémas sont appliqués aux cas réalistes du réseau veineux cérébral premièrement, puis du polygone de Willis, dans le chapitre 9. Le chapitre 10 présente les conclusions et perspectives de cette partie "simulation".

## Collaboration

Durant cette thèse, j'ai eu le privilège de travailler avec de nombreux membres du projet VIVABRAIN. La doctorante Alice Dufour et la post-doctorante Anna Jezierska m'ont guidée sur la partie segmentation au début de cette thèse. Nous avons travaillé conjointement avec la doctorante Odysée Merveille sur le modèle amélioré de segmentation incluant à la fois la tubularité et les directions. Dans cette thèse, les directions sont extraites à partir de la tubularité classique de Frangi, et les résultats sont comparables au travail d'Odysée qui extrait les directions à partir du filtre RORPO qu'elle a développé. Avec le doctorant Simon Garnotel, nous avons créé le site internet "Numerical Tour in Fluid Dynamics"<sup>2</sup> fournissant un tour d'horizon des méthodes d'éléments finis et des codes FreeFem++ pour résoudre les équations de Stokes et Navier-Stokes. Enfin, lors du CEMRACS 2015, nous avons travaillé en équipe avec Alexandre Ancel, Simon Garnotel, Alexandre Fortin et Ranine Tarabay sur le projet PHANTOM, qui consiste à valider toute la boucle du projet sur un fantôme physique représentant des artères cérébrales. Plus particulièrement, nous avons comparé avec Ranine Tarabay les codes de Stokes en FreeFem++ et en Feel++ sur notre géométrie du réseau veineux.

---

2. <http://numtourcfd.univ-reims.fr>

## **Première partie**

# **Segmentation d'images angiographiques**



# Chapitre 1

## État de l'art sur les méthodes variationnelles

*« La nature agit toujours par les voies les plus courtes et les plus simples. »*

---

Pierre de Fermat

Ce chapitre propose un état de l'art relatif aux méthodes variationnelles. En traitement d'images, ces méthodes consistent notamment à minimiser une énergie. Pour le débruitage d'images, une telle énergie est constituée d'un terme d'attache aux données et d'un terme de régularité. Les trois principaux modèles de débruitage sont présentés : Tikhonov, ROF et TV-L1. Les deux derniers modèles sont notamment utilisés dans notre méthode de débruitage hybride (incluant un a priori de tubularité dans les modèles), décrite dans le chapitre 4. Pour la segmentation d'images, deux principaux modèles sont présentés : les modèles de contours actifs (qui comportent un ou plusieurs termes internes préservant les propriétés géométriques de l'image, et un terme externe d'attache aux données) et les modèles d'intensité (qui comportent un ou plusieurs termes d'attache aux données, et un terme représentant le contour de l'image). Le modèle d'intensité de Chan-Vese sera utilisé dans notre méthode de segmentation hybride (incluant un a priori de direction dans le modèle), décrite dans le chapitre 4.

### Sommaire

---

<b>1.1</b>	<b>Introduction</b>	<b>8</b>
<b>1.2</b>	<b>Modèles de débruitage</b>	<b>11</b>
1.2.1	Modèle de Tikhonov	11
1.2.2	Modèle ROF	13
1.2.3	Modèle TV-L1	14
<b>1.3</b>	<b>Modèles de segmentation</b>	<b>15</b>
1.3.1	Modèles de contours actifs	15
1.3.1.1	Modèle snake	15
1.3.1.2	Formulation en courbes de niveau	16
1.3.1.3	Modèle géodésique	17
1.3.2	Modèles d'intensité	18
1.3.2.1	Modèle de Mumford-Shah	18
1.3.2.2	Modèle de Chan-Vese	19
1.3.2.3	Convexification du problème de Chan-Vese	20

---

## 1.1 Introduction

Les premiers problèmes variationnels remontent aux considérations de calcul différentiel lié au principe de Fermat en 1657. Tandis que Descartes explique les lois de l'optique géométrique, telles que la propagation de la lumière, en comparant la lumière à une balle soumise à des forces, Fermat cherche à démontrer proprement ce phénomène en énonçant le principe suivant : "La lumière se propage d'un point à un autre sur des trajectoires telles que la durée du parcours soit localement minimale". Ce principe précise que la lumière prend le chemin le plus court (en temps de parcours) et non pas la distance la plus courte. À l'aide de ce principe, on démontre immédiatement la loi de propagation rectiligne de la lumière dans un milieu homogène.

Pour mieux comprendre ce phénomène, considérons l'exemple connu du maître nageur qui court plus vite qu'il ne nage, de même que la lumière se propage moins vite dans l'eau que dans l'air. Il se trouve au point A lorsqu'il aperçoit un nageur qui se noie en B. La question qui se pose consiste à savoir quel chemin est le plus rapide (et non le plus court) pour sauver le nageur et quel est le point I où le maître nageur doit plonger. S'il choisit le parcours en ligne droite, il perdra du temps en nageant trop longtemps. Il doit trouver le bon compromis entre le temps de nage et le temps de course.

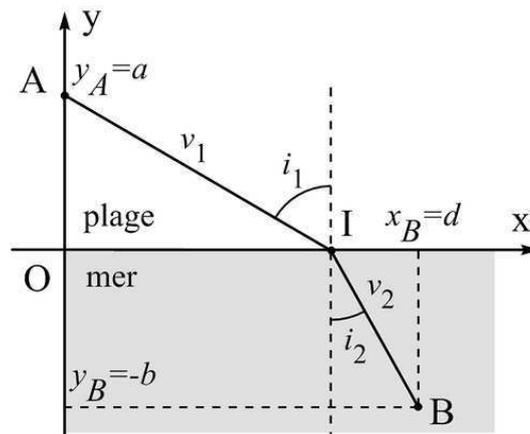


FIGURE 1.1 – Le problème du maître nageur en A et du nageur qui se noie en B.

Comme on le voit sur la figure 1.1, la trajectoire du maître nageur est constituée de deux droites AI et IB. Sachant qu'il court à la vitesse  $v_1$  et nage à la vitesse  $v_2$ , le temps  $T(x)$  mis par le maître nageur pour aller de A à B est :

$$T = \frac{AI}{v_1} + \frac{IB}{v_2}$$

soit,

$$T(x) = \frac{\sqrt{x^2 + a^2}}{v_1} + \frac{\sqrt{(d-x)^2 + b^2}}{v_2}$$

Le temps est minimal lorsque

$$\frac{dT}{dx} = \frac{1}{v_1} \frac{x}{\sqrt{x^2 + a^2}} - \frac{1}{v_2} \frac{d-x}{\sqrt{(d-x)^2 + b^2}} = 0$$

soit,

$$\frac{1}{v_1} \sin(i_1) = \frac{1}{v_2} \sin(i_2)$$

On retrouve alors la loi de réfraction de Snell-Descartes.

Maupertuis généralisa l'approche de Fermat à la mécanique et postula, en 1744, le principe de moindre action : "Maintenant, voici ce principe, si sage, si digne de l'être suprême : lorsqu'il arrive quelque changement dans la Nature, la quantité d'action employée pour ce changement est toujours la plus petite qu'il soit possible". Autrement dit, le principe de moindre action implique que la dynamique d'une quantité physique (position d'une particule, valeur d'un champ, ...) se déduit d'une unique grandeur, appelée action, qui doit être minimale pour représenter la réalité.

C'est véritablement en 1696 que le mathématicien suisse Jean Bernoulli pose l'un des premiers problèmes que l'on qualifie aujourd'hui de variationnels. Il s'agissait de trouver la forme de la brachistochrone, c'est-à-dire de la courbe reliant deux points fixés, A et B, le long de laquelle un solide tombant sous l'effet de la gravité partant de A arrive le plus vite en B. La réponse est connue, elle l'était d'ailleurs au moment où le problème fut posé, il s'agit d'une arche de cycloïde (voir figure 1.2). Le calcul des variations fut ensuite développé au XVIIIe siècle par Euler et Lagrange et au XIXe siècle par Legendre, Jacobi, Hamilton et Weierstrass entre autres. La première moitié du XXe siècle vit les contributions importantes de Bolza, Bliss, Carathéodory, etc.

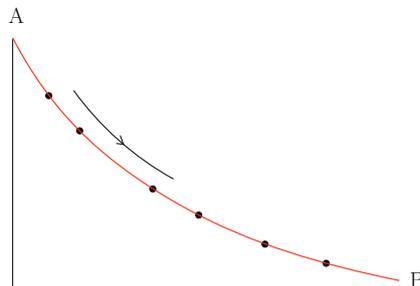


FIGURE 1.2 – Courbe brachistochrone : courbe qui minimise le temps de parcours d'un solide soumis à l'action de la pesanteur d'un point A à un point B.

Aujourd'hui, on retrouve le calcul variationnel dans de nombreuses branches des mathématiques, en particulier pour résoudre des problèmes inverses. On parle de problèmes inverses lorsque l'on cherche à déduire l'état d'un système (la solution  $x$ ) à partir de multiples observations (la donnée  $y$ ). Les problèmes inverses peuvent être formulés comme des problèmes variationnels puisqu'ils peuvent se réécrire sous la forme de la minimisation d'une fonctionnelle d'erreur entre les données et la solution du système. Les problèmes inverses sont typiquement mal posés, c'est-à-dire qu'au moins l'une des trois conditions suivantes n'est pas vérifiée :

- *existence* : la solution  $x$  du problème existe ;
- *unicité* : la solution  $x$  est unique ;
- *stabilité* : un changement infinitésimal  $\delta y$  sur la donnée implique une erreur infinitésimale  $\delta x$  sur la solution.

En effet, les problèmes inverses ne vérifient généralement pas la condition d'unicité (des paramètres différents peuvent conduire aux mêmes observations), ni la condition de stabilité (par exemple, même un faible niveau de bruit dans les observations peut conduire à une résolution incorrecte). En 1943, Tikhonov proposa une méthode pour résoudre les problèmes mal posés de la forme  $Ax = y$ , où  $A$  est un opérateur linéaire représentant la relation entre la solution  $x$  et l'observation  $y$ . La méthode standard pour résoudre ces systèmes est la méthode des moindres carrés qui consiste à minimiser le résidu :

$$f(x) = \|Ax - y\|^2 = x^T A^T A x - 2y^T A x + y^T y$$

Un point  $x$  minimise  $f$  si et seulement si :

$$f'(x) = 2A^T A x - 2A^T y = 0$$

i.e., si et seulement si  $x$  satisfait l'équation normale :

$$A^T A x = A^T y$$

La solution est alors donnée par :

$$\hat{x} = (A^T A)^{-1} A^T y$$

Cependant, la matrice  $A$  est souvent de grande taille et peut être mal conditionnée, ce qui rend difficile le calcul de la matrice pseudo-inverse de Moore-Penrose  $(A^T A)^{-1} A^T$ . Dans le but d'améliorer le conditionnement et de privilégier une solution particulière, Tikhonov ajouta un terme de régularisation dans la minimisation :

$$\|Ax - y\|^2 + \lambda \|\Gamma x\|^2$$

où  $\Gamma$ , la matrice de Tikhonov, doit être judicieusement choisie :

- $\Gamma = I$  (avec  $I =$  identité) favorise les solutions dont la norme est petite ;
- $\Gamma = \nabla$  (avec  $\nabla =$  gradient) favorise les solutions de faible variation ;
- $\Gamma = \Delta$  (avec  $\Delta =$  laplacien) favorise les solutions de faible courbure.

Cette régularisation améliore le conditionnement du problème, permettant ainsi de trouver une solution numérique donnée alors par :

$$\hat{x} = (A^T A + \lambda \Gamma^T \Gamma)^{-1} A^T y$$

En 1963, Tikhonov utilisa son modèle pour minimiser une énergie composée de deux termes :

$$\min_u \mathcal{R}(u) + \lambda \mathcal{D}(u, f) \tag{1.1}$$

L'observation est notée  $f$ , et la solution  $u$ . Le terme de droite est la *fidélité* aux données, qui représente le lien entre la solution  $u$  et l'observation  $f$ . Pour convertir un problème mal posé en un problème bien posé, on doit imposer une régularité sur la solution donnée par le terme de *régularisation*  $\mathcal{R}(u)$ . Le paramètre  $\lambda$  est utilisé pour trouver un compromis entre la régularisation et la fidélité. Ce modèle a été alors adapté au traitement d'images pour le débruitage et la segmentation.

Le but du débruitage est de supprimer le bruit, c'est-à-dire les informations parasites provenant de l'acquisition numérique, tout en préservant les propriétés essentielles comme les contours. Dans la section 1.2, les trois modèles classiques utilisés pour le débruitage d'images sont décrits :

- le modèle de Tikhonov [Tikhonov, 1963] ;
- le modèle ROF [Rudin *et al.*, 1992] ;
- le modèle TV-L1 [Alliney, 1992].

Le but de la segmentation est de partitionner l'image en un certain nombre de régions appartenant à des objets distincts (on peut regrouper, par exemple, les pixels ayant une même intensité). On peut également segmenter une image en la séparant en deux régions, l'une représentant l'objet d'intérêt et l'autre le fond. Dans la section 1.3, les deux principaux modèles pour la segmentation sont décrits :

- les modèles de contours actifs [Kass *et al.*, 1988, Casselles *et al.*, 1997] ;
- les modèles d'intensité [Mumford et Shah, 1989, Chan et Vese, 2001].

## 1.2 Modèles de débruitage

L'un des problèmes inverses les plus fondamentaux est le débruitage d'images. On suppose que l'image originale  $u : \Omega \rightarrow \mathbb{R}$  est dégradée par un bruit gaussien  $n : \Omega \rightarrow \mathbb{R}$ . Ici,  $\Omega$  représente le domaine de l'image, c'est-à-dire n'importe quel ensemble ouvert (borné) de  $\mathbb{R}^2$ , ou plus généralement  $\mathbb{R}^N$ ,  $N \geq 1$ . Alors l'image observée  $f : \Omega \rightarrow \mathbb{R}$  est telle que  $f = u + n$ .

### 1.2.1 Modèle de Tikhonov

Le modèle de Tikhonov de 1963 peut être utilisé comme un filtre de débruitage. Le but est alors de trouver une image qui est la plus proche de l'image originale, mais avec des variations gaussiennes, excluant ainsi le bruit. Le problème revient alors à minimiser une énergie composée de deux termes : un premier terme de régularisation et un d'attache aux données.

$$\min_u \left\{ \frac{1}{2} \int_{\Omega} |\nabla u|^2 + \frac{\lambda}{2} \int_{\Omega} (u - f)^2 dx \right\} \quad (1.2)$$

Dans la figure 1.3, on peut constater que le modèle de Tikhonov supprime efficacement le bruit dans le cas d'une image avec un bruit gaussien artificiel et d'une image avec un bruit naturel. Son principal inconvénient est qu'il rend floue l'image ; de petits détails sont ainsi perdus, contrairement aux deux modèles ROF et TV-L1 introduits par la suite.

Ce modèle peut être reformulé à l'aide d'une approche bayésienne comme la probabilité *a posteriori* de trouver l'hypothèse  $u$  connaissant l'observation  $f$ . La probabilité de  $u$  sachant  $f$  est calculée à l'aide du théorème de Bayes [Bayes, 1763]

$$p(u|f) = \frac{p(f|u)p(u)}{p(f)} \quad (1.3)$$

où  $p(u)$ , la probabilité *a priori* de  $u$ , correspond à la régularisation dans l'équation (1.1) ou l'information connue *a priori*. La probabilité *a posteriori* d'avoir  $f$  sachant  $u$ , notée  $p(f|u)$  et appelée probabilité conditionnelle, correspond à la fidélité dans l'équation (1.1).

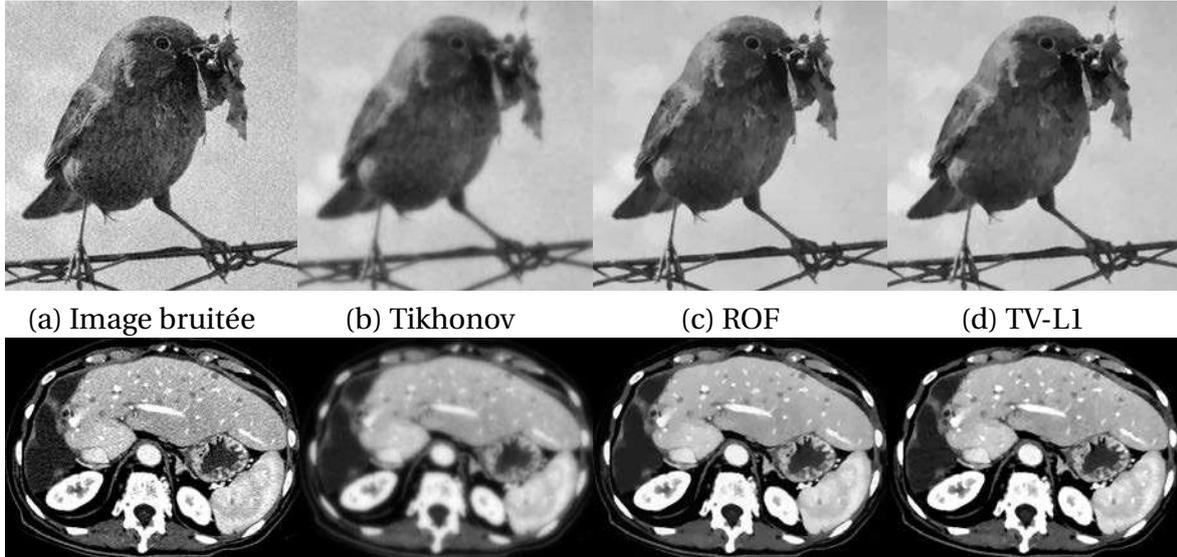


FIGURE 1.3 – Comparaison de différents modèles variationnels pour le débruitage. Haut : image naturelle avec un bruit gaussien artificiel. Bas : image CT avec un bruit dû à l'acquisition et à la reconstruction tomographique [Unger, 2012].

L'idée du maximum a posteriori (MAP) est de trouver l'image  $u$  qui maximise la probabilité  $p(u|f)$  ; la variable  $p(f)$  restant constante au cours de l'optimisation, on peut alors l'ignorer :

$$\max_u \{p(u|f)\} = \max_u \{p(f|u)p(u)\} \quad (1.4)$$

Comme il est montré dans [Chambolle *et al.*, 2010], résoudre l'équation (1.4) n'est pas nécessairement une bonne idée, car l'image  $u$  avec la plus grande probabilité peut être très rare. Nous allons néanmoins établir le lien entre le problème de minimisation initial et l'approche bayésienne. Reprenons le modèle de Tikhonov et supposons que  $n$  est un bruit gaussien de variance  $\sigma^2$  ; on a la probabilité *a posteriori* :

$$p(f|u) = p(n = f - u) = \prod_{x \in \Omega} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{\|u(x) - f(x)\|^2}{2\sigma^2}}$$

Supposons maintenant que l'on se trouve dans un modèle d'image avec contours flous et que le gradient de l'image est alors une distribution gaussienne de variance  $\mu^2$  ; on a la probabilité *a priori* :

$$p(u) = \prod_{x \in \Omega} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{|\nabla u(x)|^2}{2\mu^2}}$$

D'après le théorème de Bayes, on a :

$$p(u|f) = p(f|u)p(u) = \prod_{x \in \Omega} \frac{1}{2\pi\sigma\mu} e^{-\frac{\|u(x) - f(x)\|^2}{2\sigma^2} - \frac{|\nabla u(x)|^2}{2\mu^2}}$$

En utilisant l'équation (1.4), on obtient le problème d'optimisation suivant :

$$\begin{aligned} \max_u \{p(u|f)\} &= \max_u \left\{ \prod_{x \in \Omega} \frac{1}{2\pi\sigma\mu} e^{-\frac{\|u(x)-f(x)\|^2}{2\sigma^2} - \frac{|\nabla u(x)|^2}{2\mu^2}} \right\} \\ &= \max_u \left\{ e^{-\int_{\Omega} \left( \frac{\|u(x)-f(x)\|^2}{2\sigma^2} + \frac{|\nabla u(x)|^2}{2\mu^2} \right) dx} \right\} \\ &= \min_u \left\{ \int_{\Omega} \left( \frac{\|u(x)-f(x)\|^2}{2\sigma^2} + \frac{|\nabla u(x)|^2}{2\mu^2} \right) dx \right\} \end{aligned}$$

Cette dernière formulation est bien équivalente au problème (1.2).

## 1.2.2 Modèle ROF

Rudin, Osher et Fatemi ont amélioré le modèle de Tikhonov, en remplaçant la régularisation quadratique par la norme L1 :

$$\min_u \left\{ \int_{\Omega} |\nabla u| + \frac{\lambda}{2} \int_{\Omega} (u-f)^2 dx \right\} \quad (1.5)$$

Le modèle ROF donne de meilleurs résultats car il préserve les bords. En effet, à cause de la régularisation quadratique dans le modèle de Tikhonov, une énergie avec plusieurs sauts de faible intensité est significativement plus basse qu'une énergie avec un saut de grande intensité, comme on peut le constater dans la figure 1.4. Le modèle favorise alors les transitions lisses, ce qui a pour effet de rendre floue l'image. En revanche, la variation totale (TV pour Total Variation) compte toutes les valeurs absolues des gradients de l'image :

$$\text{TV}(u) = \int_{\Omega} |\nabla u|$$

La variation totale donne pour résultat une énergie à 1 pour les trois exemples de la figure 1.4. Elle ne favorise donc aucune de ses trois fonctions et préserve à la fois les bords et les transitions lisses. Un effet secondaire est que la régularisation L1 favorise les images en escalier.

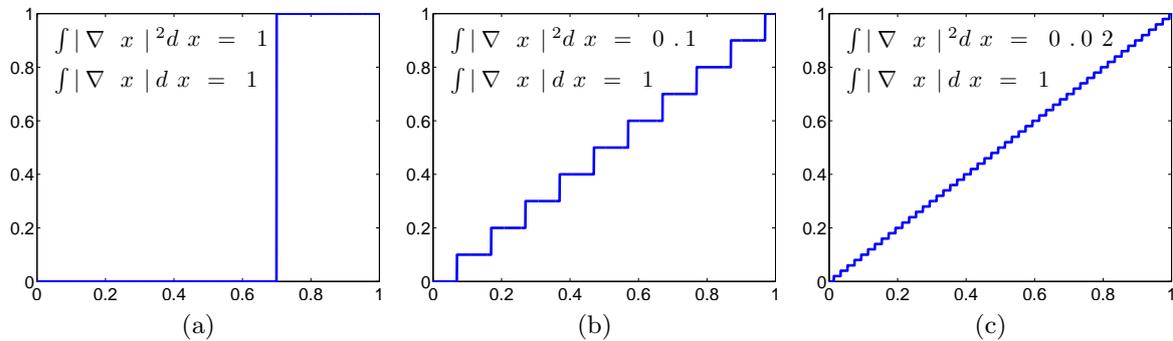


FIGURE 1.4 – Comparaison de la régularisation quadratique et de la variation totale. La variation totale compte toujours la somme totale de tous les sauts, qui est 1 dans les trois exemples. Pour la régularisation quadratique, plus les sauts sont de faible intensité, plus l'énergie est faible [Unger, 2012].

### 1.2.3 Modèle TV-L1

La norme L1 peut être utilisée non seulement pour la régularisation, mais aussi pour le terme de fidélité. Quand le terme de fidélité L2 du modèle ROF est remplacé par la norme L1, on parle de modèle TV-L1 :

$$\min_u \left\{ \int_{\Omega} |\nabla u| + \lambda \int_{\Omega} |u - f| dx \right\} \quad (1.6)$$

Quand on compare, sur la figure 1.3, le modèle ROF au modèle TV-L1, les différences sont marginales. Néanmoins, une propriété intéressante pour le modèle TV-L1 est l'effet de dépendance au contraste de l'image qui est presque supprimé par rapport au modèle ROF. Effectivement, en utilisant une norme L2 pour l'attache aux données, les faibles variations de  $u$  et  $f$  sont moins coûteuses que les fortes variations. Par conséquent, les structures de faible contraste sont supprimées. Bien que cette observation reste vraie en utilisant la norme L1, la dépendance au contraste n'est plus quadratique mais linéaire.

En estimant le maximum de vraisemblance [Boyd et Vandenberghe, 2004], on peut également montrer que le modèle TV-L1 est mieux adapté au bruit laplacien (bruit exponentiel ou impulsionnel, fréquent en thermique et en électricité), tandis que l'on préférera le modèle ROF pour un bruit gaussien (bruit blanc additif stationnaire, fréquent en traitement d'images). Considérons une famille de distributions indexées par  $x \in \mathbb{R}^n$ , avec des densités  $p_x(\cdot)$ . Lorsqu'elle est considérée comme une fonction de  $x$ , pour  $y \in \mathbb{R}^m$  fixé, la fonction  $p_x(y)$  est appelée vraisemblance. Il est plus commode de travailler avec son logarithme, appelé *log-vraisemblance* et noté  $l$  :

$$l(x) = \log p_x(y)$$

Maintenant, supposons que l'on dispose d'un certain nombre d'échantillons de  $y$ ,  $[y_1, \dots, y_m]$ . On cherche alors à estimer  $x$  comme un paramètre des  $p_x(y)$  en fonction des valeurs  $y$  observées. Une méthode très utilisée, appelée maximum de vraisemblance (MV) consiste à estimer  $x$  comme :

$$\hat{x} = \arg \max_x p_x(y) = \arg \max_x l(x)$$

Considérons un modèle de mesure linéaire

$$y_i = a_i^T x + \varepsilon_i, \quad i = 1, \dots, m$$

où  $x \in \mathbb{R}^n$  est un vecteur de paramètres à estimer,  $y_i \in \mathbb{R}$  sont les valeurs observées et  $\varepsilon_i$  les erreurs de mesure (bruit). Supposons que les  $\varepsilon_i$  sont indépendantes et identiquement distribuées (iid) ; la vraisemblance est alors :

$$p_x(y) = \prod_{i=1}^m p(y_i - a_i^T x)$$

et la log-vraisemblance est :

$$l(x) = \log p_x(y) = \sum_{i=1}^m \log p(y_i - a_i^T x)$$

L'estimateur du MV est alors solution du problème :

$$\max_x \sum_{i=1}^m \log p(y_i - a_i^T x)$$

Comparons cet estimateur pour un bruit gaussien et un bruit laplacien.

- Si les  $\varepsilon_i$  sont des variables gaussiennes de moyenne nulle et de variance  $\sigma^2$ , la densité est  $p(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{z^2}{2\sigma^2}}$  et la log-vraisemblance est :

$$l(x) = \sum_{i=1}^m \log \left( \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - a_i^T x)^2}{2\sigma^2}} \right) = \left( \sum_{i=1}^m \log \frac{1}{\sqrt{2\pi\sigma^2}} \right) - \frac{1}{2\sigma^2} \sum_{i=1}^m (y_i - a_i^T x)^2$$

Alors l'estimateur du MV est :

$$\hat{x} = \arg \max_x l(x) = \arg \min_x \sum_{i=1}^m (y_i - a_i^T x)^2 = \arg \min_x \|Ax - y\|^2$$

où A est la matrice composée des lignes  $a_1^T, \dots, a_m^T$ , et on retrouve bien le terme de fidélité du modèle ROF.

- Si les  $\varepsilon_i$  sont des variables laplaciennes, i.e. ont une densité  $p(z) = \frac{1}{2\alpha} e^{-\frac{|z|}{\alpha}}$  (où  $\alpha > 0$ ), la log-vraisemblance est :

$$l(x) = \sum_{i=1}^m \log \left( \frac{1}{2\alpha} e^{-\frac{|y_i - a_i^T x|}{\alpha}} \right) = \left( \sum_{i=1}^m \log \frac{1}{2\alpha} \right) - \frac{1}{\alpha} \sum_{i=1}^m |y_i - a_i^T x|$$

Alors l'estimateur du MV est :

$$\hat{x} = \arg \max_x l(x) = \arg \min_x \sum_{i=1}^m |y_i - a_i^T x| = \arg \min_x |Ax - y|$$

Dans ces conditions, on retrouve bien le terme de fidélité du modèle TV-L1.

## 1.3 Modèles de segmentation

Les méthodes de segmentation utilisant une approche variationnelle, peuvent être décomposées en deux grandes catégories : celles qui utilisent les informations du gradient de l'image, i.e. le contour, appelées modèles déformables ou contours actifs, et celles qui utilisent des informations de régions, appelées modèles d'intensité.

### 1.3.1 Modèles de contours actifs

#### 1.3.1.1 Modèle snake

L'idée principale des contours actifs est de faire évoluer une courbe dans le but de détecter les contours de l'image, sous l'influence de plusieurs forces généralement antagonistes : celles, *externes*, qui dépendent du contenu de l'image et celles, *internes*, visant à préserver certaines propriétés géométriques. Dans les modèles classiques, cette évolution s'achève grâce à un critère d'arrêt dépendant du gradient de l'image observée  $f$ . Soit  $\mathcal{C}(s) : [0, 1] \rightarrow \Omega$  la courbe paramétrée représentant les contours de l'image et supposée suffisamment régulière. Le modèle snake, introduit dans [Kass *et al.*, 1988], consiste à minimiser la fonctionnelle d'énergie  $J_1(\mathcal{C})$  :

$$J_1(\mathcal{C}) = \alpha \int_0^1 |\mathcal{C}'(s)|^2 ds + \beta \int_0^1 |\mathcal{C}''(s)|^2 ds - \lambda \int_0^1 |\nabla f(\mathcal{C}(s))|^2 ds \quad (1.7)$$

où  $\alpha, \beta$  et  $\lambda$  sont des constantes réelles positives. Les deux premiers termes contrôlent la régularité de la courbe (énergie interne), qui est composée de l'élasticité  $\mathcal{C}'(s)$  et de la rigidité  $\mathcal{C}''(s)$ . Le troisième terme est le terme d'attache aux données (énergie externe), permettant d'attirer la courbe vers les contours de l'image. L'un des inconvénients majeurs de cette approche est qu'elle n'autorise pas les changements de topologie : la topologie de la courbe initiale sera la même que celle de la courbe finale, potentiellement fautive (dans le cas, par exemple, où l'on cherche deux contours distincts, voir figure 1.5). La formulation en courbes de niveau a permis de surmonter ce problème, et permet de détecter plusieurs objets simultanément.

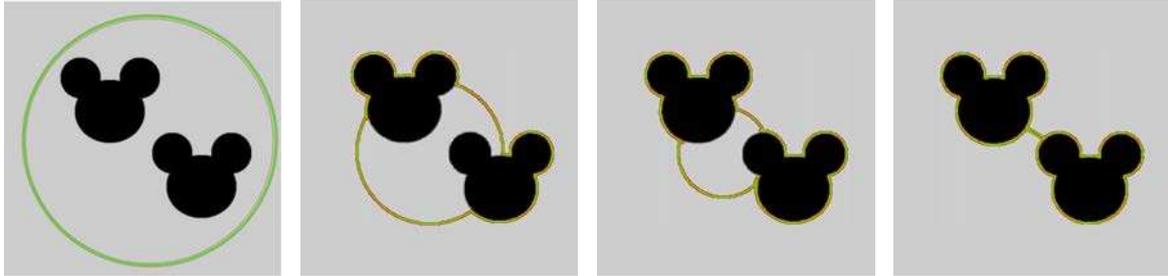


FIGURE 1.5 – Modèle de contour actif snake : le snake se déforme par itérations successives afin de converger vers le contour réel. Ici, le contour n'est pas exact puisque le snake ne peut pas se scinder pour retrouver chacun des contours des deux objets.

### 1.3.1.2 Formulation en courbes de niveau

L'évolution du contour actif dans le temps est obtenue par la minisation de la fonctionnelle d'énergie. Cette tâche est effectuée en s'appuyant sur l'équation d'Euler-Lagrange. Ainsi, l'équation d'évolution de contour du modèle snake (1.7) s'écrit sous la forme :

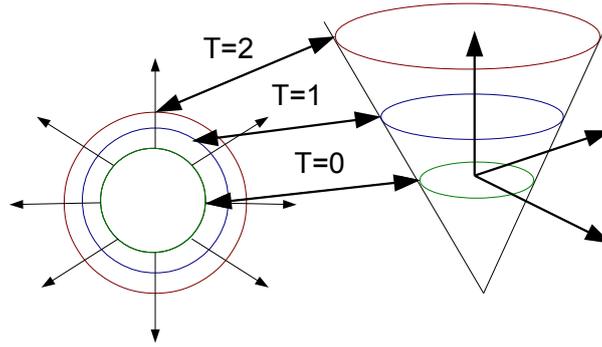
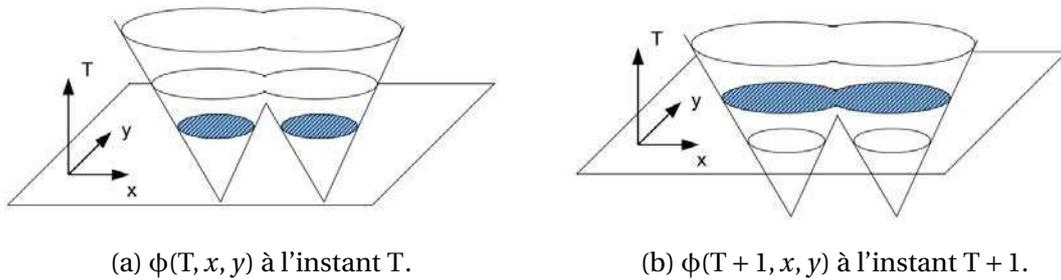
$$\frac{\partial \mathcal{C}}{\partial t} = \operatorname{div} \left( \frac{\nabla \mathcal{C}}{|\nabla \mathcal{C}|} \right) |\nabla \mathcal{C}|$$

Cependant, ce modèle est celui de l'évolution d'une courbe sous sa propre courbure, et il tend à faire disparaître la courbe. Pour contrebalancer cet effet, une constante réelle  $v$  est ajoutée, et la fonction de détection  $g$  est introduite dans [Casselles *et al.*, 1993] :

$$\frac{\partial \mathcal{C}}{\partial t} = g(|\nabla f|) \operatorname{div} \left( \frac{\nabla \mathcal{C}}{|\nabla \mathcal{C}|} + \mathbf{v} \right) |\nabla \mathcal{C}|$$

La constante  $v$  induit une force semblable à la force ballon introduite dans [Cohen, 1991]. Grâce à cette contrainte, la courbe évolue en tout point selon sa normale ; elle se comporte comme un ballon qui se dilate ou se contracte autour de l'objet.

Dans la formulation en courbes de niveau (ou *levels sets* en anglais), proposée dans [Osher et Sethian, 1988], la courbe à propager  $\mathcal{C}(T)$  est considérée comme le niveau 0 d'une fonction de trois variables  $\phi(x, y, T)$ , i.e.  $\phi(x, y, T) = 0$ . En d'autres termes, la position du contour actif à chaque instant  $T$  correspond à l'intersection de la fonction  $\phi$  avec le plan  $\phi = 0$  (voir figure 1.6). Ceci permet au contour actif de ne plus être contraint topologiquement, mais de pouvoir se scinder ou fusionner au cours de l'évolution du processus : les changements topologiques sont ainsi possibles (voir figure 1.7).


 FIGURE 1.6 – Représentation des courbes de niveau  $\phi(T, x, y)$  dans le plan à  $T = 0, 1, 2$ .

 FIGURE 1.7 – Mise en évidence de changement topologique à l'aide de la formulation en courbes de niveau : la courbe de niveau  $\phi$  (en bleu) qui est scindée à l'instant  $T$ , a fusionné à l'instant  $T + 1$ .

### 1.3.1.3 Modèle géodésique

Si l'on annule la rigidité ( $\beta = 0$ ), jugée trop contraignante et empêchant les discontinuités du second ordre (le contour n'est pas autorisé à développer des coins), et si l'on remplace le détecteur de contour  $|\nabla f(\mathcal{C}(s))|^2$  par une fonction plus générale dépendant du gradient de  $f$ ,  $g(|\nabla f(\mathcal{C}(s))|^2)$ , alors la fonctionnelle (1.7) devient :

$$J_1(\mathcal{C}) = \alpha \int_0^1 |\mathcal{C}'(s)|^2 ds + \lambda \int_0^1 g(|\nabla f(\mathcal{C}(s))|^2) ds \quad (1.8)$$

où  $g$  est une fonction de potentiel de contour, strictement décroissante, telle que  $\lim_{z \rightarrow +\infty} g(z) = 0$ . Un exemple est la fonction de détection suivante :

$$g(|\nabla f|) = \frac{1}{1 + |\nabla G_\sigma * f|^p}, \quad p \geq 1$$

où le gradient de l'image  $f$  est convolué avec un filtre gaussien  $G_\sigma$  de variance  $\sigma$ . C'est une attache aux données s'annulant sur les zones de l'image de fort gradient, c'est-à-dire aux frontières de l'objet.

En utilisant le principe de Maupertuis, il est montré dans [Casselles *et al.*, 1997] que les contours actifs consistent à minimiser la fonctionnelle d'énergie  $J_2(\mathcal{C})$  :

$$J_2(\mathcal{C}) = \int_0^1 g(|\nabla f(\mathcal{C}(s))|) dq \quad (1.9)$$

où  $dq = |\mathcal{C}'(s)|ds$ . Le problème s'interprète alors géométriquement comme la recherche de la longueur minimale du contour, c'est-à-dire la *géodésique*, dans l'espace de Riemann défini par une métrique dépendant du gradient de l'image :

$$L_R = \int_0^{L(\mathcal{C})} g(|\nabla f(\mathcal{C}(s))|) dq$$

où  $L(\mathcal{C})$  est la longueur euclidienne de la courbe  $\mathcal{C}$  donnée par  $L = \oint dq$ . La nouvelle fonctionnelle est intrinsèque, c'est-à-dire qu'elle ne dépend pas de la paramétrisation.

L'équation d'évolution du modèle géodésique (1.9) est :

$$\frac{\partial \mathcal{C}}{\partial t} = (g(|\nabla f|)\kappa - \nabla g(|\nabla f|) \cdot \vec{n}) \vec{n}$$

où  $\kappa = \operatorname{div} \frac{\nabla f}{|\nabla f|}$  est la courbure du contour actif et  $\vec{n}$  le vecteur unitaire normal intérieur au contour. Avec cette formulation, deux forces s'appliquent sur le contour actif : la première fait se déplacer la courbe vers les contours de l'objet, sous une contrainte de courbure qui tend à éviter les irrégularités ; la seconde ne s'applique qu'à proximité des contours de l'objet, et est utilisée comme une force d'ajustement. Dans ce contexte, bien qu'il existe une équation d'évolution, le modèle intrinsèque n'est pas explicite, et l'optimisation par ce moyen correspond à une descente de gradient inefficace. C'est pourquoi d'autres modèles plus génériques pouvant être optimisés par d'autres moyens ont été proposés.

## 1.3.2 Modèles d'intensité

### 1.3.2.1 Modèle de Mumford-Shah

Dans beaucoup de cas pratiques, l'information de gradient n'est pas suffisante pour réaliser une bonne segmentation. En effet, cette information n'est pas applicable dans le cas d'une image qui n'affiche pas de frontière nette entre les différentes régions qui la composent. C'est pourquoi des approches prenant en compte des informations d'intensité ont été développées. Un modèle très étudié est celui de [Mumford et Shah, 1989]. Il consiste à représenter une image en regroupant dans une même région les pixels ayant des intensités proches et des variations lisses tout en gardant une contrainte de régularité sur les contours de l'image. Cela se traduit par la recherche d'une approximation lisse par morceaux de l'image  $f$ . La fonctionnelle de Mumford-Shah est donnée par la minimisation du problème :

$$\min_{\Lambda, u} \left\{ v\mathcal{H}^{c-1}(\Lambda) + \lambda \int_{\Omega} (u - f)^2 dx + \int_{\Omega \setminus \Lambda} |\nabla u|^2 dx \right\} \quad (1.10)$$

où  $\lambda$  est une constante réelle positive,  $\Lambda$  est l'ensemble des contours de l'image et  $\mathcal{H}^{c-1}(\Lambda)$  la mesure de Hausdorff de dimension  $c - 1$  représentant la longueur de  $\Lambda$ . Le premier terme minimise la longueur des contours, le second correspond à l'attache aux données et le dernier est le terme de diffusion qui applique une homogénéité à l'intérieur de la région  $\Omega \setminus \Lambda$ .

Une variation connue du modèle (1.10) est la fonctionnelle constante par morceaux de Mumford-Shah définie par

$$\min_u \left\{ v\mathcal{H}^{c-1}(\Gamma_u) + \frac{\alpha}{2} \int_{\Omega} (u - f)^2 dx \right\} \quad (1.11)$$

où  $\alpha$  est une constante réelle positive et  $\Gamma_u$  correspond à l'ensemble des sauts de  $u$ . Le terme de diffusion lisse est ignoré, ce qui rend le problème convexe et plus facile à résoudre.

### 1.3.2.2 Modèle de Chan-Vese

Dans ce contexte, une importante contribution est celle de [Chan et Vese, 2001] qui utilise le formalisme des ensemble de niveaux pour réduire le modèle de Mumford-Shah à un problème de segmentation d'image en deux phases. Supposons que l'image  $f$  est constituée de deux régions constantes :

- la région d'intérêt  $\Sigma$ , d'intensité moyenne de valeur  $c_1$  ;
- le fond  $\Omega \setminus \Sigma$ , d'intensité moyenne de valeur  $c_2$ .

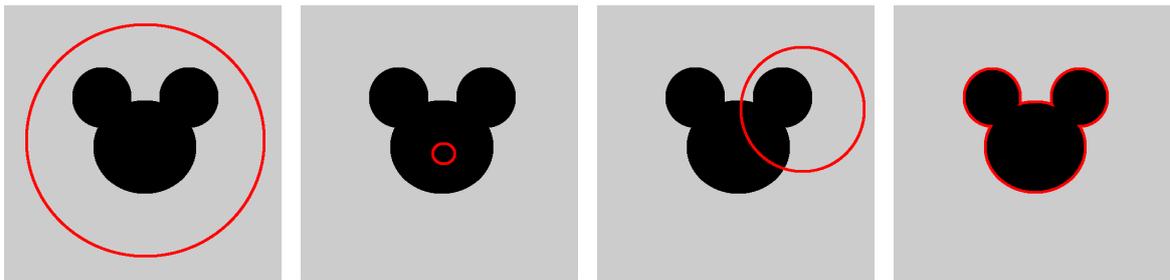
Alors le problème de détection de la région d'intérêt dans l'image revient au problème de minimisation suivant :

$$\min_{\Sigma, c_1, c_2} \left\{ \text{Per}(\Sigma) + \lambda \int_{\Sigma} (c_1 - f)^2 dx + \lambda \int_{\Omega \setminus \Sigma} (c_2 - f)^2 dx \right\} \quad (1.12)$$

où  $\text{Per}(\Sigma)$  est le périmètre de l'ensemble  $\Sigma$ . Il est alors possible de montrer que  $C_0$ , la frontière de l'objet, est le minimum de l'attache aux données :

$$F = F_1(\Sigma) + F_2(\Sigma) = \int_{\Sigma} (c_1 - f)^2 dx + \int_{\Omega \setminus \Sigma} (c_2 - f)^2 dx$$

En effet, on peut remarquer sur la figure 1.8a que si la courbe d'évolution est à l'extérieur de l'objet, alors  $F_1(\Sigma) > 0$ ,  $F_2(\Sigma) \approx 0$ , tandis que si la courbe est à l'intérieur de l'objet, comme sur la figure 1.8b, alors  $F_1(\Sigma) \approx 0$ ,  $F_2(\Sigma) > 0$ . Sur la figure 1.8c, on constate que si la courbe est à la fois à l'intérieur et à l'extérieur de l'objet, alors  $F_1(\Sigma) > 0$ ,  $F_2(\Sigma) > 0$ . Finalement, l'attache aux données est minimisée lorsque la courbe épouse parfaitement la frontière  $C_0$  (voir figure 1.8d).



(a)  $F_1(\Sigma) > 0$ ,  
 $F_2(\Sigma) \approx 0$ ,  $F > 0$

(b)  $F_1(\Sigma) \approx 0$ ,  
 $F_2(\Sigma) > 0$ ,  $F > 0$

(c)  $F_1(\Sigma) > 0$ ,  
 $F_2(\Sigma) > 0$ ,  $F > 0$

(d)  $F_1(\Sigma) \approx 0$ ,  
 $F_2(\Sigma) \approx 0$ ,  $F = 0$

FIGURE 1.8 – Fidélité du modèle de Chan-Vese pour plusieurs positions de la courbe. Elle est minimisée seulement dans la cas où la courbe est sur le contour de l'objet.

La figure 1.9 compare les modèles de Mumford-Shah et de Chan-Vese. Le modèle de Chan-Vese a segmenté l'image originale en deux régions : le fond et l'objet d'intérêt (l'oiseau et le grillage) tandis que plusieurs régions apparaissent avec le modèle de Mumford-Shah.

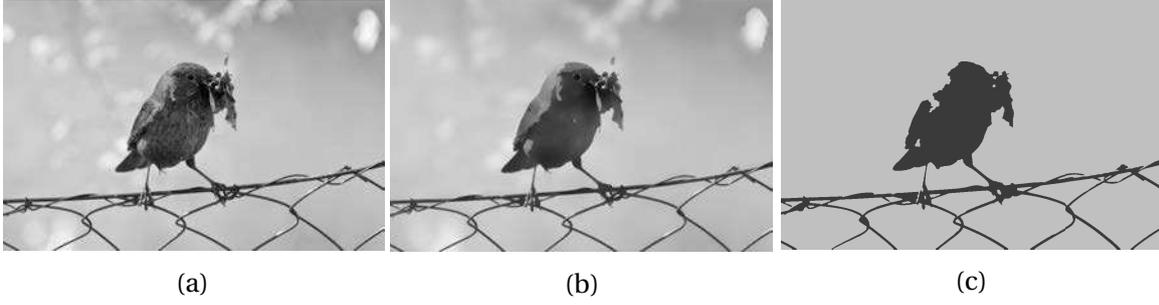


FIGURE 1.9 – Comparaison de différents modèles d'intensité. (a) Image originale. (b) Modèle de Mumford-Shah constant par morceaux donné par l'équation (1.11). (c) Modèle de Chan-Vese [Unger, 2012].

### 1.3.2.3 Convexification du problème de Chan-Vese

Le problème de Chan-Vese revient à chercher la meilleure approximation  $u$  de  $f$  prenant seulement deux valeurs ( $\{0, 1\}$  pour une image binaire). Le problème (1.12) est donc non convexe, car l'ensemble admissible  $\{0, 1\}$  de ce problème est lui-même non convexe. Néanmoins, dans [Chan *et al.*, 2006], il est démontré que le problème peut simplement être convexifié en minimisant l'énergie sur une fonction  $u$  prenant des valeurs intermédiaires entre 0 et 1 :

**Théorème 1.3.1** (Chan, Esedoğlu, Nikolova, 2006). *Le problème (1.12) est équivalent au problème de minimisation convexe :*

$$\min_{0 \leq u \leq 1} \int_{\Omega} |\nabla u| + \lambda \int_{\Omega} \{(c_1 - f(x))^2 - (c_2 - f(x))^2\} u(x) dx \quad (1.13)$$

Et alors  $\Sigma = \{x : u(x) \geq \mu\}$  pour tout  $\mu \in ]0, 1]$ .

*Démonstration.* La formule de la co-aire, démontrée dans [Fleming et Rishel, 1960], permet de montrer facilement que les termes de régularité des problèmes (1.12) et (1.13) sont équivalents :

$$\int_{\Omega} |\nabla u| = \int_0^1 \text{Per}(\{x : u(x) > \mu\}) d\mu$$

Cela signifie que la variation totale de  $u$  peut être décomposée comme la somme de tous les périmètres des courbes de niveaux de  $u$ .

En utilisant la fonction indicatrice d'un sous-ensemble, on montre que le terme d'attache aux données du problème (1.13) devient :

$$\begin{aligned} \int_{\Omega} \{(c_1 - f(x))^2 - (c_2 - f(x))^2\} u(x) dx &= \int_{\Omega} A(x) u(x) dx \\ &= \int_{\Omega} A(x) \int_0^1 \mathbb{1}_{[0, u(x)]}(\mu) d\mu dx \\ &= \int_0^1 \int_{\Omega} A(x) \mathbb{1}_{[0, u(x)]}(\mu) dx d\mu \\ &= \int_0^1 \int_{\Omega \cap \{x : u(x) > \mu\}} A(x) dx d\mu \end{aligned}$$

En posant  $\Sigma(\mu) = \{x : u(x) > \mu\}$ , on obtient pour chaque terme d'attache aux données :

$$\begin{aligned} \int_{\Omega} (c_1 - f(x))^2 u(x) dx &= \int_0^1 \int_{\Omega \cap \{x:u(x)>\mu\}} (c_1 - f(x))^2 dx d\mu \\ &= \int_0^1 \int_{\Sigma(\mu)} (c_1 - f(x))^2 dx d\mu \\ \int_{\Omega} (c_2 - f(x))^2 u(x) dx &= \int_{\Omega} (c_2 - f)^2 - \int_0^1 \int_{\Omega \cap \{x:u(x)>\mu\}^c} (c_2 - f(x))^2 dx d\mu \\ &= C - \int_0^1 \int_{\Omega \setminus \Sigma(\mu)} (c_2 - f(x))^2 dx d\mu \end{aligned}$$

où  $C = \int_{\Omega} (c_2 - f)^2$  est indépendante de  $u$ . En soustrayant les deux termes, on montre que minimiser l'attache aux données du problème (1.13) revient à minimiser l'attache aux données du problème (1.12). □

Maintenant que nous avons vu certains modèles en imagerie, nous nous intéressons aux moyen de les optimiser en pratique.



# Chapitre 2

## Vue d'ensemble sur les algorithmes d'optimisation

*« La vraie question c'est : "Comment profiter de la vie au maximum ?" La vraie réponse, c'est : "En évitant de se poser les vraies questions!" »*

Georges Wolinski

Ce chapitre présente une vue d'ensemble des algorithmes d'optimisation, organisée en quatre grandes classes : l'optimisation linéaire, l'optimisation différentiable sans contraintes, l'optimisation différentiable sous contraintes et l'optimisation non différentiable convexe. Cette dernière classe est la mieux adaptée pour résoudre les problèmes variationnels du chapitre 1. En effet, l'énergie des modèles de débruitage ou de segmentation peut être divisée en somme d'énergies convexes, dont au moins l'une n'est pas différentiable (la régularité, en général). Une descente de gradient est alors effectuée sur l'énergie différentiable, suivie d'une projection sur l'énergie non différentiable, appelée opérateur proximal. Deux algorithmes proximaux seront utilisés dans le chapitre 4 : l'algorithme explicite-implicite et l'algorithme primal-dual.

### Sommaire

<b>2.1 Introduction</b>	<b>25</b>
<b>2.2 Algorithmes de programmation linéaire</b>	<b>28</b>
2.2.1 Méthode du simplexe	28
2.2.2 Méthode de points intérieurs	29
<b>2.3 Algorithmes à directions de descente</b>	<b>30</b>
2.3.1 Recherche linéaire	30
2.3.1.1 Recherche linéaire exacte	31
2.3.1.2 Règles d'Armijo et de Goldstein	31
2.3.1.3 Règle de Wolfe	31
2.3.2 Algorithme du gradient	32
2.3.3 Algorithme du gradient conjugué	32
2.3.4 Algorithme de Newton	34
2.3.5 Algorithme de quasi-Newton	34
<b>2.4 Algorithmes de minimisation sous contraintes</b>	<b>36</b>
2.4.1 Algorithme du gradient projeté	36

2.4.2	Algorithme de pénalisation . . . . .	37
2.4.3	Méthode d'Uzawa . . . . .	38
<b>2.5</b>	<b>Algorithmes d'optimisation convexe . . . . .</b>	<b>39</b>
2.5.1	Notion de sous-différentielle . . . . .	40
2.5.2	Opérateur proximal : définition et exemples . . . . .	41
2.5.3	Opérateur proximal de la variation totale . . . . .	42
2.5.3.1	Transformation de Legendre-Fenchel . . . . .	43
2.5.3.2	Opérateur proximal et opérateur de projection . . . . .	43
2.5.3.3	Cas discret . . . . .	44
2.5.3.4	Algorithme de Chambolle . . . . .	45
2.5.4	Algorithme explicite-implicite (FB : Forward-Backward) . . . . .	46
2.5.5	Algorithme de Douglas-Rachford (DR) . . . . .	48
2.5.6	Algorithme Proximal Parallèle (PPXA) . . . . .	51
2.5.7	Algorithme Primal-Dual . . . . .	51
2.5.8	Application au débruitage d'images . . . . .	53

---

## 2.1 Introduction

De manière générale, un problème d'optimisation s'écrit sous la forme :

$$\begin{aligned} \min_x f(x) \\ \text{sous contraintes } g_i(x) = 0 \text{ et } h_j(x) \leq 0, \quad i = 1, \dots, p, j = 1, \dots, q \end{aligned} \quad (2.1)$$

où  $x$  est un vecteur de  $\mathbb{R}^n$  représentant les variables que l'on cherche à optimiser dans le problème considéré,  $f$  une fonction de coût traduisant le critère selon lequel on souhaite évaluer les diverses valeurs de  $x$ ,  $g_i$  les contraintes d'égalité et  $h_i$  les contraintes d'inégalité, représentant les restrictions sur les valeurs de  $x$  que l'on peut accepter. Par la suite, on utilisera l'abréviation "s.c." pour "sous contraintes".

Un vecteur  $x^*$  est solution du problème, appelé *minimum*, s'il possède la plus petite valeur de coût parmi tous les vecteurs satisfaisant les contraintes (voir figure 2.1). On écrit :

$$x^* = \underset{x}{\operatorname{argmin}} f(x)$$

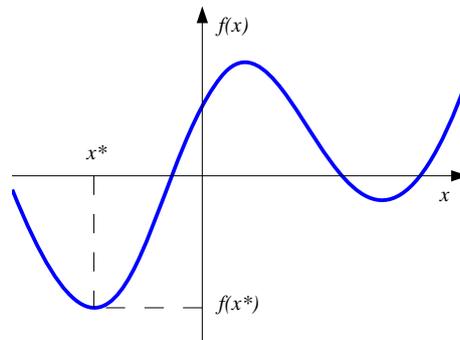


FIGURE 2.1 – Exemple de minimum d'un problème différentiable, non-convexe et sans contrainte.

Le type de la fonction de coût et des contraintes détermine le type de problème d'optimisation. De manière générale, le problème (2.1) n'admet pas de solutions. Nous allons voir quatre types de problèmes résolubles, du plus simple au plus complexe :

- l'optimisation linéaire sous contraintes linéaires ;
- l'optimisation différentiable sans contraintes ;
- l'optimisation différentiable sous contraintes ;
- l'optimisation non différentiable convexe sans contraintes.

La programmation linéaire a permis de résoudre des problèmes dans le cas simple d'une fonction et de contraintes linéaires. La fonction de coût est alors une somme pondérée des composantes de  $x$ , et les contraintes  $g_i(x) = 0$ ,  $h_j(x) \geq 0$  peuvent alors se représenter sous forme matricielle, pour donner le problème suivant, appelé *programme linéaire* :

$$\begin{aligned} \min_x f(x) = c^T x \\ \text{s.c. } Ax = b \text{ et } Dx \geq e \end{aligned} \quad (2.2)$$

où  $b, c, e$  et  $x$  sont des vecteurs colonnes de dimensions respectives  $p, n, q$  et  $n$ , alors que  $A$  et  $D$  sont des matrices de dimensions  $(p \times n)$  et  $(q \times n)$ .

Dans la section 2.2, deux méthodes de programmation linéaire sont présentées :

- la méthode du simplexe [Dantzig, 1963] ;
- la méthode de points intérieurs [Karmarkar, 1984].

Un second cas simple est le problème d'optimisation sans contrainte :

$$\min_x f(x) \tag{2.3}$$

où  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  est supposée régulière, i.e. de classe  $C^2$ . On suppose donné un produit scalaire  $\langle \cdot, \cdot \rangle$  sur  $\mathbb{R}^n$  et sa norme associée  $\| \cdot \| = \langle \cdot, \cdot \rangle^{\frac{1}{2}}$ . On note aussi  $\nabla f(x)$  et  $\nabla^2 f(x)$  le gradient et le hessien de  $f$  en  $x$ .

On s'intéresse ici à une classe d'algorithmes qui sont fondés sur la notion de direction de descente. On dit que  $d$  est une *direction de descente* de  $f$  en  $x$  si :

$$\langle \nabla f(x), d \rangle < 0$$

Par définition de la dérivée, si  $d$  est une direction de descente alors pour tout  $\alpha$  suffisamment petit, on a :

$$f(x + \alpha d) < f(x) \tag{2.4}$$

et donc  $f$  décroît strictement dans la direction  $d$ . Les méthodes à directions de descente utilisent cette idée, et construisent la suite des itérés  $\{x_k\}_{k \geq 1}$  approchant une solution  $x^*$  par la récurrence :

$$x_{k+1} = x_k + \alpha_k d_k, \quad \forall k \geq 1$$

où  $\alpha_k > 0$  est appelé le *pas*, et  $d_k$  est une direction de descente de  $f$  en  $x_k$ .

Un algorithme est donc déterminé par les paramètres  $d_k$  et  $\alpha_k$  : la façon dont la direction de descente  $d_k$  est calculée donne son nom à l'algorithme et la façon dont le pas  $\alpha_k$  est déterminé est appelé *recherche linéaire*. Un modèle d'algorithme basé sur les directions de descente est donné par l'algorithme 2.1.

---

**Algorithme 2.1 :** Algorithme de directions de descente

---

**Données :** Soient  $x_1 \in \mathbb{R}^n$  et  $\varepsilon > 0$

**tant que**  $\|\nabla f(x_k)\| > \varepsilon$  **faire**

1. Déterminer une direction de descente  $d_k$ .
2. Déterminer un pas  $\alpha_k > 0$  assez petit tel que  $f$  décroisse suffisamment.
3.  $x_{k+1} = x_k + \alpha_k d_k$
4.  $k = k + 1$

**fin**

---

Dans la section 2.3, quatre algorithmes de directions de descente sont présentés :

- l'algorithme du gradient ;
- l'algorithme du gradient conjugué ;
- l'algorithme de Newton ;
- l'algorithme de quasi-Newton.

Un problème plus complexe est la résolution du problème sous contraintes (2.1) où  $f$ ,  $g_i$  et  $h_j$  sont supposées de classe  $C^1$ . Le théorème de Kuhn-Tucker fournit une condition nécessaire pour que ce problème ait un minimum global.

**Théorème 2.1.1** (Kuhn-Tucker). *Soit  $f, g_i, i \in \{1, \dots, p\}, h_j, j \in \{1, \dots, q\}$  des fonctions de classe  $C^1$ . On veut minimiser  $f$  sur l'ensemble :*

$$K = \{x \in \mathbb{R}^n, g_i(x) = 0, i \in \{1, \dots, p\}, h_j(x) \leq 0, j \in \{1, \dots, q\}\}$$

*On suppose les contraintes qualifiées au point  $x^*$ , i.e.  $g_i(x^*) = 0$  et  $h_j(x^*) \leq 0$ . Alors une condition nécessaire pour que  $x^*$  soit un minimum de  $f$  est qu'il existe des nombres positifs  $\lambda_1^*, \dots, \lambda_q^*$  et des nombres réels  $\mu_1^*, \dots, \mu_p^*$  tels que*

$$\begin{cases} \nabla f(x^*) + \sum_{i=1}^p \mu_i^* \nabla g_i(x^*) + \sum_{j=1}^q \lambda_j^* \nabla h_j(x^*) = 0 \\ \lambda_j^* \nabla h_j(x^*) = 0, \quad 1 \leq j \leq q \end{cases} \quad (2.5)$$

On introduit le *lagrangien* du problème général (2.1) comme étant la fonction définie sur  $\mathbb{R}^n \times \mathbb{R}^p \times (\mathbb{R}_+)^q$  par :

$$\mathcal{L}(x, \mu, \lambda) = f(x) + \sum_{i=1}^p \mu_i g_i(x) + \sum_{j=1}^q \lambda_j h_j(x)$$

La relation (2.5) s'écrit alors :

$$\nabla \mathcal{L}(x^*, \mu^*, \lambda^*) = 0$$

Ces conditions deviennent suffisantes si la fonction de coût et les contraintes sont convexes.

Dans la section 2.4, trois algorithmes de minimisation sous contraintes sont présentés :

- l'algorithme du gradient projeté ;
- l'algorithme de pénalisation ;
- l'algorithme d'Uzawa.

Les problèmes variationnels d'analyse d'image peuvent souvent s'interpréter comme des problèmes avec contrainte sous forme lagrangienne. Dans notre cas, nous considérons uniquement les problèmes de minimisation convexe sans contraintes de la forme :

$$\min_x f(x) \quad (2.6)$$

où  $f : \mathbb{R}^n \rightarrow ]-\infty, +\infty]$  est une énergie convexe. Graphiquement, une fonction est dite convexe, si tout arc  $\widehat{AB}$  de la courbe représentative est situé au-dessous de sa corde  $[AB]$  (voir figure 2.2). L'avantage des problèmes convexes est que la solution globale peut être calculée, en général avec une bonne précision et dans un temps raisonnable, indépendamment de l'initialisation.

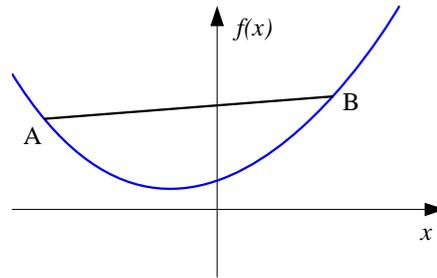


FIGURE 2.2 – Interprétation graphique d'une fonction convexe.

Plus précisément, en traitement d'images, l'énergie est divisée en une somme de fonctions convexes  $f_i$ ,  $i \in \{1, \dots, m\}$ . Une difficulté majeure de ce type de problème est que toutes les fonctions ne sont donc pas nécessairement différentiables et les méthodes classiques d'optimisation différentiable ne sont pas applicables. Les méthodes de point proximal sont alors utilisées pour résoudre ce type de problème. Un algorithme du gradient permet notamment d'obtenir  $\tilde{x}$  qui est ensuite projetée sur le domaine admissible en résolvant :

$$\min_x \left\{ f_i(x) + \frac{\|x - \tilde{x}\|^2}{2} \right\}$$

Dans la section 2.5, quatre classes d'algorithmes de point proximal pour résoudre le problème (2.6) sont décrits :

- l'algorithme Forward-Backward [Combettes et Wajs, 2005] ;
- l'algorithme Douglas-Rachford [Douglas et Rachford, 1956] ;
- l'algorithme PPXA [Combettes et Pesquet, 2008] ;
- l'algorithme primal-dual [Chambolle et Pock, 2011].

Enfin, nous appliquerons ces algorithmes de point proximal au débruitage d'images dans la section 2.5.8.

## 2.2 Algorithmes de programmation linéaire

### 2.2.1 Méthode du simplexe

La méthode du simplexe, établie en 1947 par Dantzig, part de la forme canonique d'un programme linéaire (2.2) :

$$\begin{aligned} \min_x f(x) &= c^T x \\ \text{s.c. } Ax &\geq b \text{ et } x \geq 0 \end{aligned}$$

On constate que chacune des contraintes définies par le couple  $(A, b)$  ampute le domaine de variation de  $x$  d'un demi espace bordé par un hyperplan. Le domaine de  $x$  résultant de l'ensemble des contraintes est un polyèdre convexe appelé polyèdre admissible (voir figure 2.3). La méthode consiste alors à partir d'un sommet initial (donc d'un point admissible) puis à passer successivement d'un sommet à un sommet adjacent, en améliorant constamment la valeur de l'objectif.

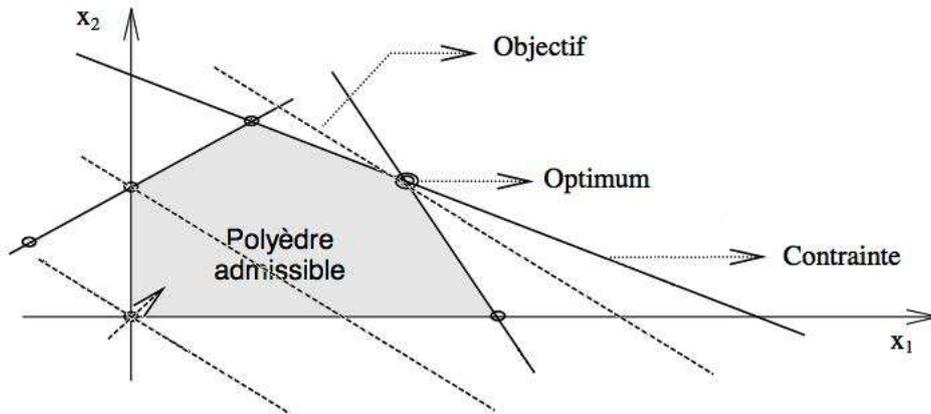


FIGURE 2.3 – Exemple de polyèdre admissible pour la méthode du simplexe.

### 2.2.2 Méthode de points intérieurs

La méthode de points intérieurs proposée en 1984 par Karmarkar, considère la forme standard d'un programme linéaire (2.2) :

$$\begin{aligned} \min_x f(x) &= c^T x \\ \text{s.c. } Ax &= b \text{ et } x \geq 0 \end{aligned}$$

À la différence de l'algorithme du simplexe, les méthodes de points intérieurs partent d'un point situé strictement à l'intérieur du polyèdre admissible :

$$x \in \mathcal{P}_0 = \{x \mid Ax = b \text{ et } x \geq 0\}$$

Le principe consiste alors à faire évoluer de façon itérative ce point vers une solution optimale du problème tout en restant à l'intérieur strict du polyèdre admissible. Plus précisément, on construit une suite  $x_0, x_1, x_2, \dots, x_i, \dots$  d'itérés intérieurs convergeant vers une solution, comme illustré sur la figure 2.4.

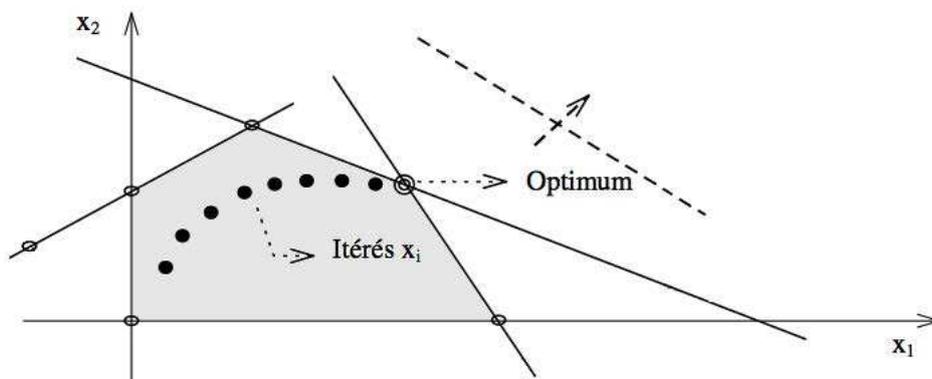


FIGURE 2.4 – Convergence des itérés d'une méthode de points intérieurs.

L'un des principaux avantages de la méthode de points intérieurs, qui a suscité un engouement en optimisation, est qu'elle a une complexité algorithmique de type polynomial, contrairement à la méthode du simplexe qui a une complexité de type exponentiel

dans le pire des cas. Cela signifie qu'il existe des cas pour la méthode du simplexe où le nombre d'itérations nécessaires pour atteindre la solution est exponentiel en fonction de la taille du problème ( $n$ ) alors que ce nombre d'itérations sera toujours borné par un polynôme fonction de  $n$  dans le cas de la méthode de points intérieurs. Cependant, il est à noter qu'il s'agit d'une complexité dans le pire des cas. En pratique, la méthode du simplexe est toujours utilisée car elle est simple à mettre en œuvre et est de complexité moyenne polynomiale.

## 2.3 Algorithmes à directions de descente

Dans cette section, les algorithmes sont testés sur une fonction test de type Rosenbrock. Cette fonction, aussi connue sous le nom de "fonction banane", n'est pas convexe mais présente un minimum global unique à l'intérieur d'une longue vallée étroite de forme parabolique (voir figure 2.5). La fonction  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  est définie par :

$$f(x, y) = (1 - x)^2 + 10(x^2 - y^2)^2$$

Le minimum global est obtenu au point  $(x,y)=(1,1)$ , pour lequel la fonction vaut 0.

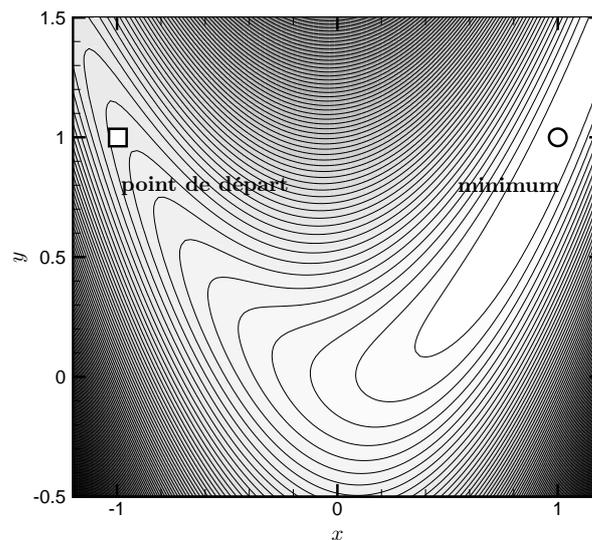


FIGURE 2.5 – Isovaleurs de la fonction de Rosenbrock. Le point de départ des algorithmes de descente est  $(x, y) = (-1, 1)$  et le minimum est localisé au point  $(x, y) = (1, 1)$  [Bergmann, 2004].

### 2.3.1 Recherche linéaire

La recherche linéaire consiste à déterminer le pas  $\alpha_k$  à effectuer le long d'une direction de descente  $d_k$ . Le premier objectif est de faire décroître  $f$  suffisamment. Cela est réalisé en forçant l'inégalité (2.4) en ajoutant un terme négatif  $\beta_k$  au second membre :

$$f(x + \alpha d) \leq f(x) + \beta_k \tag{2.7}$$

Le second objectif est d'empêcher le pas  $\alpha$  d'être trop petit. L'inégalité (2.7) est en général satisfaite pour des  $\alpha$  petits. Or, de trop petits pas peuvent entraîner une fausse convergence.

### 2.3.1.1 Recherche linéaire exacte

La recherche linéaire exacte consiste à déterminer le pas optimal, c'est-à-dire le pas qui minimise la fonction  $f$  le long de  $d_k$ . Le pas  $\alpha_k$  est donc solution du problème :

$$\alpha_k = \underset{\alpha \geq 0}{\operatorname{argmin}} f(x_k + \alpha d_k) \quad (2.8)$$

C'est ce qu'on appelle la *règle de Cauchy*, et le pas déterminé ainsi est appelé *pas de Cauchy* ou *pas optimal*. Il est parfois préférable de choisir le premier minimum local atteint par la fonction  $f$ , soit :

$$\alpha_k = \inf\{\alpha \geq 0 \mid f'_\alpha(x_k + \alpha d_k) = 0\} \quad (2.9)$$

On parle alors de *règle de Curry*, et le pas déterminé par cette règle est appelé *pas de Curry*. Ces deux règles ne sont utilisées que dans des cas particuliers, par exemple lorsque la fonction à minimiser est quadratique. Ce type de recherche linéaire demande en général un temps de calcul élevé sans permettre d'améliorer grandement la convergence de l'algorithme. D'autres règles de calcul, moins restrictives, peuvent alors être utilisées. Les règles d'Armijo, de Goldstein et de Wolfe n'autorisent plus une unique solution pour le pas  $\alpha_k$  mais un ensemble d'intervalles solutions.

### 2.3.1.2 Règles d'Armijo et de Goldstein

Ce type de méthodes se base sur la condition (2.7). Le principe consiste alors à faire décroître  $f$  autant qu'une portion  $\omega_1 \in ]0, 1[$  de ce que ferait le modèle linéaire de  $f$  en  $x_k$ . Cela conduit à l'inégalité suivante, appelée règle d'Armijo :

$$f(x_k + \alpha d_k) \leq f(x_k) + \omega_1 \alpha_k \langle \nabla f_k, d_k \rangle \quad (2.10)$$

où  $\omega_1$  est une constante habituellement choisie très petite (de l'ordre de  $10^{-4}$ ). Il est évident qu'un pas  $\alpha_k$  très petit satisfait la condition (2.10). Or, il est dangereux d'accepter un pas  $\alpha_k$  trop petit, il faut donc utiliser un mécanisme supplémentaire qui empêche le pas d'être trop petit, tel que la méthode d'Armijo à rebroussement et la méthode de Goldstein.

La méthode d'Armijo à rebroussement (*backtracking* en anglais) consiste à prendre  $\alpha_k = \tau^{i_k}$ , où  $\tau \in ]0, 1[$  est une constante et  $i_k$  est le plus petit entier positif vérifiant la condition (2.10). C'est le fait de prendre pour  $\alpha_k$  le plus grand réel dans  $\{1, \tau, \tau^2, \dots\}$  vérifiant (2.10) qui garantit que ce pas ne sera pas trop petit. Cette technique porte le nom de rebroussement, car on essaie d'abord  $\alpha_k = 1$  puis, si ce n'est pas acceptable, on rebrousse chemin en essayant des pas plus petits  $\tau, \tau^2, \dots$

Dans la méthode de Goldstein, la détermination du pas  $\alpha_k$  doit satisfaire l'équation (2.10) et l'équation suivante :

$$f(x_k + \alpha d_k) \geq f(x_k) + \omega'_1 \alpha_k \langle \nabla f_k, d_k \rangle \quad (2.11)$$

où  $\omega'_1$  est une constante dans l'intervalle  $] \omega_1, 1[$ . C'est cette inégalité qui empêche le pas d'être trop petit. Le pas déterminé par cette règle est appelé *pas de Goldstein*.

### 2.3.1.3 Règle de Wolfe

La méthode de Wolfe consiste toujours à déterminer un pas  $\alpha_k$ , appelé *pas de Wolfe*, vérifiant la condition (2.10) et la condition supplémentaire :

$$\langle \nabla f(x_k + \alpha d_k), d_k \rangle \geq \omega_2 \alpha_k \langle \nabla f_k, d_k \rangle \quad (2.12)$$

où les constantes  $\omega_1$  et  $\omega_2$  sont choisies telles que  $0 < \omega_1 < \omega_2 < 1$ . C'est cette dernière inégalité qui empêche le pas d'être trop petit.

### 2.3.2 Algorithme du gradient

L'algorithme utilise pour direction de descente l'opposé du gradient de la fonction  $f$ , appelé *direction du gradient* ou de *la plus profonde descente* :

$$d_k = -\nabla f(x_k)$$

La direction du gradient est évidemment une direction de descente, car on a  $\langle \nabla f(x_k), d_k \rangle = -\|\nabla f(x_k)\|^2 < 0$ .

Supposons maintenant que la fonction  $f$  est de classe  $C^1$  et de gradient lipschitzien de constante  $L > 0$ , i.e. :

$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, \|\nabla f(y) - \nabla f(x)\| \leq L\|y - x\|$$

On peut alors montrer que la fonction objectif  $f(x_k)$  décroît toujours pour l'algorithme de descente à pas optimal, tandis que dans le cas d'un pas fixe elle décroît pour un  $\alpha$  bien choisi ( $\alpha < \frac{2}{L}$ ).

Cependant, cet algorithme présente deux inconvénients majeurs. Le premier est la convergence lente lorsque l'on se rapproche du minimum de la fonction. Dans un tel cas, la suite des itérés  $(x_k)_k$  oscille orthogonalement et progresse laborieusement vers la solution (voir figure 2.6). Le second est la recherche du pas optimal  $\alpha_k$  qui peut se révéler très longue. Inversement, utiliser un pas  $\alpha$  fixe peut conduire à de mauvais résultats.

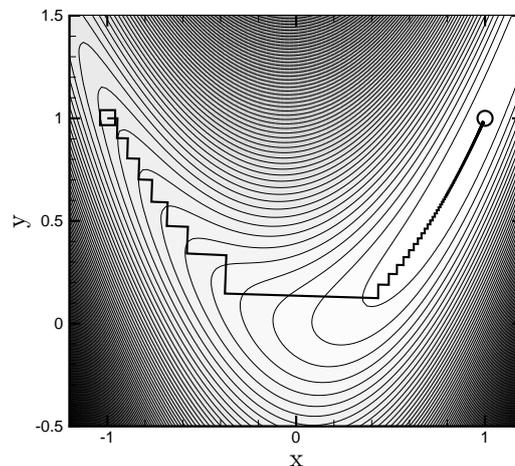


FIGURE 2.6 – Algorithme du gradient à pas optimal appliqué à la fonction de Rosenbrock (convergence : 406 itérations) [Bergmann, 2004].

### 2.3.3 Algorithme du gradient conjugué

Il s'agit d'une variante de la méthode du gradient introduite tout d'abord pour le problème de minimisation quadratique :

$$\min_x f(x) := \frac{1}{2}x^T Ax - b^T x \quad (2.13)$$

où  $A$  est une matrice carrée d'ordre  $n$ , symétrique, définie positive, et  $b \in \mathbb{R}^n$ . En dérivant cette expression, on peut montrer que la solution du problème est la solution du système  $Ax = b$ .

**Définition 2.3.1.** Des vecteurs  $u_1, \dots, u_k$  de  $\mathbb{R}^n$  sont dits  $A$ -conjugués si pour tout  $i \neq j$ , on a :

$$\langle u_i, u_j \rangle_A := \langle u_i, Au_j \rangle = u_i^T Au_j = 0$$

**Théorème 2.3.2.** Soient  $u_1, \dots, u_k$  des vecteurs non nuls de  $\mathbb{R}^n$ ,  $A$ -conjugués, alors le système est libre.

L'idée est alors de définir un algorithme itératif utilisant  $n$  directions conjuguées et de parcourir ainsi tout l'espace  $\mathbb{R}^n$  dans toutes les directions possibles (ce qui n'était pas le cas pour l'algorithme du gradient). Ainsi la solution  $x^*$  de  $Ax = b$  dans cette base est  $x^* = \sum_{i=1}^n \alpha_i u_i$  avec  $\alpha_i = \frac{\langle b, u_i \rangle}{\langle u_i, u_i \rangle_A}$ . On construit itérativement les  $n$  directions conjuguées à partir des gradients calculés à chaque itération par le procédé d'orthogonalisation de Gram-Schmidt :

$$d_k = \begin{cases} -\nabla f(x_1) & \text{si } k = 1 \\ -\nabla f(x_k) + \sum_{j=1}^{k-1} \beta_{k,j} d_j \text{ avec } \beta_{k,j} = \frac{\langle \nabla f(x_k), d_j \rangle_A}{\langle d_j, d_j \rangle_A} & \text{si } k \geq 2 \end{cases}$$

Ce procédé est applicable puisque la famille des  $\nabla f(x_k)$  est libre. On veut  $\langle d_k, d_i \rangle_A = 0$  pour tout  $i < k$ , soit :

$$-\langle \nabla f(x_k), d_i \rangle_A + \sum_{j=1}^{k-1} \beta_{k,j} \langle d_j, d_i \rangle_A = 0$$

Or,  $\langle d_j, d_i \rangle_A = 0$  si  $j \neq i$ , d'où :

$$\beta_{k,j} = \frac{\langle \nabla f(x_k), d_j \rangle_A}{\langle d_j, d_j \rangle_A}$$

Finalement, on peut généraliser l'algorithme aux fonctions non quadratiques en démontrant que les directions conjuguées  $d_k$  sont données par une formule plus simple :

$$d_k = -\nabla f(x_k) + \beta_k d_{k-1} \quad (2.14)$$

où le terme  $\beta_k$  peut être calculé par plusieurs méthodes :

- la méthode de Fletcher-Reeves :  $\beta_k = \frac{\|\nabla f(x_k)\|^2}{\|\nabla f(x_{k-1})\|^2}$  ;
- la méthode de Polack-Ribière :  $\beta_k = \frac{\langle \nabla f(x_k), \nabla f(x_k) - \nabla f(x_{k-1}) \rangle}{\|\nabla f(x_{k-1})\|^2}$  ;
- la méthode de Hestenes-Stiefel :  $\beta_k = \frac{\langle \nabla f(x_k), \nabla f(x_k) - \nabla f(x_{k-1}) \rangle}{\langle \nabla f(x_k) - \nabla f(x_{k-1}), d_{k-1} \rangle}$ .

La figure 2.7 compare l'évolution de l'algorithme du gradient conjugué à pas optimal pour ces trois différentes méthodes. On peut constater que les méthodes de Polack-Ribière et de Hestenes-Stiefel convergent en moins d'itérations que la méthode de Fletcher-Reeves dans le cas de la fonction de Rosenbrock.

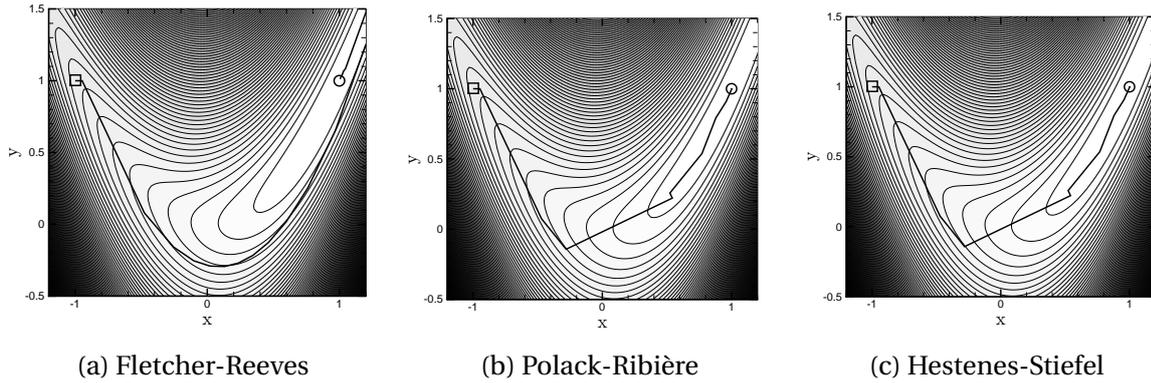


FIGURE 2.7 – Algorithme du gradient conjugué à pas optimal appliqué à la fonction Rosenbrock (convergence : (a) 25 itérations, (b) 10 et (c) 10) [Bergmann, 2004].

### 2.3.4 Algorithme de Newton

Dans le cas unidimensionnel, la méthode de Newton est une méthode itérative pour résoudre l'équation  $f(x) = 0$  où  $f$  est de classe  $C^2(\mathbb{R}, \mathbb{R})$  avec  $f'(x)$  inversible. Considérons le développement de Taylor à l'ordre 1 de la fonction  $f$  en  $x_0$  :

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + O((x - x_0)^2)$$

Pour trouver un zéro de cette fonction, on ne garde que la partie linéaire du développement :

$$0 = f(x_0) + (x - x_0)f'(x_0)$$

On obtient alors le point  $x_1$  :

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Graphiquement, cela revient à tracer la tangente à la courbe représentative de  $f$  et à chercher où elle croise l'axe des  $x$ . À partir d'un point  $x_0$  proche de la solution, on construit alors par récurrence la suite :

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

En optimisation sans contrainte dans le cas multidimensionnel, l'algorithme de Newton cherche les solutions de l'équation  $\nabla f(x) = 0$ , autrement dit, les points critiques de la fonction  $f$  à minimiser. En suposant que la matrice hessienne  $\nabla^2 f(x_k)$  est inversible, on détermine une direction de descente, appelée *direction de Newton*, par la formule suivante :

$$d_k = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$$

La figure 2.8 représente l'évolution de l'algorithme de Newton à pas optimal. On peut remarquer que la convergence est réalisée en moins d'itérations que pour les méthodes de gradient. Le principal atout de cet algorithme est la convergence quadratique ; en contrepartie, le principal inconvénient est le calcul de la matrice hessienne qui n'est pas toujours réalisable ou très coûteux.

### 2.3.5 Algorithme de quasi-Newton

Pour pallier la difficulté du calcul de la hessienne dans l'algorithme de Newton, l'algorithme de quasi-Newton utilise une approximation  $B_k$ . La direction de Newton s'écrit

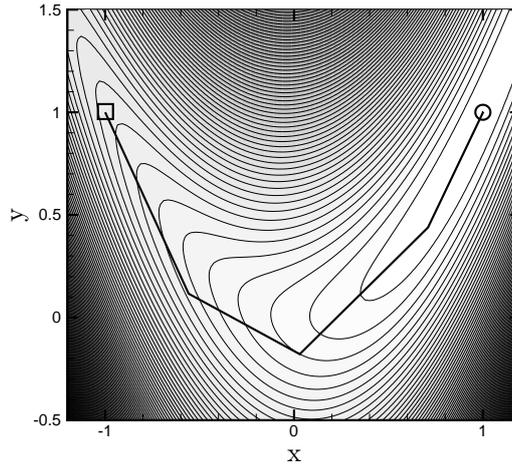


FIGURE 2.8 – Algorithme de Newton à pas optimal, appliqué à la fonction Rosenbrock (convergence : 7 itérations) [Bergmann, 2004].

alors :

$$d_k = -B_k^{-1} \nabla f(x_k)$$

Étant donné que  $\nabla f(x_k) \simeq \nabla f(x_{k+1}) + \nabla^2 f(x_k)(x_k - x_{k+1})$ , on construit une approximation  $B_{k+1}$  de  $\nabla^2 f(x_{k+1})$  comme solution de l'équation de quasi-Newton :

$$\nabla f(x_{k+1}) - \nabla f(x_k) = B_{k+1}(x_{k+1} - x_k)$$

Parmi les méthodes d'approximation de la hessienne, on peut en retenir trois :

- la méthode *BFGS* pour Broyden-Fletcher-Goldfarb-Shanno :

$$B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k};$$

- la méthode *DFP* pour Davidon-Fletcher-Powell :

$$B_{k+1} = B_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{B_k y_k y_k^T B_k}{y_k^T B_k y_k};$$

- la méthode *SR1* pour Matrice Symétrique de Rang 1 :

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}.$$

où  $s_k = x_{k+1} - x_k$  et  $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ .

La figure 2.9 compare l'évolution de l'algorithme de quasi-Newton à pas optimal pour ces trois différentes méthodes.

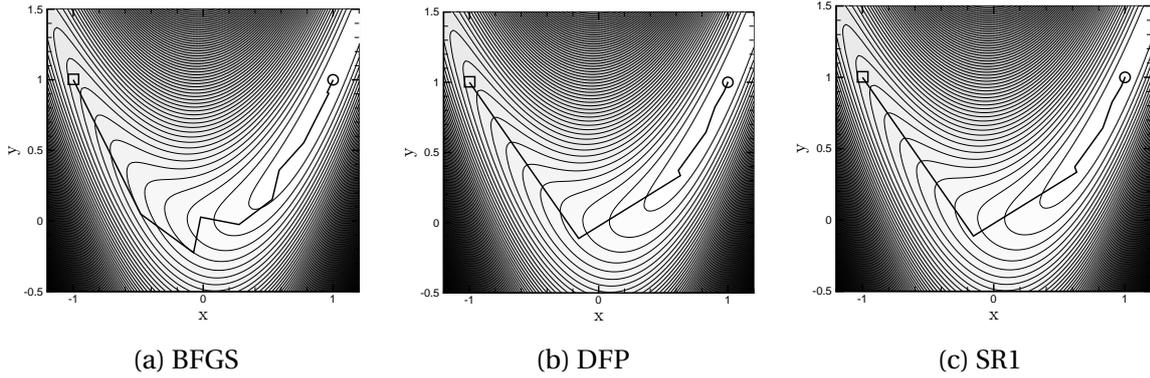


FIGURE 2.9 – Algorithme de quasi-Newton à pas optimal appliqué à la fonction Rosenbrock (convergence : (a) 13 itérations, (b) 10 et (c) 10) [Bergmann, 2004].

## 2.4 Algorithmes de minimisation sous contraintes

Dans cette section, on cherche à résoudre le problème de minimisation sous contrainte :

$$\begin{aligned} \min_x f(x) \\ \text{s.c. } x \in K \end{aligned} \quad (2.15)$$

où  $K = \{x \in \mathbb{R}^n, g_i(x) = 0, i \in \{1, \dots, p\}, h_j(x) \leq 0, i \in \{1, \dots, q\}\}$

### 2.4.1 Algorithme du gradient projeté

L'algorithme du gradient projeté s'inspire de l'algorithme du gradient décrit dans la section précédente. L'idée de base consiste à suivre la direction de plus profonde descente, comme dans le cas sans contraintes :

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

où  $\alpha_k > 0$  est choisi de sorte que  $f(x_{k+1}) < f(x_k)$ . Toutefois, si  $x_k \in K$ , rien ne garantit que  $x_{k+1}$  appartienne également à  $K$ . Il est donc nécessaire de projeter  $x_{k+1}$  sur  $K$  à chaque étape.

**Définition 2.4.1** (Projection sur un convexe fermé). *Soit  $K$  un convexe fermé et soit  $x \in \mathbb{R}^n$ . On appelle projeté de  $x$  sur  $K$ , l'unique élément  $\tilde{x} \in K$  solution du problème :*

$$\min_{y \in K} \|x - y\|$$

Le projeté de  $x$  est noté  $\tilde{x} = P_K(x)$ , et est caractérisé par :

$$\langle \tilde{x} - x, \tilde{x} - y \rangle \leq 0, \forall y \in K$$

On obtient alors l'algorithme 2.2 ci-après.

Une première remarque est que l'on ne peut plus choisir comme test d'arrêt  $\|\nabla f(x_k)\| < \varepsilon$ , car pour le problème avec contraintes, le minimum  $x$  ne satisfait pas forcément  $\nabla f(x) = 0$ . On peut également remarquer que le calcul du projeté sur  $K$  peut parfois être aussi difficile que le problème initial. En effet  $x_{k+1}$  est obtenu en résolvant le problème :

$$\min_{y \in K} \|x_k - \alpha_k \nabla f(x_k) - y\|$$

**Algorithme 2.2 :** Algorithme du gradient projeté

**Données :** Soient  $x_1 \in K$  et  $\varepsilon > 0$   
**tant que**  $\|x_{k+1} - x_k\| > \varepsilon$  **faire**

1. Déterminer un pas  $\alpha_k > 0$  par recherche linéaire.
2.  $y_k = x_k - \alpha_k \nabla f(x_k)$
3.  $x_{k+1} = P_K(y_k)$
4.  $k = k + 1$

**fin**

Il s'agit donc de résoudre un problème d'optimisation sur un convexe, avec une fonction de coût convexe. Lorsque le domaine  $K$  des contraintes est simple (contraintes de bornes en particulier), une telle résolution est réalisable. Néanmoins, dès que les contraintes ne sont pas des contraintes de bornes, le calcul de la projection devient beaucoup plus délicat.

De même que pour l'algorithme du gradient, on peut montrer que si  $f$  est convexe et Lipschitz différentiable de constante  $L$ , alors la suite des itérés  $(x_k)_k$  converge pour un pas optimal ou pour un pas fixe  $\alpha < \frac{2}{L}$ .

### 2.4.2 Algorithme de pénalisation

Les méthodes de pénalisation sont très utilisées en pratique, car elles sont très simples. Elles partent du principe suivant : on remplace le problème avec contraintes (2.15) par le problème sans contraintes suivant :

$$\min_{x \in \mathbb{R}^n} f_\varepsilon(x) = f(x) + \frac{1}{\varepsilon} p(x) \quad (2.16)$$

où  $\varepsilon > 0$  est le paramètre de pénalisation et  $p : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction de pénalisation. Le but est de trouver des fonctions  $p$  telles que les problèmes (2.15) et (2.16) soient équivalents, c'est-à-dire qu'ils aient les mêmes solutions. Dans ce cas, on dit que la pénalisation est exacte. On peut, par exemple, choisir :

$$p(x) = \begin{cases} 0 & \text{si } x \in K \\ +\infty & \text{sinon} \end{cases}$$

Néanmoins,  $p$  n'a pas de bonnes propriétés mathématiques (notamment la continuité) et les algorithmes de minimisation sans contraintes ne sont pas applicables. On modifie alors  $p$  pour avoir une fonction de *pénalisation extérieure* qui satisfait les conditions suivantes :

1.  $p$  est continue sur  $\mathbb{R}^n$
2.  $\forall x \in \mathbb{R}^n, p(x) \geq 0$
3.  $p(x) = 0 \Leftrightarrow x \in K$

Le qualificatif "extérieur" vient de la propriété (3), qui exprime que  $f_\varepsilon$  ne modifie  $f$  qu'à l'extérieur de l'ensemble admissible. En effet, quand  $\varepsilon$  devient petit, un minimum  $x_\varepsilon$  de  $f_\varepsilon$  va avoir tendance à rendre  $p$  petit et donc proche de 0, c'est-à-dire que  $x_\varepsilon$  sera presque dans  $K$ .

Le tableau 2.1 fournit quelques exemples de fonctions de pénalisation extérieure pour différentes contraintes.

Contraintes $x \in K$	Choix de $p$
$x \leq 0$	$p(x) = \ \max(0, x)\ ^2$
$g(x) = 0$	$p(x) = \ g(x)\ ^2$
$h(x) \leq 0$	$p(x) = \ \max(0, h(x))\ ^2$

TABLEAU 2.1 – Exemples de pénalisation extérieure.

Dans certains cas, le fait que les itérés  $x_\epsilon$  générés par la pénalisation extérieure ne soient pas admissibles peut être un inconvénient, en particulier quand la fonction  $f$  n'est pas définie à l'extérieur de  $K$ . On peut introduire des méthodes de pénalisation dans lesquelles les itérés  $x_\epsilon$  restent dans  $K$ . L'idée est d'utiliser un terme de pénalisation qui tend vers  $+\infty$  lorsque  $x$  s'approche de la frontière  $\partial K$  de  $K$ . On obtient alors la fonction de *pénalisation intérieure* qui satisfait les conditions suivantes :

1.  $p$  est continue sur  $\text{int} K$
2.  $\forall x \in \text{int} K, p(x) \geq 0$
3.  $p(x) \rightarrow +\infty$  lorsque  $x$  se rapproche de  $\partial K$

La condition (3) crée une "barrière" au bord de l'ensemble admissible. Le tableau 2.2 donne des exemples de fonctions de pénalisation intérieure pour différentes contraintes.

Contraintes $x \in K$	Choix de $p$	Nom de la pénalisation
$h(x) \leq 0$	$p(x) = \sum_{i=1}^q \frac{1}{-h_i(x)}$	intérieure inverse
$h(x) \leq 0$	$p(x) = -\sum_{i=1}^q \log(-h_i(x))$	logarithmique

TABLEAU 2.2 – Exemples de pénalisation intérieure.

### 2.4.3 Méthode d'Uzawa

L'idée générale est de considérer le lagrangien  $\mathcal{L}$  au lieu de la fonction  $f$  ; ce choix est motivé (au moins) par deux raisons :

- la fonction lagrangienne englobe à la fois la fonction  $f$  et les contraintes  $g$  et  $h$ , et représente bien le problème ;
- une condition nécessaire du premier ordre, pour que  $x^*$  soit un minimum de  $f$  avec contraintes, est que  $x^*$  (associé aux multiplicateurs de Lagrange) soit un point critique de  $\mathcal{L}$ .

**Définition 2.4.2.** On appelle *point-selle* de  $\mathcal{L}$  sur  $\mathbb{R}^n \times \mathbb{R}^p \times (\mathbb{R}_+)^q$  tout triplet  $(x^*, \mu^*, \lambda^*)$  vérifiant l'équation :

$$\mathcal{L}(x^*, \mu, \lambda) \leq \mathcal{L}(x^*, \mu^*, \lambda^*) \leq \mathcal{L}(x, \mu^*, \lambda^*), \forall (x, \mu, \lambda) \in \mathbb{R}^n \times \mathbb{R}^p \times (\mathbb{R}_+)^q \quad (2.17)$$

On a alors le résultat suivant, dont une démonstration est donnée dans [Sorin, 2013, Théorème 4.16] :

**Théorème 2.4.3.** *Supposons que  $f, g$  et  $h$  soient des fonctions  $C^1$  et que le triplet  $(x^*, \mu^*, \lambda^*) \in \mathbb{R}^n \times \mathbb{R}^p \times (\mathbb{R}_+)^q$  soit un point-selle de  $\mathcal{L}$  sur  $\mathbb{R}^n \times \mathbb{R}^p \times (\mathbb{R}_+)^q$ . Alors ce triplet vérifie les conditions de Kuhn-Tucker.*

Dans le cas convexe, on a une condition suffisante. Ce théorème nous aide à comprendre que, pour chercher le triplet  $(x^*, \mu^*, \lambda^*) \in \mathbb{R}^n \times \mathbb{R}^p \times (\mathbb{R}_+)^q$  vérifiant les conditions de Kuhn-Tucker, on peut procéder de la façon suivante :

- Pour  $(\mu^*, \lambda^*) \in \mathbb{R}^p \times (\mathbb{R}_+)^q$ , fixés, on peut chercher le minimum sans contrainte (i.e. sur tout l'espace  $\mathbb{R}^n$ ) de la fonction  $x \mapsto \mathcal{L}(x, \mu^*, \lambda^*)$ . C'est ce que traduit le terme de droite de l'équation (2.17), et on obtient le problème primal suivant :

$$\inf_{x \in \mathbb{R}^n} \left( \sup_{(\mu, \lambda) \in \mathbb{R}^p \times (\mathbb{R}_+)^q} \mathcal{L}(x, \mu, \lambda) \right)$$

Pour minimiser sans contraintes, on peut utiliser les méthodes du gradient, gradient conjugué, etc.

- Pour  $x^* \in \mathbb{R}^n$  fixé, on cherche le maximum sur  $\mathbb{R}^p \times (\mathbb{R}_+)^q$  de la fonction  $(\mu, \lambda) \mapsto \mathcal{L}(x^*, \mu, \lambda)$ . C'est ce que traduit le terme de gauche de l'équation (2.17), et on obtient le problème dual suivant :

$$\sup_{(\mu, \lambda) \in \mathbb{R}^p \times (\mathbb{R}_+)^q} \left( \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \mu, \lambda) \right)$$

Pour maximiser avec la contrainte plus facile  $\lambda_j > 0$ , on peut utiliser la méthode du gradient projeté, par exemple.

## 2.5 Algorithmes d'optimisation convexe

En règle générale, la fonction  $\text{TV}(u)$  des modèles ROF et TV-L1 n'est pas différentiable. Nous allons donc étudier quatre classes d'algorithmes permettant de résoudre un problème général d'optimisation convexe de la forme :

$$\min_{x \in \mathbb{R}^n} f_1(x) + \dots + f_m(x) \tag{2.18}$$

où  $f_1, \dots, f_m$  sont des fonctions convexes de  $\mathbb{R}^n$  dans  $]-\infty, +\infty]$  et au moins une fonction  $f_i$  n'est pas différentiable. L'avantage principal de l'optimisation convexe est que le minimum local correspond toujours au minimum global.

Ces algorithmes sont décrits dans [Combettes et Pesquet, 2011] comme des *algorithmes proximaux découplés*. Le terme *découplé* signifie que l'énergie à minimiser peut être divisée en plusieurs fonctions  $f_1, \dots, f_m$ , qui sont plus simples à manipuler individuellement. Le terme *proximal* provient du fait que chaque fonction non-différentiable est traitée à l'aide de son opérateur proximal. Dans un premier temps, nous allons rappeler la notion de sous-différentielle, la définition d'opérateur proximal et donner quelques exemples.

### 2.5.1 Notion de sous-différentielle

On rappelle que la fonction  $f : \mathbb{R}^n \mapsto ]-\infty, +\infty]$  est :

- propre si et seulement si elle n'est pas identiquement égale à  $+\infty$  (i.e. son domaine  $\text{dom } f = \{x \in \mathbb{R}^n : f(x) < +\infty\}$  est non vide).
- semi-continue inférieurement (s.c.i.) si et seulement si pour tout  $x_0 \in \mathbb{R}^n$

$$f(x_0) \leq \liminf_{x \rightarrow x_0} f(x)$$

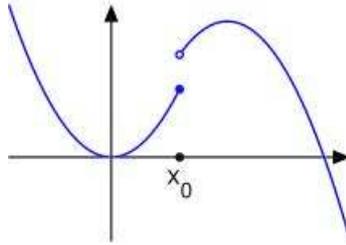


FIGURE 2.10 – Fonction semi-continue inférieurement.

- convexe si et seulement si pour tout  $x, y \in \mathbb{R}^n$  et  $t \in [0, 1]$

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$$

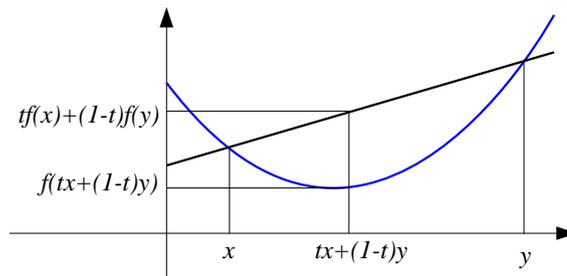


FIGURE 2.11 – Fonction convexe.

On pose  $\Gamma^0(\mathbb{R}^n)$  l'ensemble de toutes les fonctions convexes, propres et s.c.i. sur  $\mathbb{R}^n$ .

**Définition 2.5.1** (Sous-différentielle). On définit comme la sous-différentielle de  $f$  au point  $x$ , l'ensemble  $\partial f(x)$  défini par :

$$\partial f(x) := \{u \in \mathbb{R}^n \mid \forall y \in \mathbb{R}^n, \langle y - x, u \rangle + f(x) \leq f(y)\}$$

**Exemple 2.5.2** (Exemples de sous-différentielle). La figure 2.12 illustre deux cas :

- Si  $f(x) = x^2$  alors  $\forall x \in \mathbb{R}^n, \partial f(x) = 2x$ . De manière générale si  $f$  est différentiable alors  $\forall x \in \mathbb{R}^n, \partial f(x) = \nabla f(x)$ .
- Si  $f(x) = |x|$ , alors

$$\forall x \in \mathbb{R}^n, \partial f(x) = \begin{cases} \text{sign}(x) & \text{si } x \neq 0 \\ [-1, +1] & \text{si } x = 0 \end{cases}$$

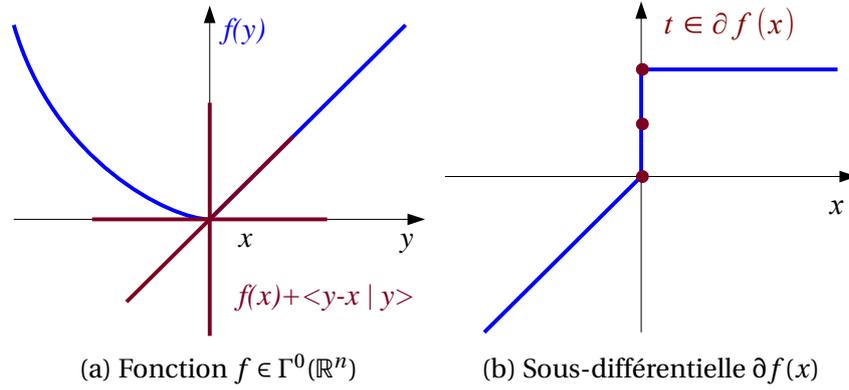


FIGURE 2.12 – Exemples de sous-différentielle.

## 2.5.2 Opérateur proximal : définition et exemples

**Exemple 2.5.3** (Projection sur un ensemble convexe). Soient  $C$  un ensemble convexe (i.e.  $\forall x, y \in C, \forall t \in [0, 1], tx + (1-t)y \in C$ ) et  $\mathbb{1}_C$  la fonction indicatrice de  $C$  définie par

$$\mathbb{1}_C(x) = \begin{cases} 0 & \text{si } x \in C \\ +\infty & \text{sinon} \end{cases}$$

Alors  $P_C(x) = \arg \min_{y \in \mathbb{R}^n} \{ \mathbb{1}_C(y) + \frac{1}{2} \|y - x\|^2 \}$ , appelée projection sur le convexe  $C$ , est l'unique point vérifiant :

$$\forall y \in C, \quad \|x - P_C(x)\| \leq \|x - y\|$$

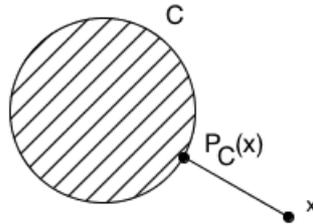


FIGURE 2.13 – Projection sur un ensemble convexe.

On étend cette notion en définissant l'opérateur proximal d'une fonction dans  $\Gamma^0(\mathbb{R}^n)$  :

**Définition 2.5.4** (Opérateur proximal). Soit  $f \in \Gamma^0(\mathbb{R}^n)$ . Pour tout  $x \in \mathbb{R}^n$ , le problème de minimisation :

$$\min_{y \in \mathbb{R}^n} f(y) + \frac{1}{2} \|x - y\|^2$$

admet une unique solution que l'on note  $\text{prox}_f(x)$ . L'opérateur  $\text{prox}_f : \mathbb{R}^n \rightarrow \mathbb{R}^n$  ainsi défini est l'opérateur proximal de  $f$ .

Si  $f \in \Gamma^0(\mathbb{R}^n)$ , alors l'opérateur proximal  $\text{prox}_f$  est caractérisé par l'inclusion :

$$x_p = \text{prox}_f(x) \Leftrightarrow x - x_p \in \partial f(x_p) \quad (2.19)$$

**Remarque :** L'opérateur proximal peut prendre la notation suivante :

$$x_p = \text{prox}_f(x) = (I + \partial f)^{-1}(x)$$

En effet, on a  $x - x_p \in \partial f(x_p) \Leftrightarrow x \in x_p + \partial f(x_p) \Leftrightarrow x = (I + \partial f)(x_p) \Leftrightarrow x_p = (I + \partial f)^{-1}(x)$  où  $(I + \partial f)^{-1}$  est appelé opérateur résolvant.

En reprenant l'inclusion (2.19), on a :

$$x_p = \text{prox}_f(x) \Leftrightarrow \exists g_p \in \partial f(x_p), x_p = x - \tau g_p$$

De manière générale, l'algorithme proximal génère une suite  $(x_k)_{k \geq 1}$  par la formule :

$$x_{k+1} = x_k - \tau_k g_{k+1} \quad \text{où } g_{k+1} \in \partial f(x_{k+1})$$

C'est une méthode de sous-gradient implicite de pas  $\tau_k$ . En effet, le sous-gradient  $g_{k+1}$  est évalué en  $x_{k+1}$  (qui est inconnu), plutôt qu'en  $x_k$  dans l'algorithme du gradient. Cet algorithme peut sembler étrange aux premiers abords puisque pour minimiser  $f$ , il faut à chaque itération minimiser la fonction  $f_k$  définie par :

$$f_k(y) = f(y) + \frac{1}{2} \|x_k - y\|^2$$

En réalité, il n'en est rien puisqu'il suffit de déterminer le sous-gradient souvent facilement calculable comme dans l'exemple suivant. De plus, la fonction  $f_k(y)$  est fortement convexe et donc de minimum unique, ce qui a un effet stabilisant dans le cas où  $f$  aurait plusieurs minima.

**Exemple 2.5.5** (Seuillage doux). Soit  $f(x) = |x|$ , calculons son opérateur proximal. Ici, le sous-gradient est défini par :

$$\partial f(x) = \begin{cases} 1 & \text{si } x > 0 \\ [-1, 1] & \text{si } x = 0 \\ -1 & \text{si } x < 0 \end{cases}$$

En utilisant la caractérisation (2.19), on obtient :

$$y = \begin{cases} x - 1 & \text{si } x \geq 1 \\ 0 & \text{si } -1 < x < 1 \\ x + 1 & \text{si } x \leq -1 \end{cases}$$

Cette fonction est appelée un seuillage doux et est représentée sur la figure 2.14.

### 2.5.3 Opérateur proximal de la variation totale

Un autre exemple de calcul d'opérateur proximal utile pour la suite est celui de la variation totale  $J(u) = |\nabla u|$  introduite dans le chapitre précédent.

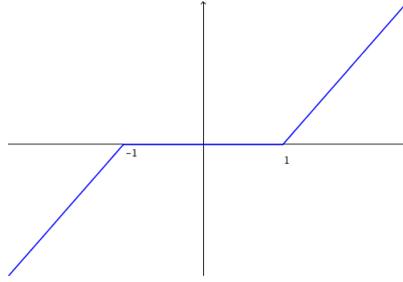


FIGURE 2.14 – Seuillage doux

### 2.5.3.1 Transformation de Legendre-Fenchel

Pour calculer l'opérateur proximal de la variation totale, nous devons tout d'abord définir la transformée de Legendre-Fenchel (voir annexe A.1 pour plus de détails) et donner une proposition, un corollaire courant d'analyse convexe et un théorème dont on trouvera la preuve dans [Hiriart-Urruty et Lemaréchal, 1993].

**Définition 2.5.6** (Transformée de Legendre-Fenchel). Soit  $J : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  une fonction propre. La fonction conjuguée, ou transformée de Legendre-Fenchel,  $J^* : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  de  $J$  est la fonction définie par :

$$J^*(p) = \sup_{u \in \mathbb{R}^n} (\langle u, p \rangle - J(u))$$

Notons que  $J^*$  est une fonction convexe s.c.i. En effet, pour chaque  $u \in \mathbb{R}^n$  fixé, l'application  $p \mapsto \langle p, u \rangle - J(u)$  est convexe et continue, donc s.c.i. Par suite, l'enveloppe supérieure de ces fonctions est convexe et s.c.i.

**Proposition 2.5.7.** Soit  $J$  une fonctionnelle homogène de degré 1 (i.e.  $\forall u \in \mathbb{R}^n$  et  $\lambda > 0$ , on a  $J(\lambda u) = \lambda J(u)$ ). Alors  $J^*$  est la fonction indicatrice d'un convexe fermé  $K$  :

$$J^*(p) = \mathbb{1}_K(p) = \begin{cases} 0 & \text{si } p \in K \\ +\infty & \text{sinon} \end{cases}$$

**Corollaire 2.5.8.** Soit  $J$  une fonctionnelle homogène de degré 1, alors  $J(u) = \sup_{p \in K} \langle u, p \rangle$  (Idée de la preuve :  $J^{**} = J^*$  d'après le théorème de Fenchel-Moreau, c.f. Annexe A.2).

**Théorème 2.5.9.** Soit  $J \in \Gamma^0(\mathbb{R}^n)$  alors  $p \in \partial J(u) \Leftrightarrow u \in \partial J^*(p)$ .

### 2.5.3.2 Opérateur proximal et opérateur de projection

Par définition, l'opérateur proximal de  $J(u) = |\nabla u|$  est donné par :

$$\text{prox}_J(f) = \underset{u}{\text{argmin}} J(u) + \frac{1}{2\lambda} \|u - f\|^2$$

L'équation de Lagrange-Euler associée est :

$$0 \in \partial J(u) + \frac{1}{\lambda}(u - f)$$

ce qui est équivalent à :

$$\frac{f - u}{\lambda} \in \partial J(u)$$

D'après le théorème 2.5.9, on a alors que :

$$u \in \partial J^* \left( \frac{f - u}{\lambda} \right)$$

Soit encore :

$$\begin{aligned} \frac{f}{\lambda} &\in \frac{f - u}{\lambda} + \frac{1}{\lambda} \partial J^* \left( \frac{f - u}{\lambda} \right) \\ \Leftrightarrow \frac{f - u}{\lambda} &= (I + \partial J^*)^{-1} \left( \frac{f}{\lambda} \right) \\ \Leftrightarrow u &= f - \lambda (I + \frac{1}{\lambda} \partial J^*)^{-1} \left( \frac{f}{\lambda} \right) \end{aligned}$$

Finalement, par la proposition 2.5.7 on a que :

$$u = f - P_{\lambda K}(f)$$

avec  $f$  l'image observée et  $P$  l'opérateur de projection sur le convexe fermé  $K$ . Nous allons maintenant chercher à déterminer cet opérateur de projection. Pour cela, nous allons nous placer dans le cas simple d'une image 2D.

### 2.5.3.3 Cas discret

Supposons que  $u$  soit une image 2D de dimension  $N \times N$ . Alors on définit son gradient  $\nabla u$  par :

$$\nabla u_{i,j} = (\nabla u_{i,j}^1, \nabla u_{i,j}^2)$$

avec

$$\begin{aligned} \nabla u_{i,j}^1 &= \begin{cases} u_{i+1,j} - u_{i,j} & \text{si } i < N, \\ 0 & \text{si } i = N \end{cases} \\ \nabla u_{i,j}^2 &= \begin{cases} u_{i,j+1} - u_{i,j} & \text{si } j < N, \\ 0 & \text{si } j = N \end{cases} \end{aligned}$$

En utilisant la formule  $\langle -\operatorname{div} p, u \rangle = \langle p, \nabla u \rangle$ , qui est l'équivalent discret de la formule de Green, on définit la divergence  $\operatorname{div} p$  par :

$$\operatorname{div} p_{i,j} = \begin{cases} p_{i,j}^1 - p_{i-1,j}^1 & \text{si } 1 < i < N, \\ p_{i,j}^1 & \text{si } i = 1, \\ -p_{i-1,j}^1 & \text{si } i = N \end{cases} + \begin{cases} p_{i,j}^2 - p_{i,j-1}^2 & \text{si } 1 < j < N, \\ p_{i,j}^2 & \text{si } j = 1, \\ -p_{i,j-1}^2 & \text{si } j = N \end{cases}$$

La variation totale s'écrit alors :

$$J(u) = \sum_{1 \leq i,j \leq N} |\nabla u_{i,j}| = \sum_{1 \leq i,j \leq N} \sqrt{(\nabla u_{i,j}^1)^2 + (\nabla u_{i,j}^2)^2} \quad (2.20)$$

Par Cauchy-Schwarz, on a

$$p_{i,j}^1 \nabla u_{i,j}^1 + p_{i,j}^2 \nabla u_{i,j}^2 \leq |\nabla u_{i,j}| |p_{i,j}|$$

et si  $|p_{i,j}| = 1$ , on a

$$p_{i,j}^1 \nabla u_{i,j}^1 + p_{i,j}^2 \nabla u_{i,j}^2 \leq |\nabla u_{i,j}|$$

d'où

$$\sum_{i,j} p_{i,j}^1 \nabla u_{i,j}^1 + p_{i,j}^2 \nabla u_{i,j}^2 \leq \sum_{i,j} |\nabla u_{i,j}|$$

et

$$\sup_{\{p \text{ tel que } |p_{i,j}| \leq 1\}} \sum_{i,j} p_{i,j}^1 \nabla_{i,j}^1 + p_{i,j}^2 \nabla_{i,j}^2 \leq \sum_{i,j} |\nabla u_{i,j}|$$

On obtient l'égalité si  $|\nabla u_{i,j}| \neq 0$  en prenant  $p_{i,j} = \frac{\nabla u_{i,j}}{|\nabla u_{i,j}|}$ . Si  $|\nabla u_{i,j}| = 0$ , l'égalité est immédiatement vérifiée. On a donc montré que :

$$J(u) = \sup_{\{p \text{ tel que } |p_{i,j}| \leq 1\}} \langle p, \nabla u \rangle$$

Par ailleurs, d'après la formule de Green, on a que :

$$J(u) = \sup_{\{p \text{ tel que } |p_{i,j}| \leq 1\}} \langle -\operatorname{div} p, u \rangle$$

Finalement d'après le corollaire 2.5.8, on en déduit que le convexe  $K$  est donné par :

$$K = \{\operatorname{div} p, |p_{i,j}| \leq 1 \forall i, j = 1 \dots N\}$$

#### 2.5.3.4 Algorithme de Chambolle

Calculer la projection  $P_{\lambda K}(f)$  revient à résoudre le problème de minimisation :

$$\min\{\|\lambda \operatorname{div} p - f\|^2 \text{ s.c. } |p_{i,j}| \leq 1, \forall i, j = 1 \dots N\} \quad (2.21)$$

Les conditions de Kuhn-Tucker impliquent l'existence de multiplicateurs de Lagrange  $\alpha_{i,j}$  associés au problème (2.21) tels que  $\forall i, j$

$$(\nabla(\lambda \operatorname{div} p - f))_{i,j} + \alpha_{i,j} p_{i,j} = 0$$

avec soit  $\alpha_{i,j} > 0$  et  $|p_{i,j}| = 1$ , soit  $\alpha_{i,j} = 0$  et  $|p_{i,j}| \leq 1$  ; dans les deux cas on a

$$\alpha_{i,j} = |(\nabla(\lambda \operatorname{div} p - f))_{i,j}|$$

En utilisant un algorithme de descente de gradient de pas  $\tau > 0$ , on obtient :

$$p_{i,j}^{k+1} = p_{i,j}^k + \tau \left( (\nabla(\lambda \operatorname{div} p^k - f))_{i,j} - |(\nabla(\lambda \operatorname{div} p^k - f))_{i,j}| p_{i,j}^{k+1} \right)$$

Et de façon explicite on retrouve l'algorithme de Chambolle [Chambolle, 2004] :

$$p_{i,j}^{k+1} = \frac{p_{i,j}^k + \tau (\nabla(\lambda \operatorname{div} p^k - f))_{i,j}}{1 + \tau |(\nabla(\lambda \operatorname{div} p^k - f))_{i,j}|}$$

### 2.5.4 Algorithme explicite-implicite (FB : Forward-Backward)

Ici, nous considérons le cas  $m = 2$  du problème (2.18), dont une fonction est différentiable.

**Problème 2.5.10.** Soient  $f_1 \in \Gamma^0(\mathbb{R}^n)$  et  $f_2 : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction convexe et différentiable de gradient  $L$ -Lipschitz  $\nabla f_2$ , i.e.

$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, \|\nabla f_2(x) - \nabla f_2(y)\| \leq L\|x - y\|, L > 0$$

Supposons que  $f_1 + f_2$  soit coercive, i.e.  $f_1(x) + f_2(x) \rightarrow +\infty$  quand  $\|x\| \rightarrow +\infty$ . Le problème posé consiste à résoudre :

$$\min_{x \in \mathbb{R}^n} f_1(x) + f_2(x) \quad (2.22)$$

On peut montrer que le problème 2.5.10 admet au moins une solution et, pour tout  $\gamma > 0$ , ses solutions sont caractérisées par l'équation du point fixe

$$x = \text{prox}_{\gamma f_1}(x - \gamma \nabla f_2(x)) \quad (2.23)$$

En effet, l'équation d'Euler-Lagrange du problème 2.5.10 est :

$$\begin{aligned} 0 &\in \partial f_1(x) + \partial f_2(x) = \partial f_1(x) + \{\nabla f_2(x)\} \\ &\Leftrightarrow -\nabla f_2(x) \in \partial f_1(x) \\ &\Leftrightarrow (x - \gamma \nabla f_2(x)) - x \in \gamma \partial f_1(x) \\ &\Leftrightarrow x = \text{prox}_{\gamma f_1}(x - \gamma \nabla f_2(x)) \end{aligned}$$

On obtient donc l'algorithme explicite-implicite :

$$x_{k+1} = \underbrace{\text{prox}_{\gamma_k f_1}}_{\text{étape implicite}} \left( \underbrace{x_k - \gamma_k \nabla f_2(x_k)}_{\text{étape explicite}} \right) \quad (2.24)$$

Cet algorithme peut être séparé en une étape explicite de descente de gradient (forward) utilisant la fonction  $f_2$ , et une étape implicite (backward) utilisant la fonction  $f_1$ . Il est caractérisé par un faible coût de calcul par itération mais une convergence lente. Une étape de relaxation est ajoutée à l'algorithme complet (cf. algorithme 2.3), afin d'obtenir un poids variable  $\lambda_k$  pour l'étape implicite et converger plus rapidement.

---

#### Algorithme 2.3 : Algorithme explicite-implicite

---

Soient  $\varepsilon \in ]0, \min(1, L^{-1})[$ ,  $x_0 \in \mathbb{R}^n$

**pour**  $k \geq 0$  **faire**

$\gamma_k \in [\varepsilon, 2L^{-1} - \varepsilon]$  (pas de l'algorithme)

$y_k = x_k - \gamma_k \nabla f_2(x_k)$

$\lambda_k \in [\varepsilon, 1]$  (paramètre de relaxation)

$x_{k+1} = x_k + \lambda_k(\text{prox}_{\gamma_k f_1} y_k - x_k)$

**fin**

---

L'algorithme explicite-implicite est efficace lorsqu'il s'agit de minimiser un critère composé d'un terme de fidélité aux données lisse, par exemple une fonction quadratique,

et un *a priori* non lisse, tel qu'une norme L1. On parle d'algorithme ISTA (Iterative Shrinkages/Thresholding Algorithm) et on cherche alors à minimiser  $F(x) = f_1(x) + f_2(x)$  où  $f_1(x) = \lambda \|x\|_1$  ( $\lambda > 0$ ) et  $f_2(x) = \|Ax - b\|^2$ . Dans cette formule,  $b$  est l'image observée,  $x$  l'image que l'on cherche à estimer et  $A : \mathbb{R}^{n^2} \mapsto \mathbb{R}^{n^2}$  un opérateur de convolution connu modélisant un processus de dégradation linéaire. Soit  $x_0 \in \mathbb{R}^n$ ; l'algorithme ISTA construit la suite  $(x_k)_{k \geq 1}$  en prenant pour tout  $k \in \mathbb{N}$  :

$$x_{k+1} = \text{prox}_{L^{-1}f_1}(x_k - 2L^{-1}A^T(Ax_k - b))$$

Ici, l'opérateur proximal correspond à un seuillage doux (voir exemple 2.5.5) défini par :

$$\text{prox}_\alpha x_i = \text{sign}(x_i) \max\{|x_i| - \alpha, 0\} = \begin{cases} x_i - \alpha & \text{si } x_i > \alpha \\ 0 & \text{si } x_i \in [-\alpha, \alpha] \\ x_i + \alpha & \text{si } x_i < -\alpha \end{cases}$$

Cet algorithme est attractif du fait de sa simplicité ; cependant son taux de convergence est faible. Nous allons décrire trois algorithmes dérivés de ISTA qui permettent d'accroître sa vitesse de convergence :

- TwIST [Bioucas-Dias et Figueiredo, 2007]

L'algorithme TwIST (Two-step Iterative Shrinkage/Thresholding) calcule l'itération  $x_{k+1}$  en fonction des deux précédentes. Cet algorithme est bien adapté lorsque l'opérateur linéaire  $A$  est mal conditionné mais inversible. Dans le cas où  $A$  est non-inversible, une seconde version nommée MTwIST (M pour Monotic) est proposée par les mêmes auteurs.

---

**Algorithme 2.4 : TwIST**

---

Soient  $x_0 \in \mathbb{R}^n$ ,  $x_1 = \text{prox}_{f_1}(x_0 - \nabla f_2(x_0))$ ,  
 $\epsilon_1$  et  $\epsilon_m$  sont deux réels tels que  $0 < \epsilon_1 \leq \lambda_i(A^T A) \leq \epsilon_m$   
 où  $\lambda_i(\cdot)$  est la  $i^{\text{e}}$  valeur propre de l'argument  $(\cdot)$ ,  
 $\kappa = \epsilon_1 / \bar{\epsilon}_m$  avec  $\bar{\epsilon}_m = \max(1, \epsilon_m)$ ,  $\rho = \frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}$ ,  $\alpha = \rho^2 + 1$ ,  $\beta = 2\alpha / (\epsilon_1 + \bar{\epsilon}_m)$

**pour**  $k \geq 1$  **faire**

|  $\gamma_k \in ]0, L^{-1}[$   
 |  $x_{k+1} = (1 - \alpha)x_{k-1} + (\alpha - \beta)x_k + \beta \text{prox}_{\gamma_k f_1}(x_k - \gamma_k \nabla f_2(x_k))$

**fin**

---

- FISTA [Beck et Teboulle, 2009b]

L'algorithme FISTA (Fast Iterative Shrinkage/Thresholding Algorithm) est une version accélérée de ISTA qui s'inspire de l'accélération de Nesterov [Nesterov, 1983]. En effet, lorsque  $f_1 \equiv 0$ , le problème 2.5.10 se réduit à une méthode du gradient et on peut alors utiliser l'accélération de Nesterov pour la descente de gradient  $x_{k+1} = y_k - \gamma_k \nabla f_2(y_k)$ , qui rajoute une combinaison linéaire des itérés  $x_k$  et  $x_{k+1}$  au calcul de l'itéré final  $y_{k+1}$ . Il est démontré que le taux de convergence est alors optimal et en  $O(1/k^2)$ , i.e.  $F(x_k) - F(x^*) \simeq O(1/k^2)$ , contrairement à ISTA pour lequel le taux est en  $O(1/k)$ .

---

**Algorithme 2.5 : FISTA**

---

Soient  $x_0 \in \mathbb{R}^n$ ,  $y_0 = x_0$  et  $t_0 = 1$   
**pour**  $k \geq 0$  **faire**  
     $\gamma_k \in ]0, L^{-1}[$   
     $x_{k+1} = \text{prox}_{\gamma_k f_1}(y_k - \gamma_k \nabla f_2(y_k))$   
     $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$   
     $y_{k+1} = x_{k+1} + \left(\frac{t_k - 1}{t_{k+1}}\right)(x_{k+1} - x_k)$   
**fin**

---

- Algorithme de Nesterov [Nesterov, 1983, Nesterov, 2009]

Contrairement à l'algorithme ISTA, où la direction de descente se base sur le gradient de  $f_2$  à l'itération courante  $x_k$ , cette version accélérée se base sur une direction de descente définie comme un élément de tout le sous-espace vectoriel engendré par l'ensemble des gradients de  $f_2$  aux itérations précédentes, i.e.  $\{\nabla f_2(x_0), \dots, \nabla f_2(x_k)\}$ .

---

**Algorithme 2.6 : Algorithme de Nesterov**

---

Soient  $\varepsilon_0 = 0$ ,  $x_0 \in \mathbb{R}^N$  et  $z_0 = x_0$   
**pour**  $k \geq 1$  **faire**  
     $\kappa_k = \frac{1}{L} + \sqrt{\frac{1}{L^2} + \frac{2}{L}\varepsilon_k}$   
     $y_{k+1} = \frac{\varepsilon_k}{\varepsilon_k + \kappa_k} x_k + \frac{\kappa_k}{\varepsilon_k + \kappa_k} \text{prox}_{\varepsilon_k f_1}(x_k - z_k)$   
     $x_{k+1} = \text{prox}_{f_1/L}(y_k - \frac{1}{L} \nabla f_2(y_k))$   
     $z_{k+1} = z_k + \kappa_k \nabla f_2(x_{k+1})$   
     $\varepsilon_{k+1} = \varepsilon_k + \kappa_k$   
**fin**

---

### 2.5.5 Algorithme de Douglas-Rachford (DR)

Contrairement à l'algorithme Forward-Backward, l'algorithme de Douglas-Rachford ne requiert pas la contrainte de différentiabilité au sens de Lipschitz.

**Définition 2.5.11** (Intérieur relatif d'un convexe). *Soit  $C \in \mathbb{R}^N$  un ensemble convexe non vide. L'intérieur relatif de  $C$  est noté  $\text{ri}C$  et est défini par*

$$\text{ri}C = \{u \in C \mid \exists \rho > 0, \mathcal{B}(u; \rho) \cap \text{aff}C \in C\}$$

où  $\mathcal{B}(u; \rho)$  désigne la boule de centre  $u$  et de rayon  $\rho$  et  $\text{aff}C$  l'enveloppe affine de  $C$ , i.e. l'ensemble des barycentres des points de  $C$ .

**Problème 2.5.12.** *Soient  $f_1$  et  $f_2$  des fonctions de  $\Gamma^0(\mathbb{R}^n)$  tels que*

$$(\text{ri dom } f_1) \cap (\text{ri dom } f_2) \neq \emptyset$$

*Supposons que  $f_1 + f_2$  soit coercive, le problème est :*

$$\min_{x \in \mathbb{R}^N} f_1(x) + f_2(x) \tag{2.25}$$

On peut montrer que le problème 2.5.12 admet au moins une solution et, pour tout  $\gamma > 0$ , ses solutions sont caractérisées par la condition à deux niveaux [Combettes et Pesquet, 2007]

$$\begin{cases} x = \text{prox}_{\gamma f_2}(y) \\ \text{prox}_{\gamma f_2} y = \text{prox}_{\gamma f_1}(2 \text{prox}_{\gamma f_2}(y) - y) \end{cases}$$

Ce qui donne l'algorithme suivant :

---

**Algorithme 2.7 : Douglas-Rachford**

---

Soient  $\varepsilon \in ]0, 1[$ ,  $\gamma > 0$  et  $y_0 \in \mathbb{R}^n$

**pour**  $k \geq 0$  **faire**

$$\begin{array}{|l} x_k = \text{prox}_{\gamma f_2}(y_k) \\ \lambda_k \in [\varepsilon, 2 - \varepsilon] \text{ (paramètre de relaxation)} \\ y_{k+1} = y_k + \lambda_k(\text{prox}_{\gamma f_1}(2x_k - y_k) - x_k) \end{array}$$

**fin**

---

Bien que l'algorithme DR ne nécessite pas que le gradient soit lipschitzien, contrairement à l'algorithme FB, il requiert l'implantation de deux opérateurs proximaux par itération. Dans certains problèmes, ces deux points peuvent être facilement implantés (voir l'exemple suivant et la figure 2.15).

**Exemple 2.5.13.** Soient  $C$  et  $D$  deux ensembles fermés convexes non vides. On considère le problème : trouver un point  $x_\infty$  dans  $C$  à une distance minimale de  $D$ , i.e.

$$\min_{x \in C} \frac{1}{2} d_D^2(x) \quad (2.26)$$

1) Soient  $f_1$  et  $f_2$  les fonctions indicatrices respectivement de  $C$  et  $D$ . Le problème (2.26) est équivalent au problème :

$$\min_{x \in C} f_1(x) + \tilde{f}_2(x) \quad (2.27)$$

où  $\tilde{f}_2$  est l'enveloppe de Moreau de  $f_2$ , i.e.  $\tilde{f}_2(x) = \inf_{y \in \mathbb{R}^N} f_2(y) + \frac{1}{2} \|x - y\|^2$ . Supposons que  $f_1(x) + \tilde{f}_2(x)$  soit coercive ; le problème (2.27) est un cas spécial du problème 2.5.10 et on a :

$$x_{k+1} = \text{prox}_{f_1}(\text{prox}_{f_2}(x_k))$$

De plus, on a  $\text{prox}_{f_1} = P_C$  et  $\text{prox}_{f_2} = P_D$  où  $P_C(x)$  est la projection de  $x \in \mathbb{R}^N$  sur  $C$ . L'algorithme Forward-Backward avec  $\gamma_k \equiv 1$  et  $\lambda_k \equiv 1$  s'écrit :

$$x_{k+1} = P_C(P_D(x_k))$$

2) Soient  $f_1 = \mathbb{1}_C$  et  $f_2 = d_D^2/2$ . Alors le problème (2.26) est équivalent au problème 2.5.12. De plus comme  $\text{prox}_{f_1} = P_C$  et  $\text{prox}_{f_2} = (x + P_D)/2$ , l'algorithme de Douglas-Rachford  $\gamma = 1$  et  $\lambda_k \equiv 1$  s'écrit :

$$\begin{aligned} x_k &= (y_k + P_D(y_k))/2 \\ y_{k+1} &= y_k + P_C(2x_k - y_k) - x_k = P_C(P_D(y_k)) + y_k - x_k \end{aligned}$$

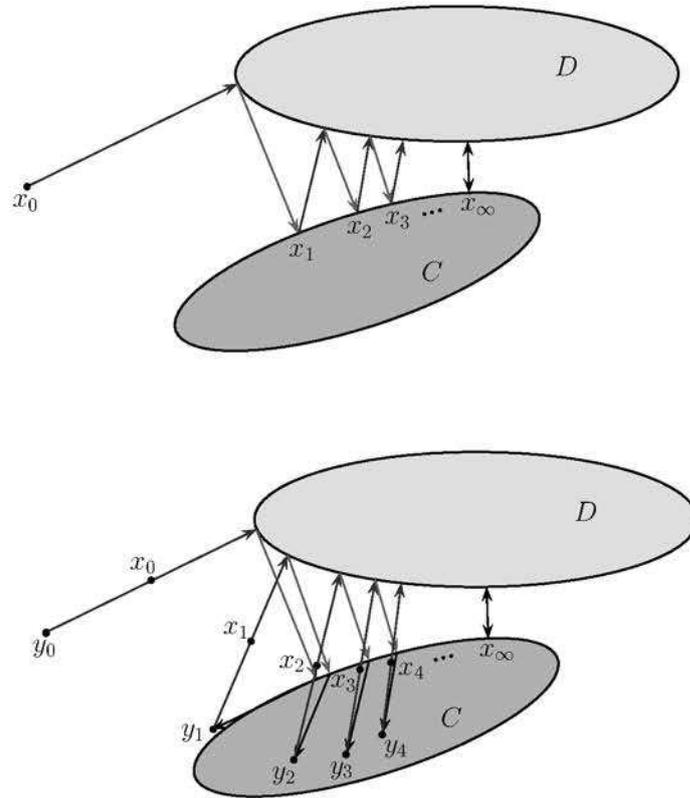


FIGURE 2.15 – Forward-Backward versus Douglas-Rachford pour résoudre le problème (2.26) qui consiste à trouver un point  $x_\infty$  dans  $C$  à une distance minimale de  $D$ . Haut : algorithme FB avec  $\gamma_k \equiv 1$  et  $\lambda_k \equiv 1$ . L'algorithme se réduit à une méthode de projection alternée  $x_{k+1} = P_C(P_D(x_k))$ . Bas : algorithme DR avec  $\gamma = 1$  et  $\lambda_k \equiv 1$ . L'algorithme génère deux suites  $(x_k)$  et  $(y_k)$  telles que  $x_k = (y_k + P_D(y_k))/2$  et  $y_{k+1} = P_C(P_D(y_k)) + y_k - x_k$  [Combettes et Pesquet, 2011]. Dans cet exemple, l'algorithme DR semble plus complexe mais il requiert des conditions moins strictes que FB pour converger.

### 2.5.6 Algorithme Proximal Parallèle (PPXA)

Ici, on considère le problème général (2.18).

**Problème 2.5.14.** Soient  $f_1, \dots, f_m$  des fonctions de  $\Gamma^0(\mathbb{R}^n)$  telles que

$$(\text{ri dom } f_1) \cap \dots \cap (\text{ri dom } f_m) \neq \emptyset$$

Supposons que  $f_1 + \dots + f_m$  soit coercive, le problème est :

$$\min_{x \in \mathbb{R}^n} f_1(x) + \dots + f_m(x) \quad (2.28)$$

Un cas particulier bien connu du problème 2.5.14 est celui où  $m = 2$  ; le problème de minimisation se réduit alors à la minimisation d'une somme de deux fonctions, un terme d'attache aux données  $f_1$  et un terme de régularisation  $f_2$ . La configuration  $m > 2$  permet par exemple :

- d'imposer des contraintes additionnelles sur l'image cible (contrainte sur la dynamique, contrainte de support, ...);
- d'introduire un terme de régularisation hybride (par exemple : L1 + TV).

Afin de réutiliser l'algorithme Douglas-Rachford (cas  $m = 2$ ), nous allons reformuler le problème (2.28) comme un problème à deux fonctions dans l'espace produit

$$\mathcal{H} = \mathbb{R}^n \times \dots \times \mathbb{R}^n$$

En effet, (2.28) peut se réécrire dans  $\mathcal{H}$

$$\min_{\substack{(x_1, \dots, x_m) \in \mathcal{H} \\ x_1 = \dots = x_m}} f_1(x_1) + \dots + f_m(x_m) \quad (2.29)$$

Soit  $x = (x_1, \dots, x_m)$  un élément de  $\mathcal{H}$ , alors (2.29) est équivalent à

$$\min_{x \in \mathcal{H}} \mathbb{1}_D(x) + f(x)$$

où

$$\begin{cases} D = \{(x, \dots, x) \in \mathcal{H} \mid x \in \mathbb{R}^n\} \\ f : x \mapsto f_1(x_1) + \dots + f_m(x_m) \end{cases}$$

On est alors ramené à un problème impliquant deux fonctions dans un espace plus large  $\mathcal{H}$  et l'on peut alors utiliser l'algorithme de Douglas-Rachford pour obtenir l'algorithme 2.8.

### 2.5.7 Algorithme Primal-Dual

Soient  $F$  et  $G$  deux fonctions de  $\Gamma^0(\mathbb{R}^n)$  ; le problème général est défini par :

$$\min_x F(Kx) + G(x) \quad (2.30)$$

où  $K$  est un opérateur linéaire.

**Algorithme 2.8 : PPXA**

Soient  $\varepsilon \in ]0, 1[$ ,  $\gamma > 0$ ,  $(\omega_i)_{1 \leq i \leq m} \in ]0, 1]^m$  tels que

$\sum_{i=1}^m \omega_i = 1$  et  $y_{1,0} \in \mathbb{R}^n, \dots, y_{m,0} \in \mathbb{R}^n$

**pour**  $k \geq 0$  **faire**

**pour**  $i = 1, \dots, m$  **faire**

$p_{i,k} = \text{prox}_{\gamma f_i / \omega_i}(y_{i,k})$

**fin**

$p_k = \sum_{i=1}^m \omega_i p_{i,k}$

$\varepsilon \leq \lambda_k \leq 2 - \varepsilon$

**pour**  $i = 1, \dots, m$  **faire**

$y_{i,k+1} = y_{i,k} + \lambda_k(2p_k - x_k - p_{i,k})$

**fin**

$x_{k+1} = x_k + \lambda_k(p_k - x_k)$

**fin**

**Théorème 2.5.15** (Fenchel-Moreau [Brézis, 1999]). *On suppose que  $F \in \Gamma^0(\mathbb{R}^n)$  ; alors  $F = F^{**}$  avec*

$$F^{**}(p) = \sup_{x \in \mathbb{R}^n} (\langle p, x \rangle - F^*(x))$$

**Remarque :** La démonstration de ce théorème est évidente dans le cas de la transformée de Legendre  $F^*(p) = \langle p, x \rangle - F(x)$  :

$$\begin{aligned} F^{**}(p) &= \langle p, x \rangle - F^*(x) \\ &= \langle p, x \rangle - (\langle x, p \rangle - F(x)) \\ &= F(p) \end{aligned}$$

Elle est moins triviale pour la transformée de Legendre-Fenchel (voir annexe A.2).

En appliquant le théorème de Fenchel-Moreau au problème général (2.30), on obtient :

$$\min_x F^{**}(Kx) + G(x) \quad (2.31)$$

soit le problème de point-selle suivant :

$$\min_x \max_y \langle Kx, y \rangle + G(x) - F^*(y) \quad (2.32)$$

L'idée est alors d'alterner une ascension de gradient sur la variable duale  $y$  et une descente de gradient sur la variable primale  $x$

$$\begin{aligned} y &= \text{prox}_{\sigma F^*}(\tilde{y}) = \arg \min_y \left\{ \frac{\|y - \tilde{y}\|^2}{2\sigma} + F^*(y) \right\} \\ x &= \text{prox}_{\tau G}(\tilde{x}) = \arg \min_x \left\{ \frac{\|x - \tilde{x}\|^2}{2\tau} + G(x) \right\} \end{aligned}$$

où  $\sigma$  et  $\tau$  sont les pas de temps.

Dans [Chambolle et Pock, 2011], il est démontré que l'algorithme primal-dual a un taux de convergence  $O(1/k)$  si  $\sigma$  et  $\tau$  sont choisis tels que  $\tau\sigma L^2 < 1$  avec la constante de Lipschitz  $L = \|K\|$ . Il est aussi montré que cet algorithme peut être accéléré si  $G$  et  $F^*$  sont uniformément convexes (i.e.  $\exists \gamma > 0$  tel que  $\forall x, x' \in \mathbb{R}^n, p \in \partial G(x) : G(x') \geq G(x) + \langle p, x' - x \rangle + \frac{\gamma}{2} \|x - x'\|^2$ ). Dans ce cas, le taux de convergence est  $O(1/k^2)$  en utilisant l'algorithme 2.10.

---

**Algorithme 2.9 :** Algorithme primal-dual général

---

Soient  $\tau, \sigma > 0, \theta \in [0, 1]$   
 $(x_0, y_0) \in \mathbb{R}^n \times \mathbb{R}^n$  et  $\bar{x}_0 = x_0$   
**pour**  $k \geq 0$  **faire**  
     $y_{k+1} = \text{prox}_{\sigma F^*}(y_k + \sigma K \bar{x}_k)$   
     $x_{k+1} = \text{prox}_{\tau G}(x_k - \tau K^T y_{k+1})$   
     $\bar{x}_{k+1} = x_{k+1} + \theta(x_{k+1} - x_k)$   
**fin**

---



---

**Algorithme 2.10 :** Algorithme primal-dual accéléré

---

Soient  $\tau_0, \sigma_0 > 0, \tau_0 \sigma_0 L^2 < 1, \gamma > 0, \theta \in [0, 1]$   
 $(x_0, y_0) \in \mathbb{R}^n \times \mathbb{R}^n$  et  $\bar{x}_0 = x_0$   
**pour**  $k \geq 0$  **faire**  
     $y_{k+1} = \text{prox}_{\sigma F^*}(y_k + \sigma_k K \bar{x}_k)$   
     $x_{k+1} = \text{prox}_{\tau G}(x_k - \tau_k K^T y_{k+1})$   
     $\theta_{k+1} = 1 / \sqrt{1 + 2\gamma \tau_k}$   
     $\tau_{k+1} = \theta_{k+1} \tau_k$   
     $\sigma_{k+1} = \sigma_k / \theta_{k+1}$   
     $\bar{x}_{k+1} = x_{k+1} + \theta_{k+1}(x_{k+1} - x_k)$   
**fin**

---

### 2.5.8 Application au débruitage d'images

La figure 2.16 compare les algorithmes explicite-implicite (FISTA), Douglas-Rachford (DR) et PPXA et Primal-Dual appliqués au modèle ROF. Ici,  $f_1 = \|\nabla x\|$  et  $f_2(x) = \frac{\lambda}{2} \|x - f\|^2$ . L'algorithme explicite-implicite nécessite le gradient de  $f_2(x)$  qui est donné par  $\nabla f_2(x) = \lambda(x - f)$ . Pour les algorithmes Douglas-Rachford et PPXA qui nécessitent l'opérateur proximal  $\text{prox}_{f_2}(x)$ , il est calculé en résolvant l'équation d'Euler-Lagrange et est donné par :

$$\text{prox}_{f_2} x = \frac{x + \lambda \tau f}{1 + \lambda \tau} \quad (2.33)$$

Pour ces trois algorithmes, l'opérateur proximal  $\text{prox}_{f_1}(x)$  est donné par l'algorithme [Chambolle, 2004] (voir 2.5.2). Enfin, pour l'algorithme primal-dual, l'opérateur proximal  $\text{prox}_{f_2}(x)$  est aussi donné par l'équation (2.33), tandis que  $\text{prox}_{f_1^*}(x)$  est calculé par dualité et se réduit à une projection sur la boule unité :

$$\text{prox}_{f_1^*} x = \frac{x}{\max(1, |x|)} \quad (2.34)$$

Ces algorithmes ont été implémenté en Matlab, en s'aidant de la boîte à outils de Gabriel Peyré "Toolbox Sparse Optimization"<sup>1</sup>. Les temps de calcul sont donnés dans le tableau 2.3 et les erreurs par rapport à la solution optimale (à  $k = 1000$ ) sont représentées sur la figure 2.17.

On peut constater que les résultats pour FISTA, Douglas-Rachford et PPXA sont similaires, alors que le résultat de Primal-Dual semble un peu plus lissé. L'avantage de l'algorithme Primal-Dual est qu'il est beaucoup plus rapide, cependant le paramètre  $\lambda$

---

1. [www.mathworks.com/matlabcentral/fileexchange/16204-toolbox-sparse-optimization](http://www.mathworks.com/matlabcentral/fileexchange/16204-toolbox-sparse-optimization)

est plus sensible et a tendance à rendre floue l'image, plus il diminue. En ce qui concerne la convergence, l'algorithme de FISTA est le plus efficace puisqu'il suffit de 5 itérations pour avoir  $F(x_k) - F(x^*) < 10^{-2}$ , contre 27 pour Douglas-Rachford, 18 pour PPXA et 1415 pour Primal-Dual. Le tableau 2.4 récapitule les avantages et les inconvénients de chaque algorithme.

Dans le chapitre suivant, nous allons voir des méthodes de filtrage et segmentation spécifiques aux images angiographiques.

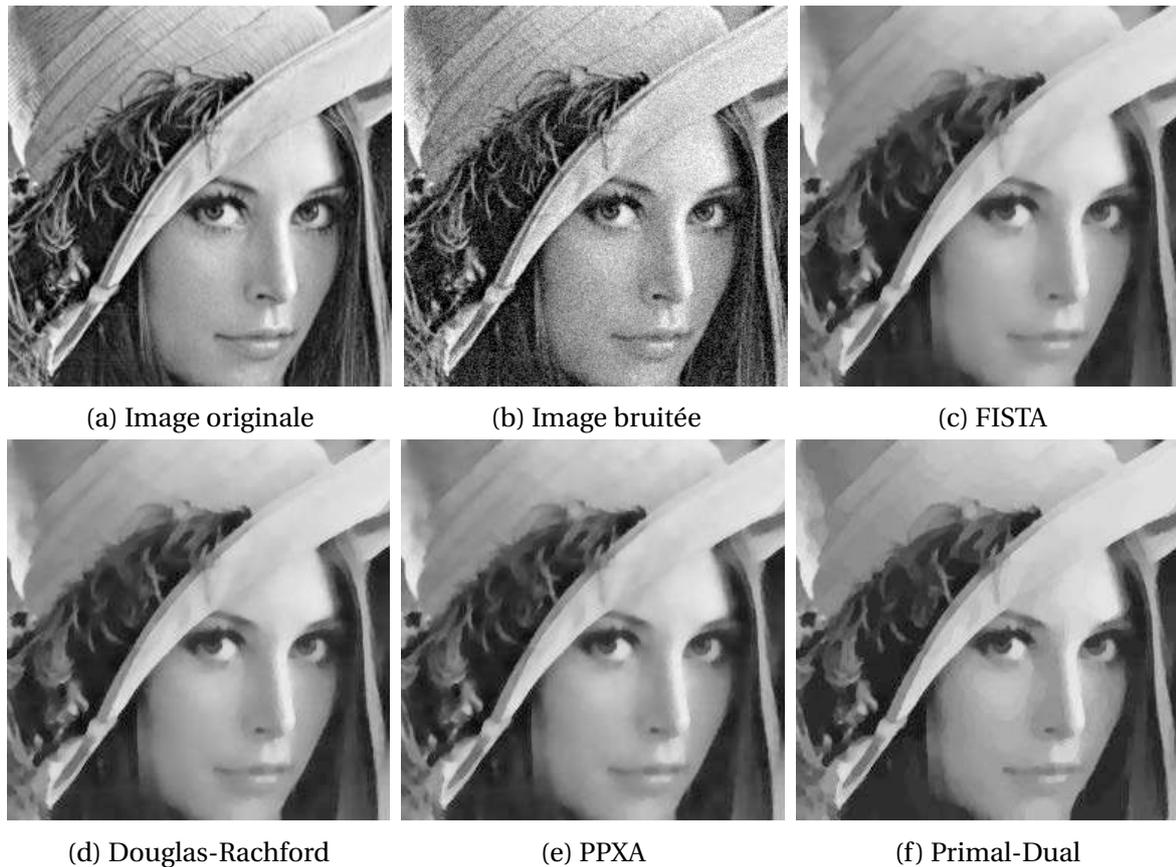


FIGURE 2.16 – Comparaison des différentes classes d'algorithmes pour débruiter l'image test de Lena (les résultats sont donnés à  $k = 1000$ ). Ici, le pas de l'algorithme est  $\gamma = L^{-1}$  pour FISTA, DR et PPXA, le paramètre de relaxation pour DR et pour PPXA est  $\lambda = 1$ .

Algorithme	FISTA	DR	PPXA	Primal-dual
CPU	132.28 s	132.98 s	134.50 s	12.59 s

TABEAU 2.3 – Temps de calcul pour débruiter l'image test de Lena ( $k = 1000$ ).

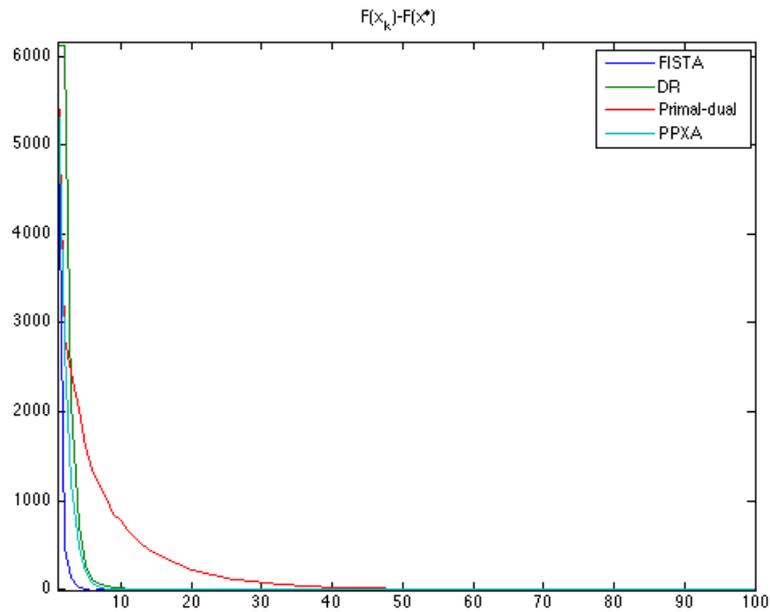


FIGURE 2.17 – Comparaison des erreurs  $F(x_k) - F(x^*)$  où  $F(x^*)$  est la solution à  $k = 1000$ .

Algorithme	Problème	Avantages et inconvénients
FB	$\min_x f_1(x) + f_2(x)$ $f_1, f_2 \in \Gamma^0(\mathbb{R}^n)$ $f_2 \text{ différentiable}$	+ Efficace pour le débruitage d'images $f_1(x) =  \cdot $ et $f_2(x) = \ x - f\ ^2$ + Convergence optimale pour FISTA – Contrainte de différentiabilité
DR	$\min_x f_1(x) + f_2(x)$ $f_1, f_2 \in \Gamma^0(\mathbb{R}^n)$ $(\text{ri dom } f_1) \cap (\text{ri dom } f_2) \neq \emptyset$	+ Aucune contrainte de différentiabilité – Calcul de deux opérateurs proximaux
PPXA	$\min_x f_1(x) + \dots + f_m(x)$ $f_1, \dots, f_m \in \Gamma^0(\mathbb{R}^n)$ $(\text{ri dom } f_1) \cap (\text{ri dom } f_2)$ $\dots \cap (\text{ri dom } f_m) \neq \emptyset$	+ Aucune contrainte de différentiabilité + Termes hybrides autorisés – Calcul de $m$ opérateurs proximaux
Primal-Dual	$\min_x f_1(Kx) + f_2(x)$ $f_1, f_2 \in \Gamma^0(\mathbb{R}^n)$	+ Efficace pour le débruitage d'images + Aucune contrainte de différentiabilité + Calcul rapide et simple de $\text{prox}_{f_1}(x)$ – Sensible au paramètre $\lambda$

TABEAU 2.4 – Avantages et inconvénients des différents algorithmes proximaux.



# Chapitre 3

## Filtrage et segmentation des vaisseaux

*« L'amour est le plus grand des vaisseaux du corps ... »*

Jacques Audibert

Dans ce chapitre, un bref tour d'horizon des méthodes de filtrage et de segmentation des vaisseaux est décrit. L'accent est mis plus particulièrement sur les méthodes de filtrage basées sur les dérivées. L'analyse des valeurs propres de la matrice hessienne permet notamment de capturer des informations relatives à la forme des objets dans l'image, tels que des tubes, des plans ou des structures isotropes. Afin de réhausser les vaisseaux dans l'image, qui correspondent aux structures tubulaires, plusieurs mesures de tubularité ou "vesselness" ont été proposées. Dans le chapitre 4, la vesselness de Frangi sera incluse à un cadre variationnel afin de segmenter les vaisseaux.

### Sommaire

---

<b>3.1</b>	<b>Introduction</b>	<b>59</b>
<b>3.2</b>	<b>Tour d'horizon des méthodes de filtrage et segmentation des vaisseaux</b>	<b>61</b>
3.2.1	Croissance de régions	61
3.2.2	Segmentation basée sur des modèles	62
3.2.2.1	Modèles d'intensité	62
3.2.2.2	Modèles géométriques	62
3.2.2.3	Modèles déformables	63
3.2.3	Recherche de chemins	63
3.2.4	Le suivi de vaisseaux	63
3.2.5	Morphologie mathématique	64
3.2.6	Approches statistiques	64
3.2.7	Filtrage bilatéral	64
3.2.8	Filtrage non-local	65
<b>3.3</b>	<b>Filtrage basé sur les dérivées</b>	<b>66</b>
3.3.1	L'opérateur gradient	66
3.3.2	L'opérateur hessien	67
3.3.3	Mesures de tubularité (vesselness)	70
3.3.3.1	Vesselness de Lorenz	70
3.3.3.2	Vesselness de Sato	70
3.3.3.3	Vesselness de Frangi	70

3.3.3.4	Cadre multi-échelle . . . . .	71
3.3.4	Tenseur de structures . . . . .	71

---

### 3.1 Introduction

On peut définir un objet fin comme un objet étant plus petit dans au moins une de ses dimensions, par rapport aux autres dimensions. En 2D, les objets fins correspondent à des lignes, tandis qu'en 3D, ils correspondent à des tubes ou à des plans. Dans la suite de ce chapitre, nous nous intéresserons particulièrement aux objets linéiques ou tubulaires. Nous parlerons d'objets tubulaires indépendamment de la dimension de l'image. En imagerie biomédicale, de telles structures sont, par exemple, des vaisseaux sanguins (cf. figure 3.1), des neurones (cf. figure 3.2) ou des fibres (cf. figure 3.3).

Malgré les progrès de l'imagerie médicale en termes de résolution et de précision, il est généralement difficile d'extraire les structures tubulaires. En particulier, les principales difficultés sont les suivantes :

- ces structures peuvent être très fines (seulement quelques pixels de large) ;
- elles peuvent être géométriquement complexes (fines, longues, curvilinéaires, ...) ;
- ces deux dernières propriétés impliquent que les structures tubulaires sont très sensibles au bruit ;
- le bruit et d'autres artefacts provoquent parfois des déconnexions le long de ces structures ;
- ces structures sont souvent multiéchelles et peuvent varier en diamètre d'un facteur de 1000 ou plus entre la plus grosse artère (l'aorte : 30 *mm*) et la plus petite artériole (30  $\mu m$ ). De ce fait, dans la plupart des modalités d'imagerie, les structures vasculaires s'étendent au delà de la limite de résolution.

C'est pourquoi les méthodes classiques de filtrage et de segmentation ne sont pas adaptées à ces structures, et que ces dernières ont suscité une attention particulière au cours des quinze dernières années. La détection des structures tubulaires présente alors un double défi : (i) le débruitage / le filtrage et (ii) le réhaussement / la segmentation. Dans la section 3.2, un bref tour d'horizon des méthodes de filtrage et de segmentation spécifiques aux vaisseaux est proposé. Enfin, dans la section 3.3, nous nous focalisons sur les méthodes de filtrage basées sur les dérivées, et notamment les mesures de tubularité qui seront utilisées dans le chapitre 4.

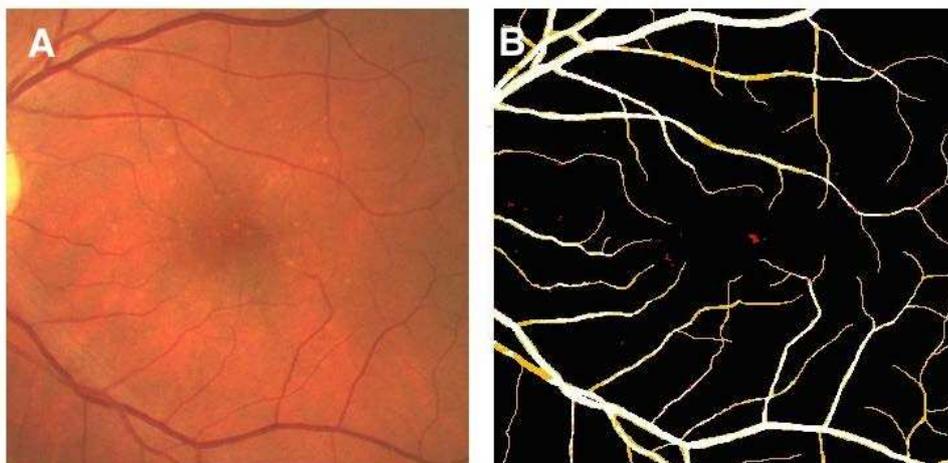


FIGURE 3.1 – Segmentation des vaisseaux sanguins de la rétine. A : image d'un fond d'œil par rétinographie. B : segmentation des vaisseaux sanguins [Migliorati, 2014].

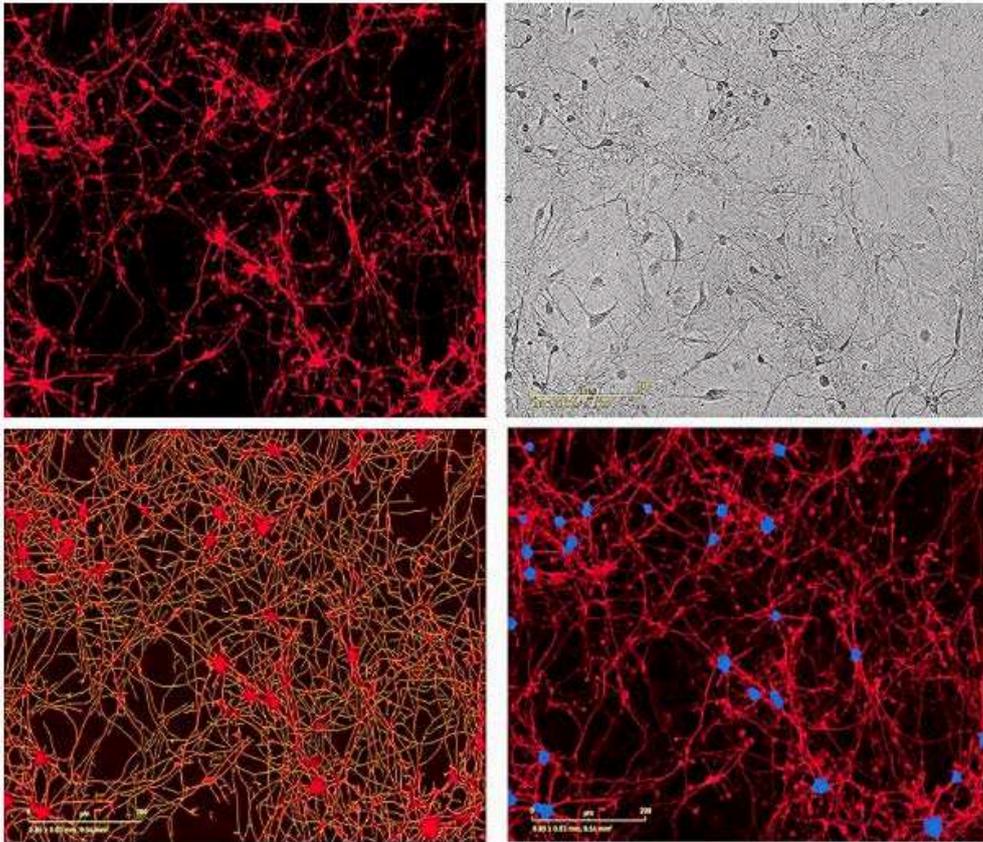


FIGURE 3.2 – Segmentation des neurites du cerveau antérieur d’un rat. A : image par microscopie fluorescente. B : image par contraste de phase. C : segmentation des neurites (jaune). D : segmentation des corps cellulaires (bleu) [Anonyme, 2015].

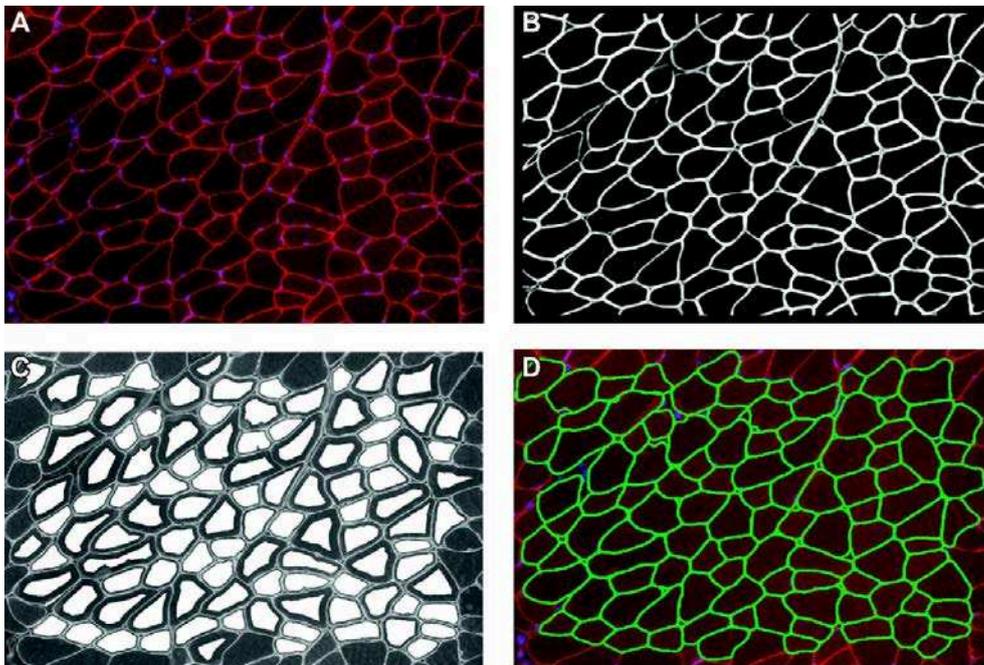


FIGURE 3.3 – Segmentation des fibres du muscle plantaire chez l’humain. A : image par microscopie fluorescente. B : détection des arêtes. C : détection des graines des fibres (blanc). D : segmentation des fibres musculaires (vert) [Mula *et al.*, 2013].

## 3.2 Tour d’horizon des méthodes de filtrage et segmentation des vaisseaux

Le filtrage consiste à réduire le niveau de bruit dans une image tout en préservant les structures d’intérêt. La segmentation consiste à détecter ces structures par rapport au reste de l’image, par exemple avec un résultat présenté sous forme d’une image binaire. Plusieurs états de l’art consacrés à ces méthodes ont été proposés ces quinze dernières années. Les plus anciens [Suri *et al.*, 2002a, Suri *et al.*, 2002b, Kirbas et Quek, 2004] présentent certains fondamentaux, mais sont aujourd’hui incomplets ; il est alors pertinent de se référer aux plus récents [Lesage *et al.*, 2009, Tankyevych, 2010] pour une vue d’ensemble plus exhaustive. La première étude [Suri *et al.*, 2002a, Suri *et al.*, 2002b] se focalise sur la segmentation des vaisseaux dans les images obtenues par ARM (Angiographie par Résonance Magnétique). Elle est divisée en deux parties : une première dédiée au préfiltrage [Suri *et al.*, 2002a], et une seconde qui compare les méthodes d’extraction de squelettes et celles qui détectent la totalité du volume vasculaire [Suri *et al.*, 2002b]. La deuxième étude [Kirbas et Quek, 2004] concerne plus généralement la segmentation des vaisseaux à partir de tous types d’images, indépendamment de leur dimension ou de leur acquisition. La troisième étude [Lesage *et al.*, 2009] s’applique aux images obtenues par ARM et ATC (Angiographie par Tomographie Calculée) et répertorie les méthodes selon trois axes : (i) les modèles vasculaires, i.e. les informations sur les vaisseaux visés, (ii) les caractéristiques spécifiques aux vaisseaux, i.e. les mesures utilisées pour détecter les vaisseaux, et (iii) les schémas d’extraction, i.e. les algorithmes utilisés pour la segmentation des vaisseaux. Enfin, la dernière étude [Tankyevych, 2010] s’applique également aux images provenant d’ARM et d’ATC, mais classe les méthodes en huit familles que nous allons décrire brièvement : la croissance de régions [Zucker, 1976, Adams et Bischof, 1994], la segmentation basée sur les modèles [Wörz et Rohr, 2004, Tyrrell *et al.*, 2007], les modèles déformables [Singh *et al.*, 1998, Lorigo *et al.*, 2001], la recherche de chemin [Wink *et al.*, 2002, Olabariaga *et al.*, 2003], le suivi de vaisseaux [Wink *et al.*, 2000, Manniesing *et al.*, 2007], la morphologie mathématique [Serra, 1982, Serra, 1988, Najman et Talbot, 2010], les approches statistiques [Hassouna *et al.*, 2006, Wink *et al.*, 2007], et enfin le filtrage basé sur les dérivées, qui sera décrit plus en détail dans la section 3.2. Les six premières familles sont des méthodes de segmentation, les approches statistiques peuvent être utilisées à la fois pour la segmentation et le filtrage, et la dernière famille est typiquement une méthode de filtrage. Nous allons compléter cette classification, en décrivant deux méthodes classiques de filtrage : le filtrage bilatéral [Tomasi et Manduchi, 1998] et le filtrage non-local [Buades *et al.*, 2005].

### 3.2.1 Croissance de régions

La croissance de régions dépend de deux éléments : une (ou plusieurs) graine(s) supposée(s) appartenir à la structure à segmenter, et un critère de croissance permettant de segmenter l’objet à partir de la graine, en incluant à chaque itération les voxels voisins vérifiant le critère (cf. figure 3.4). Dans le cas de la segmentation de vaisseaux, les graines peuvent être détectées automatiquement, en particulier dans le cas où elles constituent la racine d’un arbre vasculaire.

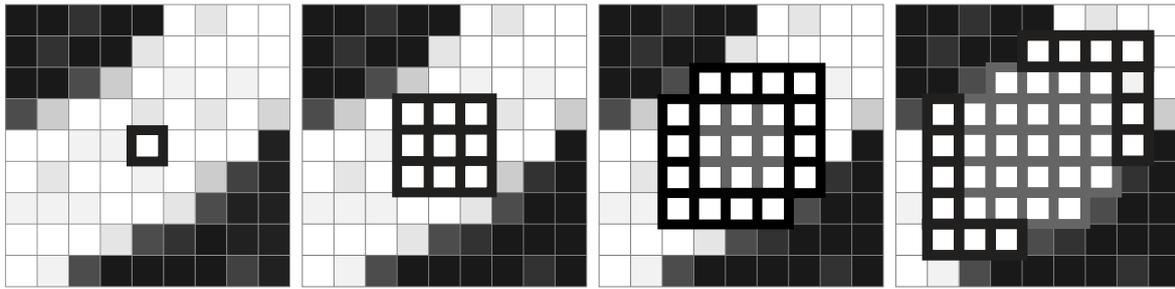


FIGURE 3.4 – Croissance de régions pour la segmentation vasculaire. De gauche à droite : partant d'un voxel "graine", l'algorithme de croissance de régions ajoute successivement des voxels voisins vérifiant le critère de croissance [Lesage *et al.*, 2009].

### 3.2.2 Segmentation basée sur des modèles

Les caractéristiques spécifiques des vaisseaux, telles que la photométrie et/ou la géométrie, peuvent être utilisées pour la segmentation. Les plus simples sont les modèles d'intensité et ceux basés sur la géométrie qui sont souvent combinés en modèles déformables.

#### 3.2.2.1 Modèles d'intensité

Les modèles d'intensité, qui sont les plus simples, dépendent fortement de la nature de l'image considérée. Ils intègrent l'intensité, le contraste et le gradient en priorité, mais aussi des propriétés d'image, comme la gamme d'intensité ou la variation d'intensité basée sur la localisation ou encore la distribution d'intensité. L'approche variationnelle de ces modèles a été établie dans le chapitre 1 (cf. section 1.3.2).

#### 3.2.2.2 Modèles géométriques

L'hypothèse que les vaisseaux sont des objets fins allongés, globalement similaires à des tubes a été utilisée pour créer plusieurs modèles géométriques, tels que des cylindres généralisés [Toumoulin *et al.*, 2001], des superellipsoïdes [Tyrrell *et al.*, 2007] (voir figure 3.5), des lignes gaussiennes [Krissian *et al.*, 2000], ou des profils en forme de barre [Boldak *et al.*, 2003]. Bien que ces modèles soient de puissants outils pour décrire et extraire les vaisseaux, ils ne sont pas adaptés à une image bruitée ou à des cas pathologiques. De plus, ils nécessitent souvent un réglage de paramètres minutieux.

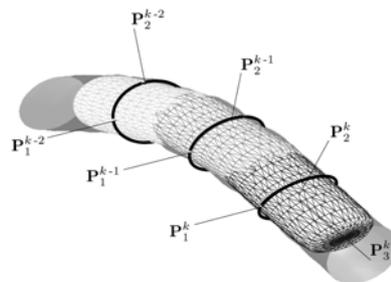


FIGURE 3.5 – Traversée d'un vaisseau par une séquence de superellipsoïdes notées  $k-2$ ,  $k-1$  et  $k$  [Tyrrell *et al.*, 2007].

### 3.2.2.3 Modèles déformables

Les modèles déformables ou de contours actifs ont pour but l'ajustement d'une hypersurface (par exemple, une surface dans une image 3D), en la déplaçant et en modifiant sa forme à partir du modèle initial, tout cela guidé par plusieurs forces : celles *internes*, qui dépendent du contenu de l'image et celles *externes*, qui servent à préserver correctement les propriétés géométriques. Ces modèles ont déjà été étudiés plus précisément dans le chapitre 1 (cf. section 1.3.1)

### 3.2.3 Recherche de chemins

Basés sur des critères d'intensité extrême et de connectivité, la détection de segments de vaisseaux peut être exprimée comme la détermination de chemin de coût minimal dans des graphes pondérés [Dijkstra, 1959], modélisant les voxels, les relations de voisinage et leur intensité. Les méthodes de recherche de chemins sont souvent bien adaptées pour la détection des axes médians des vaisseaux, spécialement dans le cas de petits vaisseaux. Malgré des essais pour segmenter des arbres vasculaires complets, ces méthodes restent dédiées à la segmentation de segments de vaisseaux et nécessitent de fournir interactivement un point initial et un point final. Dans ce cas d'utilisation, ils peuvent être robustes au bruit et à la baisse du signal dans le vaisseau, particulièrement dans le cas des sténoses.

### 3.2.4 Le suivi de vaisseaux

L'idée principale de cette méthode est le suivi de l'objet d'intérêt le long de sa direction principale. Il nécessite une graine, c'est-à-dire le point de départ du procédé, et une direction donnée. C'est un procédé itératif qui donne à chaque étape un petit segment du vaisseau (voir figure 3.6). Malgré quelques tentatives pour gérer les cas des bifurcations, qui peuvent permettre le traitement récursif d'un arbre vasculaire, le suivi de vaisseaux est spécialement adapté à la segmentation de vaisseaux uniques. Il faut noter que, similairement aux autres approches locales (qui ont pour but de détecter une partie des vaisseaux, et/ou qui sont guidées par une graine donnée), de telles méthodes présentent généralement un coût algorithmique faible.

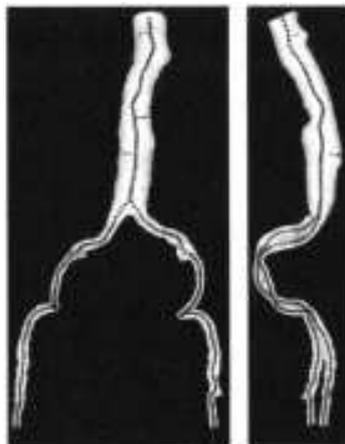


FIGURE 3.6 – Rendu volumique et axe médian d'une aorte ascendante et de branches iliaques provenant de l'ACT, suite à la méthode de suivi de vaisseaux [Wink *et al.*, 2000].

### 3.2.5 Morphologie mathématique

La morphologie mathématique est une théorie du traitement et de l'analyse d'images, établie en 1964 par Georges Matheron et Jean Serra [Serra, 1982, Serra, 1988]. Fondamentalement, la morphologie mathématique repose sur des fondements algébriques, et la donnée d'un treillis  $(E, <)$ . Sur cette base, des opérations élémentaires d'érosion et de dilatation sont définies, qui commutent respectivement avec les supremum et infimum sur ce treillis. D'un point de vue géométrique, ces opérateurs de base sont généralement définis à partir d'éléments structurants, qui peuvent être assimilés à de petits objets définissant le support spatial de noyaux de convolution. De manière incrémentale, des opérateurs de plus haut niveau peuvent être définis à partir de ces opérateurs fondamentaux : ouvertures et fermetures, filtres alternés séquentiels, transformées en tout-ou-rien [Naegel *et al.*, 2007], etc. Ils sont intrinsèquement conçus pour traiter des images binaires, ou à niveaux de gris ; des extensions aux images multivaluées ont néanmoins également été proposées.

Une seconde branche de la morphologie mathématique repose sur la notion de connectivité, établissant des liens forts avec la topologie discrète et la théorie de graphes. Dans ce contexte, de nombreux travaux ont été menés sur les opérateurs connexes [Salember *et Wilkinson*, 2009], qui présentent généralement la spécificité de manipuler les images sans en altérer les contours. Dans cette famille d'approches, on retrouve notamment des opérateurs classiques, tels que la transformée par ligne de partage des eaux (watersheds) qui est une méthode de segmentation, ainsi que de nombreuses méthodes de filtrage anti-extensif et de segmentation, basés sur des modèles hiérarchiques d'images (arbres des coupes, arbres des formes, etc.) [Dufour *et al.*, 2013, Perret *et al.*, 2015]. Les méthodes de manipulation discrète d'images à base topologique (croissance de région, amincissements, squelettisation, ainsi que les transformations homotopiques) peuvent également être assimilées à cette branche de la morphologie mathématique.

### 3.2.6 Approches statistiques

La segmentation des vaisseaux basée sur des approches statistiques dépend des hypothèses spécifiques liées à la distribution de l'intensité des signaux vasculaires ou non dans les données d'ARM. Si le nombre et la nature de ces distributions sont connus, il est alors possible d'estimer leurs paramètres à l'aide de l'algorithme standard EM, *espérance-maximisation* [Dempster *et al.*, 1977]. Les méthodes statistiques héritent des forces et des faiblesses de l'algorithme EM. Premièrement, elles nécessitent généralement d'établir des hypothèses sur la distribution du signal. De plus, l'initialisation de celles-ci demande une attention particulière. En effet, elles nécessitent le réglage de plusieurs paramètres tels que le poids, la moyenne et la variance de la distribution.

### 3.2.7 Filtrage bilatéral

De la même façon que la convolution par la fonction gaussienne, le filtre bilatéral est défini comme une moyenne pondérée des pixels dans une fenêtre carré  $\Omega_x$ , centrée sur le pixel cible  $\mathbf{x}$ . La différence est que le filtre bilatéral prend en compte la variation d'intensité pour préserver les bords. Les poids utilisés dans le calcul de la moyenne dépendent non seulement de la distance euclidienne entre le pixel central  $\mathbf{x}$  et le pixel de la fenêtre  $\mathbf{y}$ , mais aussi des similarités spectrales entre les pixels. A partir de l'image originale  $I$ , le résultat du filtre bilatéral est donné par :

$$FB(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \sum_{\mathbf{y} \in \Omega_x} G(\mathbf{x} - \mathbf{y}, \sigma_s) G(I(\mathbf{x}) - I(\mathbf{y}), \sigma_r) I(\mathbf{y}) \quad (3.1)$$

où  $G(\mathbf{x}, \sigma)$  est la fonction gaussienne centrée de variance  $\sigma^2$  définie en dimension  $D$  par :

$$G(\mathbf{x}, \sigma) = \frac{1}{(2\pi\sigma^2)^{D/2}} e^{-\frac{|\mathbf{x}|^2}{2\sigma^2}} \quad (3.2)$$

et  $C(\mathbf{x})$  un coefficient de normalisation défini par :

$$C(\mathbf{x}) = \sum_{\mathbf{y} \in \Omega_x} G(\mathbf{x} - \mathbf{y}, \sigma_s) G(I(\mathbf{x}) - I(\mathbf{y}), \sigma_r)$$

### 3.2.8 Filtrage non-local

Le filtrage non-local a été introduit avec le filtre non-local means [Buades *et al.*, 2005]. Contrairement aux filtres locaux qui réalisent une moyenne des intensités des pixels localisés autour du pixel cible, le filtre non-local means effectue une moyenne des pixels situés dans un même contexte. Le contexte d'un pixel est caractérisé par une petite fenêtre centrée nommée *patch* et seule la similarité des patches est prise en compte (voir figure 3.7). Le résultat du filtre non-local means est donné par :

$$NL(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \sum_{\mathbf{y} \in \Omega_x} w(\mathbf{x} - \mathbf{y}) I(\mathbf{y}) \quad (3.3)$$

où le poids s'écrit :

$$w(\mathbf{x} - \mathbf{y}) = \sum_{\mathbf{z} \in \Omega_z} G(I(\mathbf{x} + \mathbf{z}) - I(\mathbf{y} + \mathbf{z}), \sigma)$$

et  $C(x)$  est un coefficient de normalisation défini par :

$$C(\mathbf{x}) = \sum_{\mathbf{y} \in \Omega_x} w(\mathbf{x} - \mathbf{y})$$

Dans [Buades *et al.*, 2005],  $\Omega_x = \Omega$ ; le choix de se restreindre à un voisinage  $\Omega_x$  est conseillé afin de limiter le temps de calcul, ce qui est toujours le cas dans la pratique.

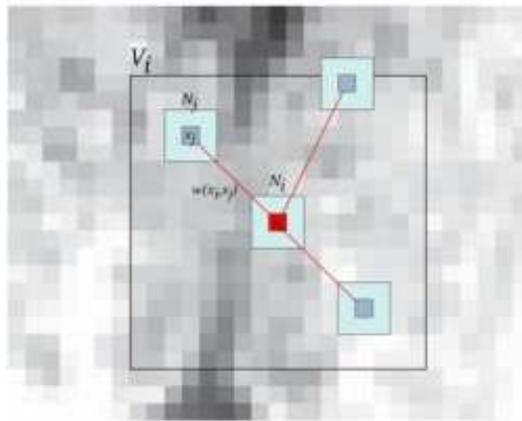


FIGURE 3.7 – Illustration du débruitage par moyennes non-locales : le pixel débruité  $x_i$  (en rouge) est la moyenne pondérée de toutes les intensités des pixels  $x_j$  dans la fenêtre  $V_i$ , basée sur la similarité entre les patches  $N_i$  et  $N_j$  [Coupé *et al.*, 2008].

Toutes les techniques récentes en débruitage d'images reposent sur les concepts de non-localité et de patches. Parmi eux, on peut citer l'algorithme K-SVD [Aharon *et al.*, 2006] qui résout le problème d'apprentissage de dictionnaire parcimonieux, et l'algorithme BM3D [Dabov *et al.*, 2007], basé sur le fait qu'une image a une représentation localement parcimonieuse dans le domaine de transformée.

### 3.3 Filtrage basé sur les dérivées

Les vaisseaux représentent généralement les zones de plus forte intensité dans l'image. Si l'on considère une image 3D en niveaux de gris comme la représentation discrète d'une fonction de  $\mathbb{R}^3$  dans  $\mathbb{R}$  (ou une image 2D comme la représentation discrète d'une fonction de  $\mathbb{R}^2$  dans  $\mathbb{R}$ ), les structures vasculaires apparaissent comme les valeurs localement maximales de cette fonction. La segmentation de telles structures peut alors être réalisée en étudiant les propriétés différentielles de l'image.

Il existe deux grandes classes de filtrage basées sur les dérivées : l'*opérateur hessien* et le *tenseur de structures*. Ces opérateurs sont tous les deux plus performants que l'opérateur gradient qui donne une information uniquement sur l'orientation des contours des objets. En effet, l'opérateur hessien et le tenseur de structures peuvent capturer des caractéristiques relatives à la forme des objets, tels que des tubes, des plans, l'absence de structure (structure isotrope ou "blob") ou du bruit. Une approche possible pour détecter les structures tubulaires est alors d'utiliser les mesures de tubularité, qui peuvent être perçues comme la probabilité pour un pixel d'appartenir à un vaisseau, i.e. un tube.

#### 3.3.1 L'opérateur gradient

L'opérateur gradient est le vecteur des dérivées premières. Il est donné en 2D par :

$$\nabla(I) = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix}$$

Et en 3D, par :

$$\nabla(I) = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial z} \end{bmatrix}$$

où  $I : \Omega \subset \mathbb{R}^2$  (ou  $\mathbb{R}^3$ )  $\mapsto \mathbb{R}$  est une fonction  $C^1$  qui est une approximation d'une fonction à variation bornée [Chambolle *et al.*, 2010].

Le gradient est caractérisé par son module et sa direction. Le module du gradient représente la pente de la surface image au point de calcul. La direction du vecteur gradient donne la direction de cette pente. Un module élevé traduit une forte variation des niveaux de gris autour de ce point. On peut donc déduire que l'on se situe dans une zone de transition entre une partie claire et une partie sombre, c'est-à-dire sur un contour. Le gradient est alors perpendiculaire au contour, comme sur la figure 3.8.

En pratique, le gradient peut être obtenu à partir de filtres de type Sobel ou Prewitt, qui calculent des différences entre voisins dans une direction donnée, après avoir effectué un lissage. Par exemple en 2D, l'image est convoluée avec les masques suivants (l'opérateur de Prewitt est défini pour  $k = 1$  et celui de Sobel pour  $k = 2$ ) :

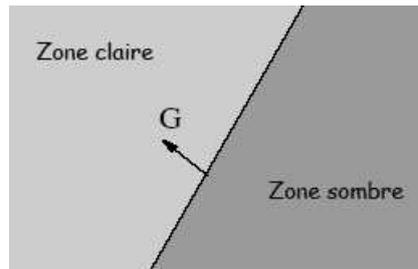


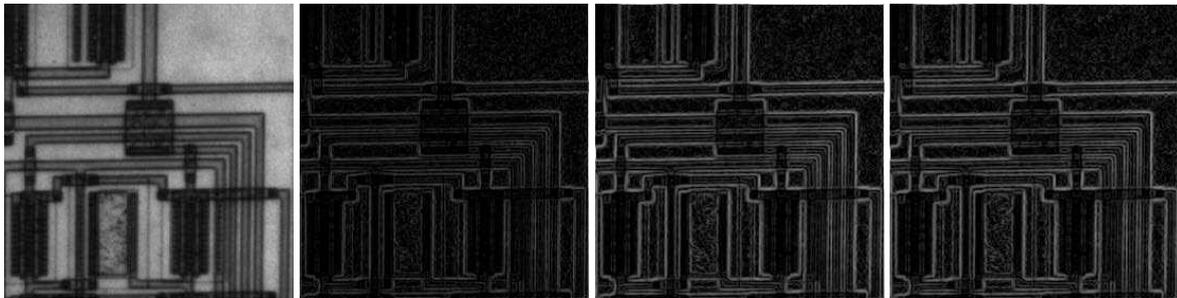
FIGURE 3.8 – Gradient d'un contour. Il représente une zone de transition entre une partie claire et une partie sombre.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -k & 0 & k \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & -k & -1 \\ 0 & 0 & 0 \\ 1 & k & 1 \end{bmatrix}$$

Ces opérateurs combinent un léger filtrage passe-bas le long des lignes et des colonnes (pondération de poids  $[1 \ k \ 1]$ ) avec un filtrage de type Roberts, qui est l'opérateur de gradient classique constitué des deux masques :

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Le filtre de Roberts produit des contours plus fins et mieux localisés que les autres opérateurs, car il constitue le filtre local différentiel de taille minimale (voir figure 3.9). Néanmoins, les filtres de Prewitt et Sobel sont plus résistants au bruit grâce au filtre passe-bas.



(a) Image originale      (b) Filtre de Roberts      (c) Filtre de Prewitt      (d) Filtre de Sobel

FIGURE 3.9 – Comparaison des différents filtres de détection de contours (Roberts, Prewitt et Sobel) sur l'image test d'un circuit électrique.

### 3.3.2 L'opérateur hessien

L'opérateur hessien est la matrice des dérivées secondes. En 2D, la matrice hessienne d'une image  $I$  peut être calculée en tout pixel comme :

$$\mathcal{H}(I) = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix}$$

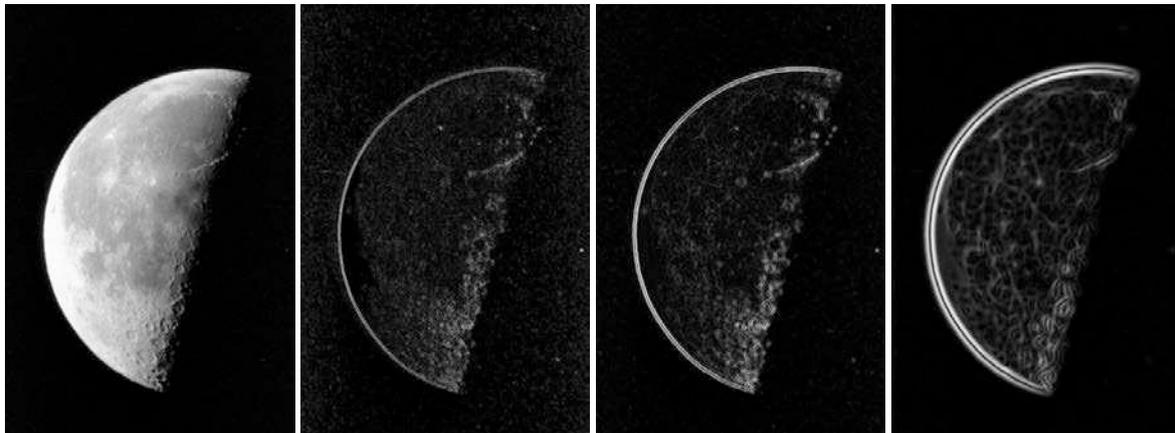
En 3D, la matrice hessienne est :

$$\mathcal{H}(I) = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial x \partial z} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y^2} & \frac{\partial^2 I}{\partial y \partial z} \\ \frac{\partial^2 I}{\partial z \partial x} & \frac{\partial^2 I}{\partial z \partial y} & \frac{\partial^2 I}{\partial z^2} \end{bmatrix}$$

Afin de détecter les vaisseaux de différentes tailles, il est important d'appliquer ces opérateurs dans un cadre multi-échelle. Pour cela, on utilise la théorie linéaire de l'espace-échelle ("scale-space") qui redéfinit les dérivées premières et secondes comme le produit de convolution avec les dérivées de la fonction gaussienne [Koenderink, 1984] :

$$\begin{aligned} \frac{\partial}{\partial x} I(\mathbf{x}, \sigma) &= \sigma^\gamma I(\mathbf{x}) * \frac{\partial}{\partial x} G(\mathbf{x}, \sigma) \\ \frac{\partial^2}{\partial x^2} I(\mathbf{x}, \sigma) &= \sigma^\gamma I(\mathbf{x}) * \frac{\partial^2}{\partial x^2} G(\mathbf{x}, \sigma) \\ \frac{\partial^2}{\partial x \partial y} I(\mathbf{x}, \sigma) &= \sigma^\gamma I(\mathbf{x}) * \frac{\partial^2}{\partial x \partial y} G(\mathbf{x}, \sigma) \end{aligned} \quad (3.4)$$

où  $G(\mathbf{x}, \sigma)$  est la fonction gaussienne définie par l'équation (3.2). Le paramètre  $\gamma$  a été introduit par Lindeberg [Lindeberg, 1998] pour définir une normalisation par rapport à l'échelle  $\sigma$ . Plus l'échelle  $\sigma$  est grande, plus le bruit dans l'image est supprimé tandis que les contours sont lissés (voir figure 3.10).



(a) Image originale      (b) Filtre hessien (norme euclidienne)      (c) Filtre hessien lissé (échelle  $\sigma = 1$ )      (d) Filtre hessien lissé (échelle  $\sigma = 3$ )

FIGURE 3.10 – Comparaison du filtre hessien et du hessien lissé sur l'image test de la lune. Le hessien lissé correspond à l'image convoluée avec les dérivées secondes de la gaussienne.

La matrice hessienne étant symétrique, elle est diagonalisable. Ses valeurs propres  $\lambda_1$ ,  $\lambda_2$  et  $\lambda_3$  ( $|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$ ) sont associées respectivement aux vecteurs propres  $e_1$ ,  $e_2$  et  $e_3$ , formant une base orthonormale. D'après la théorie de la décomposition en valeurs singulières, les vecteurs  $e_1$ ,  $e_2$  et  $e_3$  sont orientés dans les directions principales du tenseur associé à la matrice hessienne. À partir des trois directions et des trois valeurs propres, on peut alors construire un ellipsoïde représentant le tenseur décrit par la matrice hessienne, c'est-à-dire la structure de l'image en ce point (voir figure 3.11).

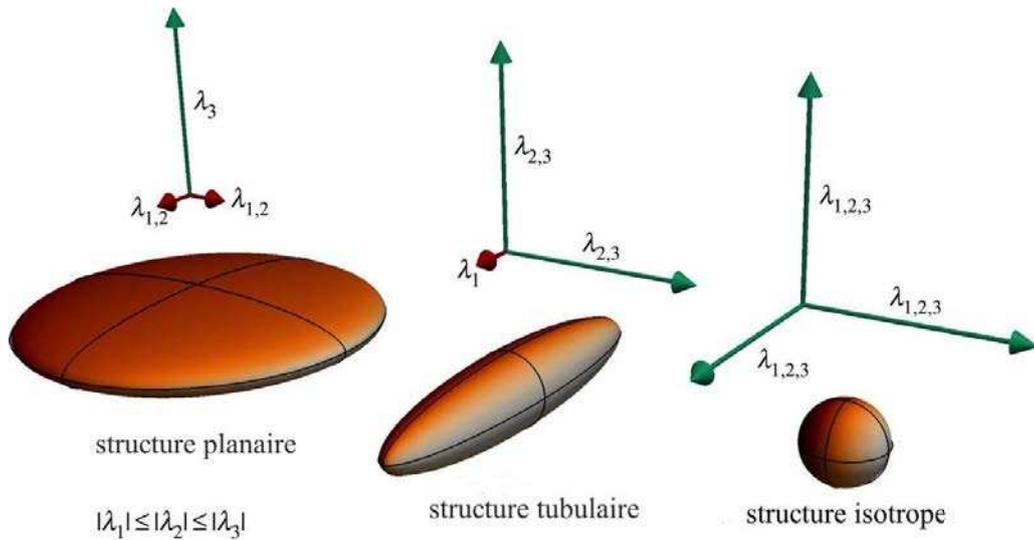


FIGURE 3.11 – Ellipsoïdes représentant la structure de l’image. Les demi-axes sont orientés selon les trois vecteurs propres  $e_1$ ,  $e_2$  et  $e_3$  et sont de dimensions  $\lambda_1$ ,  $\lambda_2$  et  $\lambda_3$ .

L’analyse des ordres de grandeur de ces valeurs propres permet donc de déterminer la forme (structure tubulaire, planaire ou isotrope) et l’intensité (claire, sombre) des structures locales. Les relations entre les valeurs propres de la matrice hessienne et les différentes formes et intensités dans des images 2D et 3D sont résumées dans le tableau 3.1. Par exemple, pour une structure tubulaire idéale dans une image 2D on a :

$$|\lambda_1| \approx 0 \tag{3.5}$$

$$|\lambda_1| \ll |\lambda_2| \tag{3.6}$$

Et en 3D, on rajoute :

$$|\lambda_2| \approx |\lambda_3| \tag{3.7}$$

2D		3D			Forme de l’objet
$\lambda_1$	$\lambda_2$	$\lambda_1$	$\lambda_2$	$\lambda_3$	
Forte+	Forte+	Forte+	Forte+	Forte+	Structure isotrope (sombre)
Forte+	Forte+	Forte-	Forte-	Forte-	Structure isotrope (claire)
Faible	Forte+	Faible	Forte+	Forte+	Structure tubulaire (sombre)
Faible	Forte-	Faible	Forte-	Forte-	Structure tubulaire (claire)
		Faible	Faible	Forte+	Structure planaire (sombre)
		Faible	Faible	Forte-	Structure planaire (claire)

TABEAU 3.1 – Différentes possibilités de formes d’objets dans une image 2D et 3D, en fonction de la norme des valeurs propres et du signe.

### 3.3.3 Mesures de tubularité (vesselness)

Lorenz [Lorenz *et al.*, 1997], Sato [Sato *et al.*, 1997] et Frangi [Frangi *et al.*, 1998] ont utilisé des *mesures de tubularité* ou *vesselness* pour détecter les structures tubulaires. Ces mesures ont toutes la particularité de calculer, à partir des valeurs propres de la matrice hessienne, un critère atteignant des valeurs fortes pour les structures tubulaires et des valeurs faibles pour les autres. Bien que nous n'utilisions que la vesselness de Frangi par la suite, nous décrivons également celles de Lorenz et Sato qui sont les travaux pionniers.

#### 3.3.3.1 Vesselness de Lorenz

Pour une image 3D  $I(x)$ , observée à l'échelle  $\sigma$ , Lorenz propose la mesure de tubularité suivante :

$$L(x) = \frac{|\lambda_1| + |\lambda_2|}{2|\lambda_3|} \quad (3.8)$$

qui prend bien des valeurs proches de 1 si l'on se trouve dans un tube, et des valeurs proches de 0 sinon.

#### 3.3.3.2 Vesselness de Sato

De même, Sato propose la mesure de tubularité suivante :

$$S(x, \sigma) = \begin{cases} |\lambda_3| \left(\frac{\lambda_2}{\lambda_3}\right)^\xi \left(1 + \frac{\lambda_1}{|\lambda_2|}\right)^\tau & \text{si } \lambda_3 < \lambda_2 < \lambda_1 < 0 \\ |\lambda_3| \left(\frac{\lambda_2}{\lambda_3}\right)^\xi \left(1 - \rho \frac{\lambda_1}{|\lambda_2|}\right)^\tau & \text{si } \lambda_3 < \lambda_2 < 0 < \lambda_1 < \frac{\|\lambda_2\|}{\rho} \\ 0 & \text{sinon} \end{cases} \quad (3.9)$$

où  $\xi \geq 0$  influence l'asymétrie des sections transversales ;  $\tau \geq 0$  contrôle la sensibilité au structure isotrope ; et  $0 < \rho \leq 1$  contrôle la sensibilité aux structures tubulaires.

#### 3.3.3.3 Vesselness de Frangi

Pour une image 3D, la vesselness de Frangi est :

$$\mathcal{V}(x, \sigma) = \begin{cases} 0 & \text{si } \lambda_2 > 0 \text{ ou } \lambda_3 > 0 \\ \left(1 - e^{-\frac{R_A^2}{2\alpha^2}}\right) \cdot e^{-\frac{R_B^2}{2\beta^2}} \cdot \left(1 - e^{-\frac{S^2}{2c^2}}\right) & \text{sinon} \end{cases} \quad (3.10)$$

avec

$$\begin{aligned} R_A &= \left| \frac{\lambda_2}{\lambda_3} \right| \\ R_B &= \frac{|\lambda_1|}{\sqrt{|\lambda_2 \lambda_3|}} \\ S &= \|\mathcal{H}(I)_\sigma\| = \sqrt{\sum_j \lambda_j^2} \end{aligned}$$

Le premier terme  $R_A$  discrimine les structures tubulaires des planaires, car  $R_A$  est proche de 1 pour un tube, alors que  $R_A$  est proche de 0 pour un plan. Le second terme  $R_B$  discrimine les structures isotropes des tubulaires, car  $R_B$  est proche de 1 pour un blob, alors que  $R_B$  est proche de 0 pour un tube. Enfin,  $S$  évalue le niveau de bruit du voisinage et donc la pertinence de l'analyse. Les paramètres  $\alpha$ ,  $\beta$  and  $c$  contrôlent respectivement

la sensibilité du filtre aux mesures  $R_A$ ,  $R_B$  et  $S$ .

Pour une image 2D, la vesselness de Frangi peut être reformulée par :

$$\mathcal{V}(x, \sigma) = \begin{cases} 0, & \text{si } \lambda_2 > 0 \\ e^{-\frac{R_B^2}{2\beta^2}} \cdot (1 - e^{-\frac{S^2}{2\sigma^2}}), & \text{sinon} \end{cases} \quad (3.11)$$

où  $R_B = \frac{|\lambda_1|}{|\lambda_2|}$  discrimine les structures isotropes des filaires.

### 3.3.3.4 Cadre multi-échelle

Ces mesures admettent également un paramètre d'échelle  $\sigma$  utilisé lors du calcul des dérivées premières et secondes à partir de la fonction gaussienne (cf. équation (3.4)). Ce paramètre étant a priori inconnu, une solution proposée par Sato et Frangi consiste alors à calculer la réponse, en chaque point de l'image,  $S(x, \sigma)$  ou  $\mathcal{V}(x, \sigma)$  pour plusieurs valeurs de  $\sigma$  et de conserver la réponse maximale (voir figure 3.12). Par exemple, pour la vesselness de Frangi, on a :

$$\mathcal{V}(x) = \max_{\sigma_{min} \leq \sigma \leq \sigma_{max}} \mathcal{V}(x, \sigma)$$

où  $\sigma_{min}$  et  $\sigma_{max}$  sont les échelles minimales et maximales pour lesquelles on s'attend à trouver les structures d'intérêt.

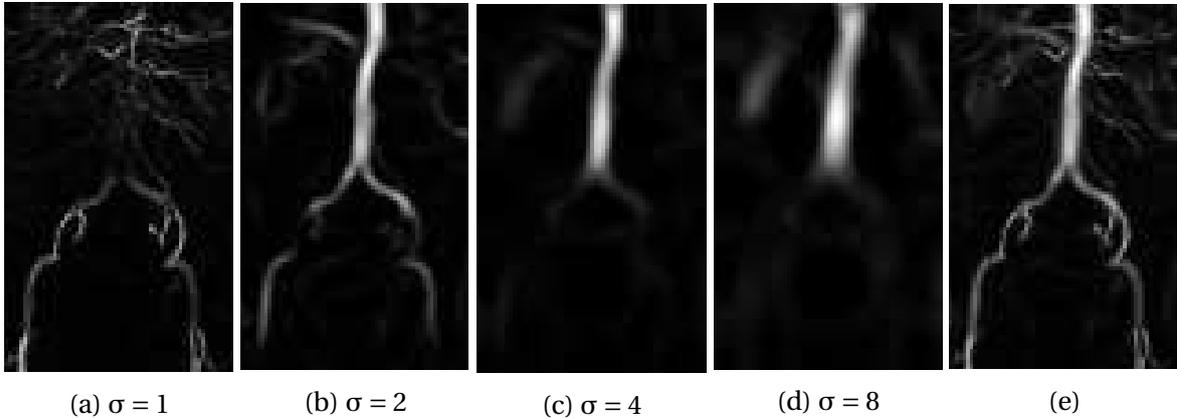


FIGURE 3.12 – (a–d) Vesselness de Frangi obtenue pour quatre facteurs d'échelle croissants, (e) réponse maximale [Frangi *et al.*, 1998].

### 3.3.4 Tenseur de structures

Le tenseur des structures  $\mathcal{J}$  est la matrice des moments du second-ordre du gradient de l'image  $I$ . En 2D, il est défini par :

$$\mathcal{J}(I) = \begin{bmatrix} \left( \frac{\partial G_\sigma(I)}{\partial x} \right)^2 & \frac{\partial G_\sigma(I)}{\partial x} \frac{\partial G_\sigma(I)}{\partial y} \\ \frac{\partial G_\sigma(I)}{\partial y} \frac{\partial G_\sigma(I)}{\partial x} & \left( \frac{\partial G_\sigma(I)}{\partial y} \right)^2 \end{bmatrix}$$

En 3D, le tenseur des structures est défini par :

$$\mathcal{J}(\mathbf{I}) = \begin{bmatrix} \left(\frac{\partial G_\sigma(\mathbf{I})}{\partial x}\right)^2 & \frac{\partial G_\sigma(\mathbf{I})}{\partial x} \frac{\partial G_\sigma(\mathbf{I})}{\partial y} & \frac{\partial G_\sigma(\mathbf{I})}{\partial x} \frac{\partial G_\sigma(\mathbf{I})}{\partial z} \\ \frac{\partial G_\sigma(\mathbf{I})}{\partial y} \frac{\partial G_\sigma(\mathbf{I})}{\partial x} & \left(\frac{\partial G_\sigma(\mathbf{I})}{\partial y}\right)^2 & \frac{\partial G_\sigma(\mathbf{I})}{\partial y} \frac{\partial G_\sigma(\mathbf{I})}{\partial z} \\ \frac{\partial G_\sigma(\mathbf{I})}{\partial z} \frac{\partial G_\sigma(\mathbf{I})}{\partial x} & \frac{\partial G_\sigma(\mathbf{I})}{\partial z} \frac{\partial G_\sigma(\mathbf{I})}{\partial y} & \left(\frac{\partial G_\sigma(\mathbf{I})}{\partial z}\right)^2 \end{bmatrix}$$

où  $G_\sigma(\mathbf{I})(x) = \mathbf{I}(x) \star G(x, \sigma)$  avec  $G(x, \sigma)$  la fonction gaussienne centrée de variance  $\sigma^2$  définie dans la section 3.3. Le tenseur de structure peut être reformulé comme le produit matriciel :

$$\mathcal{J}(\mathbf{I}) = \nabla(G_\sigma(\mathbf{I})) \cdot \nabla(G_\sigma(\mathbf{I}))^T$$

Le tenseur de structure contient les directions principales du gradient dans le voisinage du pixel  $x$ . Il peut alors être utilisé de la même manière que pour l'opérateur hessien, bien que son interprétation soit différente. Puisque son déterminant est une forme quadratique, et en raison de sa forme, le tenseur de structure est semi-défini positif. Toutes ses valeurs propres sont positives et peuvent être ordonnées comme précédemment, par exemple,  $0 \leq \lambda_1 \leq \lambda_2 \leq \lambda_3$  en 3D.

Dans cet ordre,  $\lambda_3$  est la plus grande valeur propre et son vecteur propre associé  $e_3$  fournit la direction principale du gradient de l'image  $\mathbf{I}$ . Si  $\lambda_3 \gg \lambda_1 \approx \lambda_2$ , alors cela signifie que les isosurfaces de  $\mathbf{I}$  sont des plans perpendiculaires à  $e_3$ . Si  $\lambda_3 \approx \lambda_2 \gg \lambda_1$ , alors les isosurfaces de  $\mathbf{I}$  sont des tubes perpendiculaires à  $e_3$ . Si les trois valeurs propres sont du même ordre de grandeur, alors les isosurfaces sont des structures isotropes. Identiquement au cas de l'opérateur hessien, on peut se ramener à un cadre multi-échelle en faisant varier le paramètre  $\sigma$ . Par contre, les valeurs propres du tenseur de structures ne permettent pas de déterminer si les structures sont claires ou sombres.

Dans le chapitre suivant, nous allons proposer deux méthodes hybrides qui incluent la vesselness de Fangi dans un cadre variationnel (la première dans le modèle de débruitage ROF ou TV-L1 et la seconde dans le modèle de segmentation de Chan-Vese).

# Chapitre 4

## Inclusion de la tubularité dans le modèle variationnel

*« Il est plus facile de faire sortir le dentifrice du tube que de l'y faire rentrer. »*

---

Pierre Dac

Ce chapitre présente nos contributions apportées à la segmentation d'images angiographiques. Nous proposons un premier modèle hybride qui inclut un a priori de tubularité dans les modèles variationnels de débruitage ROF et TV-L1. Néanmoins, bien que ces modèles permettent de réhausser les vaisseaux dans l'image, ils ne permettent pas de les segmenter. C'est pourquoi nous proposons un deuxième modèle hybride, qui inclut à la fois un a priori de tubularité et de direction dans le modèle variationnel de segmentation de Chan-Vese. Des résultats sont fournis sur des images synthétiques 2D, et sur des images rétiniennes de la base DRIVE.

### Sommaire

---

<b>4.1 Introduction</b> . . . . .	<b>74</b>
<b>4.2 Inclusion de la tubularité dans les modèles de débruitage ROF et TV-L1</b> .	<b>74</b>
4.2.1 Méthodologie . . . . .	74
4.2.2 Algorithme primal-dual . . . . .	75
4.2.2.1 Modèle ROF . . . . .	75
4.2.2.2 Modèle TV-L1 . . . . .	76
4.2.3 Résultats . . . . .	77
<b>4.3 Inclusion de la tubularité et des directions dans le modèle de segmentation de Chan-Vese</b> . . . . .	<b>81</b>
4.3.1 Inclusion de la tubularité . . . . .	81
4.3.1.1 Méthodologie . . . . .	81
4.3.1.2 Algorithme FGP . . . . .	81
4.3.1.3 Résultats . . . . .	82
4.3.2 Inclusion de la tubularité et des directions . . . . .	86
4.3.2.1 Méthodologie . . . . .	86
4.3.2.2 Implémentation . . . . .	87
4.3.2.3 Résultats . . . . .	89

---

## 4.1 Introduction

Dans ce chapitre, nous proposons de combiner les modèles variationnels, présentés dans le chapitre 1, avec les mesures de tubularité, présentées dans le chapitre 3. Ainsi, nous souhaitons adapter la restauration d'images (débruitage / segmentation) aux structures spécifiques que sont les structures tubulaires. Dans la section 4.2, un premier modèle hybride est décrit [Miraucourt *et al.*, 2015] ; celui-ci pondère la fidélité dans les modèles variationnels de débruitage ROF et TV-L1 par un a priori de tubularité. Néanmoins, bien que ces modèles permettent de réhausser les vaisseaux dans l'image, un seuillage est nécessaire afin d'obtenir une segmentation. C'est pourquoi nous proposons un deuxième modèle hybride qui inclut à la fois un a priori de tubularité et de direction dans le modèle variationnel de segmentation de Chan-Vese [Merveille *et al.*, 2016a]. Dans le contexte des structures tubulaires, des extensions du modèle de Chan-Vese ont déjà été proposées en ajoutant des a priori de tubularité, comme par exemple les superellipsoïdes [Tyrrell *et al.*, 2007], les frames d'ondelettes B-splines [Tai *et al.*, 2013], les dictionnaires adaptatifs [Rigamonti et Lepetit, 2012] et la régularisation élastique [El-Zehiry et Grady, 2016]. Nous avons développé un modèle pondérant le modèle de Chan-Vese par la tubularité, décrit dans la section 4.3.1. Bien que ce modèle soit plus performant que les modèles ROF et TV-L1, il présente toutefois des déconnexions. Dans la section 4.3.2, nous proposons donc un modèle incluant à la fois la tubularité et les directions des vaisseaux dans le modèle de Chan-Vese afin de pallier ce problème. Les directions sont extraites de la tubularité (ce sont les vecteurs propres associés aux plus petites valeurs propres en valeur absolue) et sont intégrées dans le calcul du gradient, appelé *gradient directionnel*. Pour extraire ces directions, nous avons comparé la vesselness de Frangi avec l'opérateur RORPO (Ranking Orientation Responses of Path Operators) [Merveille *et al.*, 2014, Merveille *et al.*, 2016b], qui est un filtre d'objets tubulaires basé sur le concept d'ouverture en morphologie mathématique. Des résultats sont fournis sur des images synthétiques 2D, et sur des images rétiniennes de la base DRIVE.

## 4.2 Inclusion de la tubularité dans les modèles de débruitage ROF et TV-L1

### 4.2.1 Méthodologie

Tout d'abord, rappelons que le modèle ROF, décrit plus en détail dans le chapitre 1, correspond au problème de minimisation suivant :

$$\min_x \int_{\Omega} |\nabla x| + \frac{\lambda}{2} \int_{\Omega} (x - f)^2 dx \quad (4.1)$$

où  $x$  est l'image débruitée,  $f$  l'image observée,  $\Omega$  le domaine de l'image, tandis que le paramètre  $\lambda$  permet de gérer la pondération entre régularisation et fidélité. Rappelons également que lorsque le terme de fidélité L2 dans le modèle ROF est remplacé par la norme L1, on obtient le modèle TV-L1 :

$$\min_x \int_{\Omega} |\nabla x| + \lambda \int_{\Omega} |x - f| dx \quad (4.2)$$

En ce qui concerne la segmentation de structures tubulaires, la régularisation de ces deux modèles (i.e. la variation totale) n'est pas adaptée puisqu'elle pénalise les contours, dont

les structures tubulaires sont essentiellement composées. Dans notre modèle hybride, nous proposons de combiner le terme de fidélité des problèmes 4.1 et 4.2 avec un a priori de tubularité, et plus particulièrement la vesselness de Frangi décrite au chapitre 3. Pour cela, le paramètre  $\lambda$  est remplacé par :

$$\lambda(x) = \alpha \lambda_{\text{reg}} + (1 - \alpha) \mathcal{V}(x) \quad (4.3)$$

où  $\mathcal{V}(x)$  est la vesselness normalisée entre 0 et 1, et le paramètre  $\alpha \in [0, 1]$  joue un rôle de balance entre régularisation et vesselness (si  $\alpha \approx 0$ , on favorise la vesselness, tandis que si  $\alpha \approx 1$ , on favorise la régularisation). En incluant ainsi la vesselness, l'effet désiré est de régulariser plus à l'extérieur des structures tubulaires qu'à l'intérieur, afin de préserver ces structures. Le paramètre  $\lambda$ , qui était précédemment une valeur constante pour toute l'image, dépend désormais de la tubularité en chaque point; néanmoins le problème reste convexe. L'algorithme primal-dual, utilisé pour résoudre les équations (4.1) et (4.2), est détaillé dans la section 4.2.2, et des résultats sont présentés dans la section 4.2.3 sur des images synthétiques 2D et des images rétinienne de la base DRIVE, qui permettent une étude quantitative puisqu'elles ont été segmentées manuellement.

## 4.2.2 Algorithme primal-dual

Considérons les fonctions  $F$  et  $G$  de  $\Gamma^0(\mathbb{R}^n)$  et l'opérateur linéaire  $K : \mathbb{R}^n \rightarrow \mathbb{R}^n$ . La formulation générale des problèmes (4.1) et (4.2) est alors donnée par :

$$\min_{x \in \mathbb{R}^n} F(Kx) + G(x) \quad (4.4)$$

Nous avons vu dans le chapitre 2 que ce problème de minimisation peut être résolu par des algorithmes d'optimisation convexe tels que les algorithmes proximaux. Nous avons choisi l'algorithme primal-dual (voir section 2.5.7) qui ne nécessite aucune contrainte de différentiabilité et est rapide. Afin d'appliquer l'algorithme primal-dual au modèle ROF et TV-L1, il est nécessaire de détailler les opérateurs proximaux  $\text{prox}_{\sigma F^*}$  et  $\text{prox}_{\tau G}$ .

### 4.2.2.1 Modèle ROF

Pour le modèle ROF, le terme de régularisation est  $F(Kx) = \|\nabla x\|$ , l'attache aux données est  $G_{\text{ROF}}(x) = \frac{\lambda}{2} \|x - f\|^2$  et l'opérateur linéaire  $K = \nabla$  implique une constante de Lipschitz  $L^2 = \|\nabla\|^2 = \|\text{div}\|^2 \leq 8$  [Chambolle, 2004, Théorème 3.1]. L'opérateur proximal pour la variable primale est donné par :

$$x = \text{prox}_{\tau G_{\text{ROF}}} \tilde{x} = \arg \min_x \left\{ \frac{\|x - \tilde{x}\|^2}{2\tau} + \frac{\lambda}{2} \|x - f\|^2 \right\} \quad (4.5)$$

Résoudre ce problème de minimisation équivaut à résoudre l'équation d'Euler-Lagrange associée :

$$\frac{1}{\tau}(x - \tilde{x}) + \lambda(x - f) = 0$$

Alors, la solution est donnée par :

$$x = \frac{\tilde{x} + \lambda \tau f}{1 + \lambda \tau} \quad (4.6)$$

Enfin, l'opérateur proximal pour la variable duale est donné par :

$$y = \text{prox}_{\sigma F^*} \tilde{y} = \arg \min_y \left\{ \frac{\|y - \tilde{y}\|^2}{2\sigma} + F^*(y) \right\}$$

Maintenant, on détermine le conjugué  $F^*$  de  $F$  par dualité, de la même façon que pour l'algorithme de Chambolle (voir section 2.5.3) :

$$\begin{aligned} F(Kx) = \|\nabla x\|_{2,1} &= \sup_{\{\xi \text{ tel que } |\xi_{i,j}| \leq 1\}} \langle \xi, \nabla x \rangle && \text{(par définition)} \\ &= \sup_{\{\xi \text{ tel que } |\xi_{i,j}| \leq 1\}} \langle -\operatorname{div} \xi, x \rangle && \text{(formule de Green)} \\ &= \sup_{p \in \mathbb{R}^n} (\langle p, x \rangle - \iota_P(p)) && \text{(corollaire 2.5.8)} \end{aligned}$$

où  $P = \{p = -\operatorname{div} \xi : |\xi_{i,j}| \leq 1 \forall i, j\}$  et  $\iota_P$  et la fonction indicatrice de  $P$  définie par :

$$\iota_P(p) = \begin{cases} 0 & \text{si } p \in P \\ +\infty & \text{si } p \notin P \end{cases}$$

Donc  $F^*(y) = \iota_P(y)$  et puisque c'est la fonction indicatrice d'un ensemble convexe, l'opérateur proximal se réduit à une projection euclidienne sur la boule unité :

$$y = \frac{\tilde{y}}{\max(1, |\tilde{y}|)} \quad (4.7)$$

L'algorithme primal-dual général pour résoudre le modèle ROF est alors résumé dans l'Algorithme 4.1.

---

**Algorithme 4.1** : Algorithme primal-dual général pour résoudre le modèle ROF

---

Soient  $\tau = \sigma = \frac{1}{L} = \frac{1}{\sqrt{8}}$ ,  $\theta = 1$

$x_0 = f$ ,  $y_0 = 0$  et  $\bar{x}_0 = x_0$

**pour**  $k \geq 0$  **faire**

$$\tilde{y}_{k+1} = y_k + \sigma \nabla \bar{x}_k$$

$$y_{k+1} = \frac{\tilde{y}_{k+1}}{\max(1, |\tilde{y}_{k+1}|)} \quad \text{(équation 4.7)}$$

$$x_{k+1} = (1 + \tau\lambda)^{-1} (x_n + \tau \operatorname{div} y_{k+1} + \tau\lambda f) \quad \text{(équation 4.6)}$$

$$\bar{x}_{k+1} = x_{k+1} + \theta(x_{k+1} - x_k) \quad \text{(étape de relaxation)}$$

**fin**

---

#### 4.2.2.2 Modèle TV-L1

Pour le modèle TV-L1, on peut reprendre le même opérateur proximal pour la variable duale  $\operatorname{prox}_{\sigma F^*}$  ; seul l'opérateur proximal pour la variable primale change :  $\operatorname{prox}_{\tau G_{\text{TV-L1}}}(\tilde{x})$  avec  $G_{\text{TV-L1}} = \lambda \|x - f\|$ . Cet opérateur proximal est donné par le problème de minimisation suivant :

$$x = \operatorname{prox}_{\tau G_{\text{TV-L1}}}(\tilde{x}) = \arg \min_x \left\{ \frac{\|x - \tilde{x}\|^2}{2\tau} + \lambda \|x - f\| \right\} \quad (4.8)$$

avec l'équation d'Euler-Lagrange associée :

$$\frac{1}{\tau}(x - \tilde{x}) + \lambda \frac{x - f}{|x - f|} = 0$$

On distingue 3 cas :

- **1<sup>er</sup> cas** :  $\tilde{x} - \tau\lambda > f$  ( $\Leftrightarrow \tilde{x} - f > \tau\lambda$ ). Alors

- Soit  $x > f$  : alors  $\frac{1}{\tau}(x - \tilde{x}) + \lambda = 0$  et  $x = \tilde{x} - \tau\lambda$ .  
On a bien  $x = \tilde{x} - \tau\lambda > f$  et c'est donc une solution.
- Soit  $x < f$  : alors  $\frac{1}{\tau}(x - \tilde{x}) - \lambda = 0$  et  $x = \tilde{x} + \tau\lambda$ . Supposons que ce soit une solution alors  $\begin{cases} \tilde{x} + \tau\lambda < f \\ \tilde{x} - \tau\lambda > f \end{cases}$ , donc  $\tau\lambda < 0$ , ce qui est absurde.
- **2<sup>e</sup> cas** :  $\tilde{x} + \tau\lambda < f$  ( $\Leftrightarrow \tilde{x} - f < -\tau\lambda$ ). Alors la démarche est identique au 1<sup>er</sup> cas et l'on trouve  $x = \tilde{x} + \tau\lambda$ .
- **3<sup>e</sup> cas** :  $f - \tau\lambda \leq \tilde{x} \leq f + \tau\lambda$  ( $\Leftrightarrow |\tilde{x} - f| \leq \tau\lambda$ ). Alors
  - Soit  $x > f$  et on a  $x = \tilde{x} - \tau\lambda$ . Alors  $\begin{cases} \tilde{x} - \tau\lambda > f \\ \tilde{x} - \tau\lambda < f \end{cases}$ , impossible.
  - Soit  $x < f$  et on a  $x = \tilde{x} + \tau\lambda$ . Alors  $\begin{cases} \tilde{x} + \tau\lambda < f \\ \tilde{x} + \tau\lambda > f \end{cases}$ , impossible.
 Comme on a une unique solution, nécessairement  $x = f$ .

Finalement, on obtient le schéma de seuillage doux suivant :

$$x = \begin{cases} \tilde{x} - \tau\lambda & \text{si } \tilde{x} - f > \tau\lambda \\ \tilde{x} + \tau\lambda & \text{si } \tilde{x} - f < -\tau\lambda \\ f & \text{si } |\tilde{x} - f| \leq \tau\lambda \end{cases} \quad (4.9)$$

L'algorithme primal-dual général pour résoudre le modèle TV-L1 est alors décrit dans l'algorithme 4.2.

---

**Algorithme 4.2** : Algorithme primal-dual général pour résoudre le modèle TV-L1

---

Soient  $\tau = \sigma = \frac{1}{L} = \frac{1}{\sqrt{9}}$ ,  $\theta = 1$

$x_0 = f$ ,  $y_0 = 0$  et  $\tilde{x}_0 = x_0$

**pour**  $k \geq 0$  **faire**

$$\tilde{y}_{k+1} = y_k + \sigma \nabla \tilde{x}_k$$

$$y_{k+1} = \frac{\tilde{y}_{k+1}}{\max(1, |\tilde{y}_{k+1}|)}$$

$$\tilde{x}_{k+1} = x_k + \tau \operatorname{div} y_{k+1}$$

$$x_{k+1} = \begin{cases} \tilde{x}_{k+1} - \tau\lambda & \text{si } \tilde{x}_{k+1} - f > \tau\lambda \\ \tilde{x}_{k+1} + \tau\lambda & \text{si } \tilde{x}_{k+1} - f < -\tau\lambda \\ f & \text{si } |\tilde{x}_{k+1} - f| \leq \tau\lambda \end{cases} \quad (\text{équation 4.9})$$

$$\tilde{x}_{k+1} = x_{k+1} + \theta(x_{k+1} - x_k) \quad (\text{étape de relaxation})$$

**fin**

---

### 4.2.3 Résultats

La figure 4.1 compare les modèles ROF et TV-L1 aux modèles ROF et TV-L1 pondérés par la vesselness de Frangi sur une image synthétique des lignes (cf. figure 4.1a). Cette image a été bruitée avec un bruit additif uniformément distribué (cf. figure 4.1d). Pour le modèle ROF (cf. figure 4.1b), nous avons utilisé l'algorithme 4.1 avec  $\lambda_{\text{reg}} = 20$  et pour le modèle ROF pondéré par la vesselness ( $\sigma = \llbracket 1, 3 \rrbracket$ ), nous avons remplacé le coefficient

$\lambda$  par celui donné dans l'équation (4.3) avec  $\lambda_{\text{reg}} = 20$  et  $\alpha = 0,5$  (cf. figure 4.1e). Pour le modèle TV-L1 (cf. figure 4.1c), nous avons appliqué l'algorithme 4.2 avec  $\lambda_{\text{reg}} = 2$  et pour le modèle ROF pondéré par la vesselness, les paramètres sont  $\sigma = \llbracket 1, 3 \rrbracket$ ,  $\lambda_{\text{reg}} = 2$  et  $\alpha = 0,5$  (cf. figure 4.1c). Les codes sont écrits en Matlab et utilisent la boîte à outils de Dirk-Jan Kroon "Hessian based Frangi Vesselness filter"<sup>1</sup> pour calculer la vesselness de Frangi.

Les paramètres utilisés ont été optimisés afin d'obtenir les meilleurs indicateurs de sensibilité et de spécificité. La sensibilité est le taux de vrais positifs (ici, la proportion de pixels blancs trouvés sur le nombre de pixels blancs total) et la spécificité est le taux de vrais négatifs (ici, la proportion de pixels noirs trouvés sur le nombre de pixels noirs total). La sensibilité et la spécificité ont été calculées après un seuillage de l'image au seuil de 0,5 (sachant qu'un pixel blanc correspond à 1 et un pixel noir à 0). On peut remarquer que le modèle TV-L1 débruite mieux le fond que le modèle ROF et que lorsque l'on inclut la vesselness, le bruit est encore mieux supprimé mais les lignes fines ont tendance à s'effacer.

La figure 4.2 compare les modèles ROF et TV-L1 pondérés par la vesselness sur une image rétinienne de la base DRIVE<sup>2</sup>. En pré-traitement, le canal vert a été choisi et la région d'intérêt a été délimitée à l'aide du masque donné par la base DRIVE (cf. figure 4.2b). Un filtre médian (avec une fenêtre de 30 pixels de large) a été également appliqué afin d'homogénéiser le fond (cf. figure 4.2c). On peut alors observer que ce filtre atténue les deux taches qui sont le disque optique à la confluence des vaisseaux et la macula au centre de l'image. Dans le cas des images rétiniennes, nous avons remplacé le paramètre spatialement variant  $\lambda(x)$  de l'équation (4.3) par le paramètre mieux adapté :

$$\lambda = \lambda_{\text{reg}}(1 - \alpha \mathcal{V}(x)) \quad (4.10)$$

Dans l'équation (4.3), lorsque  $\lambda$  est grand, la vesselness normalisée entre 0 et 1 a très peu de poids alors que dans l'équation (4.10), la vesselness garde son importance puisqu'elle est en facteur. La vesselness de Frangi est paramétrée par  $\sigma = \llbracket 1, 3 \rrbracket$  (cf. figure 4.2d). Les paramètres utilisés pour le modèle ROF + Vesselness sont  $\lambda_{\text{reg}} = 60$  et  $\alpha = 0,5$ , tandis que ceux pour TV-L1 + Vesselness sont  $\lambda_{\text{reg}} = 2$  et  $\alpha = 0,5$ . Les résultats de ces deux modèles sont donnés respectivement par les figures 4.2e et 4.2f. On peut constater que, même si les vaisseaux sont réhaussés (i.e. ils sont plus clairs que l'image originale), ces vaisseaux ne sont pas segmentés comme sur la vérité terrain (cf. figure 4.2g).

Pour obtenir une segmentation, un seuillage a été effectué au seuil de 0,75 (cf. figures 4.2h et 4.2i). Les sensibilités et spécificités ont alors pu être calculées, ainsi que la précision, qui est le rapport de pixels correctement attribués (vrais positifs et vrais négatifs) sur le nombre de pixels total. Cette précision reste correcte par rapport à la littérature (cf. tableau 4.1). Néanmoins, la sensibilité, qui a le plus d'enjeu, reste faible. Cela est certainement dû au seuillage, qui a supprimé les vaisseaux les plus sombres. C'est pourquoi dans la prochaine section, l'a priori de tubularité est intégré au modèle de segmentation de Chan-Vese pour éviter cette étape de seuillage.

1. [www.mathworks.com/matlabcentral/fileexchange/24409](http://www.mathworks.com/matlabcentral/fileexchange/24409)

2. [www.isi.uu.nl/Research/Databases/DRIVE/](http://www.isi.uu.nl/Research/Databases/DRIVE/)

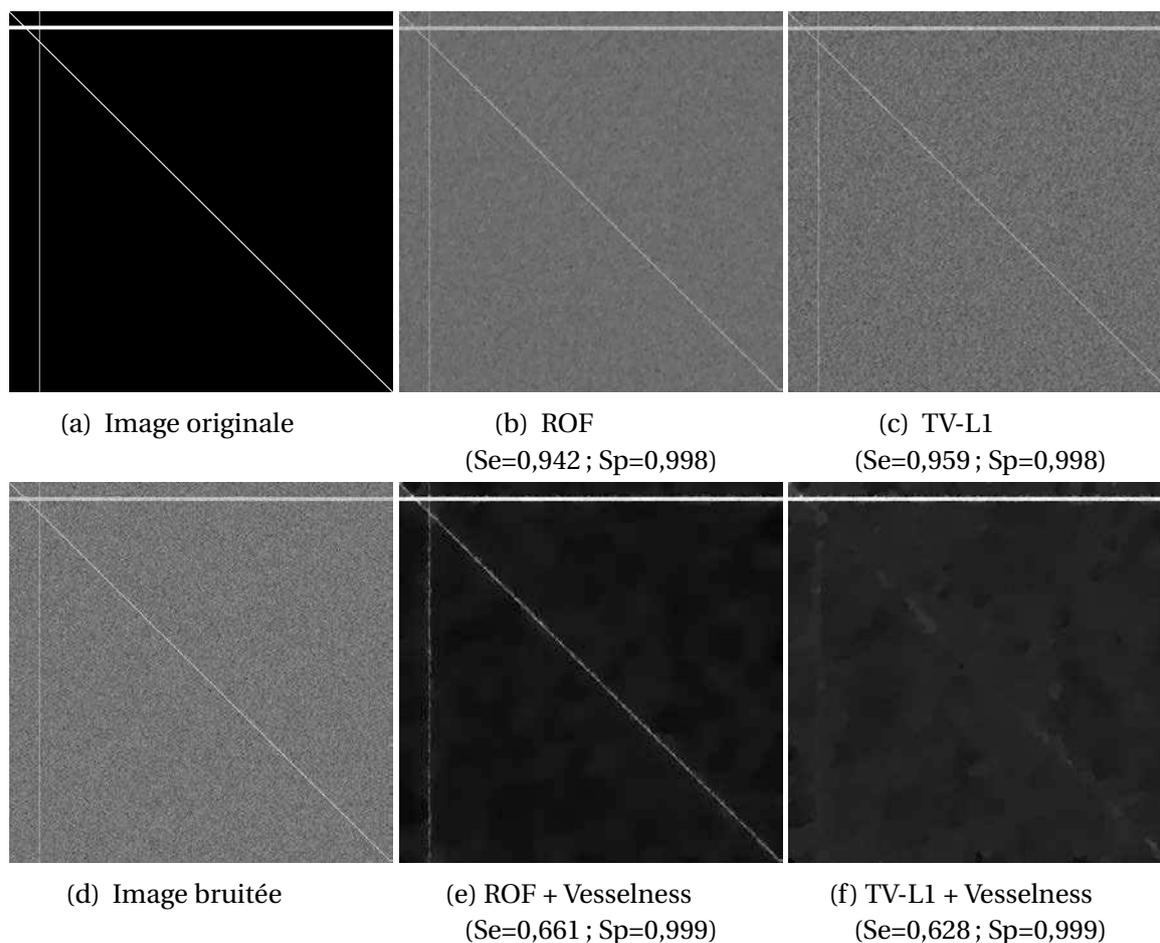


FIGURE 4.1 – Comparaison des modèles ROF et TV-L1 aux modèles ROF et TV-L1 pondérés par la vesselness, appliqués à une image synthétique. La sensibilité (Se) est le taux de vrais positifs ( $Se = VP / (VP + FN)$ , avec VP les vrais positifs et FN les faux négatifs) et la spécificité (Sp) est le taux de vrais négatifs ( $Sp = VN / (VN + FP)$ , avec VN les vrais négatifs et FP les faux positifs).

Méthode	ROF	TVL1	Lupaşcu	Staal	Niemeijer	Zana
Sensibilité	0,646	0,639	0,740	0,778	0,719	0,773
Spécificité	0,984	0,989	0,985	0,971	0,972	0,975
Précision	0,939	0,942	0,963	0,946	0,939	0,949

TABEAU 4.1 – Comparaison des indicateurs pour les modèles ROF et TV-L1 pondérés par la vesselness avec différentes méthodes de la littérature [Lupaşcu *et al.*, 2010, Staal *et al.*, 2004, Niemeijer *et al.*, 2004, Zana et Klein, 2001] (image 1 de la base DRIVE).

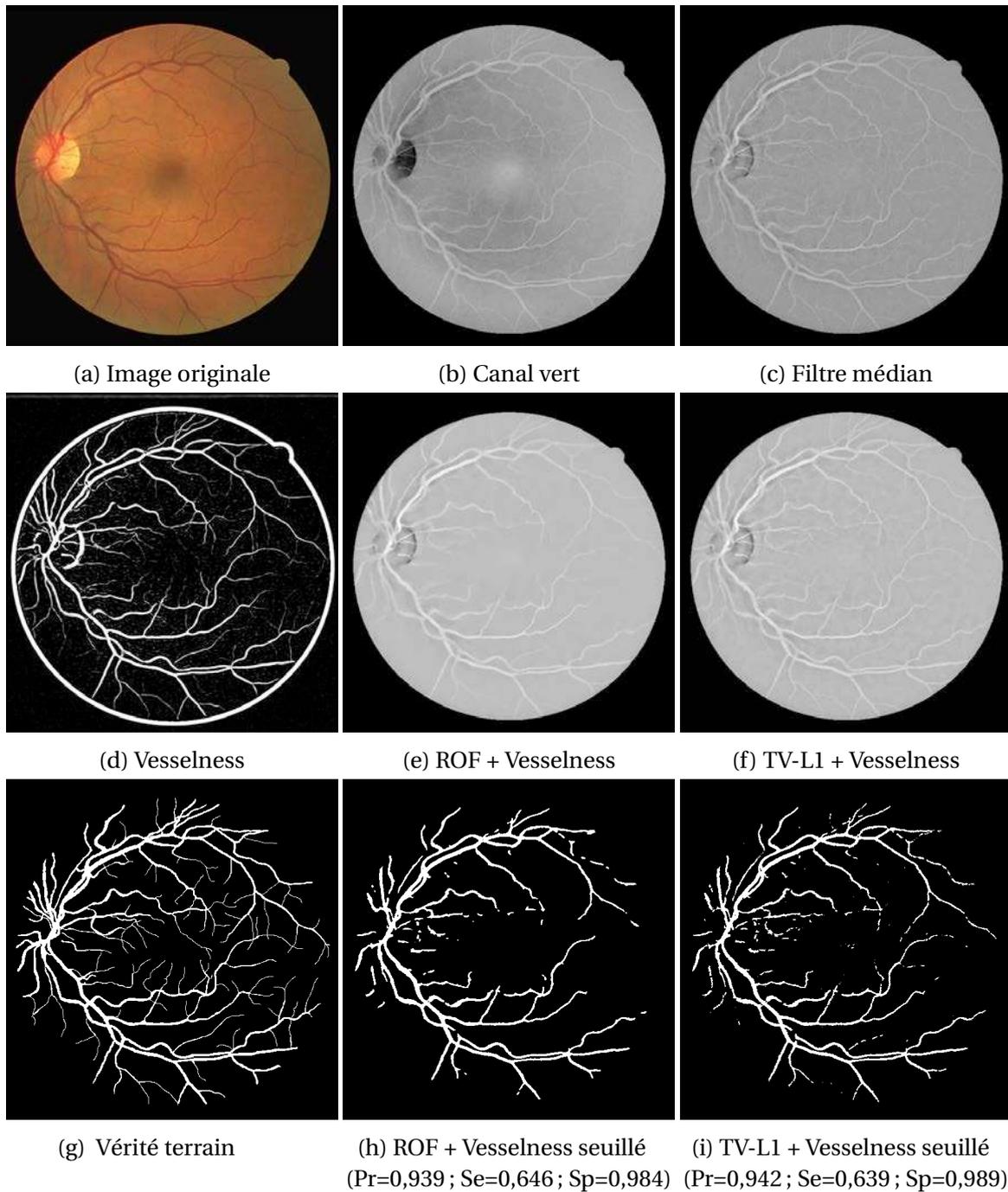


FIGURE 4.2 – Comparaison des modèles ROF et TV-L1 pondérés par la vesselness, appliqués à une image rétinienne de la base DRIVE (image 1). La précision (Pr) est le taux de vrais positifs et vrais négatifs ( $Pr = (VP + VN) / (VP + VN + FP + FN)$ ).

## 4.3 Inclusion de la tubularité et des directions dans le modèle de segmentation de Chan-Vese

### 4.3.1 Inclusion de la tubularité

#### 4.3.1.1 Méthodologie

Nous avons vu dans le chapitre 1 (cf. section 1.3.2.2) que le problème de segmentation de Chan-Vese correspond au problème de minimisation suivant :

$$\min_{\Sigma, c_1, c_2} \lambda \text{Per}(\Sigma) + \underbrace{\int_{\Sigma} (c_1 - f)^2 dx}_{\text{région d'intérêt}} + \underbrace{\int_{\Omega \setminus \Sigma} (c_2 - f)^2 dx}_{\text{fond}} \quad (4.11)$$

où  $c_1$  est la valeur de l'intensité moyenne dans la région d'intérêt  $\Sigma$ ,  $c_2$  est la valeur de l'intensité moyenne dans le fond  $\Omega \setminus \Sigma$  et  $\text{Per}(\Sigma)$  est le périmètre de  $\Sigma$ .

Ce problème n'étant pas convexe (minimisation sous contrainte d'être binaire), nous avons montré, dans le chapitre 1, qu'il est néanmoins équivalent au problème de minimisation convexe [Chan *et al.*, 2006] :

$$\min_{0 \leq u \leq 1} \lambda \int_{\Omega} |\nabla u| + \int_{\Omega} ((c_1 - f(x))^2 - (c_2 - f(x))^2) u(x) dx \quad (4.12)$$

Et alors  $\Sigma = \{x : u(x) \geq \mu\}$  pour tout  $\mu \in ]0, 1]$ .

En posant  $\text{TV}_c(u) = \lambda |\nabla u| + \mathbb{1}_{[0,1]}(u)$  la variation totale sous contraintes, et  $A(x) = (c_1 - f(x))^2 - (c_2 - f(x))^2$  l'attache aux données, on peut utiliser l'algorithme explicite-implicite, décrit dans le chapitre 2 (cf. section 2.5.4), pour résoudre l'équation (4.12) :

$$u(x) = \text{prox}_{\gamma \text{TV}_c}(u(x) - \gamma A(x)) \quad (4.13)$$

où  $\gamma$  est le pas de l'algorithme. L'algorithme complet est donné par l'algorithme 2.3. L'algorithme explicite-implicite se réduit à une descente de gradient sur la fonction différentiable  $A(x)$  et au calcul de l'opérateur proximal de la fonction non différentiable  $\text{TV}_c(u)$ . La difficulté réside dans le calcul de l'opérateur proximal  $\text{TV}_c$  sous contraintes donné par :

$$\text{prox}_{\text{TV}_c}(f) = \arg \min_{0 \leq u \leq 1} \|u - f\|^2 + 2\lambda \text{TV}_c(u)$$

Dans la section 4.3.1.3, nous allons décrire l'algorithme FGP (pour Fast Gradient Projection) permettant de minimiser la variation totale sous contraintes [Beck et Teboulle, 2009a]. De la même façon que pour les modèles précédents ROF et TV-L1 qui incluent la tubularité, le coefficient  $\lambda$  de l'équation (4.12) est remplacé par un paramètre  $\lambda(x)$  dépendant localement de la tubularité, donné par l'équation (4.10). Les résultats sont présentés dans la section 4.3.1.3.

#### 4.3.1.2 Algorithme FGP

Dans un premier temps, rappelons que l'opérateur proximal de la variation totale TV, dans le cas non contraint, peut être calculé à l'aide de l'algorithme de Chambolle [Chambolle, 2004] (voir section 2.5.3 pour plus de détails) :

$$\begin{aligned} u^{k+1} &= f - \lambda \text{div } p^k \\ p^{k+1} &= P(p^k + \tau \nabla u^{k+1}) \end{aligned}$$

où  $P(\xi) = \frac{\xi}{\max(1, |\xi|)}$  est la projection sur la boule unité,  $\tau$  le pas de l'algorithme et  $\text{div}$  la divergence discrète. Dans [Beck et Teboulle, 2009a], cet algorithme est adapté à la variation totale sous contrainte  $TV_c$ , en projetant l'étape de descente de gradient  $f - \text{div} p^k$  sur l'ensemble admissible, et on obtient l'algorithme GP (pour Gradient Projection) :

$$\begin{aligned} u_{k+1} &= P_C(f - \lambda \text{div}(p_k)) \\ p_{k+1} &= P(p_k + \tau \nabla u_{k+1}) \end{aligned}$$

où  $P_C$  est la projection sur l'ensemble convexe  $C$  (ici,  $C = [0, 1]$  pour satisfaire la contrainte  $0 \leq u \leq 1$ ). En ajoutant l'accélération de Nesterov [Nesterov, 1983], qui permet d'atteindre un taux de convergence optimal de  $O(1/k^2)$ , au lieu de  $O(1/k)$ , on obtient finalement l'algorithme FGP (voir algorithme 4.3).

La figure 4.3 compare les algorithmes de Chambolle, GP et FGP sur une image synthétique 2D bruitée. Les paramètres utilisés sont  $\lambda = 0.2$ ,  $\tau = \frac{1}{8\lambda}$  [Beck et Teboulle, 2009a, Lemme 4.2] et  $k = 1000$  itérations. On peut remarquer que les résultats sont similaires pour les trois algorithmes. Cependant, on peut constater sur la courbe de convergence (cf. figure 4.4), que ces algorithmes diffèrent de par leur convergence ; l'algorithme FGP a la convergence la plus rapide, ce qui est dû à l'accélération de Nesterov, et celui de Chambolle a la convergence la plus lente.

---

**Algorithme 4.3 :** Algorithm FGP (Fast Gradient Projection)

---

Soient  $u_0 = f$ ,  $p_0 = 0$ ,  $\eta_0 = p_0$ ,  $t_0 = 1$  et  $\tau > 0$  (pas de l'algorithme)

**pour**  $k \geq 0$  **faire**

$$\left| \begin{aligned} u_{k+1} &= P_C(f - \lambda \text{div}(\eta_k)) \\ p_{k+1} &= P(\eta_k + \tau \nabla u_{k+1}) \\ t_{k+1} &= \frac{1 + \sqrt{1 + 4t_k^2}}{2} \\ \eta_{k+1} &= p_{k+1} + \frac{t_k - 1}{t_k + 1}(p_{k+1} - p_k) \end{aligned} \right.$$

**fin**

---

### 4.3.1.3 Résultats

Dans cette section, les paramètres utilisés sont optimisés pour obtenir la meilleure précision. Pour les images rétiniennes, les paramètres  $c_1$  et  $c_2$  sont obtenus en calculant l'intensité moyenne dans la région d'intérêt et le fond de l'image originale. Ces régions sont déterminées à l'aide de la vérité terrain.

La figure 4.5 compare les modèles de Chan classique et Chan pondéré par la vesselness, appliqués à la même image synthétique 2D bruitée que sur la figure 4.3. Les paramètres utilisés sont  $\gamma = 10$ ,  $c_1 = 1$  et  $c_2 = 0$  pour l'algorithme explicite-implicite (4.13),  $\lambda = 1$  et  $\tau = \frac{1}{8\lambda}$  et  $\sigma = \llbracket 1, 2 \rrbracket$  et  $\alpha = 0,7$  pour la pondération par la vesselness. On peut remarquer que les deux modèles segmentent l'image (le résultat est binaire) et dans le cas de la pondération par la vesselness, la ligne diagonale du haut est légèrement plus visible que dans le modèle classique. Les figures 4.6a, b, c comparent ces modèles sur la même image rétinienne que la figure 4.2a (avec  $c_1 = 0,65$ ,  $c_2 = 0,49$ ,  $\lambda = 0,02$ ,  $\sigma = 1$  et

$\alpha = 1$ ). Le filtre médian, permettant d'homogénéiser le fond, a été également appliqué et ainsi les pixels du fond ont tous une intensité proche de l'intensité moyenne  $c_2$ . On peut remarquer que ces modèles donnent des résultats identiques, bien meilleurs que pour les modèles ROF et TV-L1 pondérés par la vesselness. La sensibilité a augmenté de plus de 0,1 et on s'aperçoit effectivement que l'on a détecté plus de petits vaisseaux. Une légère amélioration entre les modèles de Chan classique et Chan pondéré par la vesselness est visible sur les figures 4.7a, b, c qui comparent ces deux modèles sur l'image 3 de la base DRIVE (avec  $c_1 = 0,75$ ,  $c_2 = 0,47$ ,  $\lambda = 0,03$ ,  $\sigma = 1$  et  $\alpha = 0,9$ ), et les sensibilités sont alors de 0,67 sans pondération et 0,712 avec pondération. Les figures 4.7d, e, f, g zooment sur quelques extrémités de vaisseaux sanguins où une reconnexion ou un début de reconnexion apparaissent.

Nous avons testé ces modèles sur les 20 images de la base DRIVE, et les indicateurs (sensibilité, spécificité, précision) sont répertoriés dans le tableau 4.2 et peuvent être comparés avec les autres indicateurs de la littérature. Bien que la précision reste correcte, on peut améliorer le modèle en supprimant notamment les déconnexions. Pour cela, un a priori de directions est ajouté dans le modèle de la section suivante.

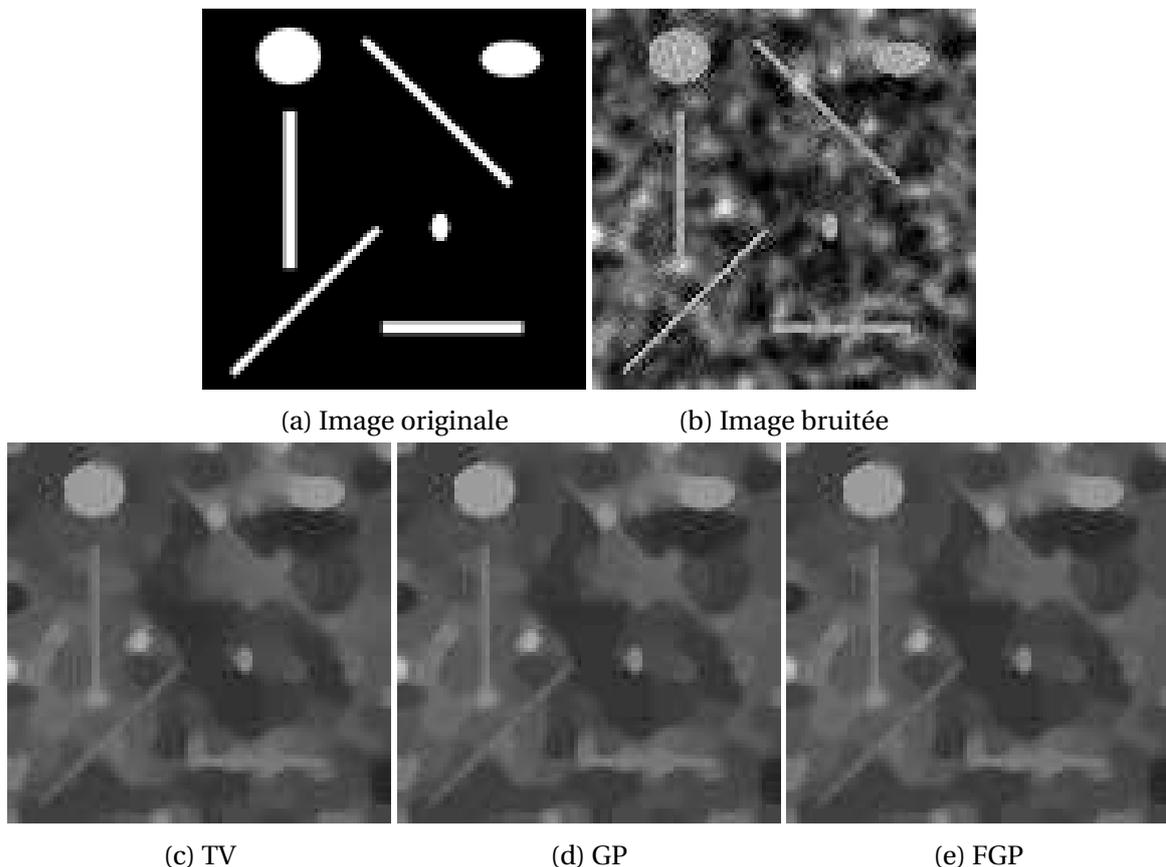


FIGURE 4.3 – Comparaison des algorithmes de Chambolle (TV), GP et FGP pour le débruitage d'une image synthétique 2D.

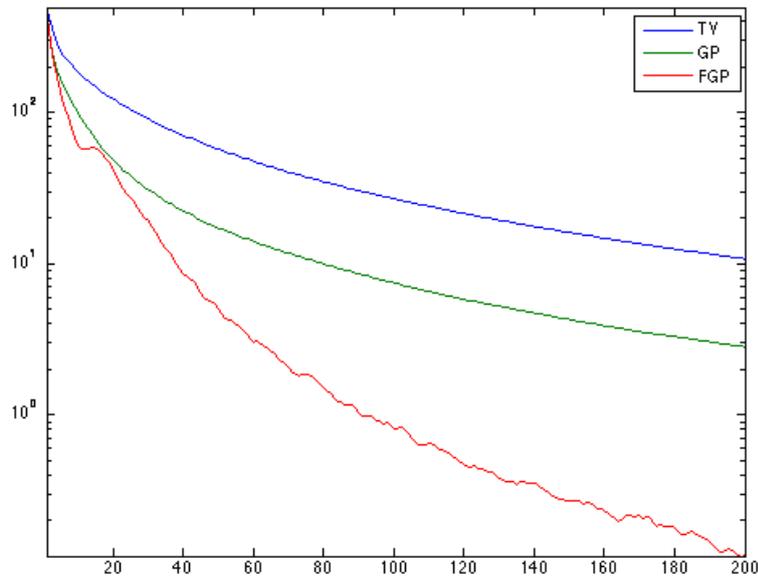


FIGURE 4.4 – Comparaison des convergences  $F(x_k) - F(x^*)$  pour les algorithmes TV, GP et FGP.  $F(x^*)$  est la solution à  $k = 1000$ .

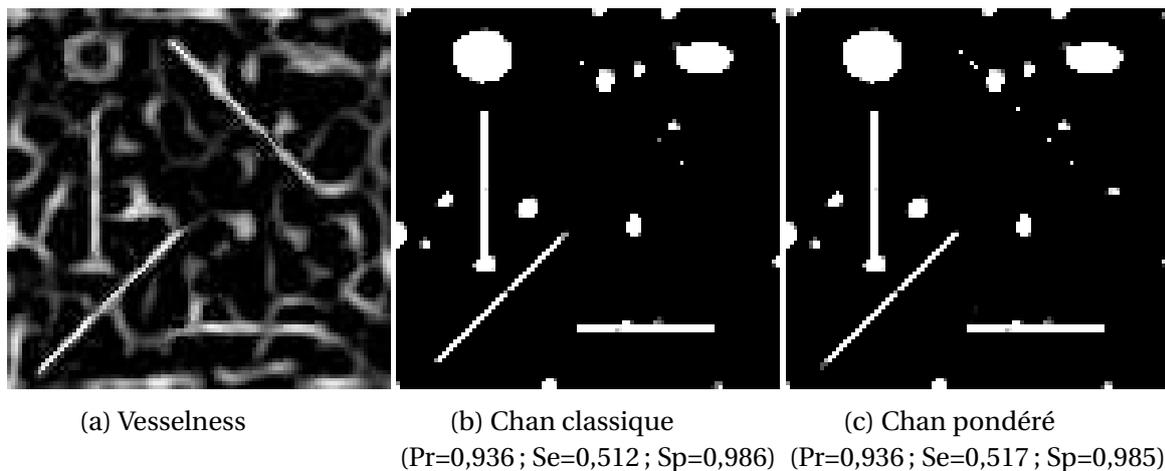


FIGURE 4.5 – Comparaison des modèles Chan classique et Chan pondéré par la vesselness appliqués à une image synthétique 2D.

Méthode	Chan	Chan pondéré	Lupașcu	Staal	Niemeijer	Zana
Sensibilité	0,6800	0,6892	0,6728	0,7193	0,6793	0,6696
Spécificité	0,9804	0,9800	0,9874	0,9773	0,9801	0,9769
Précision	0,9416	0,9423	0,9597	0,9442	0,9416	0,9377

TABEAU 4.2 – Comparaison des indicateurs pour les modèles Chan classique et Chan pondéré par la vesselness avec différentes méthodes de la littérature [Lupașcu *et al.*, 2010, Staal *et al.*, 2004, Niemeijer *et al.*, 2004, Zana et Klein, 2001] (moyenne sur les 20 images).

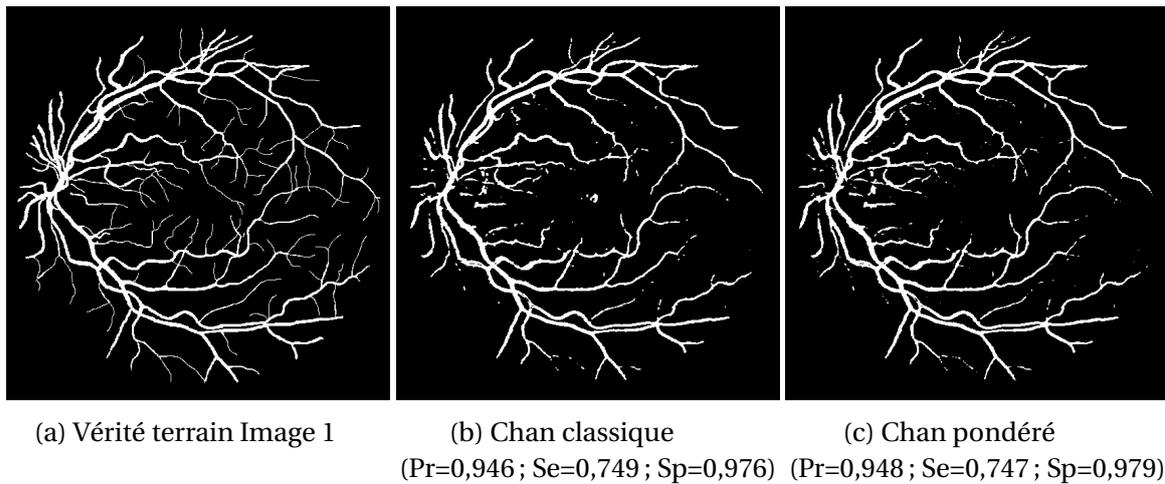


FIGURE 4.6 – Comparaison des modèles Chan classique et Chan pondéré par la vesselness appliqués à une image rétinienne (image 1 de la base DRIVE).

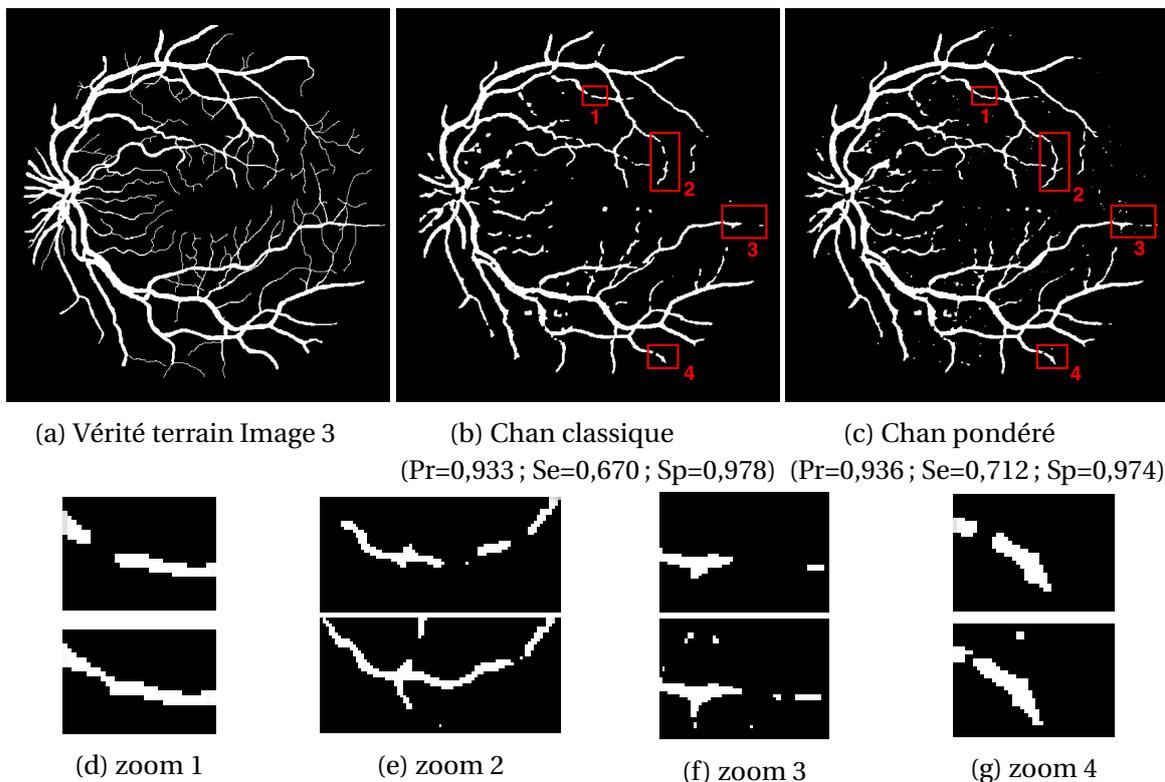


FIGURE 4.7 – 1<sup>re</sup> ligne : comparaison des modèles Chan classique et Chan pondéré par la vesselness appliqués à une image rétinienne (image 3 de la base DRIVE). 2<sup>e</sup> ligne : zoom sur les extrémités de quelques vaisseaux sanguins pour les modèles Chan classique (en haut) et Chan pondéré (en bas).

### 4.3.2 Inclusion de la tubularité et des directions

#### 4.3.2.1 Méthodologie

Rappelons que la variation totale, qui régularise de façon isotrope dans toute l'image, n'est pas adaptée aux structures tubulaires puisqu'elle va avoir tendance à les effacer (voir figure 4.8a). Nous avons donc proposé de la pondérer par la tubularité, afin d'éviter la disparition de telles structures. Néanmoins, ce modèle empêche le débruitage dans les vaisseaux et entraîne donc de nombreuses déconnexions et la perte des plus petits vaisseaux (voir figure 4.8b). Dans la suite, nous considérons alors à la fois la vesselness et les directions des vaisseaux. Au lieu de moins régulariser dans les vaisseaux, nous souhaitons régulariser le long de leurs axes principaux (voir figure 4.8c).

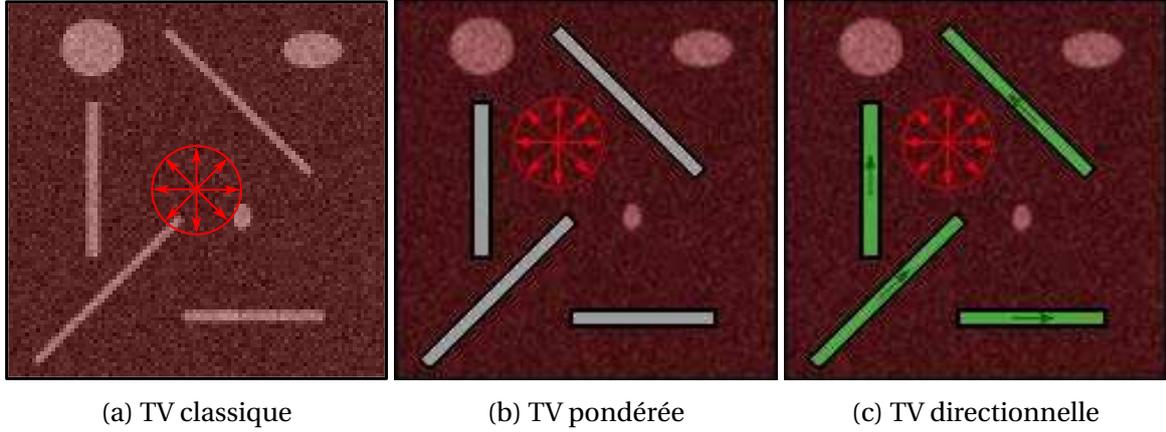


FIGURE 4.8 – Principe de régularisation dans le cas de la variation totale (TV) classique, la TV pondérée par la vesselness et la TV directionnelle. Les zones rouges correspondent à une régularisation isotrope, les zones grises à aucune régularisation et les zones vertes à une régularisation directionnelle. Les flèches représentent la direction de la régularisation.

Définissons l'opérateur de gradient directionnel  $\nabla_D$  incluant à la fois l'a priori de la tubularité et les directions. Peyré et al. ont proposé un gradient calculé sur un graphe pondéré [Peyré et al., 2008]. Le graphe et ses poids sont alors calculés à partir de l'image initiale à chaque itération de l'algorithme d'optimisation. Ici, nous proposons un gradient directionnel, où le graphe et les poids sont calculés une seule fois au début de l'algorithme, à partir de la vesselness et ses directions. Pour définir ce gradient directionnel, considérons une image 2D de taille  $N \times N$  et la famille de vecteurs unitaires  $(v_1, v_2, v_3, v_4)$  avec  $v_1 = (\vec{i} + \vec{j}) / \|\vec{i} + \vec{j}\|$ ,  $v_2 = \vec{i}$ ,  $v_3 = (\vec{j} - \vec{i}) / \|\vec{i} + \vec{j}\|$  et  $v_4 = \vec{j}$  (voir figure 4.9).

Le gradient directionnel est alors défini par :

$$(\nabla_D u) = (D^1(\nabla_d u)^1, D^2(\nabla_d u)^2, D^3(\nabla_d u)^3, D^4(\nabla_d u)^4)$$

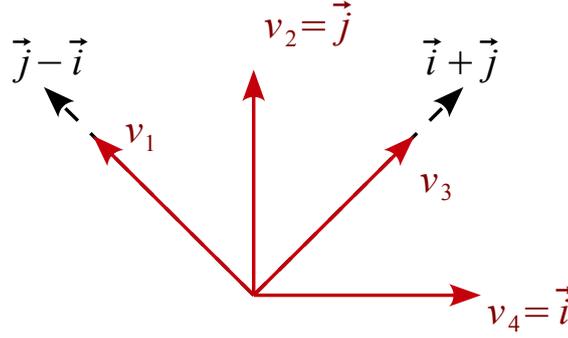
$$(\nabla_D u)_{i,j} = D^1_{i,j}(\nabla_d u)_{i,j}^1 v_1 + \dots + D^4_{i,j}(\nabla_d u)_{i,j}^4 v_4$$

avec  $D^q$ ,  $1 \leq q \leq 4$ , la matrice définie par :

$$D^q_{i,j} = d^q_{i,j} \mathcal{V}_{i,j} + (1 - \mathcal{V}_{i,j}) \quad (4.14)$$

$$\sum_{q=1}^4 D^q_{i,j} = 1 \quad \forall (i, j) \in \llbracket 0, N-1 \rrbracket^2 \quad (4.15)$$

où  $\mathcal{V}$  est la vesselness et  $(d^q)_{q \in \llbracket 1, 4 \rrbracket}$  les coefficients de direction dont le calcul est détaillé dans la section 4.3.2.2. Intuitivement, si  $\mathcal{V}_{i,j} = 0$ , i.e. le pixel  $(i, j)$  n'appartient pas à un


 FIGURE 4.9 – Famille de vecteurs  $(v_1, v_2, v_3, v_4)$ .

tube, alors  $(\nabla_D u)_{i,j} = (\nabla_d u)_{i,j}^1 v_1 + \dots + (\nabla_d u)_{i,j}^4 v_4$  est un gradient isotrope. Si  $\mathcal{V}_{i,j} = 1$ , i.e. le pixel  $(i, j)$  appartient à un tube, alors  $(\nabla_D u)_{i,j} = d_{i,j}^1 (\nabla_d u)_{i,j}^1 v_1 + \dots + d_{i,j}^4 (\nabla_d u)_{i,j}^4 v_4$  est un gradient directionnel, entraînant une régularisation dans l'axe du tube. Optimiser ce nouveau gradient ne provoque plus la perte des contours des structures tubulaires étant donné que seules les extrémités des contours sont capturées par le gradient directionnel (voir figure 4.10).

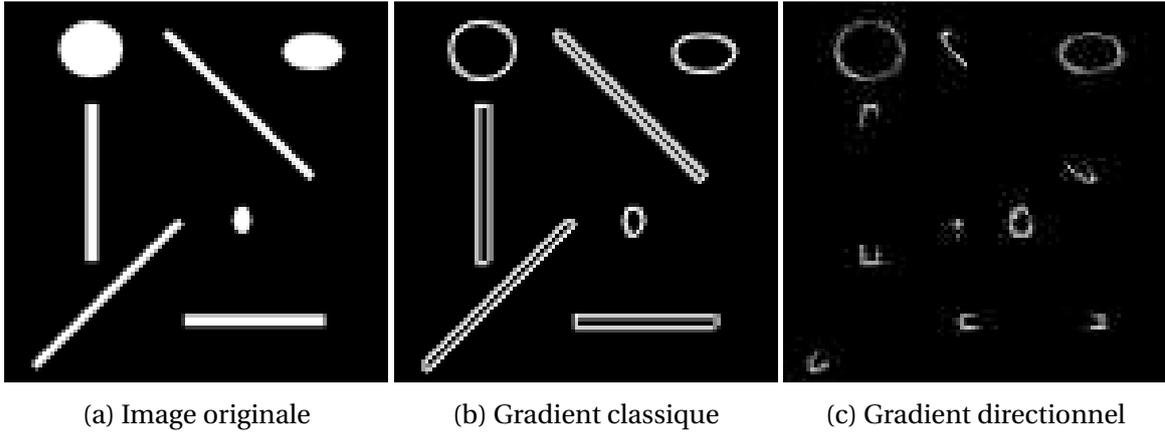


FIGURE 4.10 – Comparaison des gradients classique et directionnel sur une image synthétique 2D.

### 4.3.2.2 Implémentation

#### Calcul des coefficients de direction

Tout d'abord, rappelons que la matrice hessienne  $\mathcal{H}$  possède trois valeurs propres  $\lambda_1$ ,  $\lambda_2$  et  $\lambda_3$  ( $|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$ ), associées respectivement aux vecteurs propres  $e_1$ ,  $e_2$  et  $e_3$ . La direction principale du vaisseau est alors donnée par  $e_1$ , correspondant à la plus petite valeur propre  $\lambda_1$ . Notons alors  $\delta_{i,j} = (e_1^x, e_1^y)_{i,j}$  le vecteur donnant la direction locale du vaisseau au pixel  $(i, j)$ . Les coefficients  $(d^q)_{i,j}$  sont alors les coefficients de  $\delta_{i,j}$  décomposés sur la famille de vecteurs  $(v_1, v_2, v_3, v_4)$ . Notons que si tous les coefficients  $(d^q)_{i,j}$  ont la même valeur, alors  $\nabla_D$  est identique au gradient isotrope  $\nabla$ . Afin d'obtenir un gradient directionnel correct, nous imposons que la décomposition implique des coefficients nuls, sauf pour les deux vecteurs encadrant  $\delta_{i,j}$  (voir figure 4.11). Plus formellement, considérons  $v_m$  et  $v_n$  les deux vecteurs encadrant  $\delta_{i,j}$ , alors :

$$\delta_{i,j} = \sum_{q=1}^4 d_{i,j}^q v_q = d_{i,j}^m v_m + d_{i,j}^n v_n \quad (4.16)$$

En multipliant l'équation (4.16) par  $v_m$ , puis par  $v_n$ , on obtient le système suivant :

$$\begin{cases} d_{i,j}^m v_m \cdot v_m + d_{i,j}^n v_n \cdot v_m = \delta_{i,j} \cdot v_m & L_1 \\ d_{i,j}^m v_m \cdot v_n + d_{i,j}^n v_n \cdot v_n = \delta_{i,j} \cdot v_n & L_2 \end{cases}$$

Or, en remplaçant  $v_m \cdot v_n = \frac{\sqrt{2}}{2}$  car  $(v_m, v_n) = \frac{\pi}{4}$ , et  $v_m \cdot v_m = v_n \cdot v_n = 1$  car  $v_m$  et  $v_n$  sont des vecteurs unitaires, on obtient :

$$\begin{cases} d_{i,j}^m + \frac{\sqrt{2}}{2} d_{i,j}^n = \delta_{i,j} \cdot v_m & L_1 \\ \frac{\sqrt{2}}{2} d_{i,j}^m + d_{i,j}^n = \delta_{i,j} \cdot v_n & L_2 \end{cases}$$

Ce qui est équivalent à

$$\begin{cases} d_{i,j}^m + \frac{\sqrt{2}}{2} d_{i,j}^n = \delta_{i,j} \cdot v_m & L_1 \\ d_{i,j}^n - \frac{1}{2} d_{i,j}^m = \delta_{i,j} \cdot v_n - \frac{\sqrt{2}}{2} \delta_{i,j} \cdot v_m & L_2 \leftarrow L_2 - \frac{\sqrt{2}}{2} L_1 \end{cases}$$

Soit encore,

$$\begin{cases} d_{i,j}^m = -\frac{\sqrt{2}}{2} d_{i,j}^n + \delta_{i,j} \cdot v_m = -\sqrt{2} \delta_{i,j} \cdot v_n + 2 \delta_{i,j} \cdot v_m \\ d_{i,j}^n = 2 \delta_{i,j} \cdot v_n - \sqrt{2} \delta_{i,j} \cdot v_m \end{cases}$$

Finalement, on a

$$\begin{cases} d_{i,j}^m = 2 \delta_{i,j} \cdot v_m - \sqrt{2} \delta_{i,j} \cdot v_n \\ d_{i,j}^n = 2 \delta_{i,j} \cdot v_n - \sqrt{2} \delta_{i,j} \cdot v_m \end{cases}$$

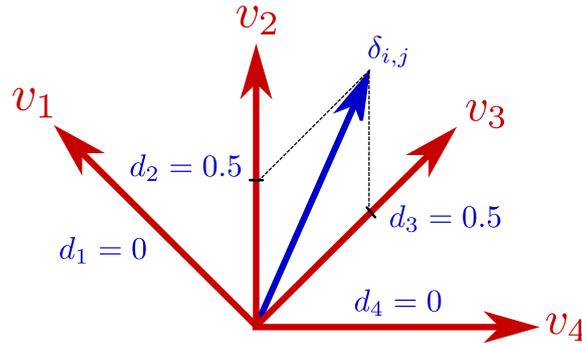


FIGURE 4.11 – Décomposition du vecteur direction au pixel  $(i, j)$ ,  $\delta_{i,j}$  sur chaque vecteur de la famille  $(v_1, v_2, v_3, v_4)$ .

### Algorithme FGP

Pour calculer l'opérateur proximal sous contrainte de la variation totale, nous avons simplement adapté l'algorithme FGP à notre gradient directionnel et les deux premières itérations de l'algorithme 4.3 deviennent alors :

$$\begin{aligned} u_{k+1} &= P_C(f - \lambda \operatorname{div}_D(p_k)) \\ p_{k+1} &= P(p_k + \tau \nabla_D u_{k+1}) \end{aligned}$$

où  $\nabla_D = D\nabla_d$  est le gradient directionnel avec  $\nabla_d$  calculé de la façon suivante :

$$\begin{aligned}
 (\nabla_d u)_{i,j}^1 &= u_{i-1,j-1} - u_{i,j} && \text{si } i, j > 0 \\
 (\nabla_d u)_{i,j}^2 &= u_{i-1,j} - u_{i,j} && \text{si } i > 0 \\
 (\nabla_d u)_{i,j}^3 &= u_{i-1,j+1} - u_{i,j} && \text{si } i > 0, j < N-1 \\
 (\nabla_d u)_{i,j}^4 &= u_{i,j+1} - u_{i,j} && \text{si } j < N-1 \\
 (\nabla_d u)_{i,j}^n &= 0 \quad \forall n \in \llbracket 1, 4 \rrbracket && \text{sinon}
 \end{aligned}$$

et  $\text{div}_D$  est la divergence discrète donnée par :

$$(\text{div } p)_{i,j} = p_{i,j}^1 - p_{i+1,j+1}^1 + p_{i,j}^2 - p_{i+1,j}^2 + p_{i,j}^3 - p_{i+1,j-1}^3 + p_{i,j}^4 - p_{i,j-1}^4$$

avec  $p_{N-1,N-1}^1 = p_{N-1,j}^2 = p_{N-1,0}^3 = p_{i,0}^4 = 0 \quad \forall i, j \in \llbracket 0, N-1 \rrbracket$ .

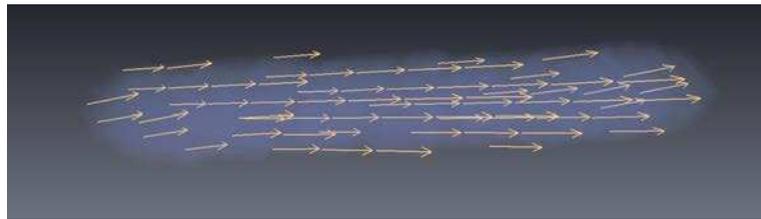
### 4.3.2.3 Résultats

La figure 4.13 compare les modèles Chan classique, Chan pondéré par la vesselness et Chan incluant les directions extraites de la vesselness sur la même image synthétique 2D bruitée que la figure 4.3. Ces résultats sont donnés en fonction du paramètre de régularisation  $\lambda$  prenant des valeurs croissantes. Les indicateurs (précision, sensibilité et spécificité) sont répertoriés dans le tableau 4.3. Avec une petite valeur de  $\lambda$ , les résultats pour les trois modèles sont équivalents, étant donné que lorsque  $\lambda$  est faible, les modèles sont dominés par le terme de fidélité qui est le même. Au fur et à mesure que  $\lambda$  augmente, la structure tubulaire fine disparaît dans le modèle de Chan classique, tandis que la pondération par la vesselness la préserve avec quelques déconnexions et le modèle de Chan incluant les directions extraites de la vesselness la préserve totalement.

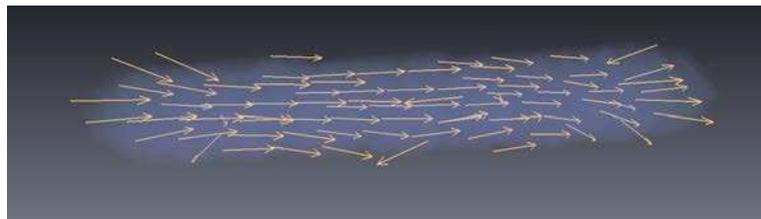
Afin d'analyser quantitativement notre dernier modèle, nous l'avons testé sur les 20 images de la base DRIVE. Les figures 4.14 et 4.15 comparent ce modèle, dont les directions sont extraites de la vesselness de Frangi, avec le modèle dont les directions sont extraites du filtre RORPO, appliqué respectivement à l'image 1 de la base DRIVE et à l'image 3. Dans [Merveille *et al.*, 2016b], il a été montré que la vesselness RORPO est une alternative robuste à la vesselness de Frangi, et les directions sont notamment plus précises (voir figure 4.12). Des zooms sur quelques extrémités des vaisseaux sanguins montrent que les deux modèles permettent de reconnecter des petits vaisseaux. De plus, en comparant les différents zooms des modèles directionnels, appliqués à l'image 3 (voir figure 4.15), avec le modèle pondéré par la vesselness (voir figure 4.7), on constate que les reconnexions sont plus complètes. Néanmoins, on peut remarquer quelques différences entre les modèles Chan + Frangi et Chan + RORPO : les extrémités semblent notamment mieux préservées pour le modèle Chan + RORPO (voir zooms 1 et 3 de la figure 4.14 et zoom 3 de la figure 4.15). Les indicateurs (sensibilité, spécificité, précision) sont répertoriés dans le tableau 4.4. On peut tout d'abord constater que les modèles Chan + Frangi et Chan + RORPO ont la même précision, ce qui signifie que la remarque précédente n'est pas une généralité et le modèle Chan + Frangi est parfois meilleur (voir zoom 2 de la figure 4.14). Étant donné que les reconnexions représentent seulement quelques pixels dans l'image, la précision de ces deux modèles directionnels (0,9434) a seulement légèrement augmenté par rapport aux modèles Chan classique (0,9416) et Chan pondéré par la vesselness (0,9423) (voir tableau 4.2).

Finalement, on peut noter que la précision de nos modèles ne dépasse pas les meilleures de la littérature. Cependant, notre but n'est pas de la dépasser mais de proposer un modèle qui reconnecte les vaisseaux, ce qui est effectivement le cas. On remarque également que le fait d'ajouter un a priori de direction robuste les résultats par rapport au choix de l'hyper-paramètre  $\lambda$ , ce qui permet de penser que les résultats obtenus seront en général meilleurs que ceux des méthodes concurrentes, dans le cas le plus réaliste où la vérité terrain n'est pas connue et où l'optimisation de cet hyper-paramètre n'est pas possible.

Le chapitre suivant dresse un bilan de nos contributions, et propose des idées pour améliorer notre dernier modèle directionnel.



(a) RORPO



(b) Vesselness de Frangi

FIGURE 4.12 – Comparaison des directions extraites de la vesselness de Frangi et du filtre RORPO sur une image représentant un tube.

$\lambda$		0,1	0,3	0,6	1
Modèle classique	Pr	<b>0,930</b>	0,933	0,935	0,937
	Se	0,558	0,552	0,533	0,513
	Sp	<b>0,974</b>	<b>0,977</b>	<b>0,982</b>	<b>0,986</b>
Modèle pondéré	Pr	0,930	<b>0,933</b>	0,935	0,936
	Se	0,558	0,555	0,543	0,517
	Sp	0,973	0,977	0,981	0,985
Modèle directionnel	Pr	0,923	0,933	<b>0,938</b>	<b>0,939</b>
	Se	<b>0,560</b>	<b>0,563</b>	<b>0,569</b>	<b>0,566</b>
	Sp	0,973	0,976	0,980	0,982

TABEAU 4.3 – Comparaison des indicateurs pour les modèles Chan classique, Chan pondéré par la vesselness, Chan incluant les directions extraites de la vesselness appliqués à une image synthétique 2D.

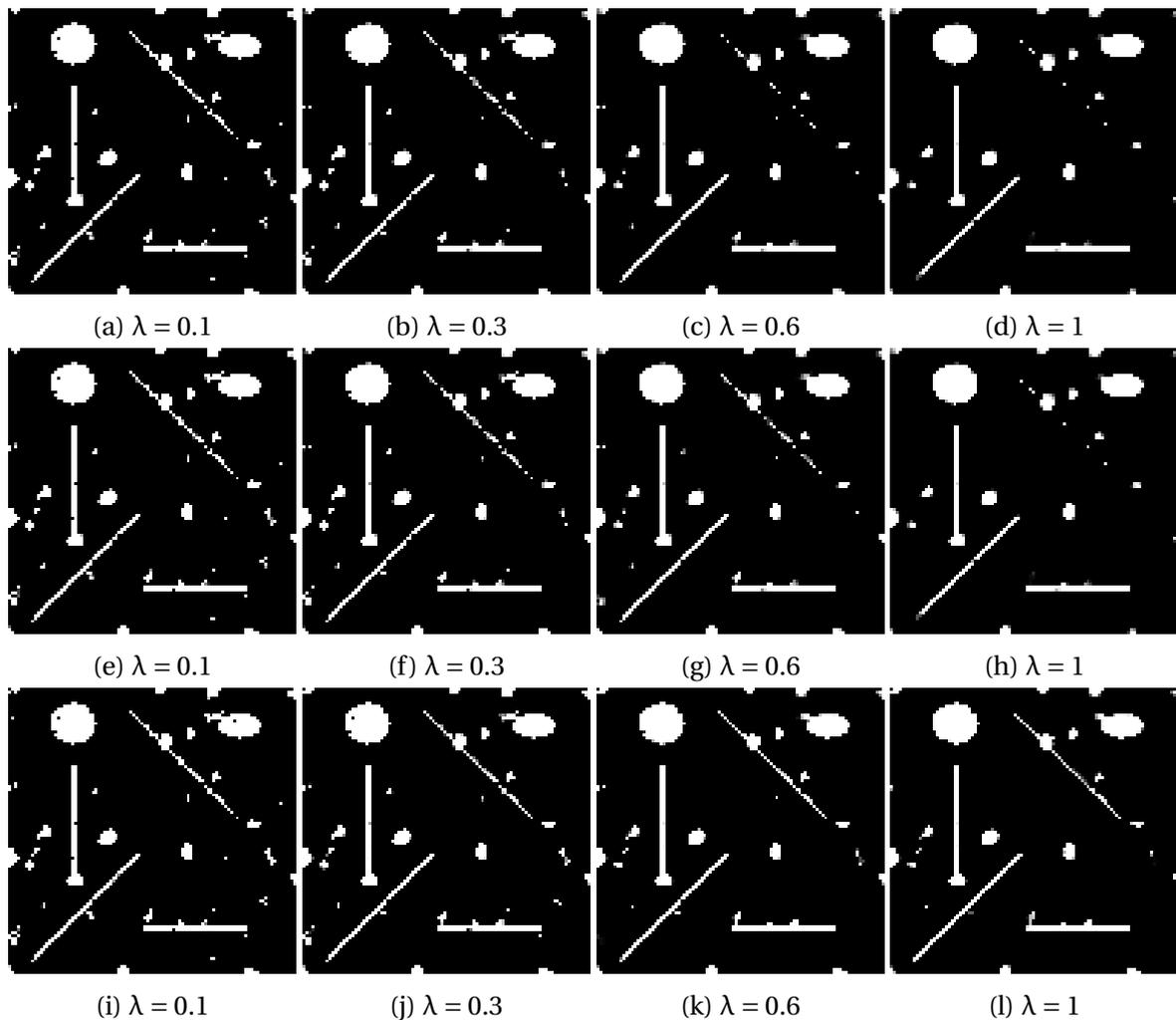


FIGURE 4.13 – Comparaison des modèles Chan classique (1<sup>e</sup> ligne), Chan pondéré par la vesselness (2<sup>e</sup> ligne), Chan incluant les directions extraites de la vesselness (3<sup>e</sup> ligne) appliqués à une image synthétique 2D, en fonction du paramètre de régularisation  $\lambda$  prenant des valeurs croissantes.

Méthode	Frangi	RORPO	Lupaşcu	Staal	Niemeijer	Zana
Sensibilité	0,6916	0,6690	0,6728	0,7193	0,6793	0,6696
Spécificité	0,9809	0,9810	0,9874	0,9773	0,9801	0,9769
Précision	0,9434	0,9434	0,9597	0,9442	0,9416	0,9377

TABEAU 4.4 – Comparaison des indicateurs pour les modèles Chan incluant les directions extraites de la vesselness de Frangi et Chan incluant les directions extraites de RORPO avec différentes méthodes de la littérature [Lupaşcu *et al.*, 2010, Staal *et al.*, 2004, Niemeijer *et al.*, 2004, Zana *et Klein*, 2001] (moyenne sur les 20 images).

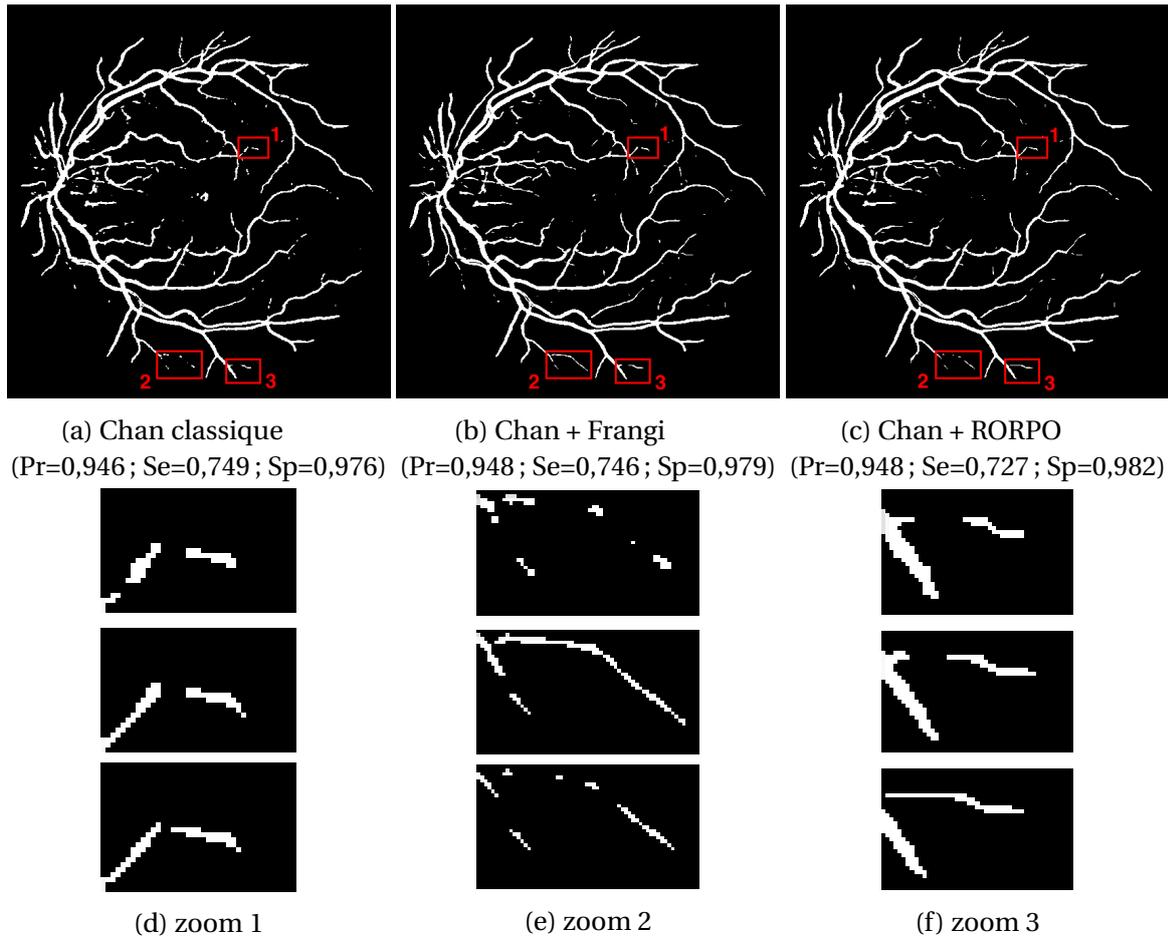


FIGURE 4.14 – Comparaison des modèles : (a) Chan classique, (b) Chan incluant les directions extraites de la vesselness de Frangi, et (c) extraites de RORPO, appliqués à une image rétinienne (image 1 de la base DRIVE). (d–f) Zoom sur les extrêmités de quelques vaisseaux sanguins pour les modèles Chan classique (en haut), Chan + Frangi (au milieu), et Chan + RORPO (en bas).

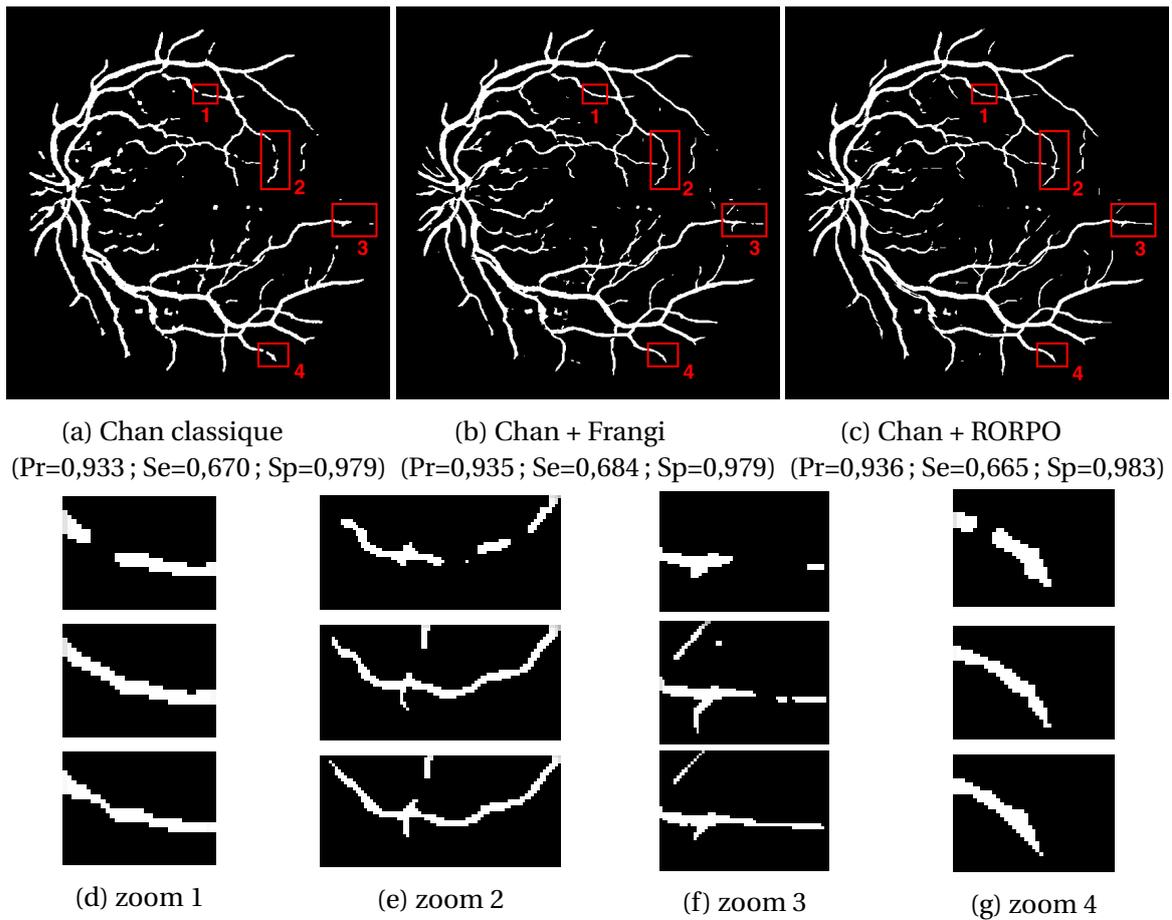


FIGURE 4.15 – Comparaison des modèles : (a) Chan classique, (b) Chan incluant les directions extraites de la vesselness de Frangi, et (c) extraites de RORPO, appliqués à une image rétinienne (image 3 de la base DRIVE). (d–g) Zoom sur les extrémités de quelques vaisseaux sanguins pour les modèles Chan classique (en haut), Chan + Frangi (au milieu), et Chan + RORPO (en bas).



# Chapitre 5

## Conclusion et perspectives

*« C'est en essayant encore et encore que le singe apprend à bondir. »*

---

Proverbe africain

Ce chapitre dresse, dans un premier temps, un bilan de nos contributions apportées à la segmentation d'images angiographiques : l'inclusion de la tubularité dans les modèles de débruitage ROF et TV-L1 et l'inclusion de la tubularité et des directions dans le modèle de segmentation de Chan-Vese. La deuxième section présente les perspectives pour améliorer ce dernier modèle de Chan-Vese : une première méthode avec des paramètres  $c_1$  et  $c_2$  spatialement variants et une seconde méthode avec un gradient directionnel à voisinage étendu. Des expérimentations sont fournies pour la première méthode. Enfin, nous décrirons comment adapter notre code pour la segmentation 3D.

### Sommaire

---

<b>5.1 Bilan</b> . . . . .	<b>96</b>
5.1.1 Inclusion de la tubularité dans les modèles de débruitage ROF et TV-L1 . . . . .	96
5.1.2 Inclusion de la tubularité et des directions dans les modèles de segmentation de Chan-Vese . . . . .	96
<b>5.2 Perspectives</b> . . . . .	<b>97</b>
5.2.1 Modèle de Chan avec les paramètres $c_1$ et $c_2$ spatialement variants . . . . .	97
5.2.2 Gradient directionnel à voisinage étendu . . . . .	98
5.2.3 Segmentation 3D . . . . .	99

---

## 5.1 Bilan

Nous allons rappeler brièvement les deux contributions apportées à la segmentation d'images angiographiques et discuter leurs limites.

### 5.1.1 Inclusion de la tubularité dans les modèles de débruitage ROF et TV-L1

Afin d'adapter la restauration d'images aux structures spécifiques que sont les structures tubulaires, nous avons proposé un premier modèle qui inclut la mesure de tubularité dans les modèles variationnels de débruitage ROF et TV-L1. La régularisation de ces deux modèles (i.e. la variation totale) n'est pas appropriée aux structures tubulaires puisqu'elle pénalise les contours. C'est pourquoi nous avons proposé de pondérer le terme de fidélité de ces modèles par la tubularité de Frangi. En incluant ainsi la tubularité, l'effet désiré est de régulariser davantage à l'extérieur des structures tubulaires plutôt qu'à l'intérieur, afin de préserver ces structures. Nous avons testé ces modèles sur une image rétinienne, et bien que ces modèles permettent de réhausser les vaisseaux dans l'image, ils ne permettent pas de les segmenter. Pour obtenir une segmentation, un seuillage est nécessaire, mais celui-ci provoque la disparition des vaisseaux les plus sombres. Nous avons alors décidé d'intégrer a priori de tubularité directement dans un modèle de segmentation pour éviter cette étape de seuillage.

### 5.1.2 Inclusion de la tubularité et des directions dans les modèles de segmentation de Chan-Vese

Le modèle de Chan-Vese est un modèle de segmentation qui permet de détecter la région d'intérêt dans les images composées de deux régions constantes : la région d'intérêt (i.e. les vaisseaux sanguins dans notre cas) et le fond. Dans un premier temps, nous avons inclus la tubularité de Frangi dans le modèle de Chan (qui est la version convexifiée du modèle de Chan-Vese), de la même façon que pour les modèles ROF et TV-L1. Nous avons testé ce modèle sur les 20 images de la base DRIVE et nous avons pu constater que ce modèle était plus performant que les modèles ROF et TV-L1. Néanmoins, il présente des déconnexions.

Dans un second temps, nous avons donc proposé un modèle incluant à la fois la tubularité et les directions des vaisseaux dans le modèle de Chan afin de pallier ce problème. Les directions sont extraites de la tubularité et sont intégrées dans le calcul du gradient, qui n'est plus un gradient classique mais un gradient directionnel. Nous avons comparé ce modèle avec les modèles Chan classique et Chan pondéré par la vesselness sur les 20 images de la base DRIVE. Nous avons pu constater que ce modèle fournit de meilleurs résultats en termes de précision et de reconnexion de vaisseaux. De plus, pour extraire les directions, nous avons comparé la vesselness de Frangi avec la vesselness de RORPO. Bien que quelques différences apparaissent sur les extrémités des petits vaisseaux sanguins, la précision est identique. Enfin, la précision de notre modèle ne dépasse pas les meilleures de la littérature, mais il permet de reconnecter les vaisseaux, et rend le choix de  $\lambda$  moins critique. Dans la suite, deux idées pour améliorer la précision sont proposées.

## 5.2 Perspectives

Nous allons décrire deux méthodes pour améliorer notre modèle : la première consiste à remplacer les paramètres constants  $c_1$  et  $c_2$  dans le modèle de Chan en des paramètres spatialement variants, et la seconde consiste à remplacer le gradient directionnel par un gradient directionnel à voisinage étendu. De premières expérimentations sont fournies pour la première méthode. Finalement, nous allons montrer qu'il est possible de développer notre modèle pour des images 3D.

### 5.2.1 Modèle de Chan avec les paramètres $c_1$ et $c_2$ spatialement variants

Rappelons tout d'abord que la fidélité dans le modèle de Chan [Chan *et al.*, 2006] est donnée par :

$$F = \int_{\Omega} \{(c_1 - f(x))^2 - (c_2 - f(x))^2\} u(x) dx$$

où  $c_1$  est la valeur de l'intensité moyenne dans la région d'intérêt et  $c_2$  la valeur de l'intensité moyenne dans le fond. Afin d'optimiser ces paramètres, on cherche à satisfaire les propriétés suivantes :

- $c_2 \ll f(x)$  et  $c_1 \approx f(x)$  dans la région d'intérêt
- $c_1 \gg f(x)$  et  $c_2 \approx f(x)$  dans le fond

Pour définir  $c_1$  et  $c_2$ , nous avons utilisé les deux principaux opérateurs de la morphologie mathématique (introduite brièvement dans le chapitre 3, section 3.2.5) : les ouvertures et les fermetures. Les ouvertures suppriment les petits objets clairs sur un fond sombre alors que les fermetures suppriment les petits objets sombres sur un fond clair. Ils sont duaux l'un de l'autre, c'est-à-dire qu'une fermeture fonctionne comme une ouverture si le contraste est inversé. Si l'on désigne par  $\gamma$  une ouverture et  $\varphi$  une fermeture, on a alors les propriétés suivantes :

- les ouvertures et fermetures sont *idempotentes* : pour toute image I,  $\gamma(\gamma(I)) = \gamma(I)$  et  $\varphi(\varphi(I)) = \varphi(I)$  (i.e. une ouverture ou une fermeture convergent en une itération, et il ne sert à rien de les utiliser plusieurs fois de suite) ;
- les ouvertures et fermetures sont *croissantes* : si une image I est plus lumineuse qu'une image J alors après soit une ouverture  $\gamma(I)$  ou une fermeture  $\varphi(I)$ , le résultat est plus lumineux que  $\gamma(J)$  ou  $\varphi(J)$  respectivement ;
- Les ouvertures sont *anti-extensives* : I est plus lumineuse que  $\gamma(I)$ . Inversement, les fermetures sont *extensives* :  $\varphi(I)$  est plus lumineuse que I.

Dans la pratique de la morphologie mathématique, on étudie souvent les images (binaires ou en niveaux de gris) à l'aide d'un ensemble B, appelé un élément structurant, que l'on connaît a priori, et que l'on adapte à nos besoins. L'ouverture par partie structurante détruit les parties "étroites" des objets, alors que la fermeture bouche les trous "étroits" ("étroit" est vu au sens de l'élément structurant).

Pour nos premières expérimentations sur l'image 1 de la base DRIVE, nous avons utilisé une ouverture  $\gamma$  et une fermeture  $\varphi$  par un élément structurant carré symétrique de taille  $11 \times 11$ . Les paramètres  $c_1$  et  $c_2$  sont alors calculés de la manière suivante :

$$\tilde{I} = I - \gamma(I) \tag{5.1}$$

$$c_1 = \varphi(\tilde{I}) \tag{5.2}$$

$$c_2 = \gamma(\tilde{I}) \tag{5.3}$$

L'équation (5.1) remplace le filtre médian et permet d'homogénéiser le fond (voir figure 5.1b). La fermeture, donnée par l'équation (5.2), est représentée sur la figure 5.1c. L'ouverture, correspondant à l'équation (5.3), est équivalente à  $c_2 = 0$  partout dans l'image. Nous avons testé ces paramètres sur les différents modèles de Chan classique, Chan pondéré par la vesselness et Chan incluant les directions de la vesselness (voir figures 5.1d, e, f). Nous avons utilisé les mêmes paramètres que dans le cas où  $c_1$  et  $c_2$  étaient constants. On peut constater que la sensibilité a augmenté, mais la spécificité a diminué (voir images 4.7 et 4.14). La meilleure précision est obtenue pour le modèle Chan directionnel. Il serait intéressant d'optimiser les paramètres  $\lambda$ , le seuil de la vesselness, ainsi que la taille de l'élément structurant pour se rendre compte de la pertinence du modèle Chan directionnel avec  $c_1$  et  $c_2$  variants.

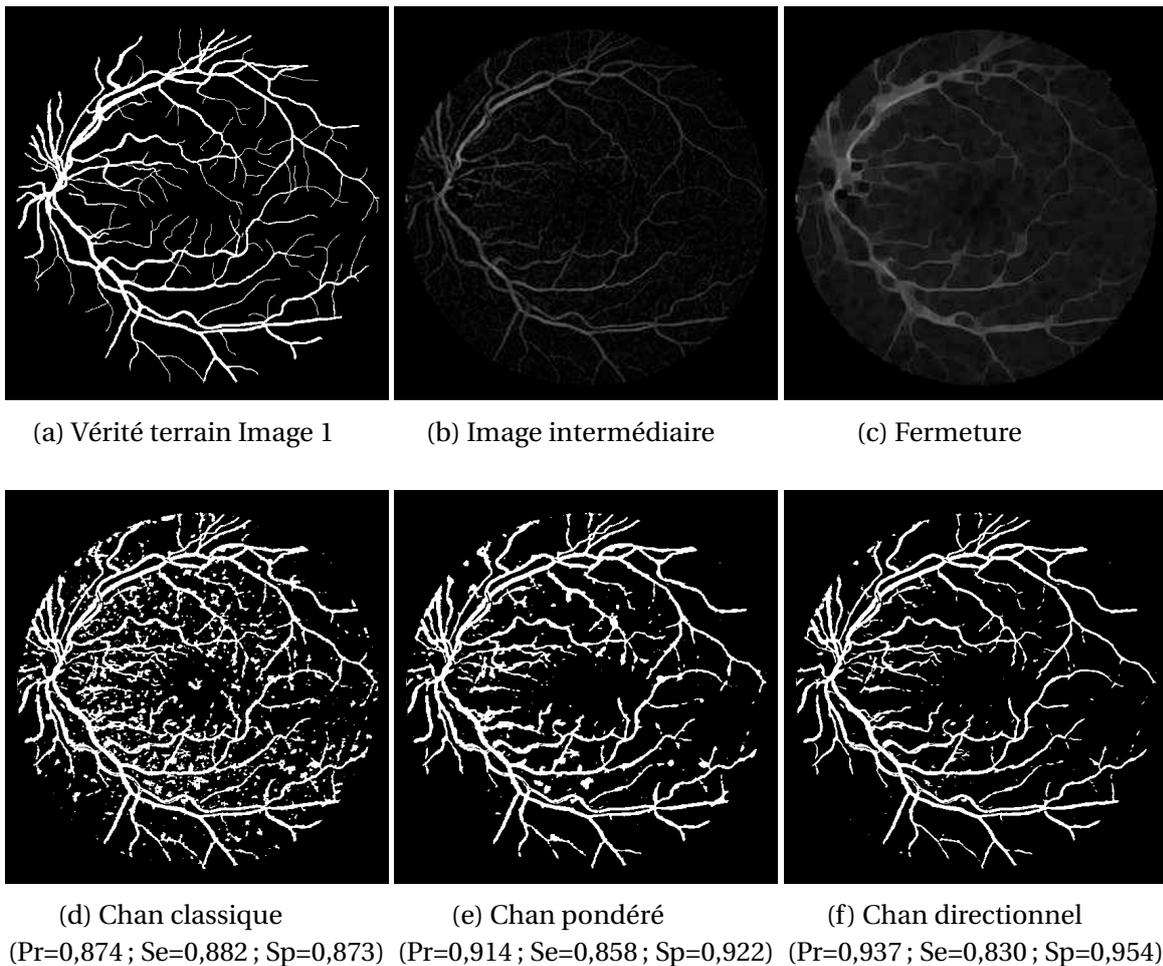


FIGURE 5.1 – Comparaison des modèles Chan classique, Chan pondéré par la vesselness et Chan incluant les directions de la vesselness avec les paramètres  $c_1$  et  $c_2$  spatialement variants (image 1 de la base DRIVE).

### 5.2.2 Gradient directionnel à voisinage étendu

De la même manière que l'on peut étendre le voisinage d'un pixel, il est possible d'étendre le gradient directionnel, qui était jusqu'à présent projeté sur une famille de vecteurs à un voisinage de  $n = 1$  pixel du pixel central, à  $n = 2$  ou  $n = 3$  pixels (voir figure 5.2). Dans le cas  $n = 2$  (respectivement  $n = 3$ ), le vecteur direction serait projeté sur les vec-

teurs adjacents en rouge, puis en bleu (respectivement en rouge, en bleu, puis en vert) et le coefficient final pourrait être une pondération par 1/2 (respectivement 1/3) de tous les coefficients. L'effet attendu serait alors d'obtenir un gradient directionnel plus précis et de supprimer les quelques déconnexions qui persistent dans notre modèle.

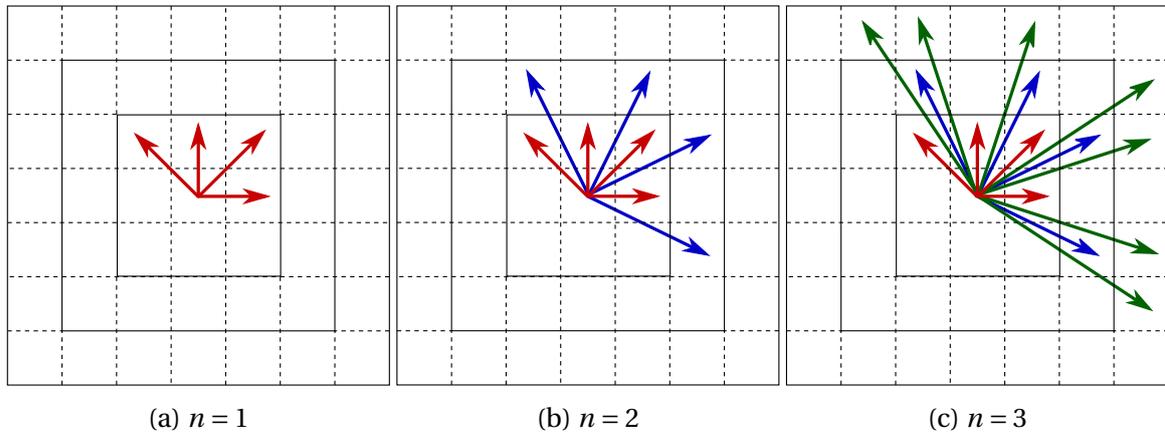


FIGURE 5.2 – Familles de vecteurs pour différentes tailles de voisinage  $n$  [Merveille *et al.*, 2016a].

### 5.2.3 Segmentation 3D

En ce qui concerne la segmentation 3D, des images synthétiques 3D fournies par le logiciel VascuSynth<sup>1</sup> permettent de valider les méthodes de segmentation en 3D (voir figures 5.3a, b). Les codes de la vesselness de Frangi et de RORPO existent déjà en 3D, il nous reste alors à coder le calcul du gradient directionnel en 3D en utilisant les 13 orientations possibles en 3D (voir figure 5.3c). Dans [Merveille, 2016], il est montré que les 7 directions (rouges et bleues) suffisent pour obtenir un filtre RORPO correct.

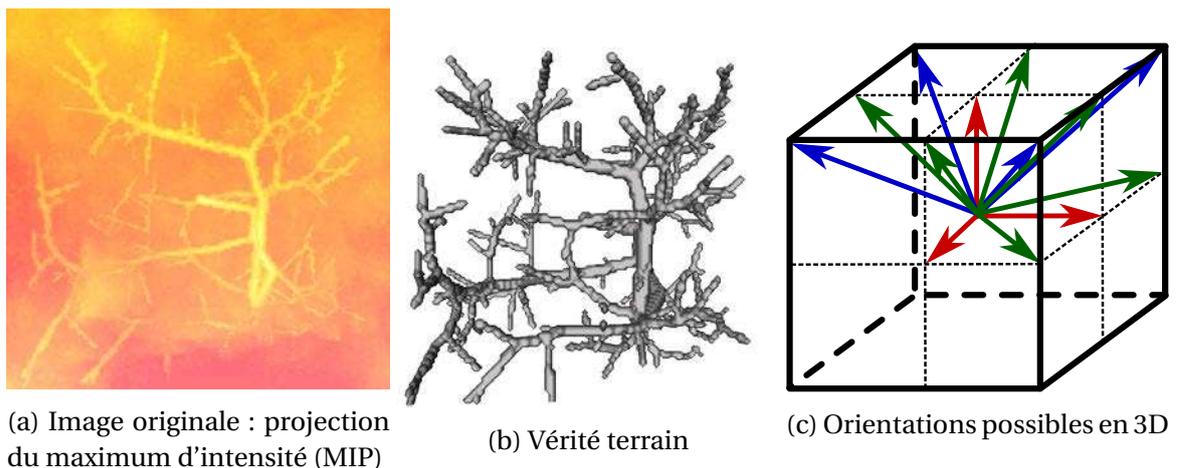


FIGURE 5.3 – Segmentation 3D : (a) image synthétique 3D fournie par le logiciel VascuSynth [Hamarneh et Jassi, 2010] (image corrompue par un bruit gaussien additif), (b) vérité terrain, (c) 13 orientations possibles en 3D : les 3 principales en rouge, les 4 diagonales en bleu et les 6 diagonales des carrés en pointillé en vert (représentant les 3 plans médians principaux) [Merveille, 2016].

1. <http://vascusynth.cs.sfu.ca>



**Deuxième partie**

**Simulation d'écoulements sanguins**



# Chapitre 6

## Modélisation pour la simulation d'écoulements sanguins cérébraux

« *Penser avec le cerveau, pas avec le sang.* »

---

Nelson Mandela

Ce chapitre fournit, en premier lieu, un aperçu des modèles utilisés pour la simulation d'écoulements sanguins cérébraux. Deux stratégies sont généralement considérées : les modèles 0D/1D, capables de traiter des réseaux complexes, et les modèles 3D, destinés uniquement à des échantillons de vaisseaux, mais permettant de capturer les structures locales contrairement aux modèles 0D/1D. L'objectif de notre travail est d'appliquer ces modèles 3D au réseau veineux cérébral, puis au polygone artériel de Willis. Dans un premier temps, la chaîne complète de la modélisation est présentée, partant de l'acquisition des images médicales et allant jusqu'à la simulation numérique. Ensuite, la vascularisation artérielle et veineuse cérébrale est décrite. Enfin, le choix du modèle biomécanique est discuté afin d'obtenir les équations adéquates qui seront résolues dans le chapitre 7, validées sur des solutions analytiques dans le chapitre 8 et appliquées à nos géométries réalistes dans le chapitre 9.

### Sommaire

---

<b>6.1 Introduction</b>	<b>104</b>
<b>6.2 Modélisation de la géométrie des vaisseaux</b>	<b>107</b>
<b>6.3 Modèle anatomique des réseaux veineux et artériel</b>	<b>108</b>
6.3.1 Réseau veineux cérébral	108
6.3.2 Réseau artériel cérébral	109
<b>6.4 Modèle biomécanique</b>	<b>112</b>
6.4.1 Hypothèses de travail	112
6.4.2 Analyse dimensionnelle	113
6.4.3 Conditions initiales et limites	114
6.4.3.1 Condition initiale	114
6.4.3.2 Conditions aux limites	114
6.4.4 Modèle final	116

---

## 6.1 Introduction

La vascularisation cérébrale est un réseau 3D très complexe, dont le rôle est de maintenir l'irrigation sanguine du cerveau. Ce réseau est composé de trois compartiments successifs : les artères, les capillaires et les veines. Les artères sont les vaisseaux sanguins qui transportent le sang oxygéné du cœur vers les tissus et les cellules de l'organisme. Les veines sont les vaisseaux chargés de ramener le sang pauvre en oxygène et les déchets du métabolisme vers le cœur. Les capillaires assurent la continuité entre les artères et les veines ; ils forment un réseau dense permettant l'irrigation des tissus. La majeure partie de la littérature actuelle sur la modélisation et la simulation d'écoulements sanguins cérébraux est focalisée sur les grosses artères cérébrales comme, par exemple, les artères carotides et le polygone de Willis [Formaggia *et al.*, 2009, et références incluses]. Par contre, le comportement du système veineux cérébral a été très peu étudié. Plusieurs difficultés interviennent, telles que le manque de parallélisme entre les circulations artérielle et veineuse [Schaller, 2004], le comportement asymétrique du réseau veineux [Schaller, 2004], et la variabilité inter-individuelle plus importante, par comparaison au réseau artériel [Stoquart-Elsankari *et al.*, 2009].

Durant cette dernière décennie, l'intérêt pour la simulation d'écoulements sanguins dans des modèles vasculaires a considérablement augmenté. En effet, outre la compréhension des phénomènes de mécanique du fluide dans le système cardiovasculaire, de telles simulations sont intéressantes pour plusieurs cas cliniques. Par exemple, elles peuvent, permettre le diagnostic et le suivi des anévrismes [Cebal *et al.*, 2005, Shojima *et al.*, 2014, Retarekar *et al.*, 2015], et prédire les effets de la pose d'endoprothèses (stents) [Wang *et al.*, 2011, Larrabide *et al.*, 2012] (voir. figure 6.1) ou de microspires (coils) [Morales *et al.*, 2013] (voir. figure 6.2). Les stents sont des petits tubes grillagés placés dans l'artère sténosée (rétrécie) et les coils sont des spires métalliques utilisées comme matériel d'occlusion endovasculaire d'un compartiment vasculaire relativement large (anévrisme). Plus généralement, les simulations peuvent fournir des informations sur l'hémodynamique à des endroits critiques des réseaux vasculaires tels que des jonctions de vaisseaux ou des bifurcations [Waechter *et al.*, 2008].

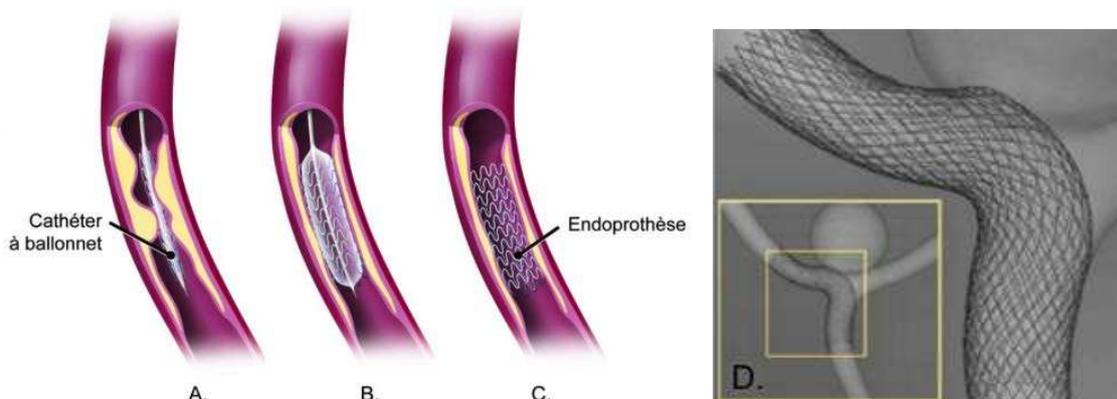


FIGURE 6.1 – Schéma de la pose d'un stent. A, cathétérisme de la lésion rétrécie. B, gonflement du ballon et ouverture du stent. C, détachement du stent et retrait du cathéter. D, modélisation d'un stent [Larrabide *et al.*, 2012].

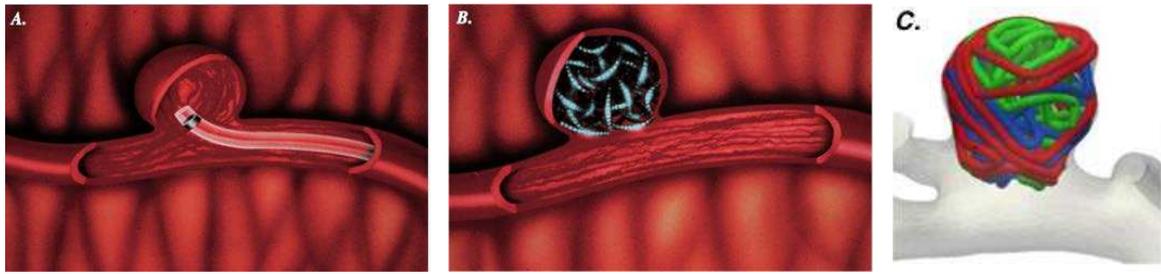


FIGURE 6.2 – Schéma de l'embolisation d'un anévrisme à l'aide de coils. A, cathétérisme du sac. B, détachement du coil et retrait du cathéter. C, modélisation d'un coil [Morales *et al.*, 2013].

Parallèlement au progrès des techniques d'imagerie médicale [Özsarlak *et al.*, 2004, Miyazaki *et Lee*, 2008], la mécanique des fluides numérique, plus souvent désignée par le terme anglais computational fluid dynamics (CFD), a évolué avec le développement de nouveaux schémas numériques et le calcul haute performance. Cependant, malgré les récents progrès dans la simulation numérique pour le vivant, des problèmes ouverts persistent. On peut notamment citer, parmi les nombreuses difficultés :

- la difficulté d'obtenir des géométries réalistes à partir d'images médicales ; les maillages qui en résultent doivent être traités pour permettre le processus de simulation tout en restant fidèle à la géométrie initiale ;
- les coûts de calculs de plus en plus élevés avec le raffinement des maillages ;
- la difficulté de modéliser précisément les hypothèses physiologiques multiples de ces phénomènes, telles que les conditions aux limites qui doivent mimer le reste du modèle et sont un champ de recherche à elles seules (problème de stabilité des schémas avec modèles 0D [Fouchet-Incaux, 2015], problème de “backflow” [Arbia *et al.*, 2016], problème de valeurs de résistance et autres paramètres [Moireau *et Chapelle*, 2011, Fabrèges, 2012], etc.) ;
- l'étape de validation, qui reste impossible à obtenir de manière non-invasive chez les patients. Les validations possibles sont alors la validation des algorithmes sur des solutions analytiques (cf. chapitre 8), la validation croisée entre plusieurs logiciels ou bibliothèques (cf. annexe B) et la validation “expérimentale” par simulation d'images IRM [Fortin *et al.*, 2016].

Dans le cadre de la simulation numérique d'écoulements sanguins, deux stratégies sont généralement considérées. La première stratégie consiste à utiliser des modèles d'ordre réduit, i.e. des modèles 0D à paramètres localisés, exploitant l'analogie avec le circuit électrique pour le circuit vasculaire, ou des modèles 1D à paramètres distribués, associés à de grands réseaux vasculaires : [Ursino *et Lodi*, 1997] (modèle 0D de l'hémodynamique intracrânienne), [Gadda *et al.*, 2015] (modèle 0D à paramètres localisés de l'écoulement veineux cérébral), [Reymond *et al.*, 2009] (modèle 1D du réseau artériel), [Ho *et al.*, 2013] (modèle 1D construit sur la géométrie anatomique du système veineux, cf. figure 6.3a) et [Müller *et Toro*, 2014b] (modèle 1D en circuit fermé de la circulation humaine, avec une attention particulière sur le système veineux). La forte diminution de la complexité spatiale permet alors aux solveurs numériques de gérer des réseaux vasculaires complexes, et de fournir un comportement précis, mais macroscopique, de l'écoulement sanguin global.

La seconde stratégie consiste à utiliser des modèles complets, c'est-à-dire à considérer la géométrie 3D des vaisseaux. Généralement, la grande taille des systèmes d'équations, induite par les modèles 3D, impose de se limiter à de petits échantillons du circuit vasculaire, éventuellement couplés avec des modèles 0D et 1D [Cebal *et al.*, 2005, Boissonnat *et al.*, 2005, Formaggia *et al.*, 2009] (cf. figure 6.3b). Des études récentes ont été proposées pour le réseau cérébral artériel : [Passerini *et al.*, 2009] (description 1D du polygone de Willis, couplée à un modèle complet 3D de l'artère carotide), [Xiao *et al.*, 2013] (modèle 3D/0D à l'intérieur du réseau artériel déformable du corps entier), et le réseau cérébral veineux : [Caiazzo *et al.*, 2015] (description 3D des veines jugulaires rigides, couplée à un modèle 1D pour les veines intracrâniennes principales), [Chabannes *et al.*, 2015] (modèle réaliste 3D pour les veines intracrâniennes, couplé à un modèle simplifié 0D pour la circulation sanguine en aval). Contrairement aux modèles réduits, les modèles 3D peuvent capturer les structures locales de l'écoulement, telles que la courbure des vaisseaux. L'inconvénient est que l'échantillonnage des vaisseaux induit un biais qui peut altérer l'information de l'écoulement [Castro *et al.*, 2006, Sato *et al.*, 2008].

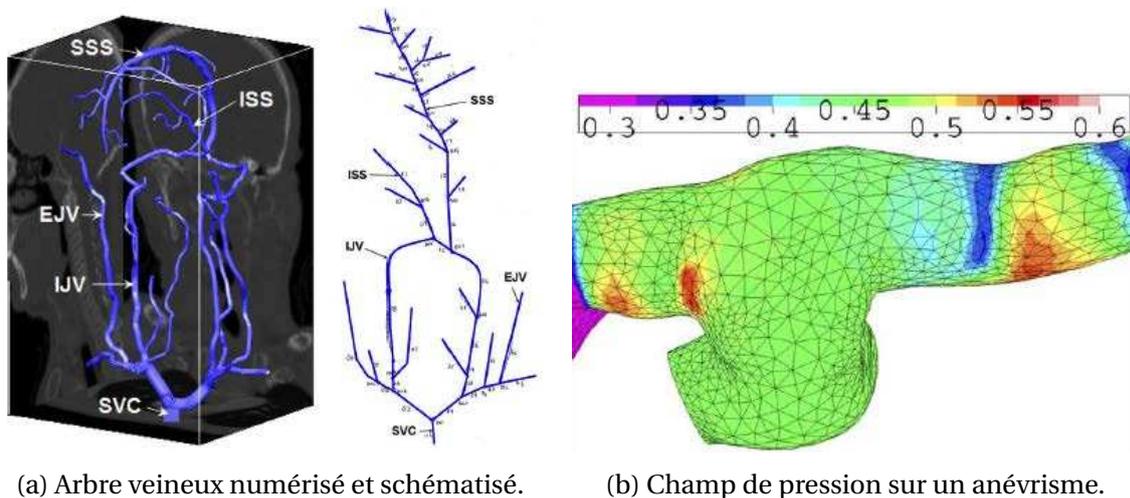


FIGURE 6.3 – Comparaison de modèles pour la simulation d'écoulements sanguins : (a), 1D [Ho *et al.*, 2013] et (b), 3D [Boissonnat *et al.*, 2005].

L'objectif de ce travail est d'appliquer des modèles 3D au réseau veineux cérébral complet, puis au polygone de Willis. La principale nouveauté réside dans une approche complète de la modélisation, partant de l'acquisition des images médicales et allant jusqu'à la résolution numérique. Cette étude suit trois directives principales :

- développer et utiliser des logiciels *open source*, à la fois en traitement du signal et en analyse numérique, afin d'avoir une gestion complète des méthodes sous-jacentes, et d'offrir des garanties en termes de disponibilité et de reproductibilité ;
- prouver l'exactitude numérique des solveurs développés sur des solutions analytiques, et fournir ainsi des garanties de qualité en termes de fiabilité, reproductibilité et incrémentalité ;
- considérer une modélisation de l'écoulement sanguin en accord avec la physique, dans le but d'obtenir des résultats réalistes et fiables.

La section 6.2 résume la chaîne permettant d'extraire une géométrie réaliste. La section 6.3 est consacrée au modèle anatomique du réseau veineux cérébral et du réseau arté-

riel, avec en particulier le polygone de Willis. Enfin, la section 6.4 présente le modèle biomécanique considéré dans ces deux géométries afin d'obtenir les équations adéquates à résoudre.

## 6.2 Modélisation de la géométrie des vaisseaux

La chaîne méthodologique allant de l'acquisition des images angiographiques 3D au modèle 3D de CFD est décrite en prenant l'exemple du réseau veineux cérébral (la chaîne étant identique dans le cas du polygone de Willis). La première étape consiste à obtenir les images 3D par ARM ; la résolution est alors suffisante à une échelle macroscopique sans injection de produit de contraste [Miyazaki et Lee, 2008], contrairement à l'ATC. Pour nos expériences, les images de volontaires sains ont été acquises sur un scanner corps entier (Siemens Magnetom Verio 3.0 T, pente de gradient =  $200T/m/s$ , séquence avec encodage des flux,  $TR = \text{Temps de Répétition} = 42,7ms$ ,  $TE = \text{Temps d'Echo} = 6,57ms$ , résolution =  $0,4 \times 0,4 \times 0,8mm^3$ ).

Premièrement, une méthode de segmentation [Dufour et al., 2013] est utilisée pour extraire le volume 3D du réseau veineux, constitué de blocs de voxels (contraction de "volumetric pixels"). Un algorithme de marching-cube est ensuite appliqué pour approcher la surface du volume sous forme d'un ensemble de triangles : le maillage 2D. Néanmoins, ces maillages ne peuvent pas être exploités immédiatement pour générer des maillages adaptés à la CFD 3D. En effet, des corrections locales de la géométrie sont requises pour supprimer les artéfacts tels que les pics à la surface du volume, les petites branches indésirables, les composantes déconnectées, etc. (voir figure 6.4, première ligne). Il est également nécessaire de labelliser les entrées et les sorties afin de pouvoir imposer les conditions aux limites du modèle mathématique, ainsi que de couper les sorties, perpendiculairement à l'axe du vaisseau, et extruder les sorties, dans l'axe du vaisseau (voir figure 6.4, deuxième ligne). La dernière étape d'extrusion permet d'éloigner l'influence des conditions aux limites et d'établir un régime permanent dans la région d'intérêt. Dans [Gambaruto et al., 2011], il est montré que la façon d'imposer les conditions aux limites en sortie a le plus d'influence sur l'écoulement.

Afin d'inspecter le maillage interactivement, ces tâches sont réalisées en utilisant à la fois le logiciel commercial de modélisation 3D BLENDER<sup>1</sup> et le code maison "Cutmesh" développé par l'un des collaborateurs du projet VIVABRAIN, Olivier Génévaux (Université de Strasbourg). Le premier logiciel est destiné à l'édition de maillages en général pour corriger les zones qui sont considérées comme problématiques, tandis que le deuxième est dédié à l'accomplissement de tâches spécifiques telles que la labellisation, la coupe et l'extrusion des entrées-sorties. Finalement, le maillage surfacique est lissé à l'aide du logiciel MMGS<sup>2</sup>, avant d'effectuer le maillage volumique avec le générateur MMG3D. La taille du maillage, i.e. le nombre de degrés de liberté, est notamment choisi à cette étape.

---

1. [www.blender.org](http://www.blender.org)

2. [www.ann.jussieu.fr/frey/software.html](http://www.ann.jussieu.fr/frey/software.html)

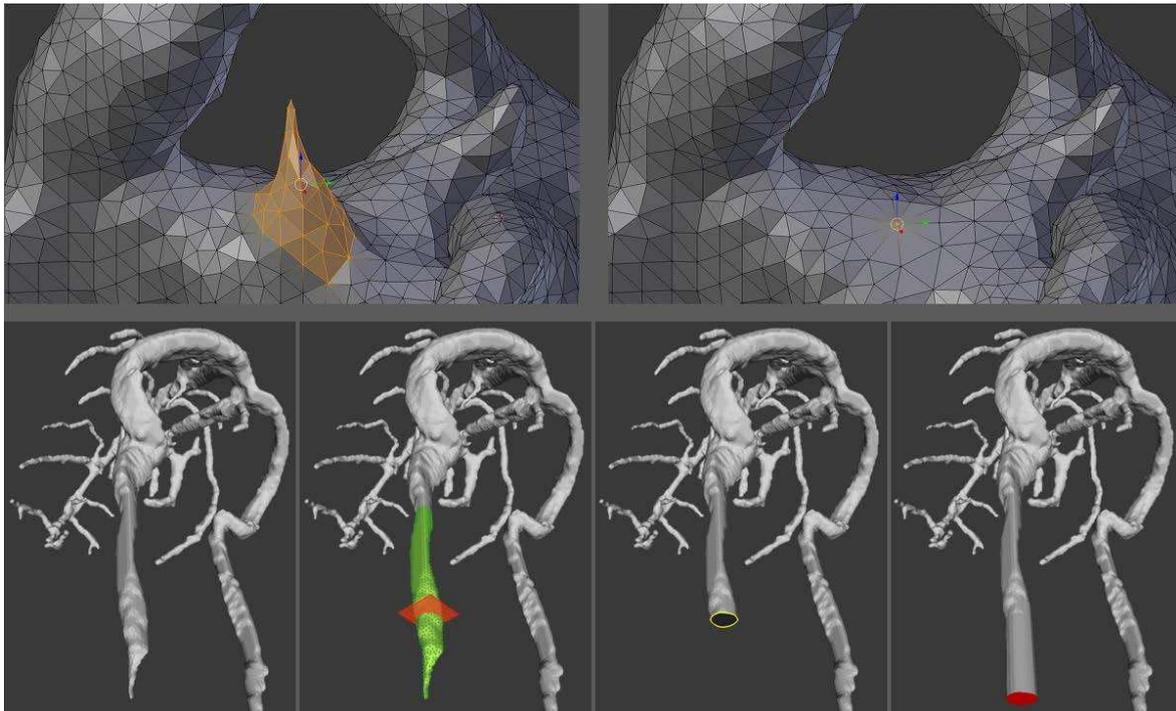


FIGURE 6.4 – 1<sup>re</sup> ligne : correction géométrique d'un maillage 2D. 2<sup>e</sup> ligne : traitement des entrées et des sorties (identification, coupe et extrusion).

## 6.3 Modèle anatomique des réseaux veineux et artériel

Même s'il ne représente que 2 % du poids total du corps (poids moyen de 1300 à 1400 grammes), le cerveau reçoit 15 % du débit cardiaque et consomme 20 % des apports en dioxygène. Une vascularisation cérébrale efficace est donc nécessaire pour assurer les apports en dioxygène et nutriments et éliminer les déchets du métabolisme.

### 6.3.1 Réseau veineux cérébral

Les principales veines cérébrales sont situées dans les leptoméninges et drainent le sang veineux du cerveau vers les veines jugulaires, ce qui assure le drainage extracrânien. Le réseau veineux cérébral est composé par des entrées (veines numérotées de 7 à 11 sur la figure 6.5) : la veine de Galien (7), la veine cérébrale interne (8), la veine basilaire (9), les veines cérébrales supérieures (10), les veines anastomotiques supérieures (11) ; drainant le sang dans le sinus sagittal supérieur (2) et le sinus droit (3), jusqu'à leur confluence (4). Le sang passe alors dans la portion transverse (5) et la portion sigmoïde (6) des sinus latéraux, et atteint une aire extracrânienne, composée des sorties : les veines jugulaires (1) [Schaller, 2004, et références incluses]. Dans cette partie du réseau veineux, les valves sont absentes, contrairement au système veineux des membres inférieurs, où un système complexe anti-retour appelé valvules veineuses empêche le sang de redescendre vers les pieds en raison de la gravité. On peut noter également que les deux veines numérotées (1) sur la figure 6.5 ne sont pas les "vraies" veines jugulaires ; ce sont les seules grosses veines visibles à l'image et le réseau secondaire est, lui, non visible. Néanmoins, ce réseau secondaire est utilisé uniquement dans la position debout (lorsque les jugulaires sont collabées), et les veines numérotées (1) pourraient correspondre aux veines jugulaires dans la position couchée (ce qui est le cas de nos images).

Des valeurs estimées des rayons de veines, obtenues à partir de notre géométrie du réseau veineux reconstruite à partir de l'ARM, sont comparées avec des valeurs de la littérature dans la table 6.1, avec un accent sur les valeurs utilisées dans les modèles 1D du réseau veineux cérébral. Ces résultats sont concordants, si l'on considère les variations inter-individuelles.

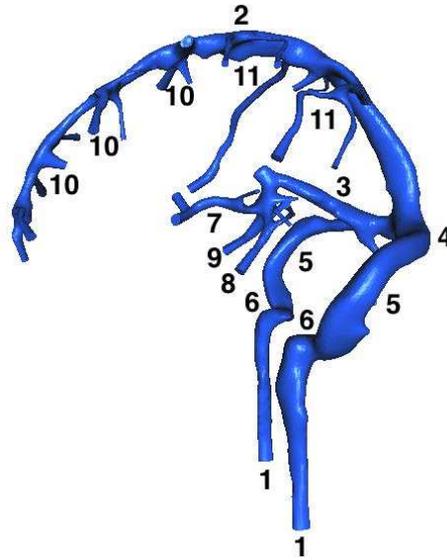


FIGURE 6.5 – Réseau veineux cérébral. 1, Veine jugulaire interne. 2, Sinus sagittal supérieur. 3, Sinus droit. 4, Confluence des sinus. 5, Sinus latéral (portion transverse). 6, Sinus latéral (portion sigmoïde). 7, Veine de Galien. 8, Veine cérébrale interne. 9, Veine basilaire. 10, Veines cérébrales supérieures. 11, Veines anastomotiques supérieures.

### 6.3.2 Réseau artériel cérébral

La vascularisation artérielle du cerveau est assurée par quatre artères : les deux *artères carotides internes* et les deux *artères vertébrales* (cf. figure 6.6). L'artère carotide gauche primitive naît directement de la crosse de l'aorte. L'artère carotide primitive droite est une branche de division du tronc brachio-céphalique droit, première branche artérielle naissant de la crosse aortique. Chaque artère carotide interne se divise en deux branches terminales au niveau du chiasma optique : d'une part l'artère cérébrale antérieure, qui passe dans la scissure inter-hémisphérique et irrigue les cortex moteur et sensitif correspondant aux membres inférieurs ; d'autre part l'artère cérébrale moyenne, qui pénètre dans la scissure de Sylvius, et irrigue la surface latérale des lobes frontaux, pariétaux et temporaux. Les artères vertébrales naissent des deux troncs axillaires ; elles ont un trajet particulier le long des vertèbres cervicales, traversant un canal osseux creusé dans l'apophyse latérale de chaque corps vertébral, et se réunissent dans le crâne pour donner le tronc basilaire.

Les deux systèmes carotidiens et vertébraux sont reliés par des ponts anastomotiques qui encadrent la selle turcique en dessinant un polygone appelé *polygone de Willis* (cf. figure 6.7) à partir duquel s'épanouissent les troncs artériels destinés à l'irrigation distincte des structures corticales et centrales. Il contribue en particulier à assurer une circulation cérébrale normale en cas d'oblitération de l'un des troncs carotidiens ou

vertébraux, d'origine physiologique ou pathologique (par exemple dans le cas d'accident vasculaire cérébral). Néanmoins, sa morphologie est variable et il n'est complet que chez 25 % des individus.

Pour estimer les rayons des artères de notre géométrie du polygone de Willis, on considère que les sections transversales des vaisseaux sont circulaires d'aire  $A$ , et on utilise alors la formule  $R = \sqrt{A/\pi}$ . Ces valeurs sont comparées avec des valeurs de la littérature dans la table 6.2. Les valeurs estimées sont légèrement plus élevées que les valeurs données par la littérature. Cependant, il est montré, dans [Albayrak *et al.*, 2007], que les aires des sections transversales des artères du cerveau varient en fonction de l'âge et du sexe.

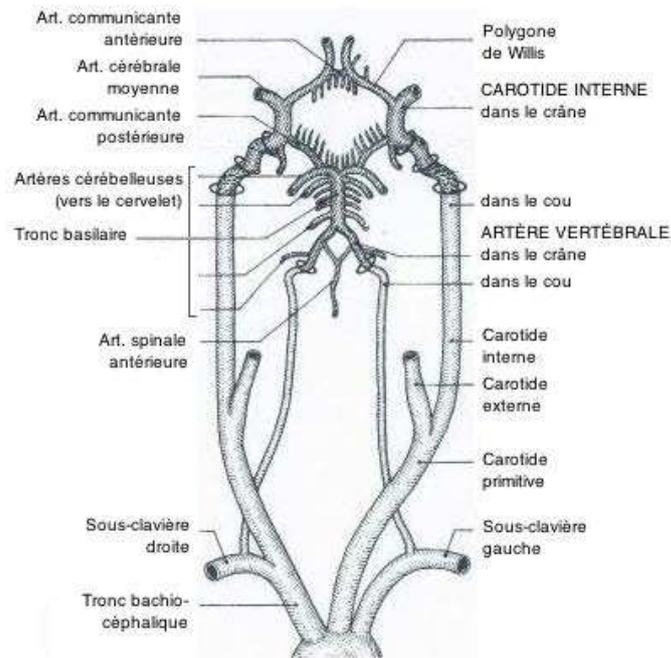


FIGURE 6.6 – Schéma des artères du cerveau [Cambier *et al.*, 2012].

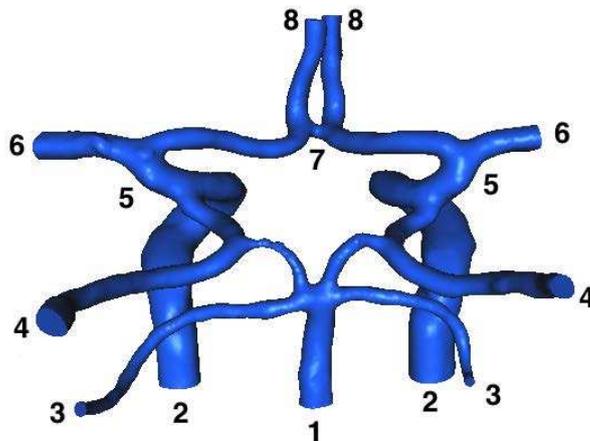


FIGURE 6.7 – Polygone de Willis. 1, Tronc basilaire. 2, Artère carotide interne. 3, Artère cérébelleuse supérieure. 4, Artère cérébrale postérieure. 5, Artère communicante postérieure. 6, Artère cérébrale moyenne. 7, Artère communicante antérieure. 8, Artère cérébrale antérieure.

Nom du vaisseau	Rayon (dans la littérature)	Rayon (ARM)
Veine jugulaire interne droite	3,0 → 7,0 [Ho <i>et al.</i> , 2013] 2,52 → 3,99 [Müller et Toro, 2014b]	2,59
Veine jugulaire interne gauche	3,0 → 7,0 [Ho <i>et al.</i> , 2013] 3,9 → 6,18 [Müller et Toro, 2014b]	2,34
Sinus sagittal supérieur	2,3 → 3,0 [Ho <i>et al.</i> , 2013] 2,0 → 3,19 [Müller et Toro, 2014b]	2,1 → 4,9
Sinus droit	1,8 [Ho <i>et al.</i> , 2013] 2,5 [Müller et Toro, 2014b]	3,3
Sinus latéral droit (portion transverse)	4 → 5 [Schaller, 2004] 1,78 → 2,52 [Müller et Toro, 2014b]	2,9
Sinus latéral gauche (portion transverse)	4 → 5 [Schaller, 2004] 3,09 → 4,37 [Müller et Toro, 2014b]	4,9
Sinus latéral droit (portion sigmoïde)	2,52 [Müller et Toro, 2014b]	3,8
Sinus latéral gauche (portion sigmoïde)	3,57 → 3,99 [Müller et Toro, 2014b]	5,2
Veine de Galien	3,09 → 4,0 [Müller et Toro, 2014b]	2,06 → 2,50
Veine cérébrale interne	1,26 [Müller et Toro, 2014b]	2,07
Veine basilaire	1,26 [Müller et Toro, 2014b]	2,03
Veines cérébrales supérieures	–	1,55 → 1,83
Veines anastomotiques supérieures	1,0 [Müller et Toro, 2014b]	1,07 → 1,86

TABLEAU 6.1 – Géométrie des veines : valeur approximative (*mm*) du rayon.

Nom du vaisseau	Rayon (dans la littérature)	Rayon (ARM)
Artère cérébrale antérieure	1,17 → 1,20 [Müller et Toro, 2014b]	2,58 → 2,62
Artère communicante antérieure	1 [Müller et Toro, 2014b]	2,36
Artère cérébrale moyenne	1,43 [Müller et Toro, 2014b]	3,02 → 3,24
Artère communicante postérieure	0,73 [Müller et Toro, 2014b]	1,85 → 3,80
Artère cérébrale postérieure	1,05 → 1,07 [Müller et Toro, 2014b]	2,03 → 2,31
Artère carotide interne	2,0 → 2,5 [Müller et Toro, 2014b] 4 [Formaggia <i>et al.</i> , 2009]	4,52 → 4,99
Tronc basilaire	1,62 [Müller et Toro, 2014b]	3,05
Artère cérébelleuse supérieure	–	0,94 → 1,16

TABLEAU 6.2 – Géométrie des artères : valeur approximative (*mm*) du rayon.

## 6.4 Modèle biomécanique

Dans cette section, l'objectif est de fournir un modèle approprié de l'écoulement sanguin dans les veines et les artères cérébrales de grande et moyenne tailles.

### 6.4.1 Hypothèses de travail

Les hypothèses suivantes sont adoptées : (1) le sang est incompressible ; (2) le sang est supposé homogène, étant donné que la taille des globules rouges (approximativement  $8\mu m$  de diamètre et  $2\mu m$  d'épaisseur au repos) est beaucoup plus petite que le calibre des vaisseaux ; (3) l'écoulement est isotherme ; (4) le sang est un fluide newtonien ; et (5) les parois des vaisseaux sont rigides. Les trois premières hypothèses sont des propriétés classiques de l'écoulement sanguin ; nous allons discuter le choix des deux dernières hypothèses.

Le sang est supposé newtonien dans la mesure où dans les vaisseaux de grande et moyenne tailles, les cellules ne se déforment pas. C'est pourquoi les caractéristiques thixotrope, viscoélastique et les effets de cisaillement sont négligés dans le contexte d'un circuit de drainage sain, qui reçoit un écoulement stationnaire et laminaire à son entrée. Le comportement newtonien du sang a déjà été considéré comme approprié dans des vaisseaux de grande et moyenne taille [Formaggia *et al.*, 2009, Thiriet, 2011], telles que ceux considérés dans notre travail (voir tables 6.1 et 6.2).

Une autre question importante porte sur la pertinence d'utiliser un modèle d'interaction fluide-structure (qui est une tâche particulièrement difficile du point de vue numérique) ou un modèle fluide seul (et alors négliger l'interaction avec la paroi des vaisseaux).

Les veines intracrâniennes qui passent à travers les leptoméniges – un tissu biologique quasi-incompressible et principalement composé d'eau – sont totalement contraintes entre le cerveau quasi-incompressible et le crâne rigide. Les veines ne sont alors pas susceptibles d'être fortement déformées. De plus, des analyses sur la sensibilité numérique [Sforza *et al.*, 2010] ont montré que négliger la déformation des parois est une hypothèse acceptable lorsque l'on étudie l'écoulement sanguin artériel intracrânien. Finalement, nous supposons dans ce travail que les veines et les artères sont rigides. Néanmoins, des études montrent que le compartiment extracrânien du drainage veineux est moins rigide que l'intracrânien [Stoquart-Elsankari *et al.*, 2009]. Des veines cervicales telles que le segment en amont de la veine jugulaire interne qui se situe le long du cou, près de l'artère carotide interne, subissent des déformations dues à la pression artérielle. Ces tensions passagères sont négligées puisque que nous désirons nous concentrer d'avantage sur l'hémodynamique intracrânienne qu'extracrânienne.

### 6.4.2 Analyse dimensionnelle

En conséquence de toutes ces hypothèses, nous considérons que les équations qui régissent l'écoulement isotherme et stationnaire d'un fluide incompressible (de masse volumique  $\rho$ , de viscosité dynamique  $\mu$ , et de viscosité cinématique  $\nu = \mu/\rho$ ) qui se déplace dans le vaisseau à une vitesse  $\mathbf{u}(\mathbf{x}, t)$  ( $\mathbf{x}$  : position eulérienne,  $t$  : temps), dérivent de la conservation de la quantité de mouvement. Elles sont connues comme les équations non-linéaires de Navier-Stokes :

$$\rho(\partial_t + \mathbf{u} \cdot \nabla)\mathbf{u} = \mathbf{f} + \nabla \cdot \mathbf{C} \quad (6.1)$$

$$\text{div} \mathbf{u} = 0 \quad (6.2)$$

où  $\mathbf{C} = -p\mathbf{I} + 2\mu\mathbf{D}$  est le tenseur de contraintes,  $p$  la pression du fluide,  $\mathbf{I}$  le tenseur métrique et  $\mathbf{D} = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)$  est le taux du tenseur des déformations pour un fluide newtonien. La seconde équation (6.2) traduit la condition d'incompressibilité. Dans des conditions isothermes, l'équation (6.1) conduit à la forme simplifiée des équations de Navier-Stokes en l'absence de forces extérieures  $\mathbf{f}$  :

$$\rho(\partial_t\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u}) = -\nabla p + \mu\Delta\mathbf{u}$$

Les équations de Navier-Stokes peuvent être transformées dans une forme adimensionnée, utilisant des échelles appropriées pour la longueur ( $L^*$ , i.e. l'épaisseur de la couche limite, ou le rayon hydraulique de la conduite pour un écoulement pleinement développé), le temps ( $T^*$ , i.e. la période de cycle du flux [ $T^* \equiv T$ ] ou l'inverse de la pulsation du flux [ $T^* \propto \omega^{-1}$ ]), la vitesse ( $U^*$ , i.e. la vitesse moyenne ou maximale dans la coupe transversale), et la pression ( $P^*$ ). On a donc :

$$\tilde{t} = t/T^*, \quad \tilde{x} = x/L^*, \quad \tilde{u} = u/U^*, \quad \tilde{p} = p/P^*,$$

et les différents opérateurs adimensionnés sont exprimés par  $\tilde{\nabla} = L^*\nabla$  et  $\tilde{\Delta} = L^{*2}\Delta$ .

On obtient alors la formulation classique des équations de Navier-Stokes :

$$\frac{\rho U^*}{T^*} \partial_{\tilde{t}} \tilde{\mathbf{u}} + \frac{\rho U^{*2}}{L^*} (\tilde{\mathbf{u}} \cdot \tilde{\nabla}) \tilde{\mathbf{u}} = -\frac{P^*}{L^*} \tilde{\nabla} \tilde{p} + \frac{\mu U^*}{L^{*2}} \tilde{\Delta} \tilde{\mathbf{u}} \quad (6.3)$$

Selon le type de force prédominant – soit des forces d'inertie convectives, soit des forces de viscosité – les coefficients adimensionnés peuvent être exprimés différemment.

Pour un écoulement stationnaire, le gradient de pression contrebalance les forces de frottements (et la gravité, quand elle a un effet considérable, comme dans les vaisseaux sanguins en position debout). Pour un écoulement quasi-stationnaire, le taux de variation des conditions aux limites est si faible que le moment a le temps de se diffuser pendant la période d'écoulement. Pour un écoulement instationnaire, les effets d'inertie locaux peuvent être prépondérants.

En multipliant l'équation (6.3) par  $\frac{L^*}{\rho U^{*2}}$  et en supprimant la notation tilde par souci de lisibilité, les équations de Navier-Stokes deviennent :

$$\frac{L^*}{U^* T^*} \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} = - \frac{P^*}{\mu U^{*2}} \nabla p + \frac{\mu}{\rho U^* L^*} \Delta \mathbf{u}$$

Puisque nous nous intéressons au cas d'une inertie convective prédominante, i.e.  $\frac{P^*}{\mu U^{*2}} \approx 1$ , on obtient finalement :

$$St \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} = - \nabla p + Re^{-1} \Delta \mathbf{u}$$

où  $Re = \frac{\rho L^* U^*}{\mu}$ ,  $St = \frac{\omega L^*}{U^*} = \frac{Sto^2}{Re}$  et  $Sto = L^* \sqrt{\frac{\omega}{\nu}}$  sont respectivement le nombre de Reynolds, le nombre de Strouhal et le nombre de Stokes.

Les paramètres mécaniques, utilisés pour calculer les paramètres adimensionnés, sont détaillés dans la table 6.3. Cependant, les données de l'écoulement sanguin spécifiques au patient ne sont généralement pas obtenues dans le cadre d'exams cliniques ordinaires. C'est pourquoi nous fournissons seulement des grandeurs de vitesse dans les plus gros vaisseaux (veines jugulaires, sinus sagittal supérieur, artère carotide interne, tronc basilaire) issues de rares données de la littérature. Les valeurs calculées pour les nombres de Reynolds, Stokes et Strouhal sont reportées dans la table 6.4. L'ordre de grandeur de  $Re^{-1}$  traduit l'importance des forces convectives, alors que la valeur de Strouhal, qui est du même ordre de grandeur, met en évidence l'importance de l'instationnarité de l'écoulement. En réunissant les hypothèses et les caractéristiques de l'écoulement précédentes, on aboutit aux équations instationnaires de Navier-Stokes, qui doivent être complétées avec des conditions initiales et limites appropriées.

### 6.4.3 Conditions initiales et limites

#### 6.4.3.1 Condition initiale

Le statut initial de la vitesse du fluide,  $\mathbf{u}_0$  doit être soigneusement choisi, puisqu'il doit être à divergence nulle pour être admissible. Malheureusement, dans les calculs hémodynamiques,  $\mathbf{u}_0$  est généralement inconnu, et alors choisi égal à zéro partout ou, en meilleure estimation, comme étant la solution du problème stationnaire de Stokes.

#### 6.4.3.2 Conditions aux limites

##### Réseau veineux cérébral

Le flux d'entrée est situé à la sortie de la microcirculation, loin du thorax et donc loin de la respiration et de la pompe cardiaque [Thiriet, 2008]. Puisque l'écoulement n'est pas pleinement développé, les écoulements de Poiseuille (cas stationnaire) et de

Paramètre		Valeur	Unité	Source
Masse volumique	$\rho$	1055	$[\frac{kg}{m^3}]$	[Thiriet, 2008]
Viscosité dynamique	$\mu$	$3,5 \cdot 10^{-3}$	$[\frac{kg}{m \cdot s}]$	[Thiriet, 2008]
Viscosité cinématique	$\nu$	$(2,8-3,8) \cdot 10^{-6}$	$[\frac{m^2}{s}]$	[Thiriet, 2008]
Pulsation du flux	$\omega$	1	$[\frac{1}{s}]$	[Thiriet, 2008]
Vitesse dans la coupe transversale (études cliniques, veines jugulaires)	$V^*$	$(10-11) \cdot 10^{-2}$ $(8,5-11,3) \cdot 10^{-2}$ $(30-50) \cdot 10^{-2}$	$[\frac{m}{s}]$	[Ho <i>et al.</i> , 2013] [Ogoh <i>et al.</i> , 2011] [Schaller, 2004]
Vitesse dans la coupe transversale (études cliniques, sinus sagittal supérieur)	$V^*$	$15 \cdot 10^{-2}$	$[\frac{m}{s}]$	[Schaller, 2004]
Vitesse dans la coupe transversale (études cliniques, artère carotide interne)	$V^*$	$(20-40) \cdot 10^{-2}$ $(16-80) \cdot 10^{-2}$ $(30-78) \cdot 10^{-2}$	$[\frac{m}{s}]$	[DeVault <i>et al.</i> , 2008] [Albayrak <i>et al.</i> , 2007] [Meckel <i>et al.</i> , 2013]
Vitesse dans la coupe transversale (études cliniques, tronc basilaire)	$V^*$	$(30-78) \cdot 10^{-2}$	$[\frac{m}{s}]$	[Meckel <i>et al.</i> , 2013]

TABLEAU 6.3 – Valeurs des paramètres mécaniques dans le modèle.

Nom du vaisseau	Nombre de Reynolds	Nombre de Stokes	Nombre de Strouhal
Veine jugulaire interne	75 → 603	1,38 → 2,20	0,008 → 0,030
Sinus sagittal supérieur	90 → 144	1,10 → 1,75	0,013 → 0,021
Artère carotide interne	96 → 1182	1,10 → 2,20	0,004 → 0,012
Tronc basilaire	147 → 717	0,89 → 1,67	0,004 → 0,005

TABLEAU 6.4 – Valeurs des paramètres adimensionnés.

Womersley (cas instationnaire) ne sont pas adaptés. Une vitesse uniforme est alors injectée en entrée. Cependant, une analyse rigoureuse des équations de Navier-Stokes requiert une régularité de la solution à  $t = 0$ ; donc cette condition doit être respectée le plus harmonieusement possible. En pratique, un multiplicateur dépendant du temps de la forme  $(1 - e^{-kt})$ , avec  $k$  constant, est introduit pour assurer que le flux d'entrée augmente régulièrement de zéro à la condition limite désirée (état stationnaire, i.e. fonction constante  $\mathbf{u}_{in}$ ) lorsque  $t \rightarrow \infty$ . Cette condition sur la vitesse en entrée est imposée aux 29 entrées du maillage. La valeur de  $\mathbf{u}_{in}$  est choisie en s'appuyant sur les vitesses de la table 6.3 et le paramètre  $k$  est choisi tel que la valeur de l'état stationnaire est atteinte au tiers du premier cycle cardiaque.

Comme les images sont acquises couchées, le principal drainage, c'est-à-dire les sorties vasculaires correspondent aux veines jugulaires. Il est à noter que dans la position debout, sous l'influence de la gravité, le drainage du sang varie [Gisolf *et al.*, 2004, Stoquart-Elsankari *et al.*, 2009]. Des chemins collatéraux alternatifs pour le drainage veineux devraient ainsi être inclus pour simuler plusieurs observations expérimentales. Par conséquent, en première approximation, nous utilisons une condition aux limites de sortie libre et nous nous focalisons sur l'hémodynamique intracrânienne pour minimiser l'influence de la section à la sortie. Pour des conditions aux limites plus réalistes, on peut se référer à un autre travail réalisé dans le projet VIVABRAIN [Chabannes *et al.*, 2015]. En résumé, les conditions aux limites que l'on impose sont les suivantes : (i) un profil de vitesse uniforme lissé en entrée, (ii) des conditions homogènes naturelles à la sortie, et (iii) une condition de non-glissement sur la paroi, supposée rigide.

### Réseau artériel cérébral

En ce qui concerne les entrées, nous possédons des profils de vitesse périodisés pour la carotide interne gauche, la carotide interne droite et le tronc basilaire. Ces profils ont été obtenus par des mesures d'IRM au laboratoire de Bio Flow Image à Amiens, à l'aide d'un logiciel dédié FLOW<sup>3</sup>. Pour ce qui est des sorties, nous considérons une condition de Neuman avec dans un premier temps, une condition homogène (sortie libre). Dans un second temps, on pourrait utiliser une condition non homogène avec un modèle 0D de type Windkessel [Vignon-Clementel *et al.*, 2006], qui imite la résistance du réseau secondaire. Finalement, nous imposons des conditions aux limites mêlées avec : (i) une condition de Dirichlet non homogène en entrée, (ii) une condition de Neuman en sortie, et (iii) une condition de non-glissement sur la paroi.

#### 6.4.4 Modèle final

Le modèle mathématique, que nous considérons dans la suite, est le problème de Navier-Stokes avec des conditions aux limites mixtes :

$$\left\{ \begin{array}{ll} \partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} & \text{dans } \Omega \times (0, T) \\ \operatorname{div} \mathbf{u} = 0 & \text{dans } \Omega \times (0, T) \\ \mathbf{u} = \mathbf{g}_1 & \text{sur } \Gamma_D \times (0, T) \\ \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} = \mathbf{g}_2 & \text{sur } \Gamma_N \times (0, T) \\ \mathbf{u}|_{t=0} = \mathbf{u}_0 & \text{dans } \Omega \end{array} \right. \quad (6.4)$$

où  $\Omega \subset \mathbb{R}^3$  est un domaine borné avec une frontière lipschitzienne  $\Gamma$ ,  $(0, T)$  est un intervalle de temps fini. On rappelle que  $\mathbf{u} = (u_1, u_2, u_3)$  est la vitesse du fluide,  $p$  sa pression et  $\nu$  sa viscosité cinématique. On suppose que  $\Gamma$  est constitué de deux parties mesurables :  $\Gamma_D$ , où les conditions aux limites de Dirichlet sont imposées, et  $\Gamma_N$ , où les conditions aux limites de Neumann sont prescrites. On note  $\mathbf{n}$  le vecteur unitaire normal à  $\Omega$  sur sa frontière  $\Gamma$ , dirigé vers l'extérieur. Les conditions aux limites de Dirichlet correspondent, dans le cas non-homogène, à une vitesse  $\mathbf{g}_1$  (imposée généralement aux entrées) et, dans le cas homogène, à une condition limite de non-glissement prescrite sur la paroi du vaisseau ; la fonction  $\mathbf{g}_2$  correspond à la composante normale du tenseur des contraintes (imposée aux sorties et fixée à zéro, dans un premier temps). Le prochain chapitre décrit les deux méthodes que nous avons utilisées pour résoudre le problème de Navier-Stokes (6.4).

3. <http://www.tidam.fr/software.htm>

# Chapitre 7

## Méthodes de résolution des équations de Stokes et Navier-Stokes

*« L'algorithme est au mathématicien ce que l'échelle de corde est à l'alpiniste. »*

Pierre Meunier

Ce chapitre décrit les méthodes que nous avons utilisées pour résoudre les équations de Navier-Stokes aux conditions limites mêlées, qui régissent notre modèle d'après le chapitre 7. Ces méthodes sont des méthodes d'éléments finis, qui consistent à affaiblir la condition d'existence des solutions du problème initial à l'aide d'une formulation variationnelle ou formulation faible, puis à discrétiser cette formulation à l'aide d'éléments finis afin d'obtenir une solution approchée. En première approximation, nous allons résoudre le problème de Stokes, dans lequel les termes inertiels sont absents. En ce qui concerne les équations de Navier-Stokes, deux méthodes sont décrites : la première est la méthode classique des caractéristiques, qui traite le terme non-linéaire convectif (une accélération par l'algorithme d'Uzawa est également détaillée), tandis que la seconde méthode est un schéma découplé d'ordre élevé qui traite la vitesse et la pression séparément.

### Sommaire

---

<b>7.1</b>	<b>Introduction</b>	<b>118</b>
<b>7.2</b>	<b>Résolution du problème de Stokes</b>	<b>119</b>
7.2.1	Formulation variationnelle	119
7.2.2	Discrétisation	120
7.2.2.1	Approximation de Galerkin	120
7.2.2.2	Espaces de Taylor-Hood	121
<b>7.3</b>	<b>Résolution du problème de Navier-Stokes</b>	<b>122</b>
7.3.1	Méthode des caractéristiques	122
7.3.1.1	Formulation variationnelle	123
7.3.1.2	Discrétisation en espace	124
7.3.2	Méthode d'Uzawa	124
7.3.2.1	Préconditionnement de Cahouet et Chabard	126
7.3.3	Schéma découplé d'ordre élevé	126
7.3.3.1	Discrétisation en temps	127
7.3.3.2	Formulation variationnelle	128
7.3.3.3	Discrétisation en espace	129

---

## 7.1 Introduction

Dans ce chapitre, les équations de Stokes et Navier-Stokes ont été résolues à l'aide de la méthode des éléments finis (MEF). La MEF est une méthode d'approximation numérique de solutions de problèmes d'équations aux dérivées partielles (EDP). À partir du problème de départ, l'objectif est de trouver une formulation variationnelle équivalente, dont on calcule une approximation de la solution en la projetant sur un espace de dimension finie, ce qui revient à résoudre au final un système linéaire. Historiquement, l'origine de la méthode peut se trouver dans le calcul variationnel (voir chapitre 1), mais il faut attendre le début du XX<sup>e</sup> siècle pour définir la méthode avec la méthode de Galerkin, qui consiste à utiliser un maillage du domaine d'étude, et considérer la restriction de la solution sur chacune des mailles. Aujourd'hui, l'ingénierie utilise la méthode des éléments finis dans de nombreuses applications, que ce soit en mécanique des fluides ou des solides avec par exemple l'industrie pétrolière, l'industrie automobile, la biomécanique et l'aéronautique (voir figure 7.1), mais aussi pour des problèmes thermiques, électromagnétiques, chimiques, physiques, astrophysiques, etc.

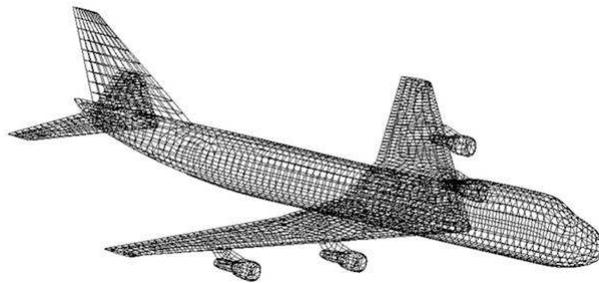


FIGURE 7.1 – Structure de l'avion B-747 et maillage produit par la MEF [Varpasuo, 2011].

En première approximation des équations de Navier-Stokes, les équations de Stokes sont résolues dans la section 7.2. Dans cette section, on définit la triangulation (le maillage) du domaine, ainsi que les éléments finis qui constituent l'espace de dimension finie. Ces éléments finis seront réutilisés dans toute la suite de ce chapitre. La section 7.3.1 décrit la méthode classique des caractéristiques, qui traite le terme convectif non linéaire des équations de Navier-Stokes. Une accélération de cette méthode est possible à l'aide de l'algorithme d'Uzawa présenté dans la section 7.3.2. La principale difficulté de la résolution de ces équations est le traitement de la pression qui n'apparaît que par son gradient. Ce terme de pression intervient dans l'équation de conservation de la quantité de mouvement comme un multiplicateur de Lagrange assurant que la vitesse vérifie la condition d'incompressibilité. Une manière de contourner le couplage vitesse-pression des équations de Navier-Stokes consiste à utiliser des schémas découplés tels que la méthode de projection. Cette méthode, introduite indépendamment par Chorin [Chorin, 1968] et Temam [Temam, 1969], sépare le problème de Navier-Stokes en deux sous-problèmes : le premier consiste à chercher une prédiction de la vitesse qui ne vérifie pas forcément la condition d'incompressibilité, puis le deuxième corrige cette vitesse et calcule le champ de pression associé. Malheureusement cette méthode présente une précision limitée à cause de la condition de Neumann artificielle. Le développement des méthodes de projection a permis de pallier ce problème (voir [Guermond *et al.*, 2006] pour un état de l'art complet). Dans la section 7.3.3, nous allons décrire un schéma proposé récemment dans [Sheng *et al.*, 2013], qui combine les avantages des schémas découplés avec la possibilité d'utiliser des conditions aux limites mêlées, qui sont nécessaires à nos cas réalistes.

## 7.2 Résolution du problème de Stokes

Soit  $\Omega \subset \mathbb{R}^d$  ( $d = 2$  ou  $3$ ) un domaine borné avec une frontière lipschitzienne  $\Gamma$ . On suppose que  $\Gamma$  est constitué de deux parties mesurables :  $\Gamma_D$  et  $\Gamma_N$ . On note  $\mathbf{n}$  le vecteur unitaire normal à  $\Omega$  sur sa frontière  $\Gamma$ . On considère le problème de Stokes, avec des conditions aux limites mêlées, suivant :

$$\left\{ \begin{array}{ll} -\nu \Delta \mathbf{u} + \nabla p = \mathbf{f} & \text{dans } \Omega \\ \operatorname{div} \mathbf{u} = 0 & \text{dans } \Omega \\ \mathbf{u} = \mathbf{g}_1 & \text{sur } \Gamma_D \\ \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} = \mathbf{g}_2 & \text{sur } \Gamma_N \end{array} \right. \quad \begin{array}{l} (7.1) \\ (7.2) \\ (7.3) \\ (7.4) \end{array}$$

où  $\mathbf{u} = (u_1, u_2, u_3)$  est la vitesse du fluide,  $p$  sa pression,  $\nu$  sa viscosité cinématique et  $\mathbf{f}$  les forces extérieures.

Les différents opérateurs sont définis par :

$$\Delta \mathbf{u} = \sum_{i=1}^d \frac{\partial^2 u_i}{\partial x_i^2} \in \mathbb{R}^d, \quad \nabla p = \left( \frac{\partial p}{\partial x_1}, \dots, \frac{\partial p}{\partial x_d} \right) \in \mathbb{R}^d, \quad \operatorname{div} \mathbf{u} = \sum_{i=1}^d \frac{\partial u_i}{\partial x_i} \in \mathbb{R} \quad (7.5)$$

### 7.2.1 Formulation variationnelle

On utilise la méthode des éléments finis pour résoudre les équations (7.1)-(7.2). Pour cela, on introduit les espaces des fonctions tests  $V = (H_0^1(\Omega))^d = \{\mathbf{v} \in (H^1(\Omega))^d \mid \mathbf{v}|_{\Gamma_D} = 0\}$  pour la vitesse (avec  $H^1(\Omega) = \{v \in L^2(\Omega) \mid \nabla v \in (L^2(\Omega))^d\}$ ), et  $Q = L^2(\Omega)$  pour la pression.

On multiplie l'équation (7.1) par  $\mathbf{v} \in V$  et l'équation (7.2) par  $q \in Q$ , et on intègre les deux équations sur  $\Omega$  :

$$\begin{aligned} -\nu \int_{\Omega} \Delta \mathbf{u} \mathbf{v} + \int_{\Omega} \nabla p \mathbf{v} &= \int_{\Omega} \mathbf{f} \mathbf{v} \\ \int_{\Omega} \operatorname{div} \mathbf{u} q &= 0 \end{aligned}$$

On applique la formule de Green :

$$\begin{aligned} \nu \int_{\Omega} \nabla \mathbf{u} \cdot \nabla \mathbf{v} - \int_{\Omega} p \operatorname{div} \mathbf{v} &= \int_{\Omega} \mathbf{f} \mathbf{v} + \int_{\Gamma_D} \left( \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \cdot \mathbf{n} \right) \mathbf{v} + \int_{\Gamma_N} \left( \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \cdot \mathbf{n} \right) \mathbf{v} \\ \int_{\Omega} \operatorname{div} \mathbf{u} q &= 0 \end{aligned}$$

Puisque  $\mathbf{v}$  appartient à  $V$ , le problème devient :

$$\begin{aligned} \nu \int_{\Omega} \nabla \mathbf{u} \cdot \nabla \mathbf{v} - \int_{\Omega} p \operatorname{div} \mathbf{v} &= \int_{\Omega} \mathbf{f} \mathbf{v} + \int_{\Gamma_N} \mathbf{g}_2 \mathbf{v} \\ \int_{\Omega} \operatorname{div} \mathbf{u} q &= 0 \end{aligned}$$

Finalement, on définit les formes bilinéaires  $a : \tilde{V} \times V \rightarrow \mathbb{R}$  et  $b : V \times Q \rightarrow \mathbb{R}$  :

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) &= \nu \int_{\Omega} \nabla \mathbf{u} \cdot \nabla \mathbf{v} \\ b(\mathbf{v}, q) &= - \int_{\Omega} p \operatorname{div} \mathbf{v} \end{aligned}$$

avec  $\tilde{V} = V$  pour une condition de Dirichlet homogène et  $\tilde{V} = \{\mathbf{v} \in (H^1(\Omega))^d \mid \mathbf{v}|_{\Gamma_D} = \mathbf{g}_1\}$  pour une condition de Dirichlet non homogène, ce qui est notre cas. On définit également la fonctionnelle linéaire  $f : V \rightarrow \mathbb{R}$  :

$$f(\mathbf{v}) = \int_{\Omega} \mathbf{f}\mathbf{v} + \int_{\Gamma_N} \mathbf{g}_2 \mathbf{v}$$

Le problème devient alors :

$$\begin{aligned} &\text{Trouver } \mathbf{u} \in \tilde{V} \text{ et } p \in Q \text{ tels que} \\ &a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = f(\mathbf{v}) \quad \forall \mathbf{v} \in V \\ &b(\mathbf{u}, q) = 0 \quad \forall q \in Q \end{aligned} \tag{7.6}$$

D'après [Quarteroni et Valli, 2008, Théorème 7.4.2], ce problème est bien défini si les conditions suivantes sont vérifiées :

1. La forme bilinéaire  $a(\mathbf{u}, \mathbf{v})$  est continue.
2. La forme bilinéaire  $a(\mathbf{u}, \mathbf{v})$  est coercive sur  $\mathcal{N}$  le noyau de la forme bilinéaire  $b$ , i.e. :

$$\exists \alpha > 0, \quad a(\mathbf{v}, \mathbf{v}) \geq \alpha \|\mathbf{v}\|_{\tilde{V}}^2 \quad \forall \mathbf{v} \in \mathcal{N}$$

3. La forme bilinéaire  $b(\mathbf{v}, q)$  est continue.
4. Les espaces  $V$  et  $Q$  vérifient la condition de l'inf-sup :

$$\exists \beta_h > 0, \quad \inf_{q_h \in Q_h} \sup_{\mathbf{v}_h \in \tilde{V}_h} \frac{b(\mathbf{v}_h, q)_h}{\|q_h\|_Q \|\mathbf{v}_h\|_V} \geq \beta_h$$

A l'aide des inégalités de Cauchy-Schwarz, de Hölder et de Poincaré, on vérifie aisément les trois premières conditions, et on peut trouver une preuve de la condition de l'inf-sup dans [Girault et Raviart, 1986].

## 7.2.2 Discrétisation

### 7.2.2.1 Approximation de Galerkin

Soient  $V_h \subset V$  et  $Q_h \subset Q$  des sous-espaces de dimension finie tels que  $V_h \rightarrow V$  et  $Q_h \rightarrow Q$  lorsque  $h \rightarrow 0$ . Le problème (7.6) s'approche de la manière suivante :

$$\begin{aligned} &\text{Trouver } \mathbf{u}_h \in \tilde{V}_h = V_h \cap \tilde{V} \text{ et } p_h \in Q_h \text{ tels que} \\ &a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = f(\mathbf{v}_h) \quad \forall \mathbf{v}_h \in V_h \\ &b(\mathbf{u}_h, q_h) = 0 \quad \forall q_h \in Q_h \end{aligned}$$

Pour montrer que ce problème est bien défini, on peut vérifier à nouveau les hypothèses du théorème 7.4.2 dans [Quarteroni et Valli, 2008]. Bien que les trois premières conditions soient alors automatiquement vérifiées pour  $V_h \subset V$  et  $Q_h \subset Q$ , ce n'est pas le cas de la condition de l'inf-sup. La difficulté du problème de Stokes est alors le choix des espaces de discrétisation qui doivent satisfaire cette dernière condition.

### 7.2.2.2 Espaces de Taylor-Hood

On considère une triangulation  $T_h$  de  $\Omega$ , telle que  $\bar{\Omega} = \bigcup_{K \in T_h} K$ , qui vérifie :

- chaque  $K$  est un polyèdre tel que  $\text{int}(K) \neq \emptyset$  ;
- $\text{int}(K_1) \cap \text{int}(K_2) = \emptyset$  pour tout  $K_1$  distinct de  $K_2$  ;
- si  $E = K_1 \cap K_2$ , alors  $E$  est un plan commun, un sommet commun ou bien une arête commune.

La figure 7.2 illustre un exemple de triangulation pour un cube.

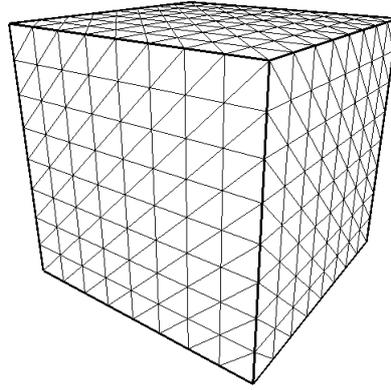


FIGURE 7.2 – Exemple de triangulation.

Finalement, on définit les espaces de dimension finie  $V_h$  et  $Q_h$  de la façon suivante :

$$\begin{aligned} V_h &= \{\mathbf{v}_h \in C^0(\bar{\Omega})^d \mid \forall K \in T_h, \mathbf{v}_h|_K \in \mathbb{P}_2\} \\ Q_h &= \{q_h \in C^0(\bar{\Omega}) \mid \forall K \in T_h, q_h|_K \in \mathbb{P}_1\} \end{aligned} \quad (7.7)$$

où  $\mathbb{P}_1$  (resp.  $\mathbb{P}_2$ ) est l'espace des polynômes de degré inférieur ou égal à 1 (resp. 2). Les espaces  $V_h$  et  $Q_h$  satisfont la condition de l'inf-sup [Girault et Raviart, 1986, Chapitre 2] et on a l'estimation de convergence suivante :

$$\|\mathbf{u} - \mathbf{u}_h\|_V + \|\mathbf{p} - \mathbf{p}_h\|_Q \leq Ch^s (\|\mathbf{u}\|_{s+1} + \|\mathbf{p}\|_s), \quad s = 1, 2 \quad (7.8)$$

On peut montrer de manière générale que les espaces de Taylor-Hood  $\mathbb{P}_N - \mathbb{P}_{N-1}$  ( $N > 1$ ) vérifient la condition de l'inf-sup discrète. L'éléments  $\mathbb{P}_1$ -bulle- $\mathbb{P}_1$ , appelé élément mini, vérifie également la condition de l'inf-sup. L'espace des vitesses  $\mathbb{P}_1$  est enrichi en lui ajoutant un degré de liberté par triangle (voir figure 7.3). L'élément mini permet de diminuer les degrés de liberté par rapport à l'élément  $\mathbb{P}_2 - \mathbb{P}_1$ , néanmoins les résultats en pression ne sont pas très bons [Salmon, 2015].

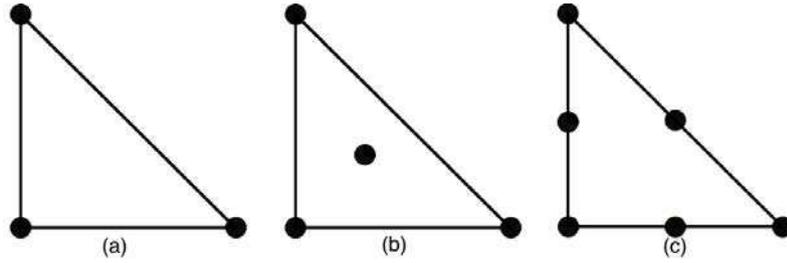


FIGURE 7.3 – Éléments finis triangulaires de type (a)  $\mathbb{P}_1$  (interpolation polynomiale sur les 3 nœuds du triangle), (b)  $\mathbb{P}_1$ -bulle (4 nœuds), et (c)  $\mathbb{P}_2$  (6 nœuds).

### 7.3 Résolution du problème de Navier-Stokes

On définit  $\Omega$ , et ses deux parties mesurables  $\Gamma_N$  et  $\Gamma_D$ , comme dans la section 7.2. On rappelle que le problème de Navier-Stokes, avec des conditions aux limites mêlées, est le suivant :

$$\left\{ \begin{array}{ll} \partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} & \text{dans } \Omega \times (0, T) & (7.9) \\ \operatorname{div} \mathbf{u} = 0 & \text{dans } \Omega \times (0, T) & (7.10) \\ \mathbf{u} = \mathbf{g}_1 & \text{sur } \Gamma_D \times (0, T) & (7.11) \\ \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} = \mathbf{g}_2 & \text{sur } \Gamma_N \times (0, T) & (7.12) \\ \mathbf{u}|_{t=0} = \mathbf{u}_0 & \text{dans } \Omega & (7.13) \end{array} \right.$$

où  $\mathbf{u} = (u_1, u_2, u_3)$  est la vitesse du fluide,  $p$  sa pression,  $\nu$  sa viscosité cinématique et  $\mathbf{f}$  les forces extérieures.

Les différents opérateurs sont définis par l'équation (7.5) et par :

$$\mathbf{u} \cdot \nabla \mathbf{u} = \sum_{i=1}^d \mathbf{u}_i \frac{\partial \mathbf{u}}{\partial x_i} \in \mathbb{R}^N$$

#### 7.3.1 Méthode des caractéristiques

L'idée principale de cette méthode est de considérer l'opérateur  $\partial_t + \mathbf{u} \cdot \nabla$  comme la dérivée totale de la vitesse  $\mathbf{u}$ , qui convertit les coordonnées eulériennes en coordonnées lagrangiennes [Pironeau, 1982, Formaggia *et al.*, 2009]. Plus précisément, on définit la caractéristique  $\mathbf{X}(\tau; t, x)$  représentant la trajectoire sous l'action du champ de vitesse  $\mathbf{u}(\cdot, \tau)$  d'une particule se trouvant en  $x$  à l'instant  $t$ . La caractéristique est solution de l'Équation Différentielle Ordinaire (EDO) définie par :

$$\left\{ \begin{array}{l} \frac{d}{d\tau} [\mathbf{X}(\tau; t, x)] = \mathbf{u}(\mathbf{X}(\tau; t, x), \tau), \quad \tau \in (0, T) \\ \mathbf{X}(\tau; t, x) = x. \end{array} \right. \quad (7.14)$$

L'introduction de la caractéristique permet de faire apparaître la dérivée totale de la vitesse :

$$\frac{d}{d\tau} [\mathbf{u}(\mathbf{X}(\tau; t, x), t)] = \frac{\partial \mathbf{u}}{\partial \tau} (\mathbf{X}(\tau; t, x), \tau) + \left( \frac{d}{d\tau} [\mathbf{X}(\tau; t, x)] \cdot \nabla \right) \mathbf{u}(\mathbf{X}(\tau; t, x), \tau) \quad (7.15)$$

et en utilisant l'EDO (7.14), on obtient :

$$\frac{d}{d\tau} [\mathbf{u}(\mathbf{X}(\tau; t, \mathbf{x}), t)]|_{\tau=t} = (\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u})(\mathbf{x}, t)$$

La dérivée totale est alors approchée de la manière suivante :

$$(\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u})(x, t^{n+1}) \approx \frac{\mathbf{u}(x, t^{n+1}) - \mathbf{u}(\mathbf{X}^n(x), t^n)}{\Delta t} + \mathcal{O}(\Delta t)$$

où  $\mathbf{X}^n(x) = x - \mathbf{u}(x, t^n)\Delta t + \mathcal{O}(\Delta t^2)$  est une approximation de  $\mathbf{X}(t^{n+1}; t^n, \mathbf{x})$ ,  $\Delta t > 0$  est le pas de temps et  $t^n = n\Delta t$ ,  $n \in \mathbb{N}$ . Par conséquent, le terme non linéaire  $\mathbf{u} \cdot \nabla \mathbf{u}$  a été "absorbé" par la dérivée totale. Finalement, on obtient la discrétisation en temps du problème de Navier-Stokes suivante :

$$\frac{1}{\Delta t} (\mathbf{u}^{n+1} - \mathbf{u}^n \circ \mathbf{X}^n) - \nu \Delta \mathbf{u}^{n+1} + \nabla p^{n+1} = \mathbf{f}^{n+1} \quad (7.16)$$

$$\operatorname{div} \mathbf{u}^{n+1} = 0 \quad (7.17)$$

avec  $\varphi^n = \varphi(x, t^n)$  et  $\mathbf{u}^n \circ \mathbf{X}^n \approx \mathbf{u}^n(x - \mathbf{u}^n(x)\Delta t)$ .

### 7.3.1.1 Formulation variationnelle

Pour les fonctions dépendantes du temps et de l'espace, on choisit l'espace des fonctions, défini sur un intervalle de temps  $[0, T]$  et dans un espace donné  $V$ , suivant :

$$L^2(0, T; V) = \left\{ \mathbf{v} : (0, T) \rightarrow V \text{ est mesurable et } \int_0^T \|\mathbf{v}(x, t)\|_V^2 dt < \infty \right\}$$

On choisit les espaces des fonctions tests  $V$  pour la vitesse et  $Q$  pour la pression, définis comme dans la section 7.2.1. On multiplie l'équation (7.16) par  $\mathbf{v} \in V$  et l'équation (7.17) par  $q \in Q$ , et on intègre les deux équations sur  $\Omega$  :

$$\begin{aligned} \frac{1}{\Delta t} \int_{\Omega} \mathbf{u}^{n+1} \mathbf{v} - \frac{1}{\Delta t} \int_{\Omega} (\mathbf{u}^n \circ \mathbf{X}^n) \mathbf{v} - \nu \int_{\Omega} \Delta \mathbf{u}^{n+1} \mathbf{v} + \int_{\Omega} \nabla p^{n+1} \mathbf{v} &= \int_{\Omega} \mathbf{f}^{n+1} \mathbf{v} \\ \int_{\Omega} \operatorname{div} \mathbf{u}^{n+1} q &= 0 \end{aligned}$$

On applique la formule de Green :

$$\begin{aligned} \frac{1}{\Delta t} \int_{\Omega} \mathbf{u}^{n+1} \mathbf{v} - \frac{1}{\Delta t} \int_{\Omega} (\mathbf{u}^n \circ \mathbf{X}^n) \mathbf{v} + \nu \int_{\Omega} \nabla \mathbf{u}^{n+1} \cdot \nabla \mathbf{v} - \int_{\Omega} p^{n+1} \operatorname{div} \mathbf{v} \\ = \int_{\Omega} \mathbf{f}^{n+1} \mathbf{v} + \int_{\Gamma_D} \left( \nu \frac{\partial \mathbf{u}^{n+1}}{\partial \mathbf{n}} - p^{n+1} \mathbf{n} \right) \mathbf{v} + \int_{\Gamma_N} \left( \nu \frac{\partial \mathbf{u}^{n+1}}{\partial \mathbf{n}} - p^{n+1} \mathbf{n} \right) \mathbf{v} \\ \int_{\Omega} \operatorname{div} \mathbf{u}^{n+1} q = 0 \end{aligned}$$

Puisque  $\mathbf{v}$  appartient à  $V$ , le terme sur  $\Gamma_D$  disparaît, et on obtient la formulation variationnelle suivante : Trouver  $\mathbf{u} \in L^2(0, T; \tilde{V})$  et  $q \in L^2(0, T; Q)$  tels que  $\forall \mathbf{v} \in V$ ,  $\forall q \in Q$  et  $\forall t \geq 0$  :

$$\begin{aligned} \frac{1}{\Delta t} \int_{\Omega} \mathbf{u}^{n+1} \mathbf{v} - \frac{1}{\Delta t} \int_{\Omega} (\mathbf{u}^n \circ \mathbf{X}^n) \mathbf{v} + \nu \int_{\Omega} \nabla \mathbf{u}^{n+1} \cdot \nabla \mathbf{v} - \int_{\Omega} p^{n+1} \operatorname{div} \mathbf{v} &= \int_{\Omega} \mathbf{f}^{n+1} \mathbf{v} + \int_{\Gamma_N} \mathbf{g}_2^{n+1} \mathbf{v} \\ \int_{\Omega} \operatorname{div} \mathbf{u}^{n+1} q &= 0 \end{aligned} \quad (7.18)$$

avec  $\mathbf{u}|_{t=0} = \mathbf{u}_0 \in \tilde{V}_{\operatorname{div}}$  où  $V_{\operatorname{div}}$  est l'espace des fonctions à divergence nulle défini par :

$$V_{\operatorname{div}} = \{ \mathbf{v} \in V \mid \operatorname{div} \mathbf{v} = 0 \}$$

### 7.3.1.2 Discrétisation en espace

Soient  $V_h$  et  $Q_h$  les espaces de dimension finie définis par l'équation (7.7); l'approximation de Galerkin du problème (7.18) s'écrit alors : Trouver  $\mathbf{u}_h^{n+1} \in \tilde{V}_h = V_h \cap \tilde{V}$  et  $p_h^{n+1} \in Q_h$  tels que :

$$\begin{aligned} \frac{1}{\Delta t} \int_{\Omega} \mathbf{u}_h^{n+1} \mathbf{v}_h + \nu \int_{\Omega} \nabla \mathbf{u}_h^{n+1} \cdot \nabla \mathbf{v}_h - \int_{\Omega} p_h^{n+1} \operatorname{div} \mathbf{v}_h &= \int_{\Omega} \mathbf{f}_h^{n+1} \mathbf{v}_h + \frac{1}{\Delta t} \int_{\Omega} (\mathbf{u}_h^n \circ \mathbf{X}^n) \mathbf{v}_h \\ &+ \int_{\Gamma_N} \mathbf{g}_{2h}^{n+1} \mathbf{v}_h \\ \int_{\Omega} \operatorname{div} \mathbf{u}_h^{n+1} q_h &= 0 \end{aligned} \quad (7.19)$$

avec  $\mathbf{u}_h|_{t=0} = \mathbf{u}_{0,h} \in (\tilde{V}_h)_{\operatorname{div}}$ .

Ce schéma est inconditionnellement stable [Formaggia *et al.*, 2009, Chapitre 12]. De plus, il satisfait l'estimation de convergence en temps suivante :

$$\|\mathbf{u}(t^n) - \mathbf{u}^n\|_0 + \left( \Delta t \sum_{k=1}^n \|\nabla(\mathbf{u}(t^k) - \mathbf{u}^k)\|_0^2 \right)^{1/2} \leq C \Delta t$$

pour une constante positive  $C$  indépendante de  $\Delta t$ . Pour la convergence en espace, l'estimation est donnée par l'équation (7.8).

### 7.3.2 Méthode d'Uzawa

Nous allons résoudre le problème discret (7.19), donné par la méthode des caractéristiques, à l'aide de l'algorithme d'Uzawa [Roberts et Thomas, 1991]. Soient  $a : \tilde{V} \times V \rightarrow \mathbb{R}$  et  $b : V \times Q \rightarrow \mathbb{R}$  les formes bilinéaires définies par :

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) &= \frac{1}{\Delta t} \int_{\Omega} \mathbf{u}^{n+1} \mathbf{v} + \nu \int_{\Omega} \nabla \mathbf{u}^{n+1} \cdot \nabla \mathbf{v} \\ b(\mathbf{v}, q) &= - \int_{\Omega} p^{n+1} \operatorname{div} \mathbf{v} \end{aligned}$$

et  $f : V \rightarrow \mathbb{R}$  la fonctionnelle linéaire définie par :

$$f(\mathbf{v}) = \int_{\Omega} \mathbf{f}^{n+1} \mathbf{v} + \frac{1}{\Delta t} \int_{\Omega} (\mathbf{u}^n \circ \mathbf{X}^n) \mathbf{v} + \int_{\Gamma_N} \mathbf{g}_{2h}^{n+1} \mathbf{v}$$

Le problème (7.18) s'approche de la manière suivante :

$$\begin{aligned} \text{Trouver } \mathbf{u}_h \in \tilde{V}_h \text{ et } p_h \in Q_h \text{ tels que} \\ a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) &= f(\mathbf{v}_h) & \forall \mathbf{v}_h \in V_h \\ b(\mathbf{u}_h, q_h) &= 0 & \forall q_h \in Q_h \end{aligned}$$

Pour chaque  $h$  fixé, on note  $M_h$  et  $N_h$  les dimensions respectives des sous-espaces  $V_h$  et  $Q_h$ , ainsi que leur base associée :

$$\{\varphi_j | j = 1, \dots, M_h\} \text{ et } \{\phi_k | k = 1, \dots, N_h\}$$

Si l'on écrit

$$\mathbf{u}_h = \sum_{j=1}^{M_h} u_j \varphi_j, \quad p_h = \sum_{k=1}^{N_h} p_k \phi_k(\mathbf{x})$$

alors on obtient le système linéaire :

$$\begin{aligned} \mathbf{A}\mathbf{u}_h + \mathbf{B}^T p_h &= \mathbf{F} \\ \mathbf{B}\mathbf{u}_h &= 0 \end{aligned} \quad (7.20)$$

où  $\mathbf{u}_h = (u_1, \dots, u_{M_h})$  et  $p_h = (p_1, \dots, p_{N_h})$  sont les vecteurs des inconnues,  $A_{i,j} = a(\varphi_j, \varphi_i)$ ,  $B_{i,k} = b(\varphi_k, \varphi_i)$  et  $F_i = f(\varphi_i)$ .

Soit  $J : V_h \rightarrow \mathbb{R}$  la fonctionnelle définie par :

$$J(\mathbf{u}_h) = \frac{1}{2} \mathbf{u}_h^T \mathbf{A} \mathbf{u}_h - \mathbf{F} \mathbf{u}_h$$

Le problème (7.20) est équivalent au problème de minimisation sous contrainte suivant :

$$\begin{cases} \min_{\mathbf{v}_h \in X} J(\mathbf{v}_h) \\ X = \{\mathbf{v}_h \in V_h, \mathbf{B}\mathbf{v}_h = 0\} \end{cases} \quad (7.21)$$

On introduit le lagrangien du problème (7.21) comme la fonction définie sur  $V_h \times \mathbb{R}^+$  par :

$$\mathcal{L}(\mathbf{u}_h, q_h) = J(\mathbf{u}_h) - q_h^T \mathbf{B}\mathbf{u}_h$$

Un point-selle  $(\mathbf{u}_h, p_h) \in V_h \times \mathbb{R}^+$  de  $\mathcal{L}$  est caractérisé par le problème min-max suivant :

$$\forall q_h \in \mathbb{R}^+ \quad \mathcal{L}(\mathbf{u}_h, q_h) \leq \mathcal{L}(\mathbf{u}_h, p_h) \leq \mathcal{L}(\mathbf{v}_h, p_h) \quad \forall \mathbf{v}_h \in V_h$$

On suppose que les fonctions  $J$  et  $b$  sont des fonctions convexes de classe  $C^1$ . Puisque les conditions de Kuhn-Tucker sont vérifiées, un point-selle  $(\mathbf{u}_h, p_h)$  de  $\mathcal{L}$  existe et le problème (7.21) admet une solution. Pour trouver le point-selle  $(\mathbf{u}_h, p_h)$ , on résout les deux sous-problèmes suivants :

- Le problème primal consiste à chercher le minimum de la fonction  $\mathbf{v}_h \rightarrow \mathcal{L}(\mathbf{v}_h, p_h)$ .

$$\text{Calculer } \mathbf{u}_h^n : \quad \mathbf{u}_h^n = \mathbf{A}^{-1}(\mathbf{F} - \mathbf{B}^T p_h^n) \quad (7.22)$$

- Le problème dual consiste à trouver le maximum de la fonction  $q_h \rightarrow \mathcal{L}(\mathbf{u}_h, q_h)$  s.c.  $q_h \geq 0$ . On introduit la fonction duale  $G(q_h) = \inf_{\mathbf{v}_h \in V_h} \mathcal{L}(\mathbf{v}_h, q_h)$ , et on cherche alors  $p_h \in \mathbb{R}^+$  qui maximise  $G$ . On en déduit que  $G(q_h) = J(\mathbf{u}_h) - q_h^T \mathbf{B}\mathbf{u}_h$ , et en dérivant, on a  $\nabla G(q_h) = -\mathbf{B}\mathbf{u}_h$ . Pour résoudre ce problème de maximisation sous contrainte, on peut utiliser la méthode de gradient projeté.

$$\text{Calculer } p_h^{n+1} : \quad p_h^{n+1} = P_{\mathbb{R}^+}(p_h^n - \rho \nabla G(p_h^n)) = \max(0, p_h^n + \rho \mathbf{B}\mathbf{u}_h^n) \quad (7.23)$$

Algébriquement, on peut résoudre le problème dual en multipliant l'équation (7.22) par la matrice  $\mathbf{B}$ , on obtient alors :

$$\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T p_h^n = \mathbf{B}\mathbf{A}^{-1}\mathbf{F} \quad (7.24)$$

et la matrice  $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$  est inversible en tant que matrice carré symétrique définie positive.

### 7.3.2.1 Préconditionnement de Cahouet et Chabard

L'implantation de l'algorithme d'Uzawa nécessite le choix d'un preconditionneur qui donne une approximation appropriée de  $p^{n+1}$ . On utilise le preconditionneur donné par Cahouet et Chabard [Cahouet et Chabard, 1998] :  $-\alpha(\Delta)^{-1} + \nu I$  où  $\alpha = \frac{1}{\Delta t}$  et  $I$  est la matrice identité.

On rappelle que la méthode des caractéristiques permet de réécrire les équations de Navier-Stokes sous la forme :

$$\begin{aligned}\alpha \mathbf{u}^{n+1} - \nu \Delta \mathbf{u}^{n+1} &= \mathbf{f}' - \nabla p^{n+1} \\ \operatorname{div} \mathbf{u}^{n+1} &= 0\end{aligned}\tag{7.25}$$

où  $\mathbf{f}' = \mathbf{f}^{n+1} + \alpha \mathbf{u}^n \circ \mathbf{X}^n$ .

Soit  $H^{-1}$  l'espace dual de  $H_0^1$ . On considère l'opérateur linéaire  $\mathbf{L}^{-1}$ , solution du problème (7.25) :

$$\begin{aligned}\mathbf{L}^{-1} : H^{-1}(\Omega) &\rightarrow H_0^1(\Omega) \\ \mathbf{v} &\mapsto \mathbf{L}^{-1}(\mathbf{v}) = (\nu \Delta - \alpha I)^{-1} \mathbf{v}\end{aligned}$$

Le complément de Schur est :

$$\begin{aligned}\mathbf{L}_0(\alpha) : L^2(\Omega) &\rightarrow L^2(\Omega) \\ p &\mapsto \mathbf{L}_0(\alpha)(p) = \operatorname{div} \mathbf{L}^{-1} \nabla p\end{aligned}$$

L'opérateur  $\mathbf{L}_0(\alpha)$  est auto-adjoint et positif. On peut alors considérer l'algorithme d'Uzawa comme la méthode itérative de Richardson du premier ordre à pas fixe appliquée à l'équation :

$$\mathbf{L}_0(\alpha)p = \operatorname{div} \mathbf{L}^{-1} \mathbf{f}'$$

Donc le preconditionneur suggéré est  $-\alpha(\Delta)^{-1} + \nu I$ .

### 7.3.3 Schéma découplé d'ordre élevé

Comparé à la méthode classique des caractéristiques, qui est fiable mais coûteuse en temps, nous allons décrire un schéma appartenant à la classe des schémas découplés [Guermond et Shen, 2003], qui découple le calcul de la vitesse et de la pression pour gagner du temps et s'affranchir de la condition de compatibilité entre les espaces, et donc de la condition de l'inf-sup. Cet algorithme récent [Sheng *et al.*, 2013] présente l'avantage de pouvoir utiliser des conditions aux limites mêlées et fournit une précision élevée. Nous allons montrer de manière formelle comment les équations de Navier-Stokes peuvent être découplées.

En calculant la divergence de l'équation (7.9), on obtient :

$$\Delta p = -\nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) + \nabla \cdot \mathbf{f} - \partial_t (\nabla \cdot \mathbf{u}) + \nu \Delta (\operatorname{div} \mathbf{u})$$

et d'après l'équation (7.10) :

$$\Delta p = -\nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) + \nabla \cdot \mathbf{f} - \partial_t (\operatorname{div} \mathbf{u})$$

Puisque la condition d'incompressibilité n'est pas satisfaite tout le temps dans les calculs numériques, on préserve le terme  $\partial_t(\operatorname{div} \mathbf{u})$ . Par contre, on peut négliger numériquement le terme  $\Delta(\operatorname{div} \mathbf{u})$ .

En prenant la composante normale de l'équation (7.9), on a :

$$\nabla p \cdot \mathbf{n} = (\mathbf{f} - \partial_t \mathbf{u} - \mathbf{u} \cdot \nabla \mathbf{u} + \nu \Delta \mathbf{u}) \cdot \mathbf{n}$$

et en utilisant l'identité vectorielle :

$$\Delta \mathbf{u} = -\nabla \times \nabla \times \mathbf{u} + \nabla(\operatorname{div} \mathbf{u})$$

on obtient :

$$\nabla p \cdot \mathbf{n} = (\mathbf{f} - \partial_t \mathbf{u} - \mathbf{u} \cdot \nabla \mathbf{u} - \nu \nabla \times \nabla \times \mathbf{u} + \nu \nabla(\operatorname{div} \mathbf{u})) \cdot \mathbf{n}$$

puis, d'après l'équation (7.10) :

$$\nabla p \cdot \mathbf{n} = (\mathbf{f} - \partial_t \mathbf{u} - \mathbf{u} \cdot \nabla \mathbf{u} - \nu \nabla \times \nabla \times \mathbf{u}) \cdot \mathbf{n}$$

En prenant la composante normale de l'équation (7.12), on a :

$$p = \nu \mathbf{n} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - \mathbf{g}_2 \cdot \mathbf{n}$$

et d'après l'équation (7.10) :

$$p = \nu \mathbf{n} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - \alpha \nu \operatorname{div} \mathbf{u} - \mathbf{g}_2 \cdot \mathbf{n}$$

où  $\alpha \in [0, 1]$  est une constante importante pour la précision et la stabilité du schéma [Sheng *et al.*, 2013].

Finalement, les équations de Navier-Stokes sont équivalentes à une équation de Poisson (7.26) pour la pression et une équation de la chaleur (7.27) pour chaque composante de la vitesse (étant donné que la pression est connue) :

$$\left\{ \begin{array}{ll} \Delta p = -\nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) + \nabla \cdot \mathbf{f} - \partial_t(\operatorname{div} \mathbf{u}) & \text{dans } \Omega \times (0, T) \\ \nabla p \cdot \mathbf{n} = (\mathbf{f} - \partial_t \mathbf{u} - \mathbf{u} \cdot \nabla \mathbf{u} - \nu \nabla \times \nabla \times \mathbf{u}) \cdot \mathbf{n} & \text{sur } \Gamma_D \times (0, T) \\ p = \nu \mathbf{n} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - \alpha \nu \operatorname{div} \mathbf{u} - \mathbf{g}_2 \cdot \mathbf{n} & \text{sur } \Gamma_N \times (0, T) \\ \operatorname{div} \mathbf{u} = 0 & \text{sur } \Omega \times (0, T) \\ \mathbf{u}|_{t=0} = \mathbf{u}_0 & \text{dans } \Omega \end{array} \right. \quad (7.26)$$

$$\left\{ \begin{array}{ll} \partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \Delta \mathbf{u} = \mathbf{f} - \nabla p & \text{dans } \Omega \times (0, T) \\ \mathbf{u} = \mathbf{g}_1 & \text{sur } \Gamma_D \times (0, T) \\ \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} = \mathbf{g}_2 & \text{sur } \Gamma_N \times (0, T) \\ \mathbf{u}|_{t=0} = \mathbf{u}_0 & \text{dans } \Omega \end{array} \right. \quad (7.27)$$

### 7.3.3.1 Discrétisation en temps

On utilise les formules de différentiation rétrograde (BDF en anglais pour "Backward Differentiation Formula") d'ordre  $k$  pour approcher  $\phi_t(t^{n+1})$ , notée  $\mathcal{D}_k$ , et l'extrapolation d'ordre  $m$  pour approcher  $\phi(t^{n+1})$ , notée  $\mathcal{E}_m$ . Ces approximations sont calculées par :

$$\mathcal{D}_k \phi^{n+1} = \frac{1}{\Delta t} \left( \beta_0^k \phi^{n+1} - \sum_{j=1}^k \beta_j^k \phi^{n+1-j} \right), \quad \mathcal{E}_m \phi^{n+1} = \sum_{j=0}^{m-1} \Upsilon_j^m \phi^{n-j}$$

Les coefficients  $\beta_j^k$  et  $\gamma_j^m$  sont donnés dans [Sheng *et al.*, 2013]. Par exemple, on a :

$$\mathcal{D}_k \phi^{n+1} = \begin{cases} \phi^{n+1} - \phi^n & \text{si } k = 1 \\ \frac{3}{2}\phi^{n+1} - 2\phi^n + \frac{1}{2}\phi^{n-1} & \text{si } k = 2 \end{cases}$$

et :

$$\mathcal{E}_m \phi^{n+1} = \begin{cases} \phi^n & \text{si } m = 1 \\ 2\phi^n - \phi^{n-1} & \text{si } m = 2 \end{cases}$$

En premier lieu, on résout la pression  $p^{n+1}$  à l'aide du schéma suivant :

$$\Delta p^{n+1} = \nabla \cdot \mathbf{f}^{n+1} - \nabla \cdot \mathcal{E}_m(\mathbf{u} \cdot \nabla \mathbf{u})^{n+1} - \mathcal{D}_k(\text{div } \mathbf{u}^{n+1}) \quad \text{dans } \Omega \quad (7.28)$$

$$\nabla p^{n+1} \cdot \mathbf{n} = (\mathbf{f}^{n+1} - \mathcal{D}_k \mathbf{u}^{n+1} - \mathcal{E}_m(\mathbf{u} \cdot \nabla \mathbf{u})^{n+1} - \nu \nabla \times \nabla \times \mathcal{E}_m \mathbf{u}^{n+1}) \cdot \mathbf{n} \quad \text{sur } \Gamma_D \quad (7.29)$$

$$p^{n+1} = \nu \mathbf{n} \cdot \frac{\partial \mathcal{E}_m \mathbf{u}^{n+1}}{\partial \mathbf{n}} - \alpha \nu \text{div}(\mathcal{E}_m \mathbf{u}^{n+1}) - \mathbf{g}_2^{n+1} \cdot \mathbf{n} \quad \text{sur } \Gamma_N \quad (7.30)$$

$$\text{div } \mathbf{u}^{n+1} = 0 \quad \text{dans } \Omega \quad (7.31)$$

Ensuite, on résout la vitesse  $\mathbf{u}^{n+1}$  par l'équation suivante :

$$\mathcal{D}_k \mathbf{u}^{n+1} + \mathcal{E}_m(\mathbf{u} \cdot \nabla \mathbf{u})^{n+1} - \nu \Delta \mathbf{u}^{n+1} + \nabla p^{n+1} = \mathbf{f}^{n+1} \quad \text{dans } \Omega \quad (7.32)$$

$$\mathbf{u}^{n+1} = \mathbf{g}_1^{n+1} \quad \text{sur } \Gamma_D \quad (7.33)$$

$$\nu \frac{\partial \mathbf{u}^{n+1}}{\partial \mathbf{n}} = \mathbf{g}_2^{n+1} + p^{n+1} \mathbf{n} \quad \text{sur } \Gamma_N \quad (7.34)$$

### 7.3.3.2 Formulation variationnelle

En multipliant l'équation (7.28) par  $q$  dans  $Q$ , en intégrant sur  $\Omega$ , et en utilisant la première formule de Green, le problème de Poisson devient :

$$\begin{aligned} - \int_{\Omega} \nabla p^{n+1} \cdot \nabla q &= \int_{\Omega} \Delta p^{n+1} q - \int_{\Gamma} \nabla p^{n+1} \cdot \mathbf{n} q \\ &= \int_{\Omega} (\nabla \cdot \mathbf{f}^{n+1} - \nabla \cdot \mathcal{E}_m(\mathbf{u} \cdot \nabla \mathbf{u})^{n+1}) q - \int_{\Gamma} (\mathbf{f}^{n+1} - \mathcal{E}_m(\mathbf{u} \cdot \nabla \mathbf{u})^{n+1}) \cdot \mathbf{n} q \\ &\quad - \int_{\Omega} \mathcal{D}_k(\text{div } \mathbf{u}^{n+1}) q + \int_{\Gamma} (\mathcal{D}_k \mathbf{u}^{n+1} + \nu \nabla \times \nabla \times \mathcal{E}_m \mathbf{u}^{n+1}) \cdot \mathbf{n} q \\ &= - \int_{\Omega} (\mathbf{f}^{n+1} - \mathcal{E}_m(\mathbf{u} \cdot \nabla \mathbf{u})^{n+1}) \cdot \nabla q - \int_{\Omega} \mathcal{D}_k(\text{div } \mathbf{u}^{n+1}) q \\ &\quad + \int_{\Gamma_D} q \mathcal{D}_k \mathbf{g}_1^{n+1} \cdot \mathbf{n} + \int_{\Gamma_D} \nu \underbrace{q \nabla \times \nabla \times \mathcal{E}_m \mathbf{u}^{n+1} \cdot \mathbf{n}}_{-\nabla \times \mathcal{E}_m \mathbf{u}^{n+1} \cdot \mathbf{n} \times \nabla q} \end{aligned}$$

avec la condition aux limites (7.30) et la condition d'incompressibilité (7.31). La dernière égalité permet d'éviter de calculer  $\nabla \times \nabla \times \mathbf{u}^{n+1}$  explicitement [Liu, 2009].

En multipliant l'équation (7.32) par  $\mathbf{v} \in V$ , en intégrant sur  $\Omega$  et en utilisant les formules de Green, le problème de la chaleur devient :

$$\begin{aligned} \int_{\Omega} \mathcal{D}_k \mathbf{u}^{n+1} \mathbf{v} + \int_{\Omega} \mathcal{E}_m(\mathbf{u} \cdot \nabla \mathbf{u})^{n+1} \mathbf{v} + \nu \int_{\Omega} \nabla \mathbf{u}^{n+1} \cdot \nabla \mathbf{v} - \int_{\Omega} p^{n+1} \text{div } \mathbf{v} \\ = \int_{\Omega} \mathbf{f}^{n+1} \mathbf{v} + \int_{\Gamma} \left( \nu \frac{\partial \mathbf{u}^{n+1}}{\partial \mathbf{n}} - p^{n+1} \mathbf{n} \right) \mathbf{v} \\ = \int_{\Omega} \mathbf{f}^{n+1} \mathbf{v} + \int_{\Gamma_N} \mathbf{g}_2^{n+1} \mathbf{v} \end{aligned}$$

avec la condition aux limites (7.33).

### 7.3.3.3 Discrétisation en espace

Soient  $V_h$  définie par l'équation (7.7) et  $\tilde{Q}_h = \{q_h \in C^0(\bar{\Omega}) \mid \forall K \in T_h, q_h|_K \in \mathbb{P}_2\} \cap Q$  (il a été observé numériquement dans [Sheng *et al.*, 2013] qu'il est possible d'utiliser des éléments finis ne vérifiant pas la condition de l'inf-sup). L'approximation de Galerkin s'écrit alors : Trouver  $\mathbf{u}_h^{n+1} \in \tilde{V}_h = V_h \cap \tilde{V}$  et  $p_h^{n+1} \in \tilde{Q}_h$  tels que :

$$\begin{aligned} - \int_{\Omega} \nabla p_h^{n+1} \cdot \nabla q_h &= - \int_{\Omega} \mathcal{D}_k(\operatorname{div} \mathbf{u}^{n+1})_h q_h + \int_{\Gamma_D} q_h \mathcal{D}_k \mathbf{g}_{1h}^{n+1} \cdot \mathbf{n} \\ &\quad - \int_{\Omega} (\mathbf{f}_h^{n+1} - \mathcal{E}_m(\mathbf{u} \cdot \nabla \mathbf{u})_h^{n+1}) \cdot \nabla q_h - \int_{\Gamma_D} \nu \nabla \times \mathcal{E}_m \mathbf{u}_h^{n+1} \cdot \mathbf{n} \times \nabla q_h \\ \int_{\Omega} \mathcal{D}_k \mathbf{u}_h^{n+1} \mathbf{v}_h + \int_{\Omega} \mathcal{E}_m(\mathbf{u} \cdot \nabla \mathbf{u})_h^{n+1} \mathbf{v}_h + \nu \int_{\Omega} \nabla \mathbf{u}_h^{n+1} \cdot \nabla \mathbf{v}_h - \int_{\Omega} p_h^{n+1} \operatorname{div} \mathbf{v}_h \\ &= \int_{\Omega} \mathbf{f}_h^{n+1} \mathbf{v}_h + \int_{\Gamma_N} \mathbf{g}_{2h}^{n+1} \mathbf{v}_h \end{aligned}$$

avec les conditions aux limites (7.30) et (7.33).

Finalement, en prenant  $(k, m) = (2, 2)$ , on a le schéma discret suivant :

**Étape 1 :** Trouver  $p_h^{n+1} \in \tilde{Q}_h$  tel que :

$$\begin{aligned} - \int_{\Omega} \nabla p_h^{n+1} \cdot \nabla q_h &= \frac{1}{\Delta t} \int_{\Omega} \left( 2 \operatorname{div} \mathbf{u}_h^n - \frac{1}{2} \operatorname{div} \mathbf{u}_h^n \right) q_h + \frac{1}{\Delta t} \int_{\Gamma_D} q_h \left( \frac{3}{2} \mathbf{g}_{1h}^{n+1} - 2 \mathbf{g}_{1h}^n + \frac{1}{2} \mathbf{g}_{1h}^{n-1} \right) \cdot \mathbf{n} \\ &\quad - \int_{\Omega} (\mathbf{f}_h^{n+1} - (2(\mathbf{u} \cdot \nabla \mathbf{u})_h^n - (\mathbf{u} \cdot \nabla \mathbf{u})_h^{n-1})) \cdot \nabla q_h \\ &\quad - \int_{\Gamma_D} \nu \nabla \times (2\mathbf{u}_h^n - \mathbf{u}_h^{n-1}) \cdot \mathbf{n} \times \nabla q_h \end{aligned}$$

avec  $p_h^{n+1} = \nu \mathbf{n} \cdot (\nabla(2\mathbf{u}_h^n - \mathbf{u}_h^{n-1})\mathbf{n}) - \alpha \nu \nabla \cdot (2\mathbf{u}_h^n - \mathbf{u}_h^{n-1}) - \mathbf{g}_{2h}^{n+1} \cdot \mathbf{n}$  sur  $\Gamma_N$  et  $\operatorname{div} \mathbf{u}_h^{n+1} = 0$  sur  $\Omega$ .

**Étape 2 :** Trouver  $\mathbf{u}_h^{n+1} \in V_h$  tel que :

$$\begin{aligned} \frac{1}{\Delta t} \int_{\Omega} \left( \frac{3}{2} \mathbf{u}_h^{n+1} - 2\mathbf{u}_h^n + \frac{1}{2} \mathbf{u}_h^{n-1} \right) \mathbf{v}_h + \nu \int_{\Omega} \nabla \mathbf{u}_h^{n+1} \cdot \nabla \mathbf{v}_h \\ + \int_{\Omega} (2(\mathbf{u} \cdot \nabla \mathbf{u})_h^n - (\mathbf{u} \cdot \nabla \mathbf{u})_h^{n-1}) \mathbf{v}_h - \int_{\Omega} p_h^{n+1} \nabla \cdot \mathbf{v}_h = \int_{\Omega} \mathbf{f}_h^{n+1} \mathbf{v}_h + \int_{\Gamma_N} \mathbf{g}_{2h}^{n+1} \mathbf{v}_h \end{aligned}$$

avec  $\mathbf{u}_h^{n+1} = \mathbf{g}_{1h}^{n+1}$  dans  $\Gamma_D$ .

Cette méthode, ainsi que celle des caractéristiques sont validées sur des solutions analytiques avec conditions aux limites mêlées, dans le chapitre suivant.



# Chapitre 8

## Validation numérique sur des solutions analytiques

*« Attention aux bugs dans le code ci-dessus. Je ne l'ai pas testé, j'ai seulement prouvé qu'il était correct. »*

---

Donald Knuth

Ce chapitre présente une validation numérique des algorithmes décrits dans le chapitre 7. Il est important de tester la performance de ces algorithmes avant de les appliquer à nos géométries réalistes, dans le chapitre 9. Pour cette étape, nous allons utiliser des solutions analytiques connues pour résoudre les problèmes de Stokes et de Navier-Stokes en 3D. Afin de s'approcher de nos cas réalistes, nous étudierons le cas physique qui simule l'écoulement de Poiseuille dans une conduite. Nous vérifierons que les solutions bien connues du problème de Stokes pour un écoulement de Poiseuille sont également solutions du problème de Navier-Stokes.

### Sommaire

---

<b>8.1 Introduction</b>	<b>132</b>
<b>8.2 Problème de Stokes</b>	<b>132</b>
8.2.1 K-test d'Ethier-Steinman	133
8.2.2 Cas-test de Bercovier-Engelman	135
8.2.3 Cas-test de Minev-Ethier	136
8.2.4 Courbes de convergence	137
<b>8.3 Problème de Navier-Stokes</b>	<b>139</b>
8.3.1 Cas-test d'Ethier-Steinman	139
8.3.1.1 Méthode des caractéristiques	139
8.3.1.2 Schéma découplé d'ordre élevé	141
8.3.2 Écoulement de Poiseuille	143
8.3.2.1 Description du modèle	143
8.3.2.2 Résultats	146

---

## 8.1 Introduction

Dans ce chapitre, les problèmes de Stokes et Navier-Stokes sont validés sur des solutions exactes définies sur le cube unité (voir figure 8.1). Nous avons également étudié le problème physique d'un écoulement de Poiseuille dans un tube pour les équations de Navier-Stokes. Pour ces validations numériques, les formulations variationnelles sont implantées à l'aide de FreeFem++<sup>1</sup>, qui est un logiciel libre, basé sur la méthode des éléments finis, développé par Frédéric Hecht (Laboratoire Jacques-Louis Lions de l'Université Pierre et Marie Curie). Les codes utilisés sont disponibles sur le site internet "Numerical Tour in Fluid Dynamics"<sup>2</sup>, créé en collaboration avec le doctorant Simon Garnotel (Université d'Amiens).

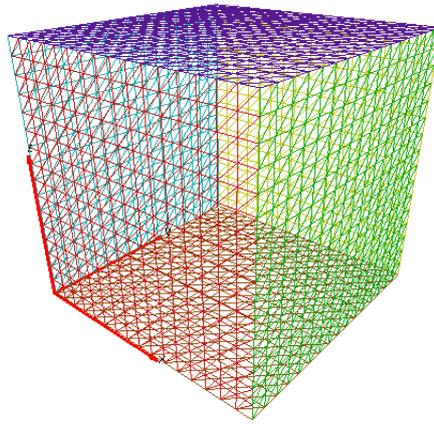


FIGURE 8.1 – Labellisation du cube :  $\Gamma_1$ = face rouge,  $\Gamma_2$ = face verte,  $\Gamma_3$ = face jaune,  $\Gamma_4$ = face bleue,  $\Gamma_5$ = face marron et  $\Gamma_6$ = face violette.

## 8.2 Problème de Stokes

On cherche à résoudre le problème de Stokes avec conditions aux limites mêlées (cf. section 7.2) sur le cube unité représenté par la figure 8.1, et partitionné comme suit :  $\Gamma_N = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$  et  $\Gamma_D = \Gamma_5 \cup \Gamma_6$ . Nous avons validé le code sur les trois solutions analytiques de :

- Ethier-Steinman [Ethier et Steinman, 1994] ;
- Bercovier-Engelman [Bercovier et Engelman, 1979] ;
- Minev-Ethier [Minev et Ethier, 1998].

Les résultats sont vérifiés en calculant les erreurs entre la pression exacte  $p$  et la pression approchée  $p_h$  pour la norme de Lebesgue ( $L^2$ ), et en calculant les erreurs entre la vitesse exacte  $\mathbf{u}$  et la vitesse approchée  $\mathbf{u}_h$  pour les normes de Lebesgue et de Sobolev ( $H^1$ ) :

---

1. <http://www.freefem.org>  
 2. <http://numtourcfd.univ-reims.fr>

$$\|p - p_h\|_{L^2(\Omega)} = \left( \int_{\Omega} (p - p_h)^2 \right)^{\frac{1}{2}}$$

$$\|\mathbf{u} - \mathbf{u}_h\|_{(L^2(\Omega))^3} = \left( \int_{\Omega} \sum_{i=1}^3 (u_i - u_{ih})^2 \right)^{\frac{1}{2}}, \quad \|\mathbf{u} - \mathbf{u}_h\|_{(H^1(\Omega))^3} = \left( \int_{\Omega} \sum_{i=1}^3 \|u_i - u_{ih}\|_{H^1(\Omega)}^2 \right)^{\frac{1}{2}} \quad (8.1)$$

$$\text{où } \|u_i - u_{ih}\|_{H^1(\Omega)} = \|u_i - u_{ih}\|_{L^2(\Omega)} + \sum_{j=1}^3 \left\| \frac{\partial u_i}{\partial x_j} - \frac{\partial u_{ih}}{\partial x_j} \right\|_{L^2(\Omega)}.$$

Finalement, les erreurs relatives sur  $p$  et  $\mathbf{u}$  sont calculées de la manière suivante :

$$\text{Er}(p) = \frac{\|p - p_h\|}{\|p\|}, \quad \text{Er}(\mathbf{u}) = \frac{\|\mathbf{u} - \mathbf{u}_h\|}{\|\mathbf{u}\|}$$

### 8.2.1 K-test d'Ethier-Steinman

La solution exacte d'Ethier-Steinman est définie sur  $\Omega = [-1, 1]^3$  par :

$$\begin{aligned} u_1 &= -a(e^{ax} \sin(ay + dz) + e^{az} \cos(ax + dy)) \\ u_2 &= -a(e^{ay} \sin(az + dx) + e^{ax} \cos(ay + dz)) \\ u_3 &= -a(e^{az} \sin(ax + dy) + e^{ay} \cos(az + dx)) \\ p &= -\frac{1}{2}a^2(e^{2ax} + e^{2ay} + e^{2az} + 2 \sin(ax + dy) \cos(az + dx)e^{a(y+z)} \\ &\quad + 2 \sin(ay + dz) \cos(ax + dy)e^{a(x+z)} + 2 \sin(az + dx) \cos(ay + dz)e^{a(x+y)}) \end{aligned} \quad (8.2)$$

et le second membre est calculé par  $\mathbf{f} = -\nu \Delta \mathbf{u} + \nabla p$ , i.e. :

$$\begin{aligned} f_1 &= -\nu(-a(a^2 e^{ax} \sin(ay + dz) - a^2 e^{az} \cos(ax + dy)) - a(-a^2 e^{ax} \sin(ay + dz) \\ &\quad - d^2 e^{az} \cos(ax + dy)) - a(-d^2 e^{ax} \sin(ay + dz) + a^2 e^{az} \cos(ax + dy))) - a^2(ae^{2ax} \\ &\quad + a \cos(ax + dy) \cos(az + dx)e^{a(y+z)} - d \sin(ax + dy) \sin(az + dx)e^{a(y+z)} \\ &\quad - a \sin(ay + dz) \sin(ax + dy)e^{a(x+z)} + a \sin(ay + dz) \cos(ax + dy)e^{a(x+z)} \\ &\quad + d \cos(az + dx) \cos(ay + dz)e^{a(x+y)} + a \sin(az + dx) \cos(ay + dz)e^{a(x+y)}) \end{aligned}$$

$$\begin{aligned} f_2 &= -\nu(-a(-d^2 e^{ay} \sin(az + dx) + a^2 e^{ax} \cos(ay + dz)) - a(a^2 e^{ay} \sin(az + dx) \\ &\quad - a^2 e^{ax} \cos(ay + dz)) - a(-a^2 e^{ay} \sin(az + dx) - d^2 e^{ax} \cos(ay + dz))) - a^2(ae^{2ay} \\ &\quad + d \cos(ax + dy) \cos(az + dx)e^{a(y+z)} + a \sin(ax + dy) \cos(az + dx)e^{a(y+z)} \\ &\quad + a \cos(ay + dz) \cos(ax + dy)e^{a(x+z)} - d \sin(ay + dz) \sin(ax + dy)e^{a(x+z)} \\ &\quad - a \sin(az + dx) \sin(ay + dz)e^{a(x+y)} + a \sin(az + dx) \cos(ay + dz)e^{a(x+y)}) \end{aligned}$$

$$\begin{aligned} f_3 &= -\nu(-a(-a^2 e^{az} \sin(ax + dy) - d^2 e^{ay} \cos(az + dx)) - a(-d^2 e^{az} \sin(ax + dy) \\ &\quad + a^2 e^{ay} \cos(az + dx)) - a(-a^2 e^{az} \sin(ax + dy) - a^2 e^{ay} \cos(az + dx))) - a^2(ae^{2az} \\ &\quad - a \sin(ax + dy) \sin(az + dx)e^{a(y+z)} + a \sin(ax + dy) \cos(az + dx)e^{a(y+z)} \\ &\quad + d \cos(ay + dz) \cos(ax + dy)e^{a(x+z)} + a \sin(ay + dz) \cos(ax + dy)e^{a(x+z)} \\ &\quad + a \cos(az + dx) \cos(ay + dz)e^{a(x+y)} - d \sin(az + dx) \sin(ay + dz)e^{a(x+y)}) \end{aligned}$$

avec  $a = \frac{\pi}{4}$ ,  $d = \frac{\pi}{2}$  et  $\nu = 1$ . Les figures 8.2 et 8.3 comparent la solution numérique et la solution exacte d’Ethier-Steinman, respectivement pour la pression et la vitesse. A l’œil nu, les solutions numériques et exactes sont identiques. Pour s’assurer de cette similarité, nous avons calculé les erreurs relatives sur la vitesse et la pression (voir table 8.1). On peut noter que ces erreurs diminuent en fonction du pas du maillage  $h$  et qu’elles sont relativement basses pour un maillage suffisamment fin. Ici, le pas du maillage est calculé de la manière suivante :

$$h = \max\{\text{diam}(K), K \in T_h\}$$

Les courbes de convergence sont données dans la section 8.2.4.

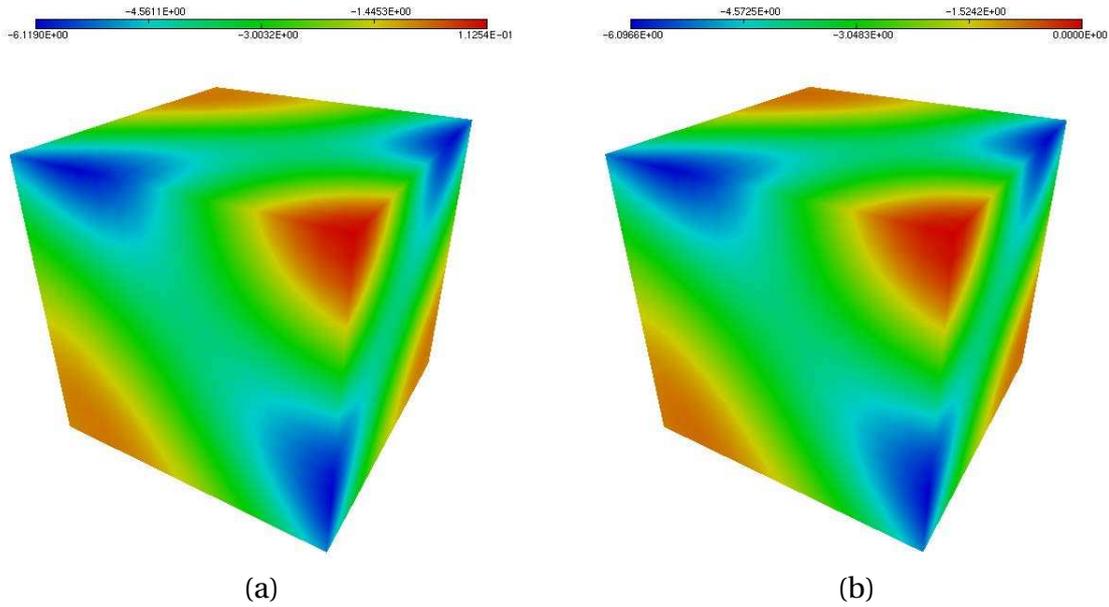


FIGURE 8.2 – Pression sur le cube unité : (a) solution numérique, (b) solution exacte d’Ethier-Steinman.

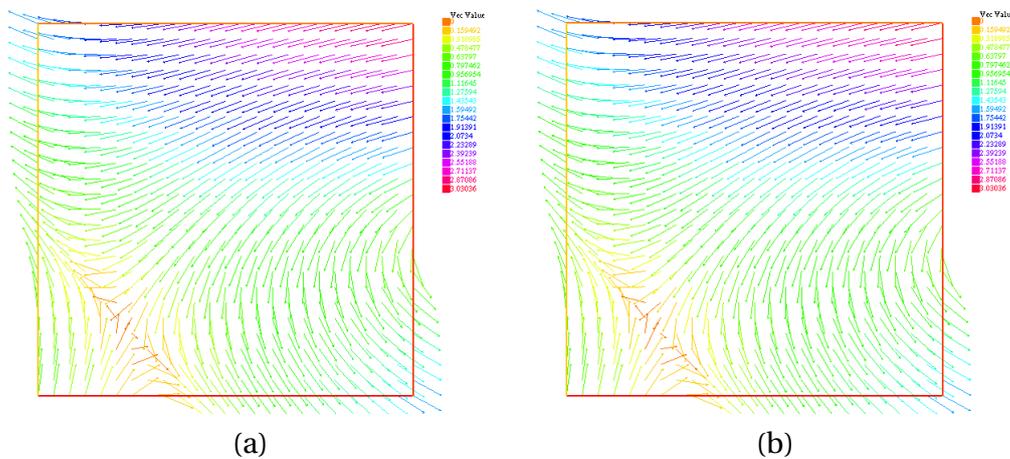


FIGURE 8.3 – Champ de vitesse sur la section  $y = 0,5$  du cube unité : (a) solution numérique, (b) solution exacte d’Ethier-Steinman.

$h$	$Er(\mathbf{u})_{L^2}$	$Er(\mathbf{u})_{H^1}$	$Er(p)_{L^2}$	Temps CPU
0,7071	$3,3387.10^{-2}$	$1,1042.10^{-1}$	$1,9911.10^{-1}$	0,08 s
0,3536	$5,1188.10^{-3}$	$2,2856.10^{-2}$	$6,1024.10^{-2}$	0,59 s
0,1768	$3,7303.10^{-4}$	$4,9048.10^{-3}$	$1,3822.10^{-2}$	5,30 s
0,0884	$2,8310.10^{-5}$	$8,0350.10^{-4}$	$3,3099.10^{-3}$	128,40 s

TABLEAU 8.1 – Erreurs  $L^2$  et  $H^1$  pour la solution exacte d’Ethier-Steinman en fonction du pas du maillage  $h$ .

### 8.2.2 Cas-test de Bercovier-Engelman

La solution exacte de Bercovier-Engelman est définie sur  $\Omega = [0, 1]^3$  par :

$$\begin{aligned}
 u_1 &= -256y(y-1)(2y-1)x^2(x-1)^2 \\
 u_2 &= 256x(x-1)(2x-1)y^2(y-1)^2 \\
 u_3 &= 0 \\
 p &= (x-0.5)(y-0.5)
 \end{aligned} \tag{8.3}$$

et le second membre est donné par :

$$\begin{aligned}
 f_1 &= 256v(x^2(x-1)^2(12y-6) + y(y-1)(2y-1)(12x^2-12x+2)) + (y-0.5) \\
 f_2 &= -256v(y^2(y-1)^2(12x-6) + x(x-1)(2x-1)(12y^2-12y+2)) + (x-0.5) \\
 f_3 &= 0
 \end{aligned}$$

Les figures 8.4 et 8.5 comparent la solution numérique et la solution exacte de Bercovier-Engelman, respectivement pour la pression et la vitesse. A l’œil nu, les solutions numériques et exactes sont similaires pour la vitesse, mais diffèrent légèrement pour la pression. Les erreurs relatives sur la vitesse et la pression sont listées dans la table 8.2. On peut noter que ces erreurs diminuent bien en fonction du pas du maillage  $h$ . Les courbes de convergence sont données dans la section 8.2.4.

$h$	$Er(\mathbf{u})_{L^2}$	$Er(\mathbf{u})_{H^1}$	$Er(p)_{L^2}$	Temps CPU
0,7071	$1,0793.10^{-1}$	$2,4262^{-1}$	$5,1503^{-0}$	0,06 s
0,35356	$1,6906.10^{-2}$	$6,3894.10^{-2}$	$2,3505^0$	0,35 s
0,1768	$1,7075.10^{-3}$	$1,3181.10^{-2}$	$3,1289.10^{-1}$	4,07 s
0,0884	$1,5299.10^{-4}$	$2,4733.10^{-3}$	$3,5778.10^{-2}$	119,77 s

TABLEAU 8.2 – Erreurs  $L^2$  et  $H^1$  pour la solution exacte de Bercovier-Engelman en fonction du pas du maillage  $h$ .

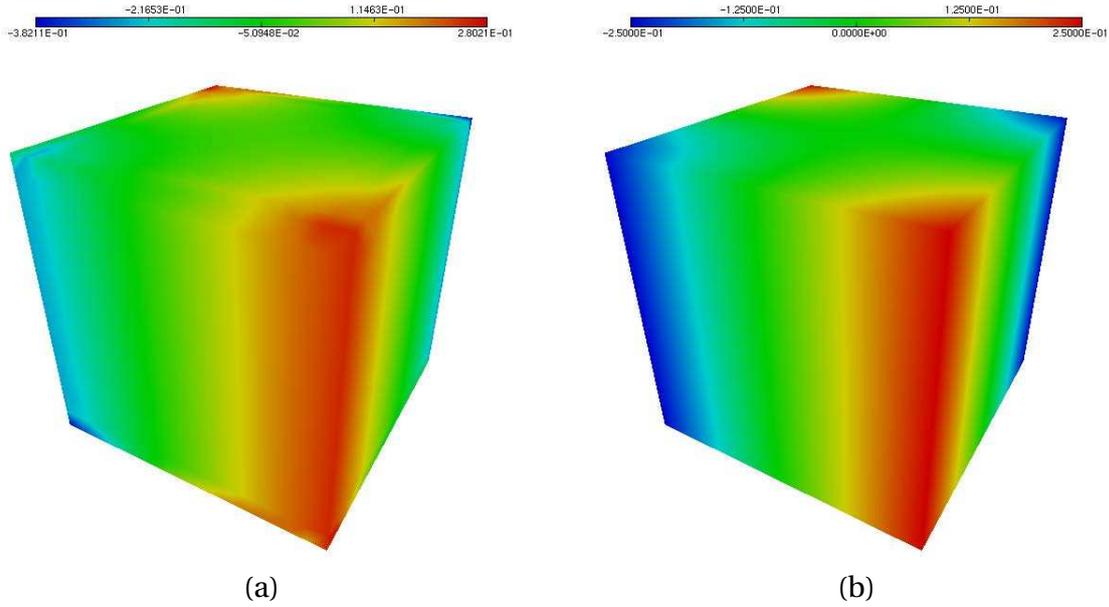


FIGURE 8.4 – Pression sur le cube unité : (a) solution numérique, (b) solution exacte de Bercovier-Engelman.

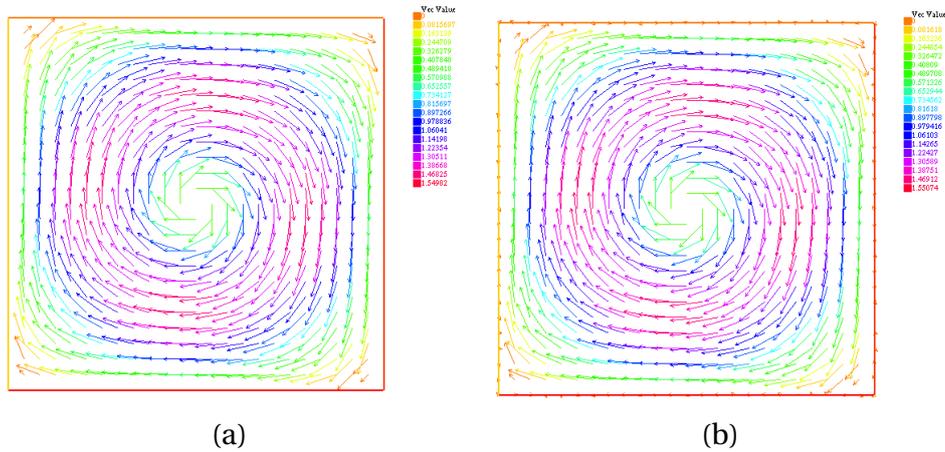


FIGURE 8.5 – Champ de vitesse sur la section  $z = 0,5$  du cube unité : (a) solution numérique, (b) solution exacte de Bercovier-Engelman.

### 8.2.3 Cas-test de Minev-Ethier

La solution exacte de Minev-Ethier est définie sur  $\Omega = [-0,5; 0,5]^3$  par :

$$\begin{aligned}
 u_1 &= -e^x \sin(y+z) - e^z \cos(x+y) \\
 u_2 &= -e^y \sin(z+x) - e^x \cos(y+z) \\
 u_3 &= -e^z \sin(x+y) - e^y \cos(z+x) \\
 p &= -0.5e^{2x} - 0.5e^{2y} - 0.5e^{2z} - \sin(x+y) \cos(z+x) e^{y+z} \\
 &\quad - \sin(y+z) \cos(x+y) e^{z+x} - \sin(z+x) \cos(y+z) e^{x+y}
 \end{aligned}
 \tag{8.4}$$

$$\begin{aligned}
 f_1 = & -v(e^x \sin(y+z) + e^z \cos(x+y)) - e^{2x} - \cos(x+y) \cos(z+x) e^{y+z} \\
 & + \sin(x+y) \sin(z+x) e^{y+z} + \sin(y+z) \sin(x+y) e^{z+x} - \sin(y+z) \cos(x+y) e^{z+x} \\
 & - \cos(z+x) \cos(y+z) e^{x+y} - \sin(z+x) \cos(y+z) e^{x+y}
 \end{aligned}$$

$$\begin{aligned}
 f_2 = & -v(e^y \sin(z+x) + e^x \cos(y+z)) - e^{2y} - \cos(x+y) \cos(z+x) e^{y+z} \\
 & - \sin(x+y) \cos(z+x) e^{y+z} - \cos(y+z) \cos(x+y) e^{z+x} + \sin(y+z) \sin(x+y) e^{z+x} \\
 & + \sin(z+x) \sin(y+z) e^{x+y} - \sin(z+x) \cos(y+z) e^{x+y}
 \end{aligned}$$

$$\begin{aligned}
 f_3 = & -v(e^z \sin(x+y) + e^y \cos(z+x)) - e^{2z} + \sin(x+y) \sin(z+x) e^{y+z} \\
 & - \sin(x+y) \cos(z+x) e^{y+z} - \cos(y+z) \cos(x+y) e^{z+x} - \sin(y+z) \cos(x+y) e^{z+x} \\
 & - \cos(z+x) \cos(y+z) e^{x+y} + \sin(z+x) \sin(y+z) e^{x+y}
 \end{aligned}$$

Les figures 8.6 et 8.7 comparent la solution numérique et la solution exacte de Minev-Ethier, respectivement pour la pression et la vitesse. A l'œil nu, les solutions numériques et exactes sont identiques. Les erreurs relatives sur la vitesse et la pression sont listées dans la table 8.3. On peut remarquer que plus le pas du maillage  $h$  est petit, plus les erreurs diminuent. Les courbes de convergence sont données dans la section 8.2.4.

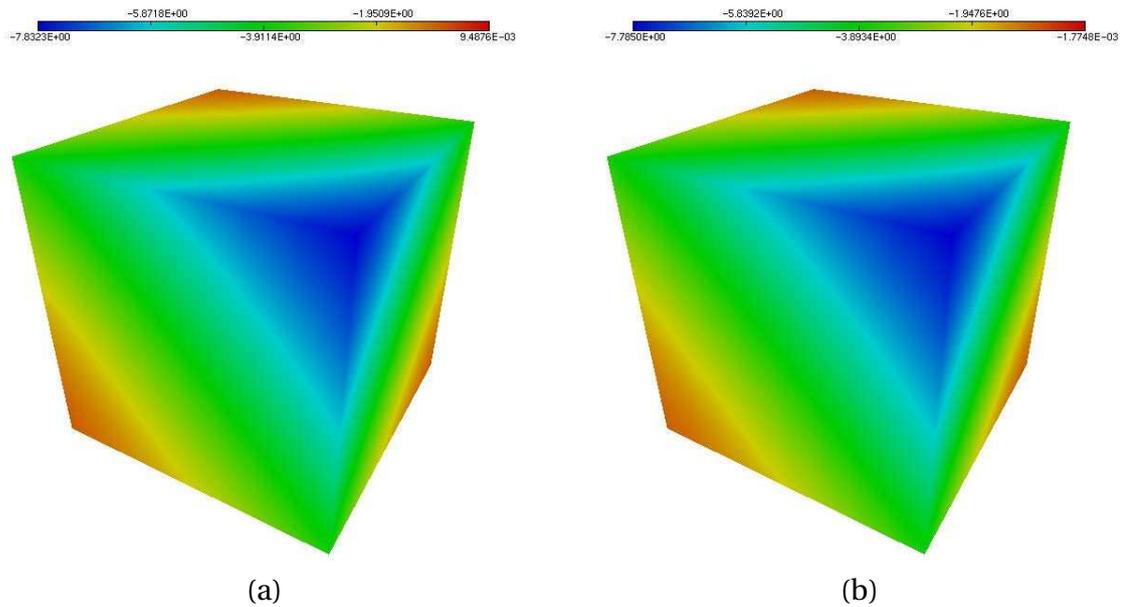


FIGURE 8.6 – Pression sur le cube unité : (a) solution numérique, (b) solution exacte de Minev-Ethier.

### 8.2.4 Courbes de convergence

Les courbes de convergence, illustrées sur la figure 8.8, représentent les erreurs en fonction du pas du maillage à l'échelle logarithmique. Ces courbes confirment la convergence en espace pour la vitesse et la pression, et ce pour les trois solutions analytiques. Les pentes de ces courbes, calculées numériquement, sont proches des pentes attendues (3 pour l'erreur  $L^2$  en vitesse, 2 pour l'erreur  $H^1$  en vitesse et 2 pour l'erreur  $L^2$  en pression).

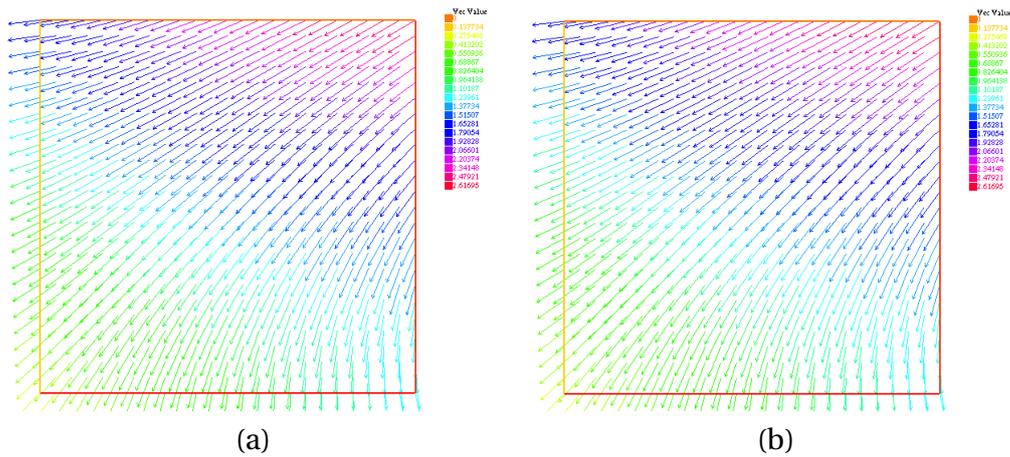


FIGURE 8.7 – Champ de vitesse sur la section  $y = 0$  du cube unité : (a) solution numérique, (b) solution exacte de Minev-Ethier.

$h$	$Er(\mathbf{u})_{L^2}$	$Er(\mathbf{u})_{H^1}$	$Er(p)_{L^2}$	Temps CPU
0,7071	$3,3883 \cdot 10^{-3}$	$2,0663 \cdot 10^{-2}$	$5,9833 \cdot 10^{-2}$	0,07 s
0,3536	$3,6296 \cdot 10^{-4}$	$4,5229 \cdot 10^{-3}$	$1,5365 \cdot 10^{-2}$	0,46 s
0,17678	$2,9771 \cdot 10^{-5}$	$8,0834 \cdot 10^{-4}$	$3,7937 \cdot 10^{-3}$	4,62 s
0,0884	$2,4859 \cdot 10^{-6}$	$1,4105 \cdot 10^{-4}$	$9,4402 \cdot 10^{-4}$	121,31 s

TABEAU 8.3 – Erreurs  $L^2$  et  $H^1$  pour la solution exacte de Minev-Ethier en fonction du pas du maillage  $h$ .

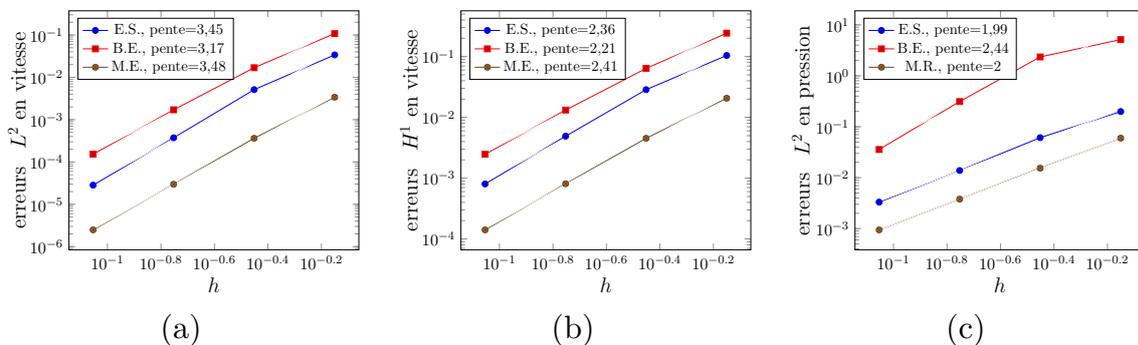


FIGURE 8.8 – Courbes de convergence, à l'échelle logarithmique, pour (a) l'erreur  $L^2$  en vitesse (pente attendue : 3), (b) l'erreur  $H^1$  en vitesse (pente attendue : 2), (c) l'erreur  $L^2$  en pression (pente attendue : 2). E.S., B.E. et M.E. sont les abréviations d'Ethier-Steinman, Bercovier-Engelman et Minev-Ethier, respectivement.

### 8.3 Problème de Navier-Stokes

Dans cette section, nous donnons deux tests numériques afin de valider les deux algorithmes permettant de résoudre les équations de Navier-Stokes décrits dans le chapitre 8. Le premier test est le cas-test d’Ethier-Steinman, vu dans la section précédente ; le second est un test plus proche de la réalité physique, simulant l’écoulement de Poiseuille dans un tube.

#### 8.3.1 Cas-test d’Ethier-Steinman

On cherche à résoudre le problème de Navier-Stokes avec conditions aux limites mêlées (cf. section 7.3) sur le cube unité représenté par la figure 8.1), et partitionné comme suit :  $\Gamma_N = \Gamma_2$  et  $\Gamma_D = \Gamma_1 \cup \Gamma_3 \cup \Gamma_4 \cup \Gamma_5 \cup \Gamma_6$ . Nous avons validé le code sur la solution analytique d’Ethier-Steinman donnée par l’équation (8.2) avec le second membre nul, i.e.  $f_1 = f_2 = f_3 = 0$ . Les erreurs en espace sur la vitesse et sur la pression sont calculées pour les normes de Lebesgue et Sobolev,  $L^2$  et  $H^1$ , données par l’équation (8.1). Pour l’erreur en temps, on note par  $(\mathbf{u}_h^n, p_h^n)$  la solution numérique au temps  $t^n$ , et on calcule l’erreur cumulée de la façon suivante :

$$\begin{aligned} \text{Er}(p)_{\text{cum}} &= \left( \Delta t \sum_{n=0}^{N_T} \|p(t^n) - p_h^n\|_{L^2(\Omega)}^2 \right)^{1/2} \\ \text{Er}(\mathbf{u})_{\text{cum}} &= \left( \Delta t \sum_{n=0}^{N_T} \|\mathbf{u}(t^n) - \mathbf{u}_h^n\|_{L^2(\Omega)}^2 \right)^{1/2} \end{aligned} \quad (8.5)$$

##### 8.3.1.1 Méthode des caractéristiques

Nous avons tout d’abord testé la performance de la méthode des caractéristiques, avec l’accélération donnée par l’algorithme d’Uzawa, décrite dans le chapitre 7, sections 7.3.1 et 7.3.2. Les figures 8.9 et 8.10 comparent la solution numérique, calculée par cette dernière méthode, et la solution exacte d’Ethier-Steinman, respectivement pour la pression et la vitesse au temps final  $T = 0,1$  ( $h = 0,35$  et  $\Delta t = 10^{-3}$ ). Alors que les résultats numériques en vitesse semblent en adéquation avec les résultats exacts, une légère différence de valeur est observable sur les isovaleurs de la pression. Néanmoins, cette erreur diminue lorsque l’on raffine le maillage ou que l’on diminue le pas de temps. On peut effectivement constater cette diminution sur les tableaux 8.4 et 8.5, qui fournissent les erreurs respectivement en espace et en temps. Pour les erreurs en espace, nous avons choisi un pas de temps suffisamment petit,  $\Delta t = 10^{-4}$ , et un pas de maillage  $h$  dans l’intervalle  $[\frac{\sqrt[3]{2}}{16}; \frac{\sqrt[3]{2}}{2}] \approx [0,18; 1,41]$ . Pour tester la précision en temps, nous avons pris  $h \approx 0,18$  et un pas de temps dans l’intervalle  $[0,0125; 0,1000]$ . Les erreurs sont données au temps final  $T = 0,5$ . La figure 8.11 illustre les courbes de convergence, à l’échelle logarithmique, représentant les erreurs en fonction du pas du maillage et du pas de temps. On peut observer que les pentes calculées de ces courbes sont proches des ordres de convergence théoriques. Les pentes attendues en espace sont pour la vitesse : 3 en norme  $L^2$  et 2 en norme  $H^1$ , et pour la pression : 2 en norme  $L^2$  et 1 en norme  $H^1$ . En ce qui concerne la précision en temps, la méthode des caractéristiques est d’ordre 1 en vitesse et en pression.

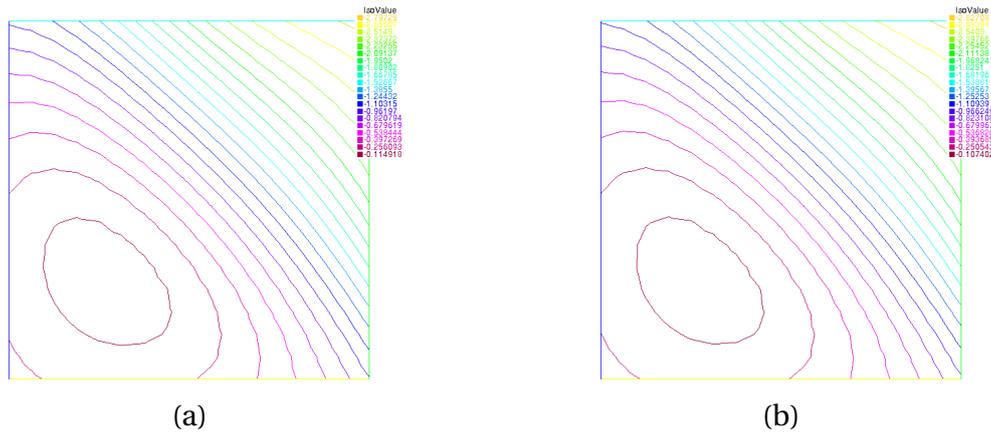


FIGURE 8.9 – Isovaleurs de la pression sur la section  $y = 0$  du cube unité pour la méthode des caractéristiques à  $T = 0, 1$  : (a) solution numérique, (b) solution exacte d’Ethier-Steinman.

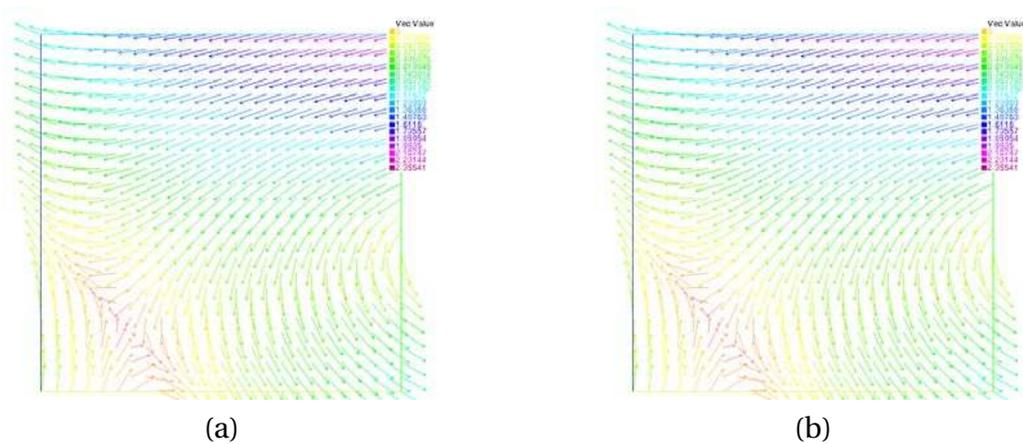


FIGURE 8.10 – Champ de vitesse sur la section  $y = 0$  du cube unité pour la méthode des caractéristiques à  $T = 0, 1$  : (a) solution numérique, (b) solution exacte d’Ethier-Steinman.

$h$	$Er(\mathbf{u})_{L^2}$	$Er(\mathbf{u})_{H^1}$	$Er(p)_{L^2}$	$Er(p)_{H^1}$	Temps CPU
1,4142	$7,4402 \cdot 10^{-2}$	$5,9287 \cdot 10^{-1}$	$5,2087 \cdot 10^{-1}$	$2,8026 \cdot 10^0$	209,21 s
0,7071	$9,1647 \cdot 10^{-3}$	$1,5034 \cdot 10^{-1}$	$5,3952 \cdot 10^{-2}$	$6,0546 \cdot 10^{-1}$	1428,19 s
0,3536	$1,1440 \cdot 10^{-3}$	$3,7971 \cdot 10^{-2}$	$8,4022 \cdot 10^{-3}$	$2,3806 \cdot 10^{-1}$	11102,40 s
0,1768	$4,0233 \cdot 10^{-4}$	$9,5799 \cdot 10^{-3}$	$1,8319 \cdot 10^{-3}$	$1,1179 \cdot 10^{-1}$	92741,30 s

TABEAU 8.4 – Erreurs en espace du problème de Navier-Stokes pour la méthode des caractéristiques en fonction du pas du maillage  $h$  (solution exacte d’Ethier-Steinman).

$\Delta t$	$Er(\mathbf{u})_{cum}$	$Er(p)_{cum}$	Temps CPU
0,1	$4,9314 \cdot 10^{-1}$	$7,9570 \cdot 10^{-1}$	110,15 s
0,05	$2,4811 \cdot 10^{-1}$	$4,7185 \cdot 10^{-1}$	185,07 s
0,025	$1,2453 \cdot 10^{-1}$	$2,5946 \cdot 10^{-1}$	300,72 s
0,0125	$6,2397 \cdot 10^{-2}$	$1,3729 \cdot 10^{-1}$	509,38 s

TABLEAU 8.5 – Erreurs cumulées en temps du problème de Navier-Stokes pour la méthode des caractéristiques en fonction du pas de temps  $\Delta t$  (solution exacte d’Ethier-Steinman).

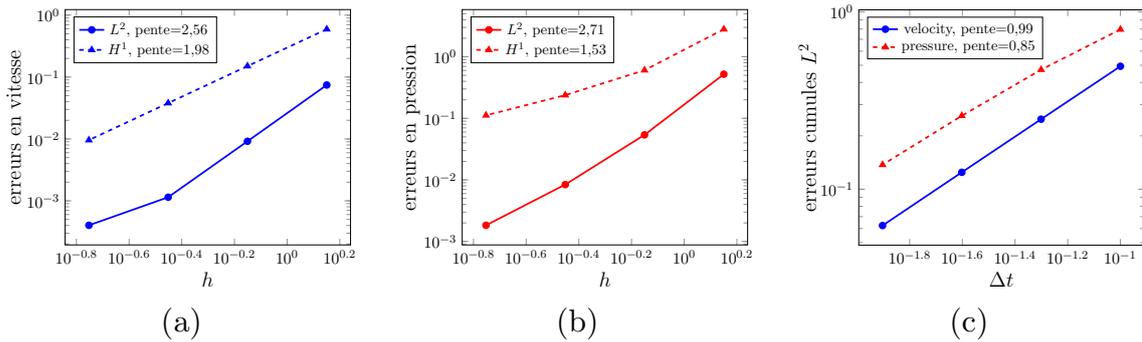


FIGURE 8.11 – Courbes de convergence, à l’échelle logarithmique, de la méthode des caractéristiques pour (a) l’erreur en espace sur la vitesse (pente attendue : 3 en  $L^2$  et 2 en  $H^1$ ), (b) l’erreur en espace sur la pression (pente attendue : 2 en  $L^2$  et 2 en  $H^1$ ), (c) l’erreur cumulée en temps (pente attendue : 1 en vitesse et 1 en pression).

### 8.3.1.2 Schéma découplé d’ordre élevé

Nous avons ensuite testé le schéma découplé d’ordre élevé, décrit dans le chapitre 7, section 7.3.3. Dans [Sheng *et al.*, 2013], ce schéma a été uniquement validé sur une solution analytique 2D du problème de Navier-Stokes. C’est pourquoi il nous a semblé important de le valider sur la solution exacte d’Ethier-Steinman 3D et de comparer les résultats à ceux de la méthode classique des caractéristiques. Les figures 8.12 et 8.13 comparent la solution numérique, calculée par ce schéma découplé, et la solution exacte d’Ethier-Steinman, respectivement pour la pression et la vitesse au temps final  $T = 0,1$  ( $h = 0,35$  et  $\Delta t = 10^{-3}$ ). La solution numérique et la solution exacte semblent identiques en vitesse et en pression. De plus, les isovaleurs de la pression sont égales pour les deux solutions, contrairement à la méthode des caractéristiques. Cette amélioration est probablement due à l’interpolation  $\mathbb{P}_2$  de la pression, et non plus  $\mathbb{P}_1$ . Les erreurs en espace et en temps sont reportés respectivement dans les tableaux 8.6 et 8.7. Pour évaluer la précision en espace et en temps, nous avons choisi les mêmes pas de maillage et de temps que pour la méthode des caractéristiques. Nous avons choisi  $\alpha = 0,5$ , comme il est recommandé dans [Sheng *et al.*, 2013]. Les courbes de convergence, à l’échelle logarithmique, sont données par la figure 8.14, et on peut constater que les pentes calculées numériquement sont proches des pentes théoriques. Les pentes attendues en espace sont 3 pour la norme  $L^2$ , et 2 pour la norme  $H^1$  (pour la vitesse et la pression dans les deux cas). On peut également noter que les pentes numériques en temps correspondent aux ordres de convergence, indiquées dans [Sheng *et al.*, 2013], plus précisément une convergence d’ordre 2 pour la vitesse et la pression.

Ces résultats prouvent que le schéma découplé permet une convergence plus élevée, à la fois en espace et en temps, que la méthode des caractéristiques. Néanmoins, on peut remarquer à l'aide des temps de calculs fournis par les tableaux 8.6 et 8.7, que la première méthode est plus rapide que le schéma découplé dans son implantation actuelle. De plus, le schéma découplé nécessite deux étapes d'initialisation et les effets de ces étapes sur la précision ne sont pas complètement clairs.

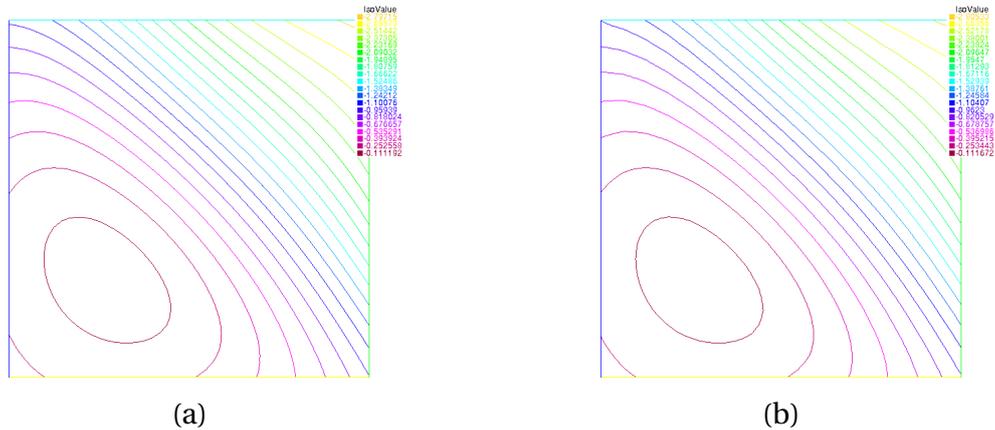


FIGURE 8.12 – Isovaleurs de la pression sur la section  $y = 0$  du cube unité pour le schéma découplé à  $T = 0, 1$  : (a) solution numérique, (b) solution exacte d’Ethier-Steinman.

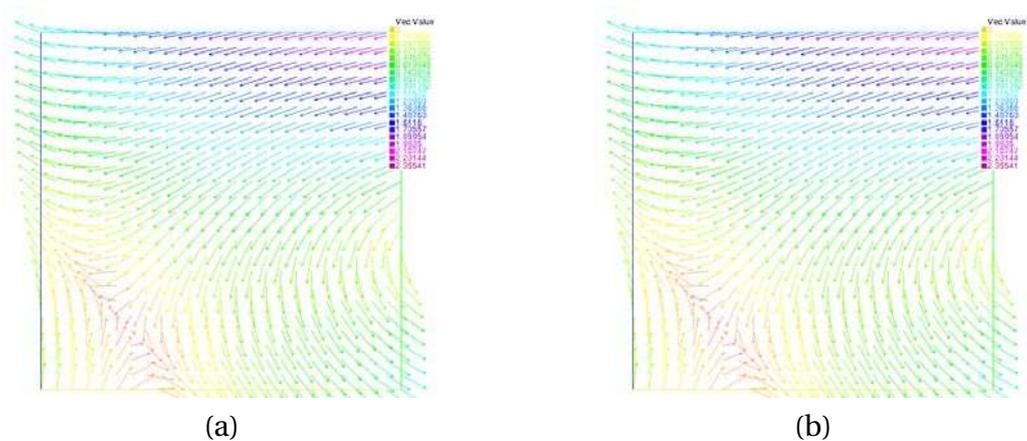
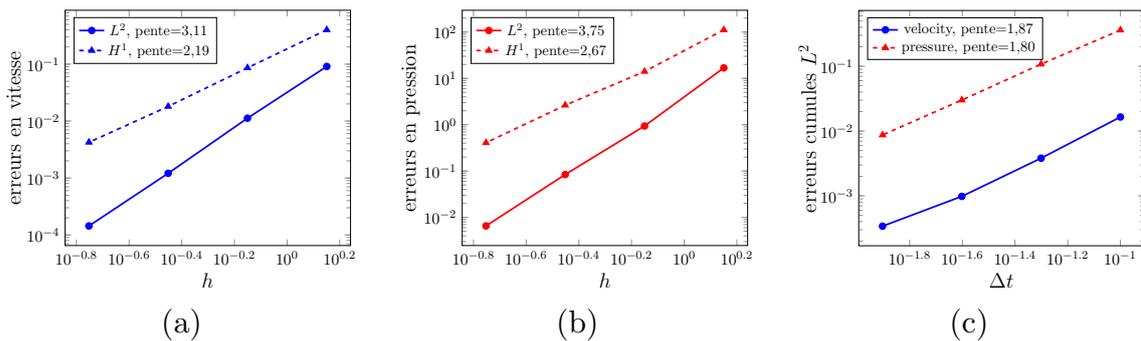


FIGURE 8.13 – Champ de vitesse sur la section  $y = 0$  du cube unité pour le schéma découplé à  $T = 0, 1$  : (a) solution numérique, (b) solution exacte d’Ethier-Steinman.

$h$	$Er(\mathbf{u})_{L^2}$	$Er(\mathbf{u})_{H^1}$	$Er(p)_{L^2}$	$Er(p)_{H^1}$	Temps CPU
1,4142	$9,1042 \cdot 10^{-2}$	$4,0051 \cdot 10^{-1}$	$1,6861 \cdot 10^1$	$1,1159 \cdot 10^2$	269,34 s
0,7071	$1,1230 \cdot 10^{-2}$	$8,6118 \cdot 10^{-2}$	$9,3735 \cdot 10^{-1}$	$1,4098 \cdot 10^1$	1901,04 s
0,3536	$1,2092 \cdot 10^{-3}$	$1,8062 \cdot 10^{-2}$	$8,4201 \cdot 10^{-2}$	$2,6374 \cdot 10^0$	14912,20 s
0,1768	$1,4385 \cdot 10^{-4}$	$4,2365 \cdot 10^{-3}$	$6,5301 \cdot 10^{-3}$	$4,0892 \cdot 10^{-1}$	98947,00 s

 TABLEAU 8.6 – Erreurs en espace du problème de Navier-Stokes pour le schéma découplé en fonction du pas du maillage  $h$  (solution exacte d’Ethier-Steinman).

$\Delta t$	$Er(\mathbf{u})_{cum}$	$Er(p)_{cum}$	Temps CPU
0,1	$1,6493 \cdot 10^{-2}$	$3,6175 \cdot 10^{-1}$	160,45 s
0,05	$3,8135 \cdot 10^{-3}$	$1,0794 \cdot 10^{-1}$	270,05 s
0,025	$9,8238 \cdot 10^{-4}$	$3,0106 \cdot 10^{-2}$	485,65 s
0,0125	$3,4062 \cdot 10^{-4}$	$8,7114 \cdot 10^{-3}$	920,26 s

 TABLEAU 8.7 – Erreurs cumulées en temps du problème de Navier-Stokes pour le schéma découplé en fonction du pas de temps  $\Delta t$  (solution exacte d’Ethier-Steinman).

 FIGURE 8.14 – Courbes de convergence, à l’échelle logarithmique, du schéma découplé pour (a) l’erreur en espace sur la vitesse (pente attendue : 3 en  $L^2$  et 2 en  $H^1$ ), (b) l’erreur en espace sur la pression (pente attendue : 3 en  $L^2$  et 2 en  $H^1$ ), (c) l’erreur cumulée en temps (pente attendue : 2 en vitesse et 2 en pression).

## 8.3.2 Écoulement de Poiseuille

### 8.3.2.1 Description du modèle

On s’intéresse à l’écoulement d’un fluide visqueux incompressible, à travers un cylindre  $]0, L[ \times S$ , où  $S$  est une section circulaire de rayon  $R$ . Nous allons vérifier qu’il existe une solution telle que la pression est constante dans chaque section du tube, et la vitesse est invariante par translation le long de la direction génératrice du cylindre (c’est-à-dire l’axe  $Ox$  dans notre cas, voir figure 8.15). Considérons tout d’abord le problème de Stokes :

$$\begin{cases} -\nu \Delta \mathbf{u} + \nabla p = 0 & \text{dans } ]0, L[ \times S \\ \operatorname{div} \mathbf{u} = 0 & \text{dans } ]0, L[ \times S \\ \mathbf{u} = 0 & \text{sur } ]0, L[ \times \partial S \end{cases} \quad (8.6)$$

avec  $\mathbf{u} = (u_x, u_y, u_z)$  la vitesse du fluide,  $p$  sa pression,  $\nu$  sa viscosité cinématique.

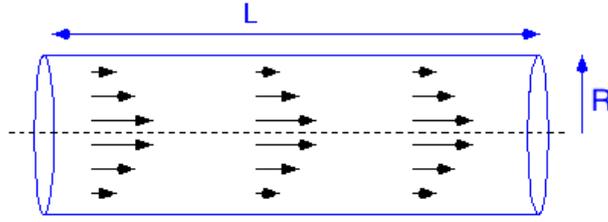


FIGURE 8.15 – Description de l'écoulement de Poiseuille dans un cylindre de longueur  $L$  et de rayon  $R$  : la vitesse est invariante par translation le long de la direction génératrice en pointillé.

Si l'on détaille le problème (8.6), on obtient :

$$\left\{ \begin{array}{l} -\nu \left( \frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial y^2} + \frac{\partial^2 u_x}{\partial z^2} \right) + \frac{\partial p}{\partial x} = 0 \quad \text{dans } ]0, L[ \times S \\ -\nu \left( \frac{\partial^2 u_y}{\partial x^2} + \frac{\partial^2 u_y}{\partial y^2} + \frac{\partial^2 u_y}{\partial z^2} \right) + \frac{\partial p}{\partial y} = 0 \quad \text{dans } ]0, L[ \times S \\ -\nu \left( \frac{\partial^2 u_z}{\partial x^2} + \frac{\partial^2 u_z}{\partial y^2} + \frac{\partial^2 u_z}{\partial z^2} \right) + \frac{\partial p}{\partial z} = 0 \quad \text{dans } ]0, L[ \times S \\ \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} = 0 \quad \text{dans } ]0, L[ \times S \\ \mathbf{u} = 0 \quad \text{sur } ]0, L[ \times \partial S \end{array} \right. \quad (8.7)$$

Étant donné que l'on cherche une solution invariante par translation par rapport à l'axe  $Ox$ , on a l'hypothèse  $\frac{\partial \mathbf{u}}{\partial x} = 0$ , et on peut donc réécrire le système sous la forme :

$$\left\{ \begin{array}{l} -\nu \left( \frac{\partial^2 u_x}{\partial y^2} + \frac{\partial^2 u_x}{\partial z^2} \right) + \frac{\partial p}{\partial x} = 0 \quad \text{dans } ]0, L[ \times S \\ -\nu \left( \frac{\partial^2 u_y}{\partial y^2} + \frac{\partial^2 u_y}{\partial z^2} \right) + \frac{\partial p}{\partial y} = 0 \quad \text{dans } ]0, L[ \times S \\ -\nu \left( \frac{\partial^2 u_z}{\partial y^2} + \frac{\partial^2 u_z}{\partial z^2} \right) + \frac{\partial p}{\partial z} = 0 \quad \text{dans } ]0, L[ \times S \\ \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} = 0 \quad \text{dans } ]0, L[ \times S \\ \mathbf{u} = 0 \quad \text{sur } ]0, L[ \times \partial S \end{array} \right. \quad (8.8)$$

Sous l'hypothèse d'une solution suffisamment régulière, et en dérivant par rapport à  $x$  chacune des équations de (8.8), on a le système suivant :

$$\left\{ \begin{array}{l} -\nu \left( \frac{\partial^3 u_x}{\partial x \partial y^2} + \frac{\partial^3 u_x}{\partial x \partial z^2} \right) + \frac{\partial^2 p}{\partial x^2} = 0 \quad \text{dans } ]0, L[ \times S \\ -\nu \left( \frac{\partial^3 u_y}{\partial x \partial y^2} + \frac{\partial^3 u_y}{\partial x \partial z^2} \right) + \frac{\partial^2 p}{\partial x \partial y} = 0 \quad \text{dans } ]0, L[ \times S \\ -\nu \left( \frac{\partial^3 u_z}{\partial x \partial y^2} + \frac{\partial^3 u_z}{\partial x \partial z^2} \right) + \frac{\partial^2 p}{\partial x \partial z} = 0 \quad \text{dans } ]0, L[ \times S \\ \frac{\partial^2 u_y}{\partial x \partial y} + \frac{\partial^2 u_z}{\partial x \partial z} = 0 \quad \text{dans } ]0, L[ \times S \\ \mathbf{u} = 0 \quad \text{sur } ]0, L[ \times \partial S \end{array} \right. \quad (8.9)$$

on en déduit que :

$$\frac{\partial}{\partial x} \left( \frac{\partial p}{\partial x} \right) = \frac{\partial}{\partial y} \left( \frac{\partial p}{\partial x} \right) = \frac{\partial}{\partial z} \left( \frac{\partial p}{\partial x} \right) = 0$$

et il existe donc une constante  $K \in \mathbb{R}$  telle que :

$$\frac{\partial p}{\partial x} = K$$

d'où :

$$p(x, y, z) = Kx + P(y, z)$$

Si l'on appelle  $P_0$  la pression en  $x = 0$  et  $P_1$  celle en  $x = L$ , on obtient :

$$\begin{aligned} p(x = 0, y, z) = P_0 &\Rightarrow P(y, z) = P_0 \\ p(x = L, y, z) &\Rightarrow K = \frac{P_1 - P_0}{L} \end{aligned}$$

et donc l'expression explicite du champ de pression est :

$$p = \frac{P_1 - P_0}{L}x + P_0$$

le système (8.8) se ramène alors à :

$$\left\{ \begin{array}{ll} -\nu \left( \frac{\partial^2 u_x}{\partial y^2} + \frac{\partial^2 u_x}{\partial z^2} \right) = -K & \text{dans } ]0, L[ \times S \\ -\nu \left( \frac{\partial^2 u_y}{\partial y^2} + \frac{\partial^2 u_y}{\partial z^2} \right) = 0 & \text{dans } ]0, L[ \times S \\ -\nu \left( \frac{\partial^2 u_z}{\partial y^2} + \frac{\partial^2 u_z}{\partial z^2} \right) = 0 & \text{dans } ]0, L[ \times S \\ \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} = 0 & \text{dans } ]0, L[ \times S \\ \mathbf{u} = 0 & \text{sur } ]0, L[ \times \partial S \end{array} \right. \quad (8.10)$$

Compte tenu de l'unicité de la solution d'un tel système [Brézis, 1999], on a  $u_y = u_z = 0$ , et le système se réduit à résoudre :

$$\left\{ \begin{array}{ll} -\nu \left( \frac{\partial^2 u_x}{\partial y^2} + \frac{\partial^2 u_x}{\partial z^2} \right) = -K & \text{dans } \omega \\ u_x = 0 & \text{sur } \partial\omega \end{array} \right. \quad (8.11)$$

où  $\omega$  est une section du cylindre et  $\partial\omega$  sa frontière. En réutilisant l'argument d'unicité de la solution d'un tel système, l'unique solution de (8.11) s'écrit :

$$u_x = \frac{(P_0 - P_1)R^2}{4\nu L} \left( 1 - \frac{y^2 + z^2}{R^2} \right)$$

En conclusion, comme il est montré dans [Soualah-Alila, 2007], on a :

$$\left\{ \begin{array}{l} u_x = \frac{(P_0 - P_1)R^2}{4\nu L} \left( 1 - \frac{y^2 + z^2}{R^2} \right) \\ u_y = 0 \\ u_z = 0 \\ p = \frac{P_1 - P_0}{L}x + P_0 \end{array} \right. \quad (8.12)$$

On remarque que le profil de vitesse est parabolique, avec un maximum au centre du tube. La pression, quant à elle, est constante dans chaque section du tube. On s'intéresse à présent à l'expression du gradient de pression  $|\nabla P| = |P_1 - P_0|$ . Tout d'abord on sait que le débit,  $Q$ , est le volume de fluide traversant une section transversale par unité de volume, d'où :

$$\begin{aligned} Q &= \int_0^R \frac{(P_0 - P_1)R^2}{4\nu L} \left( 1 - \frac{r^2}{R^2} \right) 2\pi r dr \\ &= \frac{\pi(P_0 - P_1)}{2\nu L} \int_0^R (rR^2 - r^3) dr \\ &= \frac{\pi(P_0 - P_1)}{2\nu L} \left[ R^2 r - \frac{r^4}{4} \right]_0^R \\ &= \frac{\pi(P_0 - P_1)R^4}{8\nu L} \end{aligned}$$

On obtient alors la loi de Poiseuille, qui montre que le gradient de pression (ou perte de charge), engendré par les frottements visqueux dans une conduite de longueur  $L$  et de rayon  $R$ , est proportionnelle à  $L$ , au débit  $Q$ , à la viscosité du fluide  $\nu$ , et inversement à  $R$  :

$$\nabla P = \frac{8\nu L Q}{\pi R^4}$$

Finalement, on peut constater que la solution donnée par le système (8.12) est également solution du problème de Navier-Stokes suivant :

$$\begin{cases} \partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = 0 & \text{dans } ]0, L[ \times S \\ \operatorname{div} \mathbf{u} = 0 & \text{dans } ]0, L[ \times S \\ \mathbf{u} = 0 & \text{sur } ]0, L[ \times \partial S \end{cases} \quad (8.13)$$

En effet, on a  $\partial_t \mathbf{u} = 0$ , puisque la vitesse  $\mathbf{u}$  ne dépend pas du temps. De plus, le terme  $\mathbf{u} \cdot \nabla \mathbf{u}$  est nul, car la vitesse est constante le long des lignes de courant et sa dérivée est dans sa propre direction. Il faut néanmoins noter qu'il est nécessaire de choisir  $Re \ll 1$  afin que l'équation (8.12) reste solution du problème. Cette condition permet de négliger les effets inertiels et d'éviter la déstabilisation des profils de vitesse.

### 8.3.2.2 Résultats

Afin de négliger les effets inertiels, nous avons choisi  $Re = 10^{-2}$ . Le cylindre est de longueur  $L = 5$ , de rayon  $R = 0,5$ , et a un pas de maillage  $h = 0,17$ . La figure 8.16 présente les résultats de simulation à  $T = 10^{-4}$  avec un pas de temps  $\Delta t = 10^{-6}$ . Tout d'abord, on peut constater que la vitesse est invariante le long des lignes de courant, parallèles à l'axe du cylindre, et que la pression est constante dans chaque section, comme prévu. On peut également noter que les solutions approchées par la méthode des caractéristiques et le schéma découplé sont quasiment identiques, et qu'elles correspondent à la solution exacte. Le tableau 8.8 fournit les erreurs dans la norme  $L^2$  pour chacune des méthodes. On peut remarquer que les erreurs sont du même ordre de grandeur pour la méthode et le schéma découplé avec des éléments de type  $\mathbb{P}_2 - \mathbb{P}_2$ ; cependant, le temps de calcul est beaucoup plus élevé pour la deuxième méthode. En relâchant la condition  $\mathbb{P}_{k+1} - \mathbb{P}_k$ , ce qui est rendu possible par la deuxième méthode, et en utilisant les éléments  $\mathbb{P}_1 - \mathbb{P}_1$ , on peut observer que le temps de calcul diminue. L'erreur en vitesse augmente, tandis que l'erreur en pression diminue. Pour obtenir un niveau d'erreur équivalent en vitesse, on peut raffiner le maillage. On obtient alors une erreur en vitesse d'ordre  $10^{-5}$  en vitesse et d'ordre  $10^{-4}$  en pression, pour un pas de maillage  $h = 0,078$  et un temps CPU de 2614 s.

En résumé, nous avons validé les deux méthodes permettant de résoudre le problème de Navier-Stokes à l'aide du cas-test d'Ethier-Steinman (la solution numérique converge vers la solution exacte avec l'ordre attendu théoriquement) et du modèle physique de l'écoulement de Poiseuille. Par la suite, nous allons considérer que ces algorithmes sont fiables pour calculer les solutions approchées du modèle réaliste. Bien que le schéma découplé d'ordre élevé semble prometteur, puisque la condition de l'inf-sup peut-être relâchée, celui-ci demande un temps de calcul plus important. C'est pourquoi les résultats du chapitre suivant sont fournis à l'aide de la première méthode uniquement. Par la suite, il serait intéressant de paralléliser le schéma d'ordre élevé afin de diminuer les temps de calculs; ceci est rendu possible par le découplage du calcul de la pression et de la vitesse, contrairement à la méthode des caractéristiques.

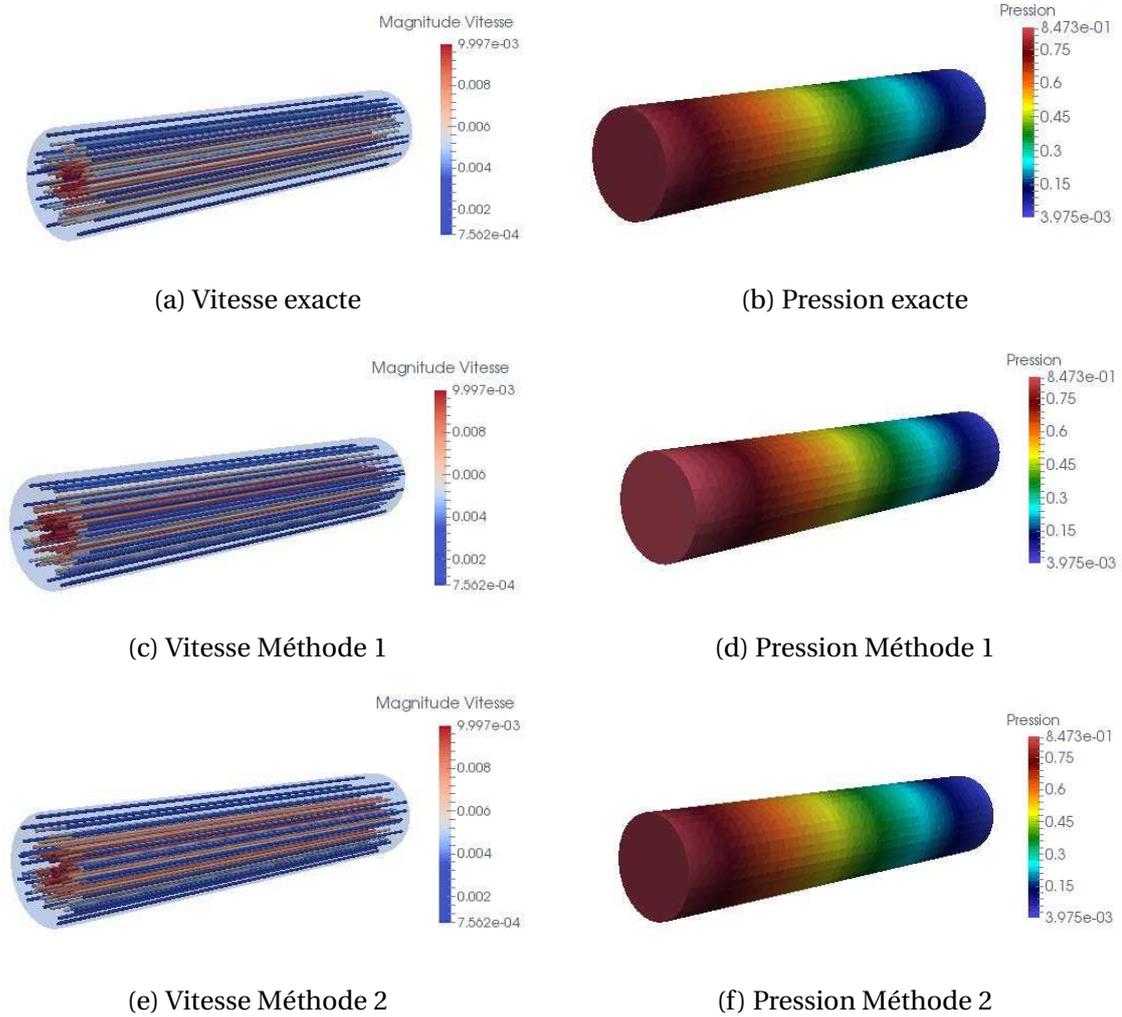


FIGURE 8.16 – Comparaison de la solution exacte pour un écoulement de Poiseuille (1<sup>re</sup> ligne) avec la solution approchée par la méthode des caractéristiques (2<sup>e</sup> ligne) et par le schéma découplé (3<sup>e</sup> ligne). La vitesse est représentée par ses lignes de courant (à gauche) et la pression par ses isovaleurs (à droite).

Méthode	$h$	$\Delta t$	$Er(\mathbf{u})_{L^2}$	$Er(p)_{L^2}$	Temps CPU
Caractéristiques $\mathbb{P}_2 - \mathbb{P}_1$	0,17	$10^{-6}$	$3,3040 \cdot 10^{-5}$	$5,4756 \cdot 10^{-2}$	237,33 s
Schéma découplé $\mathbb{P}_2 - \mathbb{P}_2$	"	"	$3,3160 \cdot 10^{-5}$	$5,6302 \cdot 10^{-2}$	727,57 s
Schéma découplé $\mathbb{P}_1 - \mathbb{P}_1$	"	"	$2,6193 \cdot 10^{-4}$	$5,3181 \cdot 10^{-3}$	314,82 s
Schéma découplé $\mathbb{P}_1 - \mathbb{P}_1$	0,078	"	$6,3014 \cdot 10^{-5}$	$7,6175 \cdot 10^{-4}$	2614,14 s

TABEAU 8.8 – Erreurs pour le problème de l'écoulement de Poiseuille dans un tube.



# Chapitre 9

## Application à des géométries réalistes

*« Soyez réalistes : demandez  
l'impossible. »*

---

Ernesto “Che” Guevara

Ce chapitre fournit des résultats de simulation d'écoulements sanguins dans le réseau vasculaire cérébral, avec, dans un premier temps, le réseau veineux cérébral complet et, dans un second temps, le polygone de Willis appartenant au réseau artériel. Pour cela, nous avons utilisé la méthode des caractéristiques, décrite dans le chapitre 7, afin de résoudre les équations de Navier-Stokes dans ces géométries. Un schéma découplé d'ordre élevé a été également présenté dans le chapitre 7, et semble prometteur. Néanmoins, nous avons montré, dans le chapitre 8, qu'il est trop coûteux en temps, dans son implantation actuelle.

### Sommaire

---

<b>9.1 Introduction</b> . . . . .	<b>150</b>
<b>9.2 Réseau veineux cérébral</b> . . . . .	<b>150</b>
9.2.1 Problème de Stokes . . . . .	150
9.2.2 Problème de Navier-Stokes . . . . .	151
<b>9.3 Polygone de Willis</b> . . . . .	<b>157</b>

---

## 9.1 Introduction

Dans ce chapitre, des résultats sont présentés sur les géométries réalistes du réseau veineux cérébral et du polygone de Willis. Les résultats sont calculés à l'aide de la méthode des caractéristiques, décrite dans le chapitre 7, et codée en FreeFem++. Ces scripts ont été exécutés sur le calculateur ROMEO<sup>1</sup> de l'Université de Reims Champagne-Ardenne. Nous travaillons avec des mesures réelles en  $mm$  et  $\nu = 3,5 \text{ mm}^2/s$ . La pression et la vitesse sont visualisées sur le logiciel ParaView<sup>2</sup>, selon un code couleur pour leur magnitude.

## 9.2 Réseau veineux cérébral

### 9.2.1 Problème de Stokes

Nous avons réussi à résoudre le problème de Navier-Stokes sur notre géométrie du réseau veineux grâce au développement du logiciel FreeFem++ et à l'implantation de la version parallélisable du solveur `sparsesolver` fourni par la librairie MUMPS en 2013. En première approximation, nous avons commencé par étudier le problème de Stokes. Nous avons imposé en entrée un vitesse constante de  $28 \text{ mm/s}$ ; le résultat pour la vitesse est illustrée sur la figure 9.1. Les débits calculés à l'entrée et à la sortie sont identiques, avec une valeur de  $5164 \text{ mm}^3/s$  (voir annexe B pour une comparaison avec la librairie Feel++).

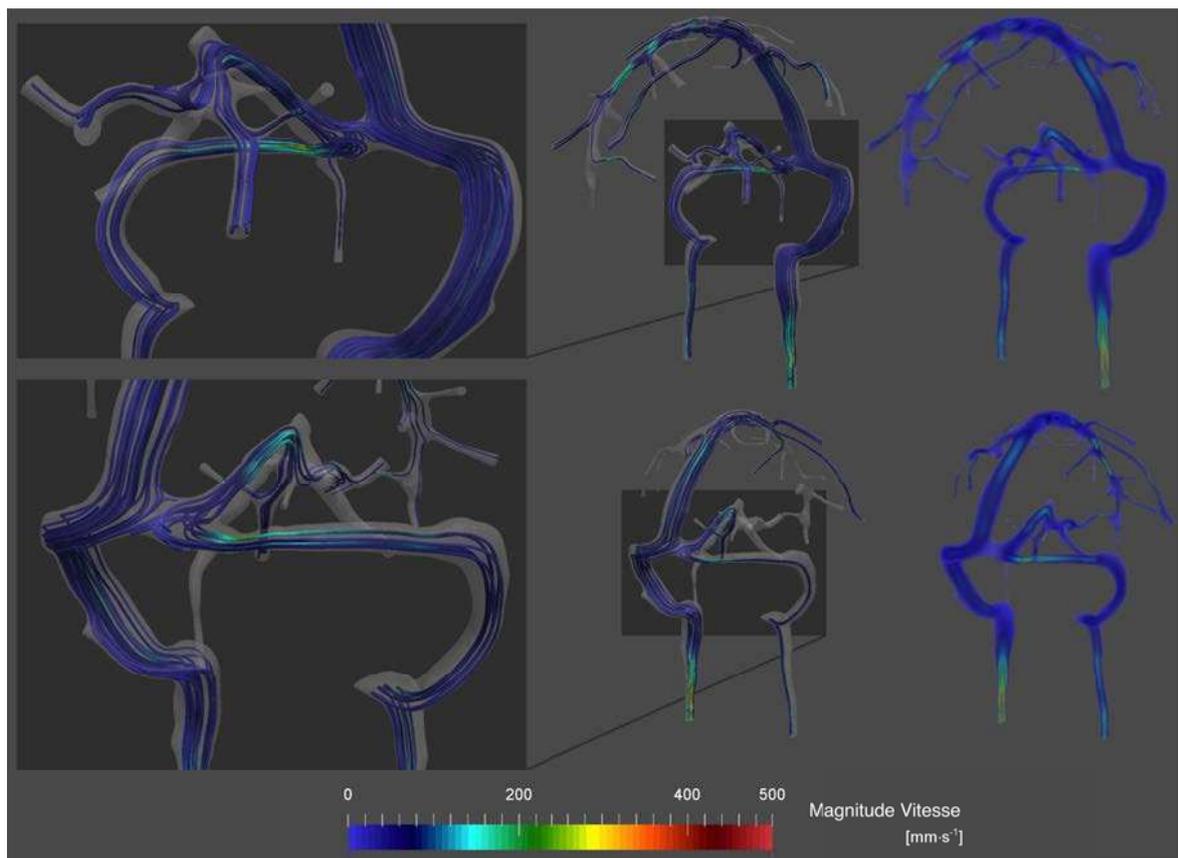


FIGURE 9.1 – Résultats de simulation pour les équations de Stokes : lignes de courant et champ de vitesse.

1. <https://romeo.univ-reims.fr>  
 2. <http://www.paraview.org>

La figure 9.2 illustre la pression, et l'on peut observer une pression anormalement élevée dans une veine anastomotique (en rouge). Ce comportement du flux sanguin non-pertinent est dû à une erreur dans la modélisation géométrique. Il est donc crucial de fournir une géométrie correcte et précise du réseau vasculaire.

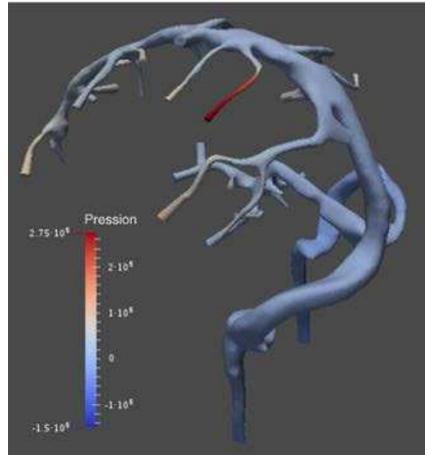


FIGURE 9.2 – Résultats de simulation pour les équations de Stokes : pression. La réduction du diamètre d'un vaisseau peut entraîner une pression anormalement élevée (en rouge).

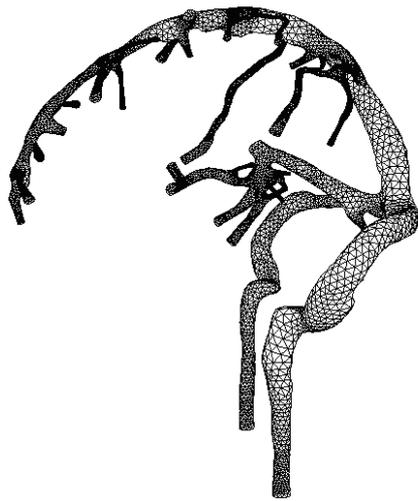
### 9.2.2 Problème de Navier-Stokes

À l'aide de la chaîne allant de l'acquisition des images à la CFD, présentée dans le chapitre 6, section 6.2, nous avons reconstruit cinq maillages différents du réseau veineux cérébral (voir figure 9.3 et tableau 9.1).

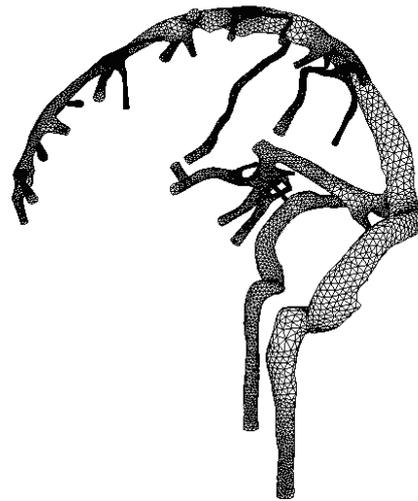
	Maillage 1	Maillage 2	Maillage 3	Maillage 4	Maillage 5
Nombre d'éléments	66 885	94 690	151 634	212 679	535 453
Degrés de liberté	405 663	552 126	884 469	1 148 127	2 692 798
Taille d'élément minimum ( <i>mm</i> )	0,10	0,12	0,08	0,07	0,05
Taille d'élément maximum ( <i>mm</i> )	7,02	6,71	5,60	4,72	3,85

TABLEAU 9.1 – Caractéristiques des 5 maillages du réseau veineux cérébral.

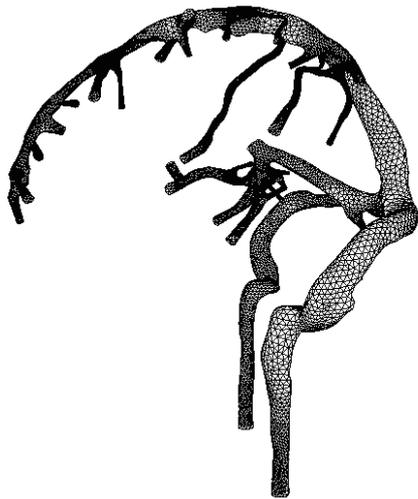
Nous avons exécuté la méthode des caractéristiques pendant 2 cycles cardiaques (i.e.  $\sim 2$  s), avec un pas de temps  $\Delta t = 10^{-2}$  s. La vitesse imposée en entrée est  $\mathbf{u}(\cdot, t) = (1 - e^{-kt})\mathbf{u}_0$ , où  $\mathbf{u}_0 = 50$  mm/s, et la constante  $k$  est choisie de telle sorte que la vitesse soit quasiment constante à 1/3 du premier cycle cardiaque (voir chapitre 6, section 6.4.3). La figure 9.4 illustre le champs de pression pour les cinq différents maillages (itération 135 sur 200, i.e 0,35 s dans le second cycle cardiaque), et on peut remarquer que ceux-ci sont quasiment identiques. La figure 9.6 présente les lignes de courant de la vitesse pour les cinq maillages différents avec des zooms sur la confluence des sinus et le sinus sagittal supérieur pour chaque maillage. On peut observer un comportement



(a) Maillage 1



(b) Maillage 2



(c) Maillage 3



(d) Maillage 4



(e) Maillage 5

FIGURE 9.3 – Comparaison des 5 maillages du réseau veineux cérébral.

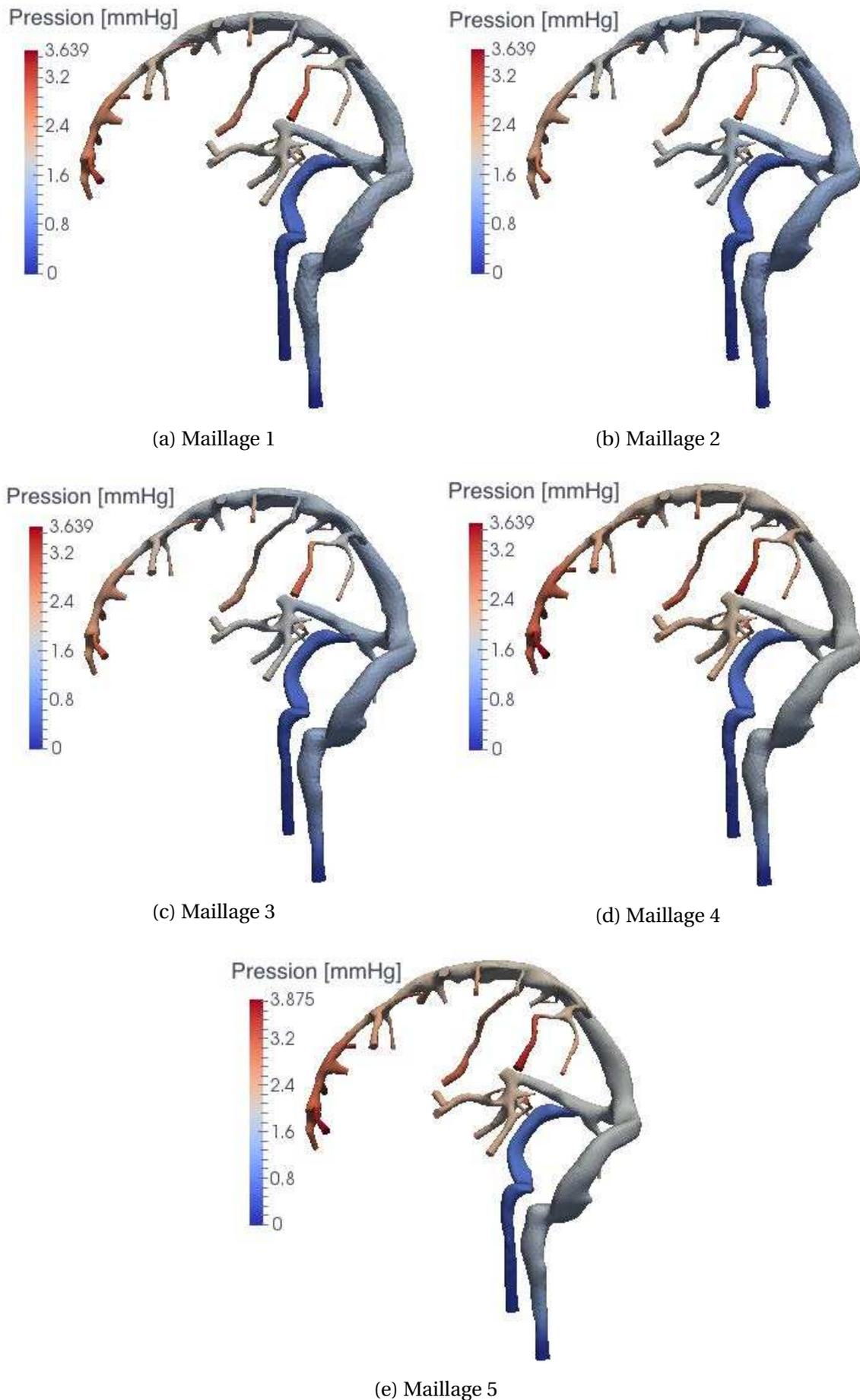


FIGURE 9.4 – Résultats de simulation pour les équations de Navier-Stokes sur le réseau veineux cérébral : champ de pression pour les 5 différents maillages.

du flux sanguin complexe, particulièrement à la confluence des sinus. En effet, une recirculation apparaît juste à l'entrée du sinus transverse droit. On peut expliquer ce comportement asymétrique du flux, au moins en partie, par l'architecture asymétrique du réseau veineux. De plus, on peut noter que plus le maillage est raffiné, plus la recirculation est remarquable.

Notons tout d'abord que pour une meilleure comparaison, nous avons extrait de la littérature uniquement des mesures par IRM dans la position couchée. Dans notre modèle, les vitesses calculées dans le sinus sagittal supérieur vont de  $12,5 \cdot 10^{-2} m/s$  à  $18 \cdot 10^{-2} m/s$ , ce qui est en accord avec les données cliniques, qui sont  $15 \cdot 10^{-2} m/s$  [Schaller, 2004], et  $15,2 \pm 3 \cdot 10^{-2} m/s$  [Gideon *et al.*, 1996] (étude portant sur 14 patients, et utilisant la vélocimétrie par IRM). Les débits calculés à l'entrée et à la sortie sont identiques, ce qui garantit la propriété de la conservation de la masse. Ceux-ci ont une valeur approximative de  $8 ml/s$ , qui correspond aux données physiologiques de la littérature [Stoquart-Elsankari *et al.*, 2009]. Les débits et les temps CPU sont reportés, en fonction des différents maillages, dans le tableau 9.2. On peut constater que le débit final dépend de la précision du maillage, mais tous les débits sont égaux à  $8 \pm 1 ml/s$ . Des valeurs de débits similaires ou légèrement plus élevées, entre  $10 ml/s$  et  $14 ml/s$ , ont été également trouvés dans la littérature [Müller et Toro, 2014a, Müller et Toro, 2014b]. La figure 9.5 compare les débits calculés, pour plusieurs coupes transversales du réseau veineux cérébral, avec les débits mesurés par IRM et extraits de la littérature.

	Maillage 1	Maillage 2	Maillage 3	Maillage 4	Maillage 5
Débit final ( $ml/s$ )	7,888	7,274	7,746	8,856	9,279
Temps CPU (s)	3 447	4 853	8 386	13 747	43 122

TABLEAU 9.2 – Débits calculés aux entrées et sorties du réseau veineux cérébral.

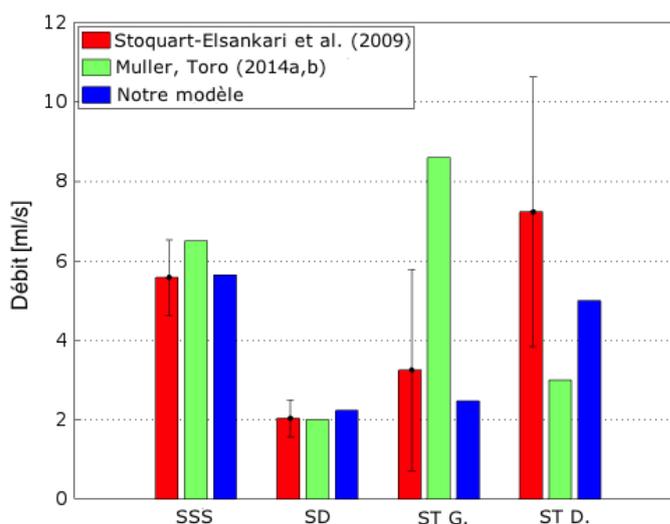
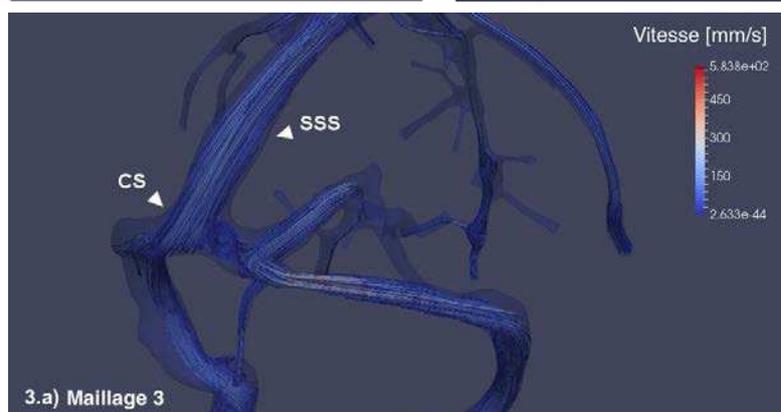
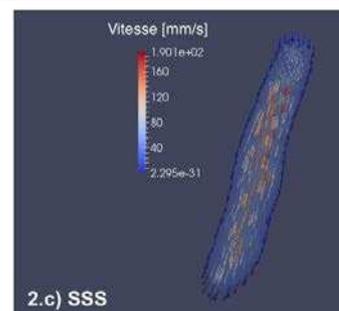
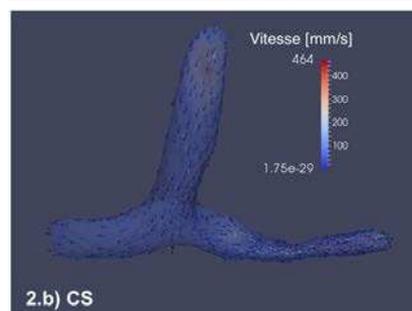
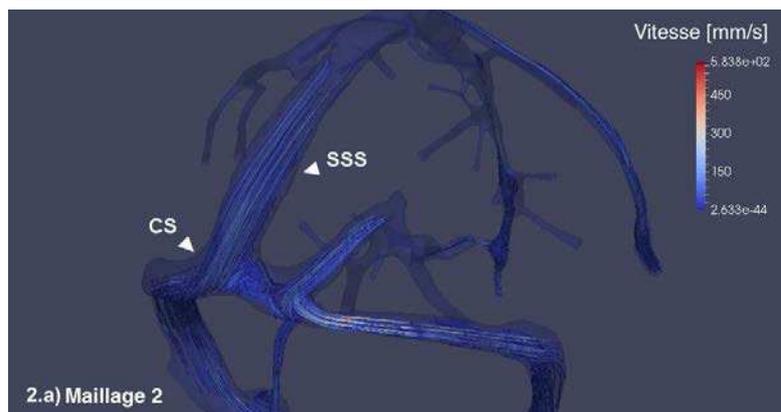
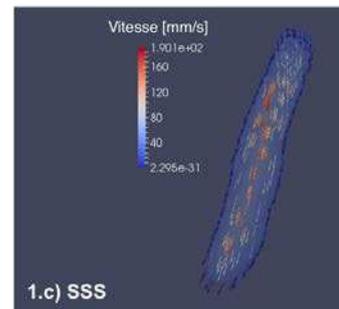
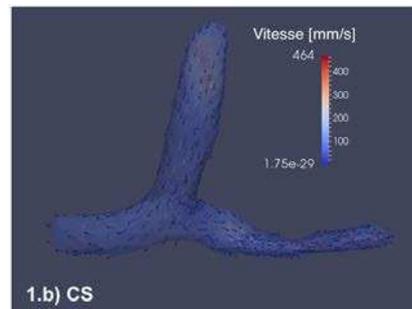
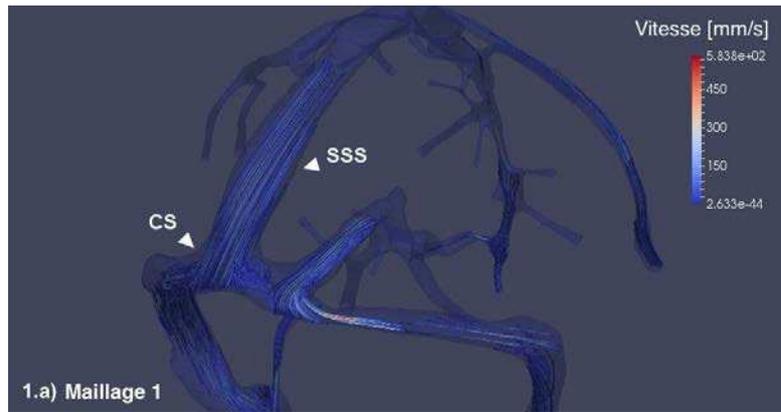


FIGURE 9.5 – Comparaison entre les mesures IRM de débits en  $ml/s$  (moyenne et écart-type), données dans la littérature, et les mesures calculées par notre modèle, à différents endroits du réseau veineux cérébral (voir figure 6.5). SSS : Sinus Sagittal Supérieur (2), SD : Sinus Droit (3), ST G. : Sinus Transverse Gauche (5), ST D. : Sinus Transverse Droit (5).



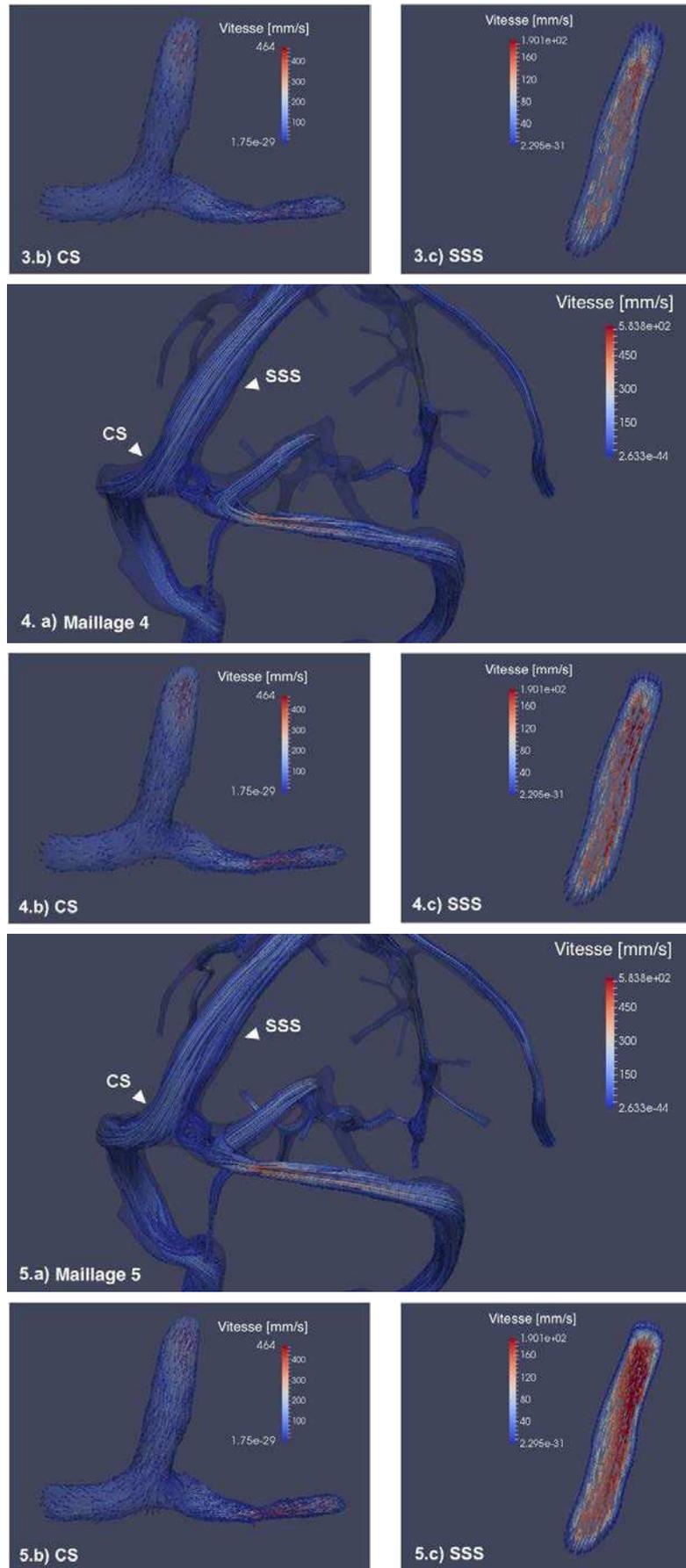


FIGURE 9.6 – Résultats de simulation pour les équations de Navier-Stokes : en haut N.a), lignes de courant de la vitesse, en bas à gauche N.b), champ de vitesse à la confluence des sinus, et en bas à droite N.c), champ de vitesse pour le sinus sagittal supérieur. N correspond au N<sup>e</sup> maillage.

### 9.3 Polygone de Willis

En utilisant la même chaîne qui a permis d'obtenir les différents maillages du réseau veineux cérébral, nous avons reconstruit un maillage du polygone de Willis (voir figure 9.7 et tableau 9.3).

Nombre d'éléments	202 734
Degrés de liberté	973 756
Taille d'élément minimum ( <i>mm</i> )	0,20
Taille d'élément maximum ( <i>mm</i> )	2,30

TABLEAU 9.3 – Caractéristiques du maillage du polygone de Willis.

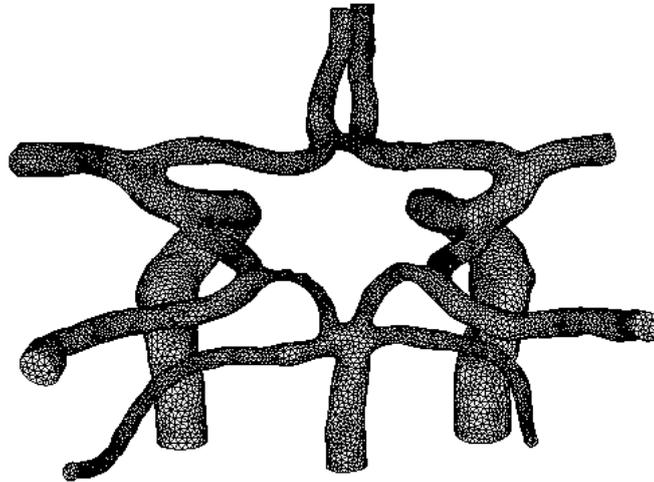
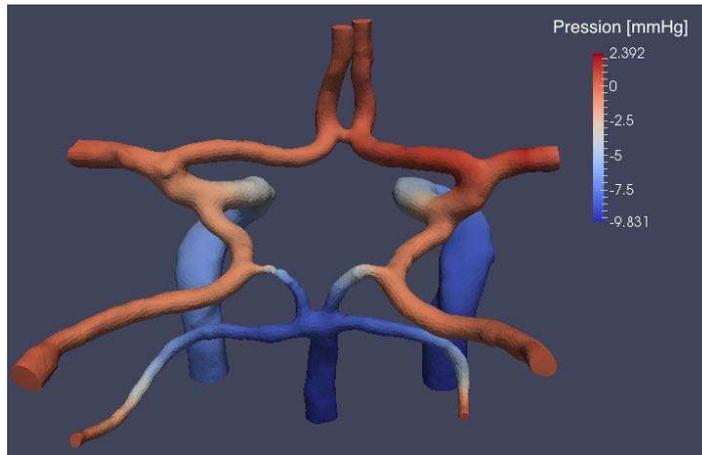


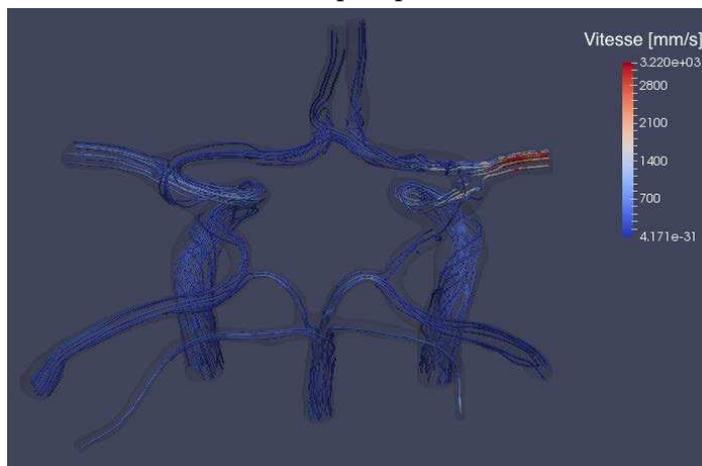
FIGURE 9.7 – Maillage du polygone de Willis.

La méthode des caractéristiques a été également appliquée pendant deux cycles cardiaques, avec le pas de temps  $\Delta t = 10^{-2}$  s. Comme il est décrit dans le chapitre 6, section 6.4.3, nous avons imposé, sur les trois entrées, des profils de vitesse obtenus par des mesures d'IRM et périodisés, de période un cycle cardiaque (voir figure 9.9). La figure 9.8 illustre le champ de pression et les lignes de courant de la vitesse à  $T = 2$  s. Les débits calculés aux entrées et sorties sont identiques ; de plus ils ont la même allure que les profils de vitesse imposés en entrée (voir figure 9.10). Le gradient de pression, qui correspond à la différence entre la pression maximale et la pression minimale, est illustré sur la figure 9.11. Son allure est également identique aux profils de vitesse sur la figure 9.9.

Nous allons discuter du choix de notre modèle dans le chapitre suivant, et fournir quelques pistes afin de le rendre plus réaliste.



(a) Champ de pression



(b) Lignes de courant de la vitesse

FIGURE 9.8 – Résultats de simulation pour les équations de Navier-Stokes sur le polygone de Willis ( $T = 2$  s).

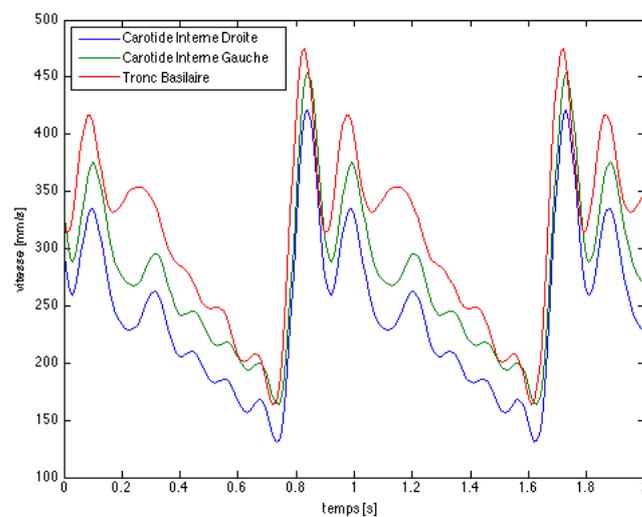


FIGURE 9.9 – Profils de vitesse imposés sur les 3 entrées du polygone de Willis : la carotide interne gauche, la carotide interne droite et le tronc basilaire (en  $mm/s$ ).

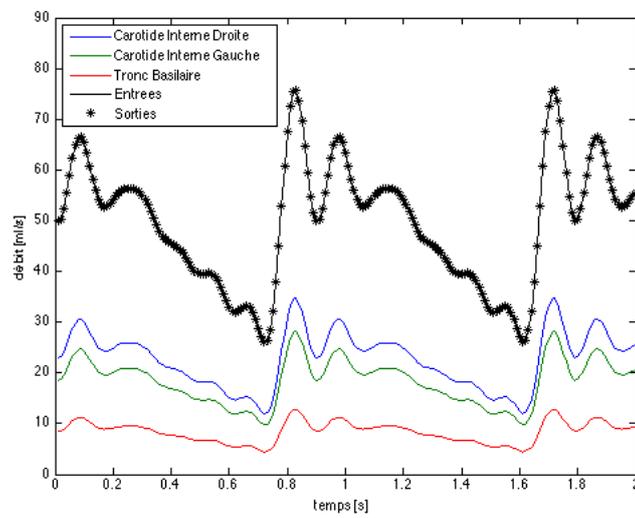


FIGURE 9.10 – Débits calculés sur chaque entrée du polygone de Willis, et débits moyens calculés sur les entrées et sorties (en  $ml/s$ ).

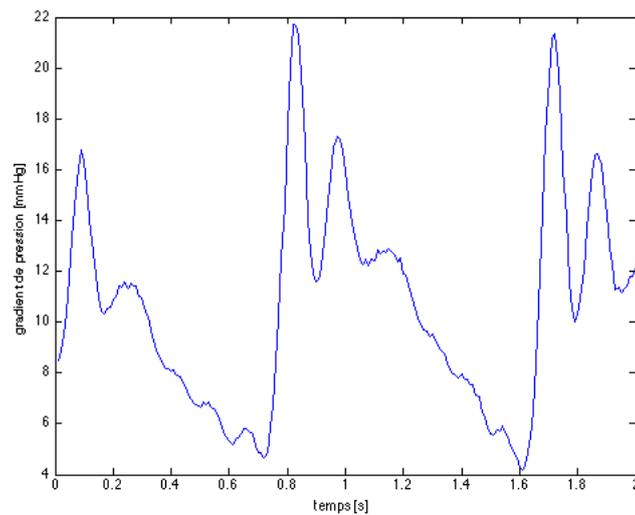


FIGURE 9.11 – Gradient de pression calculé sur le polygone de Willis (en  $mmHg$ ).



# Chapitre 10

## Conclusion et perspectives

*« Ce n'est pas la fin. Ce n'est même pas le commencement de la fin. Mais, c'est peut-être la fin du commencement. »*

---

Winston Churchill

Ce chapitre présente, dans un premier temps, un bilan de la chaîne utilisée pour la simulation avec le choix du modèle (voir chapitre 6) et le choix des algorithmes (voir chapitre 7 pour la description, et chapitre 8 pour la validation). Les résultats du chapitre 9 sont également discutés. Dans un second temps, des pistes sont fournies pour améliorer le modèle (conditions aux limites réalistes, parallélisation de l'algorithme découplé, validation, et liaison entre les outils de segmentation et simulation), et une application clinique est présentée.

### Sommaire

---

<b>10.1 Bilan</b>	<b>162</b>
10.1.1 Modélisation	162
10.1.2 Algorithmes	163
<b>10.2 Perspectives</b>	<b>164</b>
10.2.1 Modélisation	164
10.2.2 Algorithmes	164
10.2.3 Validation	164
10.2.4 Liaison entre segmentation et simulation	164
10.2.5 Application à des cas cliniques	164

---

## 10.1 Bilan

Nous allons discuter, tout d'abord, du choix de notre modèle pour simuler des écoulements sanguins à l'intérieur du réseau vasculaire cérébral. Nous allons ensuite dresser un bilan des algorithmes utilisés pour résoudre les équations de Navier-Stokes qui décrivent le mouvement du fluide.

### 10.1.1 Modélisation

#### Modèle d'ordre réduit vs modèle complet

Les modèles précédents se sont principalement concentrés sur la simulation d'écoulements sanguins, dans les réseaux cérébraux artériel ou veineux, au moyen de modèles d'ordre réduits 0D ou 1D. Néanmoins, les valeurs locales de résistances vasculaires et de compliances restent, en général, inconnues dans le contexte d'études personnalisées. De plus, il a été récemment démontré dans [Xiao *et al.*, 2014] que les résultats des modèles 1D sont satisfaisants pendant la phase de décélération diastolique, mais des différences significatives apparaissent avec les modèles 3D pendant la phase d'accélération systolique ; ceci est en partie expliqué par l'incapacité de ces modèles à prendre en compte le mouvement hélicoïdal induit par la courbure des vaisseaux. Dans le réseau artériel intracrânien, les résultats des modèles 1D concordent avec ceux des modèles 3D lorsque l'élasticité de la paroi artérielle est ignorée (modèle fluide seul) [Grinberg *et al.*, 2011]. Cependant, des fluctuations dans la pression apparaissent et les débits calculés par le modèle fluide seul sont significativement plus élevés que ceux calculés par le modèle fluide-structure. Dans notre étude, qui utilise un modèle 3D, on peut observer un comportement asymétrique du flux avec l'apparition d'une recirculation juste à l'entrée du sinus transverse droit, au début de la phase de décélération. Ce comportement local du flux n'aurait pas pu être capturé par un modèle 1D.

#### Hypothèse newtonienne

Pour la rhéologie du sang, nous avons adopté un modèle newtonien. En plus des arguments développés dans le chapitre 6, section 6.4.1, on peut noter que des résultats récents [Gambaruto *et al.*, 2011] ont évalué la sensibilité des simulations hémodynamiques d'anévrismes cérébraux en fonction de la rhéologie du sang, et que les différences observées entre les solutions sont très petites. De plus, le modèle de viscosité du sang influence dans une moindre mesure les calculs de flux sanguin que les hypothèses de conditions aux limites, en particulier les conditions en entrée, sauf quand un calcul précis de contrainte de cisaillement de la paroi est nécessaire [Chabannes *et al.*, 2015].

#### Modèle fluide seul vs modèle fluide structure

Dans notre modèle, nous avons supposé que les veines et les artères sont rigides. Un modèle pertinent décrivant la déformabilité de la paroi des vaisseaux sanguins reste une question ouverte pour plusieurs raisons : (i) les évaluations *in vivo* des propriétés matérielles ne sont pas parfaitement maîtrisées ; (ii) des vaisseaux de taille moyenne ne sont pas accessibles à partir de l'élastographie par ultrasons pour l'estimation des paramètres rhéologiques ; (iii) les mesures *in vitro* dans des échantillons d'individus décédés sont traitées minutieusement, mais ne sont pas réalisées dans des tissus vivants

irrigués, et (iv) l'état de l'art des interactions fluide-structure ne prend pas en compte la paroi à plusieurs couches constituée de matériaux composites. À notre connaissance, une étude récente de l'écoulement veineux cérébral (jugulaires uniquement) est aussi limitée au cas des parois rigides [Caiazzo *et al.*, 2015].

### Conditions aux limites

En ce qui concerne le réseau veineux, nous avons imposé une vitesse en entrée quasi-stationnaire. Dans [Chabannes *et al.*, 2015], une analyse de sensibilité de l'amplitude d'une telle vitesse a permis de montrer que les conditions aux limites jouent un rôle important sur les forces créées par le flux sanguin dans les veines cérébrales. Il est donc nécessaire de récupérer ces valeurs de vitesse à partir de données cliniques. Néanmoins, les limitations techniques ne permettent pas ces mesures dans des vaisseaux de petites tailles comme les veines d'entrée de notre modèle. Par contre, en ce qui concerne le polygone de Willis, nous avons modelé le profil de vitesse en entrée sur des mesures obtenues par IRM, et les profils de débit du flux sanguin et de pression sont alors fidèles au profil initial. Dans les deux cas, les conditions artificielles d'entrée et de sortie ont été traitées en coupant le domaine de calcul du reste du réseau. Dans les perspectives, nous allons proposer des idées pour améliorer ces conditions.

## 10.1.2 Algorithmes

### Méthode des caractéristiques vs schéma découplé

Nous avons comparé deux algorithmes permettant de résoudre les équations de Navier-Stokes avec des conditions aux limites mêlées. Ici, nous nous ne sommes pas intéressés à l'existence et l'unicité de ces équations avec de telles conditions. Cependant, il a déjà été montré que la méthode des caractéristiques [Fouchet-Incaux, 2015] et le schéma découplé d'ordre élevé [Sheng *et al.*, 2013] donnent des résultats stables. Nous avons validé ces algorithmes sur des solutions analytiques; les résultats obtenus sont quasiment identiques pour les deux méthodes et concordent avec les solutions exactes. Néanmoins, l'algorithme découplé, dans son implantation actuelle, a pour inconvénient d'exiger plus de temps de calculs. Par conséquent, nous avons appliqué les équations de Navier-Stokes sur nos géométries réalistes uniquement avec la méthode des caractéristiques.

### Résultats

Les résultats de vitesse et de débit, dans le réseau veineux cérébral et le polygone de Willis, coïncident avec les données de la littérature. De plus, les débits calculés aux entrées et aux sorties sont identiques, ce qui garantit la loi de la conservation de la masse. Par contre, on peut constater que les résultats de pression ne sont souvent pas physiologiques comme pour le polygone de Willis (voir figure 9.8), où la pression est négative. Ceci est en partie dû à la difficulté de calculer exactement la pression à partir des équations de Navier-Stokes. En effet, celle-ci est fournie uniquement par son gradient et est donc définie à une constante additive près. Néanmoins le gradient de pression calculé pour le polygone de Willis semble correct (voir figure 9.11).

## 10.2 Perspectives

Nous allons décrire les possibilités pour améliorer la simulation : opter pour des conditions aux limites plus réalistes, utiliser le schéma découplé, valider nos résultats, et connecter l'outil de segmentation à celui de la simulation. Une autre perspective envisageable serait d'appliquer notre simulation à des cas cliniques.

### 10.2.1 Modélisation

Une prochaine étape de ce travail pourrait être de coupler le modèle 3D dans la région d'intérêt avec un modèle d'ordre réduit dans d'autres compartiments du réseau vasculaire (par exemple, la pompe cardiaque dans le cas du réseau veineux et le lit vasculaire dans le cas du réseau artériel). Bien que la pertinence d'un modèle simulant la traction due à l'appel du cœur reste incertaine pour le réseau veineux, le couplage de la solution en sortie du réseau artériel avec un modèle 1D, ou un modèle à paramètres distribués, modélisant le réseau secondaire en aval est primordial [Vignon-Clementel *et al.*, 2006].

### 10.2.2 Algorithmes

Bien que nous avons utilisé la méthode des caractéristiques pour nos géométries réalistes, le schéma d'ordre découplé semble prometteur. Il permet effectivement de relâcher la condition de l'inf-sup, et converge en  $O(\Delta t^2)$ , ce qui permettrait d'utiliser des pas de temps plus grands et de simuler plusieurs cycles cardiaques. Il serait donc intéressant de paralléliser ce schéma afin de diminuer les temps de calcul.

### 10.2.3 Validation

Une des questions ouvertes principales est la validation des résultats en utilisant des fantômes ou, mieux, des mesures *in vivo* par imagerie fonctionnelle. Néanmoins, la résolution en espace et en temps des appareils de mesures, et l'accessibilité des mesures dans les vaisseaux sanguins sont limitées. Le développement d'angiographie virtuelle, qui simule des séquences d'acquisition d'images physiques pour créer des données synthétiques réalistes, fournit une perspective prometteuse [Ford *et al.*, 2005, Jurczuk *et al.*, 2008], notamment au sein du projet VIVABRAIN [Fortin *et al.*, 2016].

### 10.2.4 Liaison entre segmentation et simulation

Comme il a été illustré sur la figure 9.2, un artefact dans la reconstruction de la géométrie peut entraîner une pression anormalement élevée dans un vaisseau sanguin. L'objectif à long terme serait de réaliser une chaîne complète avec les outils de segmentation et de simulation. Cet outil permettrait d'extraire le volume vasculaire, de simuler l'écoulement sanguin à l'intérieur du volume maillé et de corriger interactivement un défaut de géométrie, si besoin.

### 10.2.5 Application à des cas cliniques

Dans [Wolff *et al.*, 2013], les complications neurovasculaires, et plus particulièrement les accidents vasculaires cérébraux, sont étudiés chez des consommateurs de cannabis. Il serait intéressant d'utiliser notre modèle pour simuler des écoulements sanguins dans

des polygone de Willis de ces sujets, afin d'étudier le comportement du flux. Un exemple de maillage est donné par la figure 10.1.

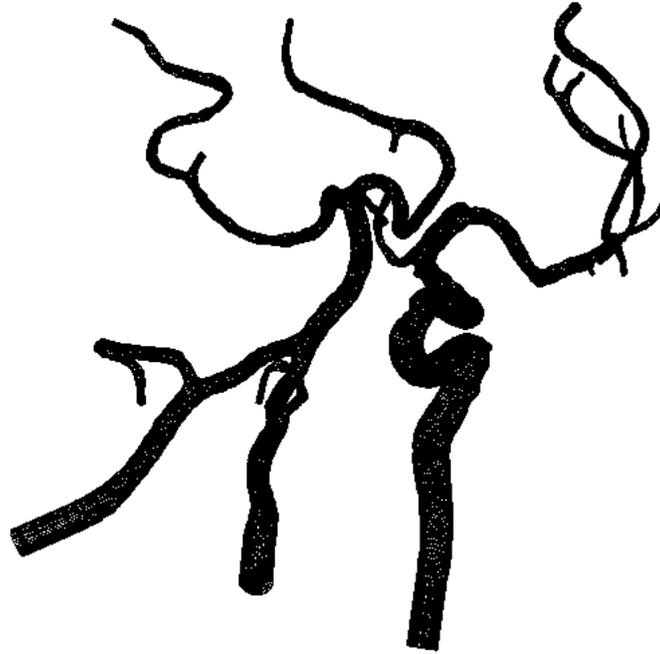


FIGURE 10.1 – Exemple d'un maillage de polygone de Willis d'un cas clinique [Wolff *et al.*, 2013].



# **Troisième partie**

## **Annexes**



# Annexe A

## Transformée de Legendre-Fenchel

### A.1 Transformée de Legendre vs transformée de Legendre-Fenchel [Gonçalves, 2013]

Soit  $F$  une fonction convexe et suffisamment régulière. Il existe une unique correspondance entre  $x$  et  $p(x) = \frac{dF(x)}{dx}$ , puisque  $p(x)$  est une fonction monotone de  $x$ . Si l'on veut une fonction de  $p$ , on peut chercher l'inverse de  $p(x)$ , appelé  $x(p)$  et l'appliquer à  $F(x)$ . La transformée de Legendre définie par  $F^*(p) = \langle p, x \rangle - F(x(p))$  est un exemple de fonction de  $p$ .

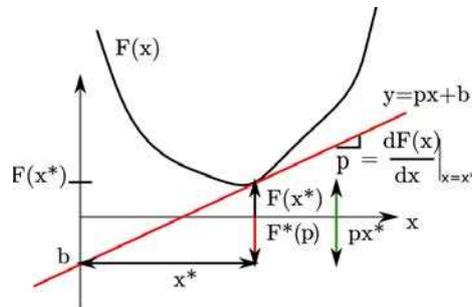


FIGURE A.1 – Transformée de Legendre. Géométriquement, la transformée de Legendre  $F^*(p)$  correspond au contraire de  $b$ , où  $b$  représente l'ordonnée à l'origine de la tangente à  $F(x)$  en  $x^*$ .

L'idée, pour retrouver cette expression, est d'écrire l'équation de la tangente en  $x$  :  $y = \langle p, x \rangle + b$ , où  $b$  représente l'ordonnée à l'origine de la droite. Or, on sait que lorsque  $x = x^*$ , la tangente passe par  $F(x^*)$ , d'où :

$$\begin{aligned} F(x^*) &= \langle p, x^* \rangle + b \\ -b &= \langle p, x^* \rangle - F(x^*) \end{aligned}$$

On peut alors poser  $-b = F^*(p)$ , car le terme dépend de  $x^*$  et donc  $p$ ; on retrouve alors l'expression de la transformée de Legendre :

$$F^*(p) = \langle p, x \rangle - F(x)$$

La figure A.1 donne une représentation géométrique de la transformée de Fenchel. Celle-ci a de bonnes propriétés (elle est convexe et la double transformée de Legendre

d'une fonction est égale à la fonction elle-même). Cependant, elle est limitée aux fonctions convexes et différentiables (voir figure A.2).

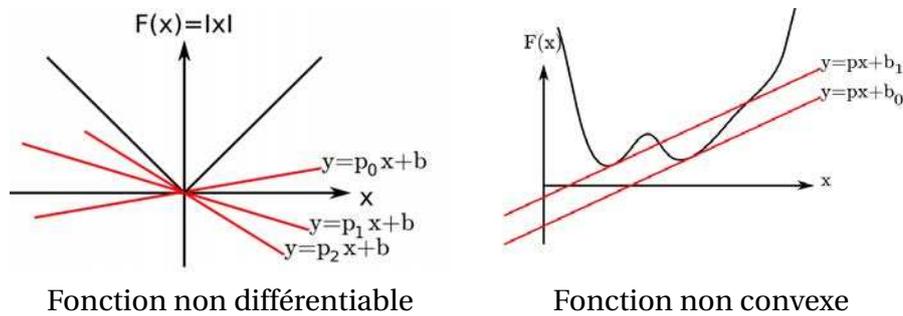


FIGURE A.2 – Exemples de fonctions non adaptées à la transformée de Legendre. Pour la fonction non différentiable  $F(x) = |x|$ , la pente en 0 n'existe pas. En ce qui concerne la fonction non convexe, il est impossible de créer une unique correspondance entre  $x$  et  $p$ .

La solution consiste alors à chercher la valeur de  $b$  la plus petite possible pour une droite de pente  $p$  tangente en  $F(x)$ . On sait que  $b = F(x) - \langle p, x \rangle$  et on cherche la plus petite valeur :

$$\begin{aligned} b &= \inf_x (F(x) - \langle p, x \rangle) \\ &= \inf_x (-(\langle p, x \rangle - F(x))) \\ &= -\sup_x (\langle p, x \rangle - F(x)) \end{aligned}$$

Or,  $-b = F^*(p)$ , on obtient alors la transformée de Legendre-Fenchel (voir figure A.3) :

$$F^*(p) = \sup_x (\langle p, x \rangle - F(x))$$

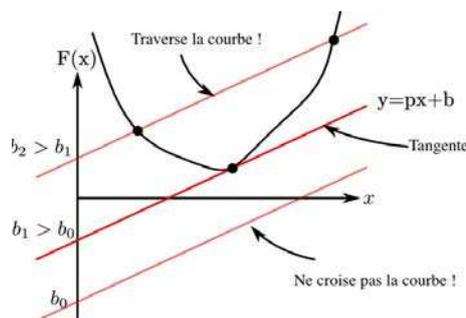


FIGURE A.3 – Transformée de Legendre-Fenchel. Géométriquement, on cherche un point  $(x, F(x))$  tel que la droite de pente  $p$  passant par ce point ait une ordonnée à l'origine  $b$  maximale.

## A.2 Théorème de Fenchel-Moreau [Brézis, 1999]

**Lemme A.1** (Théorème de Hahn-Banach, deuxième forme géométrique). *Soient  $A \in E$  et  $B \in E$  deux ensembles convexes, non vides disjoints. On suppose que  $A$  est fermé et que  $B$  est*

compact. Alors il existe un hyperplan fermé  $H[f = \alpha]$  qui sépare  $A$  et  $B$  au sens strict ( $\exists \varepsilon > 0$  tel que  $f(x) \leq \alpha - \varepsilon, \forall x \in A$  et  $f(x) \geq \alpha + \varepsilon, \forall x \in B$ ).

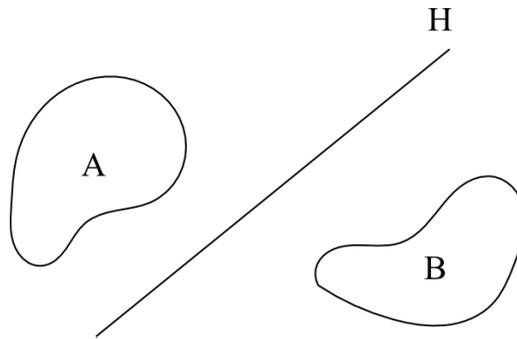


FIGURE A.4 – Hyperplan fermé  $H$  qui sépare  $A$  et  $B$  au sens strict.

**Théorème A.2** (Fenchel-Moreau). On suppose que  $F \in \Gamma^0(\mathbb{R}^n)$ ; alors  $F = F^{**}$  avec

$$F^{**}(p) = \sup_{x \in \mathbb{R}^n} (\langle p, x \rangle - F^*(x))$$

*Démonstration.* La preuve est organisée en deux étapes : la première étape démontre le théorème dans le cas où  $F \geq 0$  et la deuxième étape construit une fonction intermédiaire  $\bar{F} \geq 0$  afin de se ramener au cas général.

**1<sup>re</sup> étape :** On suppose que  $F \geq 0$ . D’abord il est clair que  $F^{**} \leq F$ , en effet d’après la définition de  $F^*$  on a :

$$\langle y, x \rangle \leq F(x) + F^*(y) \quad \forall x \in \mathbb{R}^n, \forall y \in \mathbb{R}^n$$

Pour prouver que  $F^{**} = F$ , on raisonne par l’absurde et on suppose qu’il existe un  $x_0 \in X$  tel que  $F^{**}(x_0) < F(x_0)$ .

Éventuellement, on a  $F(x_0) = +\infty$ , mais on a toujours  $F^{**}(x_0) < +\infty$ . On applique le théorème de Hahn-Banach dans l’espace  $\mathbb{R}^n \times \mathbb{R}$  avec  $A = \text{epi}(F)$  et  $B = [x_0, F^{**}(x_0)]$ .

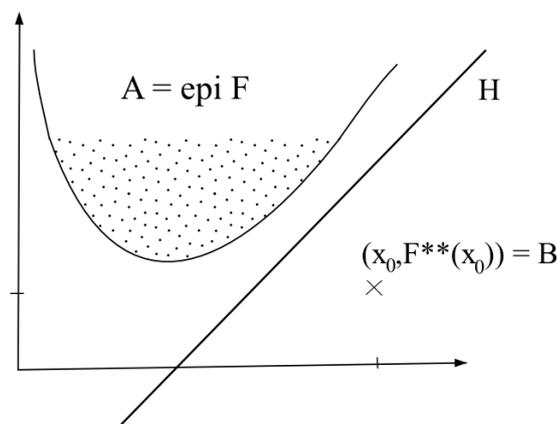


FIGURE A.5 – Application du théorème de Hahn-Banach dans l’espace  $\mathbb{R}^n \times \mathbb{R}$  avec  $A = \text{epi}(F)$  et  $B = [x_0, F^{**}(x_0)]$ .

Il existe donc un hyperplan fermé  $H$  dans  $\mathbb{R}^n \times \mathbb{R}$ , d'équation  $[\phi = \alpha]$ , qui sépare strictement  $A$  et  $B$ . Noter que l'application  $x \in \mathbb{R}^n \mapsto \phi([x, 0])$  est une forme linéaire et continue sur  $\mathbb{R}^n$  et donc  $\phi([x, 0]) = \langle y, x \rangle$  pour un certain  $y \in \mathbb{R}^n$ . En posant  $k = \phi([0, 1])$ , on a alors

$$\phi([x, \lambda]) = \langle y, x \rangle + k\lambda \quad \forall [x, \lambda] \in \mathbb{R}^n \times \mathbb{R}$$

Ecrivant que  $\phi > \alpha$  sur  $A$  et  $\phi < \alpha$  sur  $B$ , on obtient :

$$\langle y, x \rangle + k\lambda > \alpha \quad \forall [x, \lambda] \in \text{epi}F \quad (1)$$

$$\langle y, x_0 \rangle + kF^{**}(x_0) < \alpha \quad (2)$$

On en déduit que  $k \geq 0$  (choisir dans (1),  $x \in \text{dom}F$  et  $\lambda = n \rightarrow +\infty$ ). D'après (1), on a

$$\langle y, x \rangle + kF(x) > \alpha \quad \forall x \in \text{dom}F$$

Soit  $\varepsilon > 0$ , comme  $F \geq 0$ , on a

$$\langle y, x \rangle + (k + \varepsilon)F(x) \geq \alpha \quad \forall x \in \text{dom}F$$

d'où

$$\begin{aligned} \left\langle -\frac{y}{k + \varepsilon}, x \right\rangle - F(x) &\leq -\frac{\alpha}{k + \varepsilon} \\ F^*\left(-\frac{y}{k + \varepsilon}\right) &\leq -\frac{\alpha}{k + \varepsilon} \end{aligned}$$

D'après la définition de  $F^{**}(x_0)$ , il vient

$$\begin{aligned} F^{**}(x_0) &\geq \left\langle -\frac{y}{k + \varepsilon}, x \right\rangle - F^*\left(-\frac{y}{k + \varepsilon}\right) \\ &\geq \left\langle -\frac{y}{k + \varepsilon}, x \right\rangle + \frac{\alpha}{k + \varepsilon} \end{aligned}$$

d'où

$$\begin{aligned} (k + \varepsilon)F^*(x_0) &\geq \langle -y, x_0 \rangle + \alpha \\ \langle y, x_0 \rangle + (k + \varepsilon)F^*(x_0) &\geq \alpha \end{aligned}$$

ce qui contredit (2).

### 2<sup>e</sup> étape: Cas général.

Soit  $y_0 \in \text{dom}F^*$  ( $\text{dom}F^* \neq \emptyset$ ). Pour se ramener au cas précédent, on introduit la fonction

$$\bar{F}(x) = F(x) - \langle y_0, x \rangle + F^*(y_0)$$

de sorte que  $\bar{F}$  est convexe, s.c.i., propre et  $\bar{F} \geq 0$  (car  $F^*(y_0) \geq \langle y_0, x \rangle - F(x)$ ). On sait que  $\bar{F}^{**} = \bar{F}$  (1<sup>re</sup> étape). Calculons maintenant  $\bar{F}^*$  et  $\bar{F}^{**}$ ; on a

$$\begin{aligned} \bar{F}^*(y) &= \sup_x \{ \langle y, x \rangle - F(x) + \langle y_0, x \rangle - F^*(y_0) \} \\ &= \sup_x \{ \langle y + y_0, x \rangle - F(x) \} - F^*(y_0) \\ &= F^*(y + y_0) - F^*(y_0) \end{aligned}$$

$$\begin{aligned} \bar{F}^{**}(x) &= \sup_x \{ \langle y, x \rangle - F^*(y + y_0) + F^*(y_0) \} \\ &= \sup_y \{ -\langle y_0, x \rangle + \langle y + y_0, x \rangle - F^*(y + y_0) \} + F^*(y_0) \\ &= F^{**}(x) - \langle y_0, x \rangle + F^*(y_0) \end{aligned}$$

D'où  $F^{**} = F$

□

## Annexe B

### Comparaison FreeFem++ et Feel++

Dans cette annexe, une validation croisée entre les bibliothèques FreeFem++ et Feel++ est présentée sur la géométrie réaliste du réseau veineux (voir chapitre 6, figure 6.5). Ce travail a été réalisé conjointement avec la doctorante Ranine Tarabay (Université de Strasbourg) lors du CEMRACS 2015. Il fait partie du projet PHANTOM, dont le but est de valider toute la boucle du projet VIVABRAIN sur un fantôme physique représentant une double bifurcation, et s'apparentant à de petits vaisseaux cérébraux de par ses dimensions (voir figure B.1). En premier lieu, les doctorants Simon Garnotel (Université d'Amiens) et Ranine Tarabay ont effectué une validation croisée sur ce fantôme et ont comparé les profils de vitesse et de pression à six différentes sections du fantôme. Les résultats sont comparables et les quelques différences observées sont probablement dues à la méthode de décomposition de domaines utilisée en Feel++.



FIGURE B.1 – Fantôme physique représentant une double bifurcation.

Nous avons voulu étendre cette validation croisée à la géométrie réaliste du réseau veineux, dont le problème reste la validation expérimentale des résultats. C'est pourquoi trouver un moyen de valider cette étape est essentielle. En première approximation, nous avons résolu le problème de Stokes avec conditions aux limites de Dirichlet :

$$\left\{ \begin{array}{ll} -\nu \Delta \mathbf{u} + \nabla p = \mathbf{f} & \text{dans } \Omega \\ \operatorname{div} \mathbf{u} = 0 & \text{dans } \Omega \\ \mathbf{u} = \mathbf{g} & \text{sur } \Gamma \end{array} \right. \quad (\text{B.1})$$

où  $\Omega \subset \mathbb{R}^3$  est un domaine borné avec une frontière lipschitzienne  $\Gamma$ . On rappelle que  $\mathbf{u} = (u_1, u_2, u_3)$  est la vitesse du fluide,  $p$  sa pression et  $\nu$  sa viscosité cinématique ( $3,5 \text{ mm}^2/\text{s}$  pour le sang). La formulation variationnelle est semblable à celle décrite dans le chapitre 7, section 7.2.1 (seuls les termes sur  $\Gamma_N$  doivent être supprimés).

En ce qui concerne les conditions aux limites de Dirichlet, il existe deux manières pour les imposer (voir figure B.2). La première consiste à imposer tout d'abord une vitesse constante en entrée, puis une condition de non-glissement sur la paroi. Dans ce cas, la vitesse est nulle sur la paroi et également sur les éléments qui sont à l'intersection entre l'entrée et la paroi. La seconde manière consiste à imposer tout d'abord la condition de non-glissement sur la paroi, puis une vitesse constante en entrée. La vitesse est alors nulle sur la paroi, à l'exception des éléments qui sont à l'intersection entre l'entrée et la paroi (la vitesse est égale à la vitesse constante imposée en entrée).

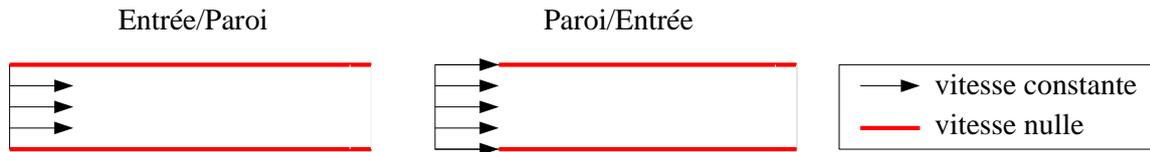


FIGURE B.2 – Deux manières d'imposer des conditions aux limites de Dirichlet.

Nous avons testé le problème de Stokes sur la géométrie réaliste, et nous avons imposé un débit initial de  $6000 \text{ mm}^3/\text{s}$ , ce qui correspond à une vitesse initiale de  $28 \text{ mm}/\text{s}$ . Dans le cas de conditions aux limites du type Paroi/Entrée, les débits ne sont pas conservés : débits en entrée ( $-5985 \text{ mm}^3/\text{s}$  en FreeFem++ et  $-5987 \text{ mm}^3/\text{s}$  en Feel++) et en sortie ( $6227 \text{ mm}^3/\text{s}$  en FreeFem++ et  $6175 \text{ mm}^3/\text{s}$  en Feel++). Étant donné que ces résultats ne vérifient pas la loi physique de la conservation de la masse, nous avons opté pour imposer des conditions aux limites du type Entrée/Paroi. La table B.1 fournit les aires et les débits en pourcentage de l'aire totale et du débit total, respectivement. Les aires calculées en entrée et sortie sont identiques pour les librairies FreeFem++ et Feel++. Les débits totaux en entrée et sortie sont proches pour les deux librairies, et en ce qui concerne les débits en pourcentage, ils sont quasiment identiques. De plus, on peut noter que pour les deux librairies, le débit en entrée est préservé en sortie dans le cas de conditions aux limites du type Entrée/Paroi (respectivement  $5164 \text{ mm}^3/\text{s}$  en FreeFem++ et  $5406 \text{ mm}^3/\text{s}$  en Feel++). Finalement, ces résultats garantissent une simulation correcte.

Label	Aire(%)	Débit	
		FreeFem++ (%)	Feel++ (%)
1	5,9	5,12	5,11
2	1,98	1,96	1,97
3	1,85	1,83	1,84
4	1,26	1,23	1,24
5	1,45	1,43	1,44
6	4,94	4,98	4,97
8	1,25	1,23	1,24
9	2,52	2,52	2,52
10	6,05	6,09	6,07
11	5,23	5,24	5,24
12	6,22	6,28	6,26
13	1,56	1,54	1,55
14	0,85	0,83	0,84
15	6,42	6,41	6,41
16	7,28	7,29	7,29
17	1,70	1,68	1,68
18	1,89	1,87	1,88
19	0,96	0,93	0,94
20	2,43	2,40	2,41
21	0,52	0,51	0,51
22	2,64	2,61	2,62
23	6,30	6,34	6,33
25	3,54	3,54	3,54
26	5,43	5,46	5,45
27	3,92	3,92	3,92
28	5,78	5,83	5,81
29	6,34	6,36	6,35
30	1,18	1,16	1,17
31	3,40	3,39	3,39
Total Entrée	214,85 <i>mm</i> <sup>2</sup>	-5163,93 <i>mm</i> <sup>3</sup> / <i>s</i>	-5405,517 <i>mm</i> <sup>3</sup> / <i>s</i>
7	45,08	69,18	69,1
24	54,92	30,82	30,9
Total Sortie	37,64 <i>mm</i> <sup>2</sup>	5163,78 <i>mm</i> <sup>3</sup> / <i>s</i>	5405,51 <i>mm</i> <sup>3</sup> / <i>s</i>

TABLEAU B.1 – Comparaison des débits dans les veines cérébrales, calculés par FreeFem++ et Feel++. Ces débits sont exprimés en pourcentage du débit total en entrée et total en sortie.



# Annexe C

## Liste des publications et communications issues des travaux de thèse

### C.1 Publications

#### C.1.1 Articles de journal

1. Olivia Miraucourt, Stéphanie Salmon, Marcela Szopos et Marc Thiriet, Blood flow in the cerebral venous system: Modeling and simulation, *Computer Methods in Biomechanics and Biomedical Engineering (CMBBE)*, accepté pour publication.
2. Alexandre Ancel, Alexandre Fortin, Simon Garnotel, Olivia Miraucourt et Ranine Tarabay, Phantom Project: Development and validation of the pipeline from MRA acquisition to MRA simulation, *ESAIM-PROCS*, accepté pour publication.

#### C.1.2 Articles dans des actes de conférences avec comité de lecture

1. Odysée Merveille, Olivia Miraucourt, Stéphanie Salmon, Nicolas Passat et Hugues Talbot, A variational model for thin structure segmentation based on a directional regularization, *23rd International Conference on Image Processing (ICIP)*, accepté pour publication.
2. Olivia Miraucourt, Anna Jezierska, Hugues Talbot, Stéphanie Salmon et Nicolas Passat, Variational method combined with Frangi vesselness for tubular object segmentation, *4th International Conference on Computational and Mathematical Biomedical Engineering (CMBE)*, pp. 485–488, 2015.
3. Olivia Miraucourt, Olivier Gènevaux, Marcela Szopos, Marc Thiriet, Hugues Talbot, Stéphanie Salmon et Nicolas Passat, 3D CFD in complex vascular systems: A case study, *6th International Symposium on Biomedical Simulation (ISBMS)*, pp. 86–94, 2014.
4. Anna Jezierska, Olivia Miraucourt, Hugues Talbot, Stéphanie Salmon et Nicolas Passat, A non-local Chan-Vese model for sparse, tubular object segmentation, *21st International Conference on Image Processing (ICIP)*, pp. 907–911, 2014.
5. Olivia Miraucourt, Stéphanie Salmon, Marcela Szopos et Marc Thiriet, Blood flow simulations in the cerebral venous network, *3rd International Conference on Computational and Mathematical Biomedical Engineering (CMBE)*, pp. 187–190, 2013.

## C.2 Communications

### C.2.1 Posters

1. Olivia Miraucourt, Méthodes variationnelles pour la segmentation d'images médicales, *Journée des Jeunes Chercheurs de la SFR CAP-Santé*, Reims, mars 2015.
2. Olivia Miraucourt, Génération de modèles vasculaires complexes pour la simulation d'écoulements sanguins, *Journée des Jeunes Chercheurs de la SFR CAP-Santé*, Amiens, mars 2014.
3. Olivia Miraucourt, Stéphanie Salmon et Marcela Szopos, Simulation d'écoulements sanguins dans le réseau veineux cérébral, *7<sup>e</sup> Biennale Française des Mathématiques Appliquées et Industrielles SMAI 2015*, Seignosse, mai 2013.

### C.2.2 Communications orales

1. Olivia Miraucourt, Génération de modèles vasculaires cérébraux, *Séminaire Jeunes Chercheurs (LMR)*, Reims, juin 2016
2. Olivia Miraucourt, Approche variationnelle pour la segmentation d'images angiographiques, *Atelier doctorant A3SI (LIGM)*, Champs-sur-Marne, juin 2016
3. Olivia Miraucourt, Génération de modèles vasculaires cérébraux : segmentation de vaisseaux et simulation d'écoulements sanguins, *Séminaire doctorant (LAMFA)*, Amiens, novembre 2015
4. Olivia Miraucourt, Génération de modèles vasculaires cérébraux : segmentation de vaisseaux et simulation d'écoulements sanguins, *Groupe de travail Modélisation numérique et Images (MAP5)*, Paris 5, septembre 2015
5. Olivia Miraucourt, Variational method combined with Frangi vesselness for tubular object segmentation, *Workshop Analyse d'images angiographiques*, Paris 5, juin 2015
6. Olivia Miraucourt, Génération de modèles vasculaires complexes pour la simulation d'écoulements sanguins dans le cerveau, *6<sup>e</sup> Biennale Française des Mathématiques Appliquées et Industrielles SMAI 2013*, Les Karellis, juin 2015.
7. Olivia Miraucourt, Méthodes variationnelles pour la segmentation d'images, *Séminaire Jeunes Chercheurs (LMR)*, Reims, février 2015.
8. Olivia Miraucourt, Le quotidien d'une étudiante en thèse au Laboratoire de Mathématiques de Reims, *MathC2+*, Reims, juin 2014.
9. Olivia Miraucourt, Génération de modèles vasculaires complexes pour la simulation de l'écoulement sanguin, *Ma thèse en 180 secondes*, Reims, mars 2014.
10. Olivia Miraucourt, Simulation d'écoulements sanguins dans le réseau veineux cérébral, *Séminaire Jeunes Chercheurs (LMR)*, Reims, novembre 2013.

## C.3 Autres

1. Co-créditation du site internet <http://numtourcfd.univ-reims.fr> présentant un tour d'horizon des méthodes d'éléments finis et des codes FreeFem++ pour résoudre les équations de Stokes et Navier-Stokes
2. Co-organisation du Séminaire Jeunes Chercheurs du Laboratoire de Mathématiques de Reims <http://jchercheurs.reims.pagesperso-orange.fr>

# Bibliographie

- [Adams et Bischof, 1994] ADAMS, R. et BISCHOF, L. (1994). Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647. 61
- [Aharon *et al.*, 2006] AHARON, M., ELAD, M. et BRUCKSTEIN, A. (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322. 66
- [Albayrak *et al.*, 2007] ALBAYRAK, R., DEGIRMENCI, B., ACAR, M., HAKTANIR, A., COLBAY, M. et YAMAN, M. (2007). Doppler sonography evaluation of flow velocity and volume of the extracranial internal carotid and vertebral arteries in healthy adults. *Journal of Clinical Ultrasound*, 35(1):27–33. 110, 115
- [Alliney, 1992] ALLINEY, S. (1992). Digital filters as absolute norms regularizers. *IEEE Transactions on Signal Processing*, 40(6):1548–1562. 11
- [Anonyme, 2013] ANONYME (2013). Cholestérol : n’avez pas peur! On en a besoin. <http://lespacedesprosdudomicile.over-blog.com/article-cholesterol-n-avez-pas-peur-on-en-a-beso-116343787.html>. 2
- [Anonyme, 2015] ANONYME (2015). Neuronal co-culture – Fluorescent live cell imaging. <http://www.essenbioscience.com/applications/neurite-dynamics-fluorescent-co-culture/>. 60
- [Arbia *et al.*, 2016] ARBIA, G., VIGNON-CLEMENTEL, I., HSIA, T. et GERBEAU, J. (2016). Modified Navier-Stokes equations for the outflow boundary conditions in hemodynamics. <https://hal.archives-ouvertes.fr/hal-01328501/document>. 105
- [Bayes, 1763] BAYES, T. (1763). An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53:370–418. 11
- [Beck et Teboulle, 2009a] BECK, A. et TEBOULLE, M. (2009a). Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Transactions on Image Processing*, 18(11):2419–2434. 81, 82
- [Beck et Teboulle, 2009b] BECK, A. et TEBOULLE, M. (2009b). A Fast Iterative Shrinkage-Thresholding Algorithm for linear inverse problem. *SIAM Journal on Imaging Sciences*, 2(1):183–202. 47
- [Bercovier et Engelman, 1979] BERCOVIER, M. et ENGELMAN, M. (1979). A finite element for the numerical solution of viscous incompressible flows. *Journal of Computational Physics*, 30:181–201. 132
- [Bergmann, 2004] BERGMANN, M. (2004). *Optimisation aérodynamique par réduction de modèle POD et contrôle optimal : application au sillage laminaire d’un cylindre circulaire*. Thèse de doctorat, Institut National Polytechnique de Lorraine. 30, 32, 34, 35, 36
- [Bioucas-Dias et Figueiredo, 2007] BIOUCAS-DIAS, J. M. et FIGUEIREDO, M. A. T. (2007). A new TwIST: Two-Step Iterative Shrinkage/Thresholding algorithms for image restoration. *IEEE Transactions on Signal Processing*, 16(12):2992–3004. 47

- [Boissonnat *et al.*, 2005] BOISSONNAT, J., CHAINE, R., FREY, P., MALANDAIN, G., SALMON, S., SALTEL, E. et THIRIET, M. (2005). From arteriographies to computational flow in saccular aneurisms: the INRIA experience. *Medical Image Analysis*, 9(2):133–143. 106
- [Boldak *et al.*, 2003] BOLDAK, C., ROLLAND, Y., TOUMOULIN, C. et COATRIEUX, J. (2003). An improved model-based vessel tracking algorithm with application to computed tomography angiography. *Journal of Biocybernetics and Biomedical Engineering*, 3(1):41–64. 62
- [Boyd et Vandenberghe, 2004] BOYD, S. et VANDENBERGHE, L. (2004). Statistical estimation. In *Convex Optimization*, chapitre 7. Cambridge University Press. 14
- [Brézis, 1999] BRÉZIS, H. (1999). *Analyse fonctionnelle, Théorie et Applications*. Dunod. 52, 145, IV
- [Buades *et al.*, 2005] BUADES, A., COLL, B. et MOREL, J. (2005). A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530. 61, 65
- [Cahouet et Chabard, 1998] CAHOUET, J. et CHABARD, J. (1998). Some fast 3D finite element solvers for the generalized Stokes problem. *International Journal for Numerical Methods in Fluids*, 8:869–895. 126
- [Caiazzo *et al.*, 2015] CAIAZZO, A., MONTECINOS, G., MÜLLER, L., HAACKE, E. et TORO, E. (2015). Computational haemodynamics in stenotic internal jugular veins. *Journal of Mathematical Biology*, 70(4):745–772. 106, 163
- [Cambier *et al.*, 2012] CAMBIER, J., MASSON, M., MASSON, C. et DEHEN, H. (2012). *Neurologie*. 110
- [Casselles *et al.*, 1993] CASSELLES, V., CATTÉ, E., COLL, T. et DIBOS, F. (1993). A geometric model for active contours in image processing. *Numerische Mathematik*, 66(1):1–31. 16
- [Casselles *et al.*, 1997] CASSELLES, V., KIMMEL, R. et SAPIRO, G. (1997). On geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79. 11, 17
- [Castro *et al.*, 2006] CASTRO, M., PUTMAN, C. et CEBRAL, J. (2006). Computational fluid dynamics modeling of intracranial aneurysms: Effects of parent artery segmentation on intra-aneurysmal hemodynamics. *American Journal of Neuroradiology*, 27(8):1703–1709. 106
- [Cebal *et al.*, 2005] CEBRAL, J., CASTRO, M., APPANABOYINA, S., PUTMAN, C., MILLAN, D. et FRANGI, A. (2005). Efficient pipeline for image-based patient-specific analysis of cerebral aneurysm hemodynamics: Technique and sensitivity. *IEEE Transactions on Medical Imaging*, 24(4):457–467. 104, 106
- [Chabannes *et al.*, 2015] CHABANNES, V., ISMAIL, M., PRUD’HOMME, C. et SZOPOS, M. (2015). Hemodynamic simulations in the cerebral venous network : A study on the influence of different modeling assumptions. *Journal of Coupled Systems and Multiscale Dynamics*, 3(1):23–37. 106, 116, 162, 163
- [Chambolle, 2004] CHAMBOLLE, A. (2004). An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1–2):89–97. 45, 53, 75, 81
- [Chambolle *et al.*, 2010] CHAMBOLLE, A., CASSELLES, V., CREMERS, D., NOVAGA, M. et POCK, T. (2010). An introduction to total variation for image analysis. In *Radon Series on Computational and Applied Mathematics*, volume 9, pages 263–340. 12, 66

- [Chambolle et Pock, 2011] CHAMBOLLE, A. et POCK, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 20(1):120–145. 28, 52
- [Chan *et al.*, 2006] CHAN, T., ESEDOGLU, S. et NIKOLOVA, M. (2006). Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66(5):1632–1648. 20, 81, 97
- [Chan et Vese, 2001] CHAN, T. F. et VESE, L. A. (2001). Active contours without edges. *IEEE Transactions on Signal Processing*, 10(2):266–277. 11, 19
- [Chorin, 1968] CHORIN, A. (1968). Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 22(104):745–762. 118
- [Cohen, 1991] COHEN, L. (1991). On active contour models and ballons: Image understanding. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 53(2):211–218. 16
- [Combettes et Pesquet, 2007] COMBETTES, P. L. et PESQUET, J.-C. (2007). A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):564–574. 49
- [Combettes et Pesquet, 2008] COMBETTES, P. L. et PESQUET, J.-C. (2008). A proximal decomposition method for solving convex variational inverse problems. *Inverse Problems*, 24(6):(article ID : 065014). 28
- [Combettes et Pesquet, 2011] COMBETTES, P. L. et PESQUET, J.-C. (2011). Proximal splitting methods in signal processing. *In Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, chapitre 10, pages 185–212. Springer New York. 39, 50
- [Combettes et Wajs, 2005] COMBETTES, P. L. et WAJS, V. R. (2005). Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4(4):1168–2000. 28
- [Coupé *et al.*, 2008] COUPÉ, P., YGER, P., PRIMA, S., HELLIER, P., KERVRANN, C. et BARILLOT, C. (2008). An optimized blockwise nonlocal means denoising filter for 3-D magnetic resonance images. *IEEE Transactions on Medical Imaging*, 27(4):425–441. 65
- [Dabov *et al.*, 2007] DABOV, K., FOI, A., KATKOVNIK, V. et EGIAZARIAN, K. (2007). Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095. 66
- [Dantzig, 1963] DANTZIG, G. (1963). *Linear Programming and Extensions*. Princeton University Press. 26
- [Dempster *et al.*, 1977] DEMPSTER, A., LAIRD, N. et RUBIN, D. (1977). Maximum likelihood from incomplete data via EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38. 64
- [DeVault *et al.*, 2008] DEVULT, K., GREMAUD, P., NOVAK, V., OLUFSEN, M., VERNIÈRES, G. et ZHAO, P. (2008). Blood flow in the circle of Willis: Modeling and calibration. *Multiscale Modeling & Simulation*, 7:888–909. 115
- [Dijkstra, 1959] DIJKSTRA, E. (1959). A note on two problems in connection with graphs. *Numerische Mathematik*, 1:269–271. 63
- [Douglas et Rachford, 1956] DOUGLAS, J. et RACHFORD, H. (1956). On the numerical solution of heat conduction problems in two or three space variables. *Transactions of the American Mathematical Society*, 82(2):421–439. 28

- [Dufour *et al.*, 2013] DUFOUR, A., TANKYEVYCH, O., NAEGEL, B., TALBOT, H., RONSE, C., BARUTHIO, J., DOKLÁDAL, P. et PASSAT, N. (2013). Filtering and segmentation of 3D angiographic data: Advances based on mathematical morphology. *Medical Image Analysis*, 17(2):147–164. 64, 107
- [El-Zehiry et Grady, 2016] EL-ZEHIRY, N. et GRADY, L. (2016). Contrast driven elastica for image segmentation. *IEEE Transactions on Image Processing*, 25(6):2508–2518. 74
- [Ethier et Steinman, 1994] ETHIER, C. et STEINMAN, D. (1994). Exact fully 3D Navier-Stokes solutions for benchmarking. *International Journal for Numerical Methods in Fluids*, 19:369–375. 132
- [Fabrèges, 2012] FABRÈGES, B. (2012). *Une méthode de prolongement régulier pour la simulation d'écoulements fluide/particules*. Thèse de doctorat, Université Paris-Sud. 105
- [Fleming et Rishel, 1960] FLEMING, W. et RISHEL, R. (1960). An integral formula for total gradient variation. *Archiv der Mathematik*, 11:218–222. 20
- [Ford *et al.*, 2005] FORD, M., STUHNE, G., NIKOLOV, H., HABETS, D., LOWNIE, S., HOLDSWORTH, D. et STEINMAN, D. (2005). Virtual angiography for visualization and validation of computational models of aneurysm hemodynamics. *IEEE Transactions on Medical Imaging*, 24(12):1586–1592. 164
- [Formaggia *et al.*, 2009] FORMAGGIA, L., PERKTOLD, K. et QUARTERONI, A. (2009). Basic mathematical models and motivations. In *Cardiovascular Mathematics. Modeling and Simulation of the Circulatory System*, MS&A, pages 47–75. Springer Milan. 104, 106, 112, 122, 124
- [Fortin *et al.*, 2016] FORTIN, A., SALMON, S., BARUTHIO, J., DELBANY, M. et DURAND, E. (2016). High performance MRI simulation of arbitrarily complex flow: A versatile framework. <https://hal.archives-ouvertes.fr/hal-01326698v1>. 105, 164
- [Fouchet-Incaux, 2015] FOUCHET-INCAUX, J. (2015). *Modélisation, analyse numérique et simulations autour de la respiration*. Thèse de doctorat, Université Paris-Sud. 105, 163
- [Frangi *et al.*, 1998] FRANGI, A., NIESSEN, W., HOOGEVEEN, R., van WALSUM, T. et VIERGEVER, M. (1998). Multiscale vessel enhancement filtering. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI. Lecture Notes in Computer Science*, 1946:130–137. 70, 71
- [Gadda *et al.*, 2015] GADDA, G., TAIBI, A., SISINI, E., GAMBACCINI, M., ZAMBONI, P. et UR-SINO, M. (2015). A new hemodynamic model for the study of cerebral venous outflow. *American Journal of Physiology-Heart and Circulatory Physiology*, 308(3):H217–H231. 105
- [Gambaruto *et al.*, 2011] GAMBARUTO, A., JANELA, J., MOURA, A. et SEQUEIRA, A. (2011). Sensitivity of hemodynamics in a patient specific cerebral aneurysm to vascular geometry and blood rheology. *Mathematical Biosciences and Engineering*, 8(2):409–423. 107, 162
- [Gideon *et al.*, 1996] GIDEON, P., THOMSEN, C., GJERRIS, F., SØRENSEN, P., STÅHLBERG, F. et HENRIKSEN, O. (1996). Measurement of blood flow in the superior sagittal sinus in healthy volunteers, and in patients with normal pressure hydrocephalus and idiopathic intracranial hypertension with phase-contrast cine mr imaging. *Acta Radiologica*, 37(2):171–176. 154
- [Girault et Raviart, 1986] GIRAULT, V. et RAVIART, P. (1986). *Finite Element Methods for Navier-Stokes Equations*. Springer-Verlag. 120, 121

- [Gisolf *et al.*, 2004] GISOLF, J., LIESHOUT, J. V., HEUSDEN, K. V., POTT, F., STOK, W. et KAREMAKER, J. (2004). Human cerebral venous outflow pathway depends on posture and central venous pressure. *The Journal of Physiology*, 560(1):317–327. 116
- [Gonçalves, 2013] GONÇALVES, H. (2013). Legendre and Legendre-Fenchel transforms. <http://www.onmyphd.com/?p=legendre.fenchel.transform>. III
- [Grinberg *et al.*, 2011] GRINBERG, L., CHEEVER, E., ANOR, T., MADSEN, J. et KARNIADAKIS, G. (2011). Modeling blood flow circulation in intracranial arterial networks: A comparative 3D/1D simulation study. *Annals of Biomedical Engineering*, 39(1):297–309. 162
- [Guermont *et al.*, 2006] GUERMOND, J., MINEV, P. et SHEN, J. (2006). An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195(44):6011–6045. 118
- [Guermont et Shen, 2003] GUERMOND, J. et SHEN, J. (2003). A new class of truly consistent splitting schemes for incompressible flows. *Journal of Computational Physics*, 192(1):262–276. 126
- [Hamarneh et Jassi, 2010] HAMARNEH, G. et JASSI, P. (2010). VascuSynth: Simulating vascular trees for generating volumetric image data with ground-truth segmentation and tree analysis. *Computerized Medical Imaging and Graphics*, 34:605–616. 99
- [Hassouna *et al.*, 2006] HASSOUNA, M. S., FARAG, A., HUSHEK, S. et MORIARTY, T. (2006). Cerebrovascular segmentation from TOF using stochastic models. *Medical Image Analysis*, 10(1):2–18. 61
- [Hiriart-Urruty et Lemaréchal, 1993] HIRIART-URRUTY, J.-B. et LEMARÉCHAL, C. (1993). *Convex Analysis and Minimization algorithms II*. Springer-Verlag. 43
- [Ho *et al.*, 2013] HO, H., MITHRARATNE, K. et HUNTER, P. (2013). Numerical simulation of blood flow in an anatomically-accurate cerebral venous tree. *IEEE Transactions on Medical Imaging*, 32(1):85–91. 105, 106, 111, 115
- [Jurczuk *et al.*, 2008] JURCZUK, K., KRETOWSKI, M., ELIAT, P., BELLANGER, J., SAINT-JALMES, H. et BÉZY-WENDLING, J. (2008). A new approach in combined modeling of MRI and blood flow: A preliminary study. In: IEEE International Symposium on Biomedical Imaging–ISBI. pages 812–815. 164
- [Karmarkar, 1984] KARMARKAR, N. (1984). A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395. 26
- [Kass *et al.*, 1988] KASS, M., WITKIN, A. et TERZOPOULOS, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331. 11, 15
- [Kirbas et Quek, 2004] KIRBAS, C. et QUEK, F. (2004). A review of vessel extraction techniques and algorithms. *ACM Computing Surveys*, 36(2):81–121. 61
- [Koenderink, 1984] KOENDERINK, J. (1984). The structure of images. *Biological Cybernetics*, 50:363–370. 68
- [Krissian *et al.*, 2000] KRISSIAN, K., MALANDAIN, G., AYACHE, N., VAILLANT, R. et TROUSSET, Y. (2000). Model-based detection of tubular structures in 3D images. *Computer Vision and Image Understanding*, 80(2):130–171. 62
- [Larrabide *et al.*, 2012] LARRABIDE, I., KIM, M., AUGSBURGER, L., VILLA-URIOL, M., RÜFENACHT, D. et FRANGI, A. (2012). Fast virtual deployment of self-expandable stents: Method and in vitro evaluation for intracranial aneurysmal stenting. *Medical Image Analysis*, 16(3):721–730. 104

- [Lesage *et al.*, 2009] LESAGE, D., ANGELINI, E., BLOCH, I. et FUNKA-LEA, G. (2009). A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes. *Medical Image Analysis*, 13(6):819–845. 61, 62
- [Lindeberg, 1998] LINDBERG, T. (1998). Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):117–156. 68
- [Liu, 2009] LIU, J. (2009). Open and traction boundary conditions for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 228(19):7250–7267. 128
- [Lorenz *et al.*, 1997] LORENZ, C., CARLSEN, I., BUZUG, T., FASSNACHT, C. et WEESE, J. (1997). Multi-scale line segmentation with automatic estimation of width, contrast and tangential direction in 2D and 3D medical images. In: Computer Vision, Virtual Reality and Robotics in Medicine and Medical Robotics and Computer-Assisted Surgery–CVRMed–MRCAS. *Lecture Notes in Computer Science*, 1205:233–342. 70
- [Lorigo *et al.*, 2001] LORIGO, L., FAUGERAS, O., GRIMSON, W., KERIVEN, R., KIKINIS, R., NABAVI, A. et WESTIN, C. (2001). Curves: Curve evolution for vessel segmentation. *Medical Image Analysis*, 5(3):195–206. 61
- [Lupaşcu *et al.*, 2010] LUPAŞCU, C., TEGOLO, D. et TRUCCO, E. (2010). FABC: Retinal vessel segmentation using AdaBoost. *IEEE Transactions on Information Technology in Biomedicine*, 14(5):1267–1274. 79, 84, 91
- [Manniesing *et al.*, 2007] MANNIESING, R., VIERGEVER, M. et NIESSEN, W. (2007). Vessel axis tracking using topology constrained surface evolution. *IEEE Transactions on Medical Imaging*, 26(3):309–316. 61
- [Meckel *et al.*, 2013] MECKEL, S., LEITNER, L., BONATI, L., SANTINI, F., SCHUBERT, T., STALDER, A., LYRER, P., MARKL, M. et WETZEL, S. (2013). Intracranial artery velocity measurement using 4D PC MRI at 3 T: Comparison with transcranial ultrasound techniques and 2D PC MRI. *Neuroradiology*, 55(4):389–398. 115
- [Merveille, 2016] MERVEILLE, O. (2016). *A morphological framework for line-like structures analysis*. Thèse de doctorat, University Paris-Est. 99
- [Merveille *et al.*, 2016a] MERVEILLE, O., MIRAUCOURT, O., SALMON, S., PASSAT, N. et TALBOT, H. (2016a). A variational model for thin structure segmentation based on a directional regularization. In: International Conference on Image Processing – ICIP. accepté pour publication. 74, 99
- [Merveille *et al.*, 2014] MERVEILLE, O., TALBOT, H., NAJMAN, L. et PASSAT, N. (2014). Tubular structure filtering by ranking orientation responses of path operators. In: European Conference on Computer Vision–ECCV. *Lecture Notes in Computer Science*, 8690:203–218. 74
- [Merveille *et al.*, 2016b] MERVEILLE, O., TALBOT, H., NAJMAN, L. et PASSAT, N. (2016b). Tubular structure analysis by ranking the orientation responses of path operators. <https://hal.archives-ouvertes.fr/hal-01262728>. 74, 89
- [Migliorati, 2014] MIGLIORATI, A. (2014). A supervised method by using gray-level and moment invariants-based features. <http://projects.i-ctm.eu/en/project/blood-vessel-segmentation-0>. 59
- [Minev et Ethier, 1998] MINEV, P. et ETHIER, C. (1998). A characteristic/finite element algorithm for the 3-D Navier-Stokes equations using unstructured grids. *Computational Methods in Applied Mathematics*, 178:39–50. 132
- [Miraucourt *et al.*, 2015] MIRAUCOURT, O., JEZERSKA, A., TALBOT, H., SALMON, S. et PASSAT, N. (2015). Variational method combined with Frangi vesselness for tubular object

- segmentation. In: International Conference on Computational and Mathematical Biomedical Engineering–CMBE. pages 485–488. 74
- [Miyazaki et Lee, 2008] MIYAZAKI, M. et LEE, V. (2008). Nonenhanced MR angiography. *Radiology*, 248(1):20–43. 105, 107
- [Moireau et Chapelle, 2011] MOIREAU, P. et CHAPELLE, D. (2011). Reduced-order unscented Kalman filtering with application to parameter identification in large-dimensional systems. *ESAIM: Control, Optimisation and Calculus of Variations*, 17(2):380–405. 105
- [Morales et al., 2013] MORALES, H., LARRABIDE, I., GEERS, A., ROMAN, L. S., BLASCO, J., MACHO, J. et FRANGI, A. (2013). A virtual coiling technique for image-based aneurysm models by dynamic path planning. *IEEE Transactions on Medical Imaging*, 32(1):119–129. 104, 105
- [Mula et al., 2013] MULA, J., LEE, J., LIU, F., YANG, L. et PETERSON, C. (2013). Automated image analysis of skeletal muscle fiber cross-sectional area. *Journal of Applied Physiology*, 114(1):148–155. 60
- [Müller et Toro, 2014a] MÜLLER, L. et TORO, E. (2014a). Enhanced global mathematical model for studying cerebral venous blood flow. *Journal of Biomechanics*, 47(13):3361–3372. 154
- [Müller et Toro, 2014b] MÜLLER, L. et TORO, E. (2014b). A global multiscale mathematical model for the human circulation with emphasis on the venous system. *International Journal for Numerical Methods in Biomedical Engineering*, 30(7):681–725. 105, 111, 112, 154
- [Mumford et Shah, 1989] MUMFORD, D. et SHAH, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685. 11, 18
- [Naegel et al., 2007] NAEGEL, B., PASSAT, N. et RONSE, C. (2007). Grey-level hit-or-miss transforms–part II: Application to angiographic image processing. *Pattern Recognition*, 40(2):648–658. 64
- [Najman et Talbot, 2010] NAJMAN, L. et TALBOT, H. (2010). *Mathematical Morphology: From theory to applications*. J. Wiley & Sons. 61
- [Nesterov, 1983] NESTEROV, Y. (1983). A method of solving a convex programming problem with convergence rate  $o(1/k^2)$ . *Soviet Mathematics–Doklady*, 27(2):372–376. 47, 48, 82
- [Nesterov, 2009] NESTEROV, Y. (2009). Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259. 48
- [Niemeijer et al., 2004] NIEMEIJER, M., STAAL, J., van GINNEKEN, B., LOOG, M. et ABRAMOFF, M. (2004). Comparative study of retinal vessel segmentation methods on a new publicly available database. In: SPIE Medical Imaging. 5370:648–656. 79, 84, 91
- [Ogoh et al., 2011] OGOH, S., SATO, K., SEIFERT, J. F. T., OVERGAARD, M. et SECHER, N. (2011). The effect of phenylephrine on arterial and venous cerebral blood flow in healthy subjects. *Clinical Physiology and Functional Imaging*, 31(6):445–451. 115
- [Olabarriaga et al., 2003] OLABARRIAGA, S., BREEUWER, M. et NIESSEN, W. (2003). Minimum cost path algorithm for coronary artery central axis tracking in CT images. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI. *Lecture Notes in Computer Science*, 2879:687–694. 61

- [Osher et Sethian, 1988] OSHER, S. et SETHIAN, J. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79(1):12–49. 16
- [Özsarlak et al., 2004] ÖZSARLAK, Ö., GOETHEM, J. V., MAES, M. et PARIZEL, P. (2004). MR angiography of the intracranial vessels: Technical aspects and clinical applications. *Neuroradiology*, 46(12):955–972. 105
- [Passat et al., 2016] PASSAT, N., SALMON, S., ARMSPACH, J.-P., NAEGEL, B., PRUD’HOMME, C., TALBOT, H., FORTIN, A., GARNOTEL, S., MERVEILLE, O., MIRAUCOURT, O., TARABAY, R., CHABANNES, V., DUFOUR, A., JEZIERSKA, A., BALÉDENT, O., DURAND, E., NAJMAN, L., SZOPOS, M., ANCEL, A., BARUTHIO, J., DELBANY, M., FALL, S., PAGÉ, G., GÉNEVAUX, O., ISMAIL, M., de SOUSA, P. L., THIRIET, M. et JOMIER, J. (2016). From Real MRA to Virtual MRA: Towards an Open-Source Framework. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI. accepté pour publication. 2
- [Passerini et al., 2009] PASSERINI, T., de LUCA, M., FORMAGGIA, L., QUARTERONI, A. et VENEZIANI, A. (2009). A 3D/1D geometrical multiscale model of cerebral vasculature. *Journal of Engineering Mathematics*, 64(4):319–330. 106
- [Perret et al., 2015] PERRET, B., COUSTY, J., TANKYEVYCH, O., TALBOT, H. et PASSAT, N. (2015). Directed connected operators: Asymmetric hierarchies for image filtering and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(6):1162–1176. 64
- [Peyré et al., 2008] PEYRÉ, G., BOUGLEUX, S. et COHEN, L. (2008). Non-local regularization of inverse problems. In: European Conference on Computer Vision–ECCV. *Lecture Notes in Computer Science*, 5304:57–68. 86
- [Pironeau, 1982] PIRONEAU, O. (1982). On the transport-diffusion algorithm and its applications to the Navier-Stokes equations. *Numerische Mathematik*, 38:309–332. 122
- [Quarteroni et Valli, 2008] QUARTERONI, A. et VALLI, A. (2008). *Numerical approximation of partial differential equations*, volume 23. Springer Science & Business Media. 120
- [Retarekar et al., 2015] RETAREKAR, R., RAMACHANDRAN, M., BERKOWITZ, B., HARBAUGH, R., HASAN, D., ROSENWASSER, R., OGILVY, C. et RAGHAVAN, M. (2015). Stratification of a population of intracranial aneurysms using blood flow metrics. *Computer Methods in Biomechanics and Biomedical Engineering*, 18(10):1072–1082. 104
- [Reymond et al., 2009] REYMOND, P., MERENDA, F., PERREN, F., RÜFENACHT, D. et STERGIOPILOS, N. (2009). Validation of a one-dimensional model of the systemic arterial tree. *American Journal of Physiology-Heart and Circulatory Physiology*, 297(1):H208–H222. 105
- [Rigamonti et Lepetit, 2012] RIGAMONTI, R. et LEPETIT, V. (2012). Accurate and efficient linear structure segmentation by leveraging ad hoc features with learned filters. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI. *Lecture Notes in Computer Science*, 7510:189–197. 74
- [Roberts et Thomas, 1991] ROBERTS, J. et THOMAS, J. (1991). Mixed and hybrid methods. *Handbook of numerical analysis*, 2:523–639. 124
- [Rochat, 2013] ROCHAT, S. (2013). En savoir davantage sur la rupture d’anévrisme. [http://www.docvadis.fr/cabinetduliondor/page/notre\\_guide\\_medical/la\\_maladie/en\\_savoirdavantage\\_sur\\_la\\_rupture\\_d\\_anevrisme.html](http://www.docvadis.fr/cabinetduliondor/page/notre_guide_medical/la_maladie/en_savoirdavantage_sur_la_rupture_d_anevrisme.html). 2
- [Rudin et al., 1992] RUDIN, L., OSHER, S. et FATEMI, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1–4):259–268. 11

- [Salembier et Wilkinson, 2009] SALEMBIER, P. et WILKINSON, M. (2009). Connected operators. *IEEE Signal Processing Magazine*, 26(6):136–157. 64
- [Salmon, 2015] SALMON, S. (2015). Compléments de méthodes numériques. Faculté des Sciences Exactes et Naturelles de Reims. 121
- [Sato *et al.*, 2008] SATO, K., YOHSUKE, I., ISHIKAWA, T., MATSUKI, N. et YAMAGUCHI, T. (2008). The importance of parent artery geometry in intra-aneurysmal hemodynamics. *Medical Engineering & Physics*, 30(6):774–782. 106
- [Sato *et al.*, 1997] SATO, Y., NAKAJIMA, S., ATSUMI, H., KOLLER, T., GERIG, G., YOSHIDA, S. et KIKINIS, R. (1997). Three dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. In: Computer Vision, Virtual Reality and Robotics in Medicine and Medical Robotics and Computer-Assisted Surgery–CVRMed–MRCAS. *Lecture Notes in Computer Science*, 1205:213–222. 70
- [Schaller, 2004] SCHALLER, B. (2004). Physiology of cerebral venous blood flow: From experimental data in animals to normal function in humans. *Brain Research Reviews*, 46(3):243–260. 104, 108, 111, 115, 154
- [Serra, 1982] SERRA, J. (1982). *Image Analysis and Mathematical Morphology*. Academic Press. 61, 64
- [Serra, 1988] SERRA, J. (1988). *Image Analysis and Mathematical Morphology II: Theoretical Advances*. Academic Press. 61, 64
- [Sforza *et al.*, 2010] SFORZA, D., LÖHNER, R., PUTMAN, C. et CEBRAL, J. (2010). Hemodynamic analysis of intracranial aneurysms with moving parent arteries: Basilar tip aneurysms. *International Journal for Numerical Methods in Biomedical Engineering*, 26(10):1219–1227. 113
- [Sheng *et al.*, 2013] SHENG, Z., THIRIET, M. et HECHT, F. (2013). A high-order scheme for the incompressible Navier–Stokes equations with open boundary condition. *International Journal for Numerical Methods in Fluids*, 73(1):58–73. 4, 118, 126, 127, 128, 129, 141, 163
- [Shojima *et al.*, 2014] SHOJIMA, M., MORITA, A., KIMURA, T., OSHIMA, M., KIN, T. et SAITO, N. (2014). Computational fluid dynamic simulation of a giant basilar tip aneurysm with eventual rupture after hunterian ligation. *World Neurosurgery*, 82(3):535.e5–535.e9. 104
- [Singh *et al.*, 1998] SINGH, A., TERZOPOULOS, D. et GOLDFOG, D. (1998). *Deformable models in medical image analysis*. IEEE Computer Society Press. 61
- [Sorin, 2013] SORIN, I. (2013). Cours optimisation. Institut de Science Financière et d’Assurances de Lyon. 39
- [Soualah-Alila, 2007] SOUALAH-ALILA, A. (2007). *Mathematical and numerical modelling of the human lung*. Thèse de doctorat, Université Paris Sud. 145
- [Staal *et al.*, 2004] STAAL, J., ABRAMOFF, M., NIEMEIJER, M., VIERGEVER, M. et van GINNEKEN, B. (2004). Ridge based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, 23(4):501–509. 79, 84, 91
- [Stoquart-Elsankari *et al.*, 2009] STOQUART-ELSANKARI, S., LEHMANN, P., VILLETTE, A., CZOSNYKA, M., MEYER, M., DERAMOND, H. et BALÉDENT, O. (2009). A phase-contrast MRI study of physiologic cerebral venous flow. *Journal of Cerebral Blood Flow and Metabolism*, 29(6):1208–1215. 104, 113, 116, 154

- [Suri *et al.*, 2002a] SURI, J., LIU, K., REDEN, L. et LAXMINARAYAN (2002a). A review on MR vascular image processing algorithms: Acquisition and prefiltering: Part I. *IEEE Transactions on Information Technology in Biomedicine*, 6(4):324–337. 61
- [Suri *et al.*, 2002b] SURI, J., LIU, K., REDEN, L. et LAXMINARAYAN (2002b). A review on MR vascular image processing algorithms: Skeleton versus nonskeleton approaches: Part II. *IEEE Transactions on Information Technology in Biomedicine*, 6(4):338–350. 61
- [Tai *et al.*, 2013] TAI, C., ZHANG, X. et SHEN, Z. (2013). Wavelet frame based multi-phase image segmentation. *SIAM Journal on Imaging Sciences*, 6(4):2521–2546. 74
- [Tankyevych, 2010] TANKYEVYCH, O. (2010). *Filtering of thin objects, applications to vascular image analysis*. Thèse de doctorat, University Paris-Est. 61
- [Temam, 1969] TEMAM, R. (1969). Sur l’approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires : Part II. *Archive for Rational Mechanics and Analysis*, 33(5):377–385. 118
- [Thiriet, 2008] THIRIET, M. (2008). *Biology and Mechanics of Blood Flows, part I: Biology of Blood Flows, part II: Mechanics and Medical Aspects of Blood Flows*. CRM Series in Mathematical Physics. Springer. 114, 115
- [Thiriet, 2011] THIRIET, M. (2011). *Cell and tissue organization in the circulatory and ventilatory systems*, volume 1 de *Biomathematical and Biomechanical Modeling of the Circulatory and Ventilatory Systems*. Springer. 112
- [Tikhonov, 1963] TIKHONOV, A. N. (1963). Regularization of incorrectly posed problems. *Soviet Mathematics–Doklady*, 4(1):1624–1627. 11
- [Tomasi et Manduchi, 1998] TOMASI, C. et MANDUCHI, R. (1998). Bilateral filtering for gray and color images. In: International Conference on Computer Vision–ICCV. pages 839–846. 61
- [Toumoulin *et al.*, 2001] TOUMOULIN, C., BOLDAK, C., DILLENSEGER, C., COATRIEUX, J. et ROLLAND, Y. (2001). Fast detection and characterization of vessels in very large 3-D data sets using geometrical moments. *IEEE Transactions on Biomedical Engineering*, 48(5):604–606. 62
- [Tyrrell *et al.*, 2007] TYRRELL, J., TOMASO, E. D., FUJA, D., TONG, R., KOZAK, K., JAIN, R. et ROYSAM, B. (2007). Robust 3-D modeling of vasculature imagery using superellipsoids. *IEEE Transactions on Medical Imaging*, 26(2):223–237. 61, 62, 74
- [Unger, 2012] UNGER, M. (2012). *Convex Optimization for Image Segmentation*. Thèse de doctorat, Graz University of Technology. 12, 13, 20
- [Ursino et Lodi, 1997] URSINO, M. et LODI, C. (1997). A simple mathematical model of the interaction between intracranial pressure and cerebral hemodynamics. *Journal of Applied Physiology*, 82(4):1256–1269. 105
- [Varpasuo, 2011] VARPASUO, P. (2011). Fortum participation in the IAEA ISSC WG7 kick-off work shop. <http://www.kolumbus.fi/pentti.varpasuo/iaea-issc-wa7-varpasuo-mar2011-ottawa-v03.htm>. International Workshop on IAEA ISSC WG7 Kick-off. 118
- [Vignon-Clementel *et al.*, 2006] VIGNON-CLEMENTEL, I., FIGUEROA, C., JANSEN, K. et TAYLOR, C. (2006). Outflow boundary conditions for three-dimensional finite element modeling of blood flow and pressure in arteries. *Computer methods in applied mechanics and engineering*, 195(29):3776–3796. 116, 164

- [Waechter *et al.*, 2008] WAECHTER, I., BREDNO, J., HERMANS, R., WEESE, J., BARRATT, D. et HAWKES, D. (2008). Model-based blood flow quantification from rotational angiography. *Medical Image Analysis*, 12(5):586–602. 104
- [Wang *et al.*, 2011] WANG, S., DING, G., ZHANG, Y. et YANG, X. (2011). Computational haemodynamics in two idealised cerebral wide-necked aneurysms after stent placement. *Computer Methods in Biomechanics and Biomedical Engineering*, 14(11):927–937. 104
- [Wink *et al.*, 2002] WINK, O., FRANGI, A., VERDONCK, B., VIERGEVER, M. et NIESSEN, W. (2002). 3D MRA coronary axis determination using a minimum cost path approach. *Magnetic Resonance in Medicine*, 47(6):1169–1175. 61
- [Wink *et al.*, 2000] WINK, O., NIESSEN, W. et VIERGEVER, M. (2000). Fast delineation and visualization of vessels in 3-D angiographic images. *IEEE Transactions on Medical Imaging*, 19(4):337–346. 61, 63
- [Wink *et al.*, 2007] WINK, O., NIESSEN, W. et VIERGEVER, M. (2007). Probabilistic vessel axis tracing and its application to vessel segmentation with stream surfaces and minimum cost paths. *Medical Image Analysis*, 11(6):567–587. 61
- [Wolff *et al.*, 2013] WOLFF, V., ARMSPACH, J., LAUER, V., ROUYER, O., BATAILLARD, M., MARESCAUX, C. et GENY, B. (2013). Cannabis-related stroke myth or reality? *Stroke*, 44(2):558–563. 164, 165
- [Wörz et Rohr, 2004] WÖRZ, S. et ROHR, K. (2004). A new 3D parametric intensity model for accurate segmentation and quantification of human vessels. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI. *Lecture Notes in Computer Science*, 3216:491–499. 61
- [Xiao *et al.*, 2014] XIAO, N., ALASTRUEY, J. et FIGUEROA, C. (2014). A systematic comparison between 1-D and 3-D hemodynamics in compliant arterial models. *International Journal for Numerical Methods in Biomedical Engineering*, 30(2):204–231. 162
- [Xiao *et al.*, 2013] XIAO, N., HUMPHREY, J. et FIGUEROA, C. (2013). Multi-scale computational model of three-dimensional hemodynamics within a deformable full-body arterial network. *Journal of Computational Physics*, 244:22–40. 106
- [Zana et Klein, 2001] ZANA, F. et KLEIN, J. (2001). Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. *IEEE Transactions on Image Processing*, 10(7):1010–1019. 79, 84, 91
- [Zucker, 1976] ZUCKER, S. (1976). Region growing: Childhood and adolescence. *Computer Graphics and Image Processing*, 5(3):382–399. 61

---

## **GÉNÉRATION DE MODÈLES VASCULAIRES CÉRÉBRAUX : SEGMENTATION DE VAISSEAUX ET SIMULATION D'ÉCOULEMENTS SANGUINS.**

---

Ce travail a pour objectif de générer des modèles vasculaires et de simuler des écoulements sanguins réalistes à l'intérieur de ces modèles. La première étape consiste à segmenter/reconstruire le volume 3D du réseau vasculaire.

Une fois de tels volumes vasculaires segmentés et maillés, il est alors possible de simuler des écoulements sanguins à l'intérieur de ceux-ci. Pour la segmentation, nous utilisons une approche variationnelle.

Nous proposons un premier modèle qui inclut un a priori de tubularité dans les modèles de débruitage ROF et TV-L1. Néanmoins, bien que ces modèles permettent de rehausser les vaisseaux, ils ne permettent pas de les segmenter.

C'est pourquoi nous proposons un deuxième modèle amélioré qui inclut à la fois un a priori de tubularité et de direction dans le modèle de segmentation de Chan-Vese.

Les résultats sont présentés sur des images synthétiques 2D, ainsi que sur des images rétinienne. En ce qui concerne la simulation, nous nous intéressons d'abord au réseau veineux cérébral, encore peu étudié. Les équations de la dynamique des fluides qui régissent les écoulements sanguins dans notre géométrie sont alors les équations de Navier-Stokes.

Pour résoudre ces équations, la méthode classique des caractéristiques est comparée avec un schéma d'ordre plus élevé.

Ces deux schémas sont validés sur des solutions analytiques avant d'être appliqués aux cas réalistes du réseau veineux cérébral premièrement, puis du polygone artériel de Willis.

---

Segmentation de vaisseaux, modèles variationnels, mesures de tubularité, simulation d'écoulements sanguins, équations de Navier-Stokes, réseau vasculaire cérébral.

---

## **GENERATION OF CEREBRAL VASCULAR MODELS: VESSEL SEGMENTATION AND BLOOD FLOWSIMULATION**

---

The aim of this work is to generate vascular models and simulate blood flows inside these models.

A first step consists of segmenting/reconstructing the 3D volume of the vascular network.

Once such volumes are segmented and meshed, it is then possible to simulate blood flows.

For segmentation purposes, we use a variational approach.

We first propose a model that embeds a vesselness prior in the denoising models ROF and TV-L1. Although these models can enhance vessels, they are not designed for segmentation.

Then, we propose a second, improved model that includes both vesselness and direction priors in the Chan-Vese segmentation model.

The results are presented on 2D synthetic images, as well as retinal images. In the second part, devoted to simulation, we first focus on the cerebral venous network, that has not been intensively studied.

The equations governing blood flows inside our geometry are the Navier-Stokes equations.

For their resolution, the classical method of characteristics is compared with a high-order scheme.

Both schemes are validated on analytical solutions before their application on the realistic cases of the cerebral venous network, and the arterial polygon of Willis.

---

Vessel segmentation, variational models, vesselness, blood flow simulation, Navier-Stokes equations, cerebral vascular network.

---

**Discipline : MATHÉMATIQUES**

---



UFR Sciences Exactes et Naturelles

Laboratoire de Mathématiques

Moulin de la Housse – 51687 REIMS CEDEX 2