



Thèse de Doctorat

Mention Mathématiques

présentée à *l'École Doctorale en Sciences Technologie et Santé (ED 585)* de

L'Université de Picardie Jules Verne

par

Simon Garnotel

pour obtenir le grade de Docteur de l'Université de Picardie Jules Verne

Modélisation numérique de la pression intracrânienne via les écoulements du liquide cérebrospinal et du sang mesurés par IRM de flux

Soutenue le 9 décembre 2016, après avis des rapporteurs, devant le jury d'examen :

M. MARTIN Sébastien, Professeur des Universités	Rapporteur
M. ZAGZOULE Mokhtar, Professeur des Universités	Rapporteur
M ^{me} GUIDOBONI Giovanna, Maître de Conférences	Examinatrice
M. SCHMIDT Éric, Praticien Hospitalier	Examineur
M ^{me} SALMON Stéphanie, Professeur des Universités	Directrice de thèse
M. BALÉDENT Olivier, Maître de Conférences	Directeur de thèse



À ma compagne Narimène

À ma fille Lila

À ma famille et mes amis

Remerciements

Je tiens en premier lieu à remercier Stéphanie Salmon, qui m'a offert la possibilité de découvrir le monde de la recherche scientifique au travers de mon stage de fin de Master 2, et qui, avec cette thèse, me permet de continuer dans cet univers passionnant.

Je remercie Olivier Balédent pour m'avoir accueilli dans son laboratoire et de m'avoir permis de découvrir l'imagerie par résonance magnétique (IRM).

Je remercie Sébastien Martin et Mokhtar Zagzoule pour avoir accepté de rapporter sur ma thèse, ainsi que les membres du jury Giovanna Guidoboni et Eric Schmidt.

Je remercie Nicolas Passat pour avoir pris le temps de relire ma thèse et d'en avoir corrigé certains points.

Je remercie Cornel Murea pour son aide dans la compréhension et la programmation de l'algorithme fluide-structure qu'il a développé.

Je remercie Olivier Goubet pour m'avoir invité à participer aux séminaires du Laboratoire Amiénois de Mathématiques Fondamentales et Appliquées (LAMFA) ainsi que pour m'avoir aidé lors de ma recherche d'une mission d'enseignement. Je remercie Nabil Bedjaoui et Vivien Desveaux pour leur encadrement lors de ma mission d'enseignement à l'INSSET de Saint-Quentin.

Je remercie Eric Schmidt pour m'avoir fourni de nombreuses données cliniques, notamment d'IRM de flux et de mesures de pression intracrânienne provenant du projet PRO-LIPHYC qu'il porte.

Je souhaite également remercier Audrey Lecompte, Virginie Pecourt et Virginie Lefevre, secrétaires de l'École Doctorale, que j'ai beaucoup sollicitées lors de mes trois années de thèse, que ce soit pour les unités d'enseignement, la mission d'enseignement ou mon congé paternité.

Je remercie finalement Roger Bouzerar et Joël Daouc, ingénieurs de recherche du laboratoire BioFlowImage, pour leur aide ponctuelle tout au long de ma thèse.

Je remercie de plus Olivia Miraucourt, Alexandre Fortin, Ranine Tarabay et Alexandre Ancel pour leur implication dans notre projet commun du CEMRACS 2015. Je remercie encore une fois Alexandre Ancel pour m'avoir aidé dans la prise en main de la librairie Feel++.

Je remercie Ihab Haidar et Thi Khuyen Le pour leur implication dans notre projet commun de la SEME 2016.

Résumé

La modélisation numérique de la pression intracrânienne est un sujet de thèse pluridisciplinaire faisant intervenir aussi bien des connaissances en mathématiques appliquées, utiles pour résoudre les équations de la mécanique des fluides et des interactions fluide-structure, qu'en anatomie ou en physiologie, afin de modéliser correctement le système cérébrospinal. C'est pourquoi cette thèse se décompose en quatre parties principales.

La première partie, celle d'introduction, définit le système étudié : le système cérébrospinal, d'un point de vue anatomique et physiologique. Ensuite sera fait l'état de l'art de la modélisation de ce système et des différentes techniques utilisées pour obtenir une indication sur la pression intracrânienne. Finalement, les modèles que nous allons considérer par la suite seront présentés.

La seconde partie concerne la modélisation mathématique et la simulation numérique des modèles. Différentes méthodes numériques seront présentées, puis validées avant d'être appliquées à nos modèles. Les résultats obtenus seront comparés à des mesures de flux et de pressions obtenues par IRM¹ de flux et monitoring de pression intracrânienne.

La troisième partie concerne l'analyse des données obtenues par IRM de flux et par monitoring de pression intracrânienne, utilisées dans nos modèles. Il sera présenté un logiciel d'analyse développé durant cette thèse afin de recueillir de nombreux paramètres caractérisés par ces données. L'analyse des données de cette population sera présentée afin de mieux appréhender le fonctionnement global du système cérébrospinal, après avoir étudié la reproductibilité de l'analyse.

La dernière partie est la conclusion générale de ce travail de thèse, évoquant les résultats obtenus grâce à nos modèles et posant les perspectives de ce travail.

Finalement, les annexes décrivent les algorithmes utilisés, codés à l'aide du logiciel FREEFEM++, présentent un aperçu du projet ANR VIVABRAIN auquel j'ai contribué tout au long de ma thèse, et fournissent les rapports du CEMRACS 2015 et de la SEME 2016, deux écoles d'été proposant des projets académiques et industriels auxquels j'ai participé, durant ma thèse.

Mots clés

Modélisation mathématique, Simulation numérique, Équations de Navier-Stokes, Interaction fluide-structure, Pression intracrânienne

1. Imagerie par Résonance Magnétique

Abstract

The numerical modeling of the intracranial pressure is a multidisciplinary PhD thesis subject involving both applied mathematics knowledge, useful to solve the fluid mechanics equations and fluid-structure interaction, and in anatomy or physiology, to correctly model the cerebrospinal system. Therefore, this PhD thesis is divided into four main parts.

The first part, the introduction, defines the studied system: the cerebrospinal system from an anatomical and physiological perspective. Then, the state of the art about numerical modeling of this system and different techniques used to obtain an indication of the intracranial pressure is provided. Finally, the models we will consider later are presented.

The second part concerns the mathematical modeling and numerical simulation models. Different numerical methods will be presented and validated before being applied to our models. The results will be compared with real measurements of flows and pressures obtained by flow MRI² and intracranial pressure monitoring.

The third part concerns the analysis of data obtained by flow MRI and intracranial pressure monitoring, used in our models. An analysis software developed during this PhD thesis is presented in order to gather several parameters characterized by the data. Data analysis of the population will be presented to better understand the overall behavior of the cerebrospinal system, after having studied the reproductibility of the analysis.

The last part is the general conclusion of this PhD thesis, referring to the results obtained with our models and laying perspectives of this work.

Finally, the appendices describe the used algorithms, coded using `FREEFEM++`, present an overview of the ANR VIVABRAIN project which I contributed through my PhD thesis, and provide reports of the CEMRACS 2015 and the SEME 2016, two summer schools offering academic and industrial projects in which I participated during my PhD thesis.

Keywords

Mathematical modeling, Numerical simulation, Navier-Stokes equations, Fluid-structure interaction, Intracranial pressure

2. Magnetic Resonance Imaging

Table des matières

Remerciements	V
Résumé	VII
Abstract	IX
Introduction	3
1 Présentation	3
1.1 Le système cérébrospinal, vue anatomique	3
1.1.1 Les méninges	4
1.1.2 Les ventricules cérébraux	4
1.1.3 Le liquide cérébrospinal	5
1.1.4 Vascularisation	5
1.2 Le système cérébrospinal, vue physiologique	6
1.2.1 Mouvements du liquide cérébrospinal	6
1.2.2 Chronologie	6
1.3 L'IRM de flux	7
1.4 Monitoring de pression intracrânienne	9
1.5 Problématique	9
1.6 Cadre de la thèse	10
1.6.1 Laboratoire BioFlowImage	10
1.6.2 Laboratoire de Mathématiques de Reims	10
2 État de l'art	11
2.1 Modélisation du système cérébrospinal	11
2.1.1 Modèles électriques	11
2.1.2 Modèles physiques	13
2.1.3 Modèles numériques	15
2.2 Études expérimentales	16
2.2.1 Monitoring de pression intracrânienne	16
2.2.2 Forme de la courbe de pression intracrânienne	17
2.3 Études cliniques	18

3	Modèles utilisés	19
3.1	Bifurcation	19
3.2	Système cérébrospinal simplifié	20
I	Modélisation numérique	21
4	Méthodes numériques	23
4.1	Simulation de la dynamique des fluides	23
4.1.1	Préliminaires	24
4.1.2	Grandeurs caractéristiques	25
4.1.3	Formulation variationnelle des équations de Stokes	26
4.1.4	Formulation variationnelle des équations de Navier-Stokes	27
4.1.5	Méthode des caractéristiques	29
4.1.6	Algorithme de splitting	31
4.2	Modèles de Windkessel	34
4.3	Stabilité	36
4.3.1	Étude théorique – Conditions de Neumann	36
4.3.2	Étude théorique – Conditions mêlées	39
4.3.3	Étude numérique	41
4.4	Simulation de l’interaction fluide-structure	45
4.4.1	FREEFEM++	45
4.4.2	FEEL++	48
5	Validation	51
5.1	Cas tests – Équations de Navier-Stokes	51
5.1.1	Vortex de Taylor-Green	51
5.1.2	Solution de Ethier-Steinman	52
5.1.3	Écoulement de Poiseuille	52
5.1.4	Écoulement de Richardson	53
5.2	Vérification et validation des algorithmes fluide	53
5.3	Modèles de Windkessel	57
5.4	Cas test – Interaction fluide-structure	59
5.5	Validation des algorithmes d’interaction fluide-structure	60
5.5.1	FREEFEM++	60
5.5.2	FEEL++	60
5.6	Validation croisée Simulation – IRM	60
5.6.1	Conditions expérimentales	60
5.6.2	Validation en fluide seul	61
5.6.3	Validation en interaction fluide-structure	61

6	Premier modèle : Bifurcation	67
6.1	Modèle	67
6.1.1	Modèles capacitifs constants	68
6.1.2	Modèles capacitifs variables	69
6.2	Résultats – Windkessel constant	70
6.2.1	Stroke volume	70
6.2.2	Paramètres	70
6.2.3	Impact de l’amplitude	72
6.2.4	Impact du cycle cardiaque	72
6.2.5	Impact de la compliance	76
6.2.6	Cas réels	78
6.2.7	Conclusion	82
6.3	Résultats – Windkessel variable	82
6.3.1	Paramètres	82
6.3.2	Impact de l’amplitude	83
6.3.3	Impact du cycle cardiaque	84
6.3.4	Impact de la compliance	85
6.3.5	Impact de la constante élastique	85
6.3.6	Cas réels	85
6.3.7	Conclusion	88
6.4	Conclusion	89
7	Second modèle : Système cérébrospinal simplifié	93
7.1	Modèle	93
7.1.1	Première amélioration	94
7.1.2	Deuxième amélioration	94
7.1.3	Construction géométrique	95
7.2	Résultats	95
7.2.1	Paramètres	95
7.2.2	Impact de l’amplitude	97
7.2.3	Impact du cycle cardiaque	97
7.2.4	Impact du module d’Young	98
7.2.5	Impact de la résistance à l’écoulement	99
7.2.6	Cas réels	99
7.3	Impact des améliorations	104
7.3.1	Structures différenciées	104
7.3.2	Compartiment sanguin	105
7.4	Conclusion	105
8	Conclusion	107

II	Analyse de données	109
9	Acquisition des données	111
9.1	Flux sanguins et de liquide cébrospinal	111
9.2	Pression intracrânienne	112
10	Traitement des données	115
10.1	Analyse de la pression	116
10.1.1	Principe	116
10.1.2	Fonctionnement	116
10.2	Analyse des flux	117
10.2.1	Principe	117
10.2.2	Fonctionnement	118
10.3	Calcul de la compliance	119
10.3.1	Principe	119
10.3.2	Fonctionnement	121
10.4	Outils statistiques	121
10.4.1	Principe	121
10.4.2	Fonctionnement	122
11	Résultats	123
11.1	Population	123
11.2	Méthodes	123
11.3	Reproductibilité des mesures	124
11.4	Résultats	125
11.4.1	Description de la population	125
11.4.2	Catégorisation de la population	126
11.4.3	Récapitulatif	128
11.5	Résultats complémentaires	129
12	Utilisation dans les modèles numériques	135
12.1	Flux	135
12.2	Pression	136
12.3	Compliance	136
13	Conclusion	137
	Conclusion générale	141
13.1	Conclusion	141
13.2	Perspectives	141
A	Abréviations	145

B Publications et congrès	147
B.1 Publications	147
B.2 Congrès	148
B.3 Écoles d'été	148
C Algorithmes FREEFEM++	149
C.1 Stokes	149
C.1.1 2D	149
C.1.2 3D	150
C.2 Navier-Stokes, méthode des caractéristiques	153
C.2.1 2D	153
C.2.2 3D	155
C.3 Navier-Stokes, algorithme d'Uzawa	157
C.3.1 2D	157
C.3.2 3D	161
C.4 Navier-Stokes, algorithme de splitting	165
C.4.1 2D	165
C.4.2 3D	168
C.5 Interaction fluide-structure	173
C.5.1 2D	173
C.5.2 3D	179
D Projet ANR VIVABRAIN	185
D.1 Présentation	185
D.2 Participation	186
D.2.1 Équipes mathématiques	186
D.2.2 Équipes physiques	186
E CEMRACS 2015	189
F SEME 2016	213
Bibliographie	231

Introduction

Chapitre 1

Présentation

La modélisation numérique de la pression intracrânienne est un sujet de thèse pluridisciplinaire faisant intervenir aussi bien des connaissances en mathématiques appliquées, utiles pour résoudre les équations de la mécanique des fluides et des interactions fluide-structure, qu'en anatomie ou en physiologie, afin de modéliser correctement le système cérébrospinal.

C'est pourquoi ce chapitre reprendra des notions générales sur le système cérébrospinal dans son ensemble. Des notes supplémentaires d'imagerie par résonance magnétique (IRM) et de monitoring de pression intracrânienne seront proposés afin de mieux comprendre l'origine des mesures d'écoulements de sang et de liquide cérébrospinal (LCS) et de la pression intracrânienne (PIC). Finalement, la problématique sera exposée, suivie d'une brève présentation des laboratoires de recherche partenaires.

1.1 Le système cérébrospinal, vue anatomique

Le système cérébrospinal [65] est constitué de l'encéphale (cerveau, tronc cérébral et cervelet) et de la moelle spinale, ou moelle épinière, Fig. 1.1. C'est le système le plus complexe du corps humain. Il assure notamment les fonctions cognitives, motrices et sensibles de l'ensemble du corps humain.

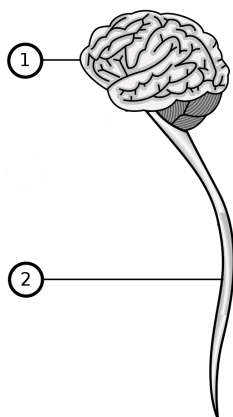


FIGURE 1.1 – **Système cérébrospinal.** 1 : encéphale. 2 : moelle spinale.

Le cerveau est composé de matière grise, ou substance grise, en périphérie, et de matière blanche, ou substance blanche, en son centre.

Le système cérébrospinal est totalement entouré par une superposition de trois membranes : **les méninges**. Au centre du cerveau se trouvent plusieurs cavités physiologiques : **les ventricules cérébraux**. Le **liquide cérébrospinal (LCS)** est présent dans ces deux structures : les méninges et les ventricules cérébraux. Le système cérébrospinal est également alimenté en sang par un réseau artériel et veineux très développé.

1.1.1 Les méninges

Les méninges sont des membranes fibreuses qui entourent le système cérébrospinal, Fig. 1.2. Elles sont au nombre de trois :

- la dure-mère, la plus externe, qui tapisse le crâne ;
- l'arachnoïde, au centre, qui forme un réseau comparable à une toile d'araignée ;
- la pie-mère, la plus interne, qui tapisse les replis du cerveau et de la moelle spinale.

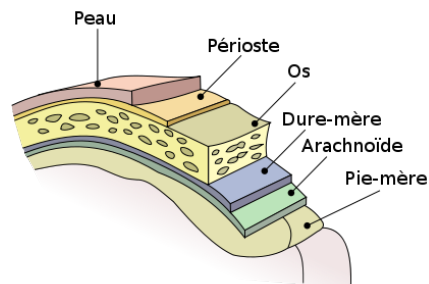


FIGURE 1.2 – **Les méninges**. Les espaces subarachnoïdiens se trouvent entre l'arachnoïde et la pie-mère.

Entre la pie-mère et l'arachnoïde se trouvent les espaces subarachnoïdiens (ESA), ou espaces sous-arachnoïdiens, qui contiennent du LCS.

1.1.2 Les ventricules cérébraux

Au centre du cerveau se trouvent quatre cavités appelées les ventricules cérébraux, Fig. 1.3.

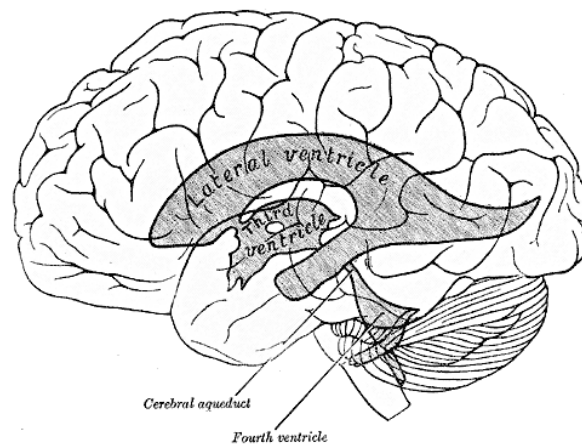


FIGURE 1.3 – **Les ventricules cérébraux**. Au nombre de quatre, ils communiquent entre eux et avec les espaces subarachnoïdiens.

Les deux premiers ventricules cérébraux sont les ventricules latéraux. L'un est présent dans l'hémisphère cérébral droit, l'autre dans l'hémisphère cérébral gauche. On y trouve les plexus

choroïdes qui ont un rôle dans la sécrétion du LCS. Le troisième ventricule cérébral, relié aux deux précédents par le foramen de Monro, est plus en bas dans le crâne.

Le quatrième ventricule cérébral, relié au troisième par l'aqueduc de Sylvius, est plus en bas que le précédent et en arrière du crâne. Le quatrième ventricule cérébral est également relié aux espaces subarachnoïdiens par le trou de Magendie, en bas, et les deux trous de Luschka, sur la face dorsale du tronc cérébral.

Le LCS est présent dans les quatre ventricules cérébraux et communique avec les espaces subarachnoïdiens.

1.1.3 Le liquide cérebrospinal

Le liquide cérebrospinal est un liquide proche de l'eau. Il est composé à 99% d'eau pour 1% de minéraux, de nutriments et d'hormones.

Le LCS a différents rôles :

- un rôle mécanique : l'absorption des chocs, le maintien du cerveau ; en effet, sans LCS, le cerveau s'écraserait sous son propre poids ;
- un rôle biologique : transport de nutriments et d'hormones, protection contre les infections.

Le volume de LCS est d'environ 150 ml. Sa production est de l'ordre de 500 à 1200 ml par jour selon l'âge ; il est donc renouvelé plusieurs fois par jour.

On appelle **pression intracrânienne** la pression du LCS.

Le système dans son ensemble est représenté de manière simplifiée Fig. 1.4.

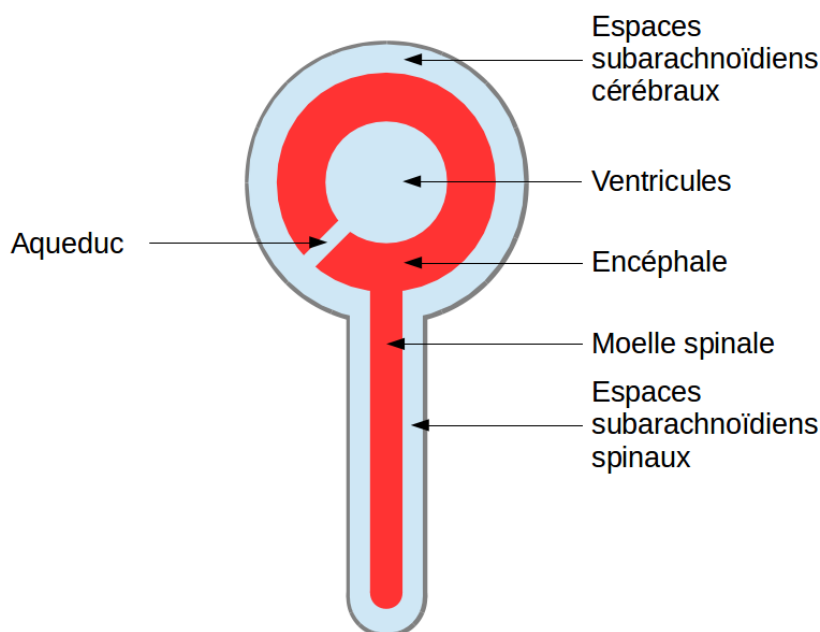


FIGURE 1.4 – **Représentation simplifiée du système cérébrospinal.** Description anatomique.

1.1.4 Vascularisation

Les arrivées de sang dans le système cérébrospinal se font principalement par :

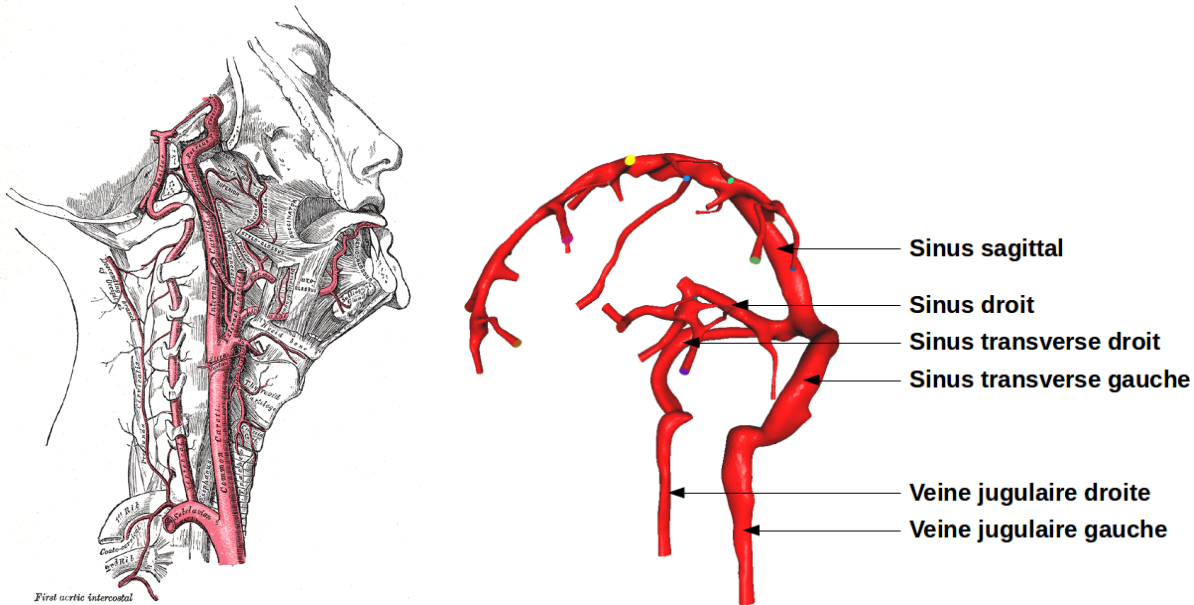
- les artères carotides communes, droite et gauche ;
- les artères vertébrales, droite et gauche.

Les artères carotides communes se divisent en artères carotides internes et externes, Fig. 1.5a.

Les sorties de sang se font principalement par :

- les veines jugulaires, droite et gauche ;
- les plexus postérieurs, droite et gauche ;
- les veines épidurales, droite et gauche.

Les veines jugulaires proviennent d'une réunion du sinus sagittal et du sinus droit, qui se divisent en sinus transverses, Fig. 1.5b.



(a) **Système artériel.** Les principales voies sont les artères carotides et les artères vertébrales.

(b) **Système veineux.** Les principales voies sont les jugulaires, le sinus sagittal et le sinus droit.

FIGURE 1.5 – **Système artério-veineux cérébral.** Description anatomique.

1.2 Le système cérébrospinal, vue physiologique

1.2.1 Mouvements du liquide cérébrospinal

D'un point de vue simplifié, le LCS est produit dans les ventricules cérébraux par les plexus choroïdes et est absorbé dans les espaces subarachnoïdiens cérébraux par les granulations de Pacchioni ou villosités arachnoïdiennes.

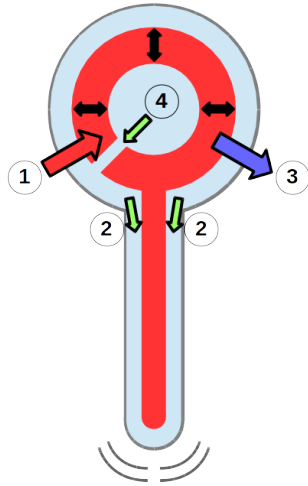
En réalité, ce comportement ne représente qu'une partie infime du mouvement du LCS. À l'arrivée de sang dans le cerveau, à chaque cycle cardiaque, le LCS est chassé du compartiment cérébral vers le compartiment spinal, afin de laisser de la place à l'expansion du cerveau qui se remplit de sang artériel avant de se vider de sang veineux [10]. Cela provoque un va-et-vient du LCS rythmé par le sang artériel, donc par la pulsation cardiaque.

À titre d'exemple, sur un sujet sain, la production de LCS est comprise entre $6 \cdot 10^{-3}$ et $1 \cdot 10^{-2} \text{ mm}^3 \cdot \text{s}^{-1}$ alors que le pic de l'oscillation de LCS est de l'ordre de $100 \text{ mm}^3 \cdot \text{s}^{-1}$.

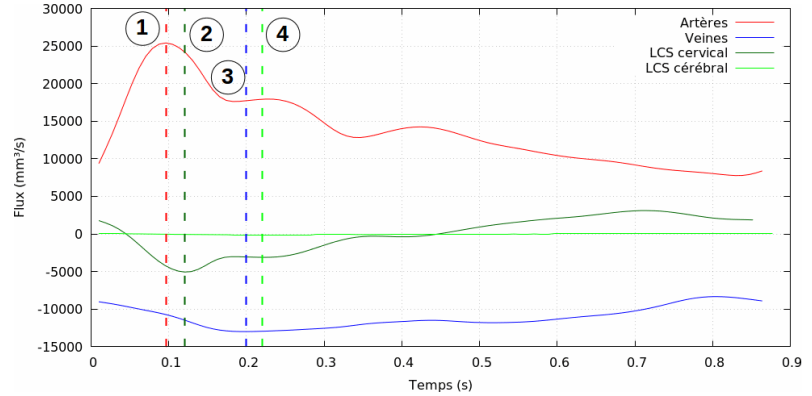
1.2.2 Chronologie

Une chronologie des flux du compartiment sanguin et du compartiment de LCS est connue [9].

Le point de départ est l'arrivée de sang artériel dans le cerveau. On observe ensuite une chasse du LCS sous-arachnoïdien vers le compartiment spinal, ensuite la chasse du sang veineux, et finalement la chasse du LCS ventriculaire vers le compartiment spinal. Après un temps de latence, le système revient à l'équilibre avant la nouvelle pulsation artérielle, Fig. 1.6.



(a) Représentation simplifiée du système cérébro-spinal. Description physiologique.



(b) Mesures de flux par IRM. Représentation des pics maximums. Le flux de LCS au niveau cérébral paraît plat au regard des autres courbes mais il présente une forme similaire au flux de LCS cervical.

FIGURE 1.6 – **Chronologie des échanges de flux.** 1 : arrivée de sang artériel dans le cerveau. 2 : mouvement de LCS des espaces subarachnoïdiens cérébraux vers les espaces subarachnoïdiens spinaux. 3 : sortie de sang veineux du cerveau. 4 : mouvement de LCS des ventricules vers les espaces subarachnoïdiens cérébraux.

1.3 L'IRM de flux

L'imagerie par résonance magnétique (IRM) est une technique d'imagerie médicale qui présente l'avantage d'être non invasive (injection de produit de contraste non nécessaire) et non irradiante. Elle repose sur la résonance magnétique nucléaire des atomes d'hydrogène en présence d'un champ magnétique.

Ce champ magnétique est généré par un anneau d'aimants au centre duquel se place le témoin / patient, Fig. 1.7. Ces aimants génèrent, en plus d'une aimantation permanente, des gradients d'aimantation nécessaires au codage des différentes séquences.

Il est plus habituel de voir des IRM morphologiques, comme celle présentée Fig. 1.8, où les nuances de gris rendent compte des différences de propriétés magnétiques des tissus. À partir de plusieurs coupes, une reconstruction 3D peut être faite pour obtenir la géométrie complète du champ de vue.

L'IRM de flux s'intéresse quant à elle aux déplacements, et donne une image représentative des vitesses, Fig. 1.10a et Fig. 1.10b. Chaque pixel fournit une information locale de la vitesse. Il est donc possible d'obtenir le flux à l'intérieur d'une structure, comme un vaisseau sanguin, en segmentant une région d'intérêt et en extrayant les vitesses à l'intérieur de cette région.

La vitesse mesurée en chaque instant est en fait une moyenne de la vitesse sur plusieurs cycles cardiaques, on a donc une séquence d'environ 3 min ; la technique IRM de flux n'étant pas suffisamment rapide pour acquérir toutes les informations d'un seul coup.

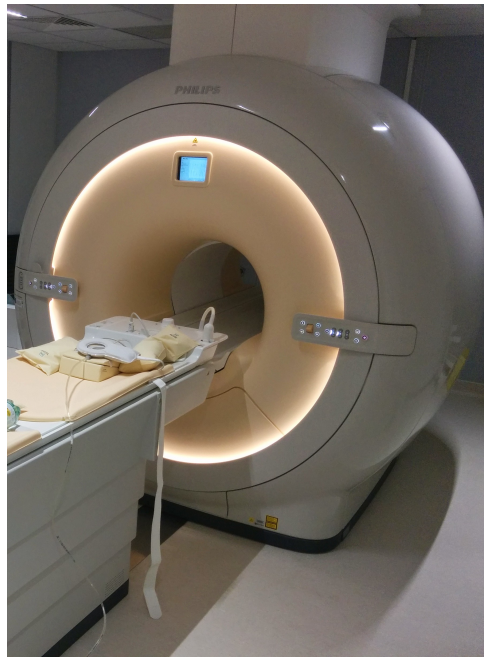


FIGURE 1.7 – Machine IRM 3T. IRM de recherche installée au CHU Amiens-Picardie.



FIGURE 1.8 – IRM morphologique.

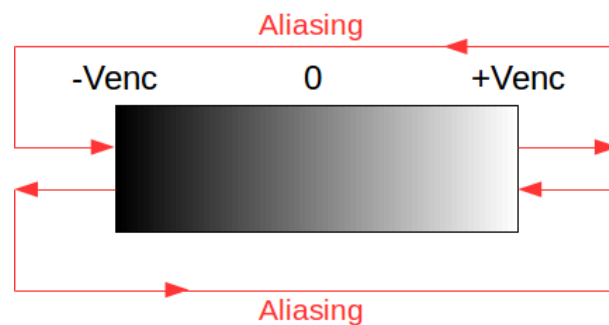
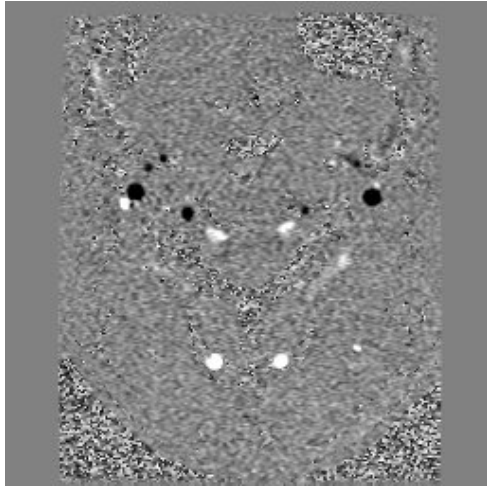


FIGURE 1.9 – **Aliasing**. Le niveau de gris représente l'intervalle de vitesse $[-V_{enc}, +V_{enc}]$; lorsque la vitesse mesurée dépasse de cet intervalle, les valeurs s'inversent.

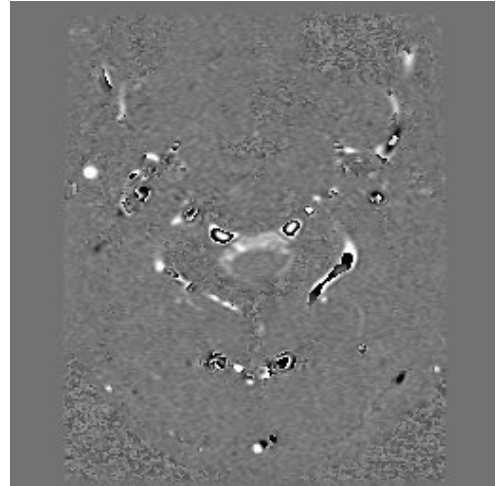
Ce type de séquence possède, entre autres paramètres, une vitesse d'encodage V_{enc} . Ce paramètre définit *a priori* l'intervalle dans lequel la vitesse est mesurée; ce qui dépasse cette vitesse apparaît inversé en niveau de gris, c'est ce que l'on appelle l'aliasing, Fig. 1.9.

L'image résultante dépend grandement de ce paramètre : on voit, Fig. 1.10a, sur un même plan de coupe, pour une grande vitesse d'encodage (de l'ordre de 800 mm.s^{-1}) apparaître les artères et veines dans lesquels le sang circule rapidement alors que pour une vitesse d'encodage plus réduite (de l'ordre de 50 mm.s^{-1}), on voit apparaître le LCS qui circule à une vitesse plus faible ; les artères et veines apparaissent alors avec de l'aliasing.

Ces mesures peuvent être réalisées dans un plan 2D (séquence 2D, comprendre 2D+t) ou dans le champ de vue complet 3D (séquence 4D, comprendre 3D+t).



(a) **IRM de flux par contraste de phase.** Niveau C2-C3, haute vitesse d'encodage (800 mm.s^{-1}).



(b) **IRM de flux par contraste de phase.** Niveau C2-C3, basse vitesse d'encodage (50 mm.s^{-1}).

FIGURE 1.10 – **IRM de flux par contraste de phase.** Plan de coupe identique pour deux vitesses d'encodage différentes. (Niveau C2-C3 : entre la deuxième et la troisième vertèbre cervicale.)

1.4 Monitoring de pression intracrânienne

Le monitoring de la pression intracrânienne est une technique invasive permettant de mesurer directement la pression intracrânienne. Cette méthode nécessite de placer un capteur de pression dans le cerveau, dans le parenchyme, les ventricules cérébraux, l'espace épidural ou les espaces subarachnoïdiens intracrâniens ou spinaux.

C'est pourquoi toutes les données sur la pression intracrânienne que nous exposerons dans ce travail sont obtenues sur des patients et non sur des témoins. En effet, cette technique présente un risque du fait de l'introduction du capteur de pression à l'intérieur du système cérébrospinal et ne peut être pratiquée sur un sujet sain.

Des informations supplémentaires sont données dans la section 9.2.

1.5 Problématique

La pression intracrânienne est un paramètre clinique important qui reflète le bon fonctionnement du cerveau ; toutefois, cette pression n'est pas accessible de manière non invasive. On se propose donc dans cette thèse d'approcher numériquement la pression intracrânienne à l'aide de simulations numériques de l'écoulement du liquide cérébrospinal et de ses interactions avec les différentes structures présentes dans le système cérébrospinal, principalement le cerveau, en

s'appuyant sur des mesures *in vivo* de flux artériels, veineux et du liquide cérebrospinal obtenues par IRM de flux.

Dans un premier temps, on considère une géométrie très simplifiée, une bifurcation, afin de mieux comprendre les mécanismes globaux du système cérebrospinal. Ce modèle fera intervenir uniquement le liquide cérebrospinal considéré comme un fluide Newtonien incompressible. On aura donc recours aux équations de Navier-Stokes que l'on approchera numériquement à l'aide de la méthode des éléments finis.

Dans un second temps, une géométrie plus réaliste sera utilisée afin de valider les premiers résultats obtenus et mieux approcher la pression intracrânienne. Cette géométrie prendra en compte le liquide cérebrospinal dans son environnement et fera intervenir des interactions entre les différentes structures du système cérebrospinal. On aura donc recours à un modèle d'interaction fluide-structure, résolu également par la méthode des éléments finis.

En parallèle de cette analyse numérique, une analyse de données réelles (pressions et flux) sera mise en place afin d'observer et de mieux comprendre le comportement réel du système dans des cas pathologiques. Un logiciel dédié sera mis en place et utilisé pour traiter les données de nombreux patients.

Dans les deux premières parties, concernant la modélisation, les données réelles analysées seront utilisées comme conditions aux limites pour nos modèles numériques ou comparées aux résultats de simulation.

1.6 Cadre de la thèse

1.6.1 Laboratoire BioFlowImage

Le laboratoire BioFlowImage (<http://www.tidam.fr/>) est un laboratoire de recherche situé au cœur du CHU Amiens-Picardie (<http://www.chu-amiens.fr/>), bien que faisant partie de l'Université de Picardie Jules Verne (<https://www.u-picardie.fr/>). Il est localisé au même endroit que le service de traitement de l'image en imagerie médicale, et de l'IRM 3T de recherche.

Ce laboratoire est spécialisé dans la physiologie et la physiopathologie des écoulements du sang et du LCS. Ses axes de recherche sont l'étude des écoulements de LCS, du sang crâniospinal, des écoulements hépatiques et cardiaques et de la vascularisation de la face.

Le laboratoire est dirigé par Olivier Balédent, mon co-directeur de thèse. Il est composé de deux ingénieurs de recherche : Roger Bouzerar et Joël Daouk, de post-doctorants et de doctorants.

1.6.2 Laboratoire de Mathématiques de Reims

Le laboratoire de Mathématiques de Reims (EA 4535) est composé d'enseignants-chercheurs du Département de Mathématiques et Informatiques de l'UFR Sciences Exactes et Naturelles de l'Université de Reims Champagne-Ardenne (<http://www.univ-reims.fr/>).

Ce laboratoire compte trois équipes de recherches distinctes :

- Analyse ;
- Groupes et Quantification ;
- Modélisation Stochastique et Numérique.

Ma co-directrice de thèse, Stéphanie Salmon, est membre de l'équipe Modélisation Stochastique et Numérique. Cette équipe travaille sur l'optimisation non-linéaire, la micro-fluidique, la séparation aveugle de signaux, les singularités en électromagnétisme, la statistique paramétrique et non-paramétrique, les systèmes dynamiques et la mécanique des fluides numérique appliquée au vivant.

Chapitre 2

État de l'art

Des travaux ont déjà été réalisés concernant le fonctionnement du système cérébrospinal et l'approche de la pression intracrânienne de manière non invasive. Différentes approches ont été utilisées, comme les modélisations électriques, physiques ou encore numériques, les études expérimentales et les études cliniques. Ce chapitre présente une liste non exhaustive de travaux utilisant ces approches.

2.1 Modélisation du système cérébrospinal

2.1.1 Modèles électriques

Historiquement, les premières modélisations du système cérébrospinal ont été réalisées à l'aide de modèles électriques. Elles ont l'avantage d'être simples à mettre en place et nécessitent peu de ressources de calcul. Ces études considèrent l'analogie électrique / mécanique présentée Tab. 2.1.

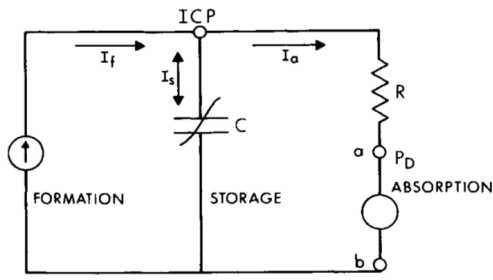
TABLE 2.1 – **Modèles de Windkessel.** Analogie Électrique / Mécanique

Électrique		Mécanique	
Courant	i	Flux	Q
Tension	U	Pression	p
Résistance	R	Résistance	R
Capacité	C	Compliance	C

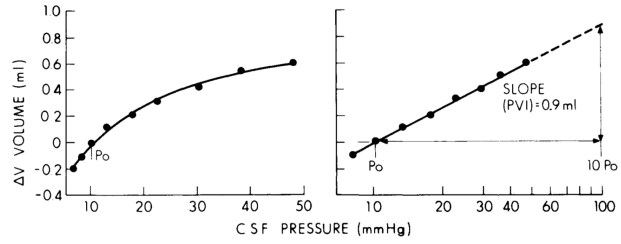
Les premières études ont été réalisées par Marmarou *et al.* [49, 50]. Il s'agit d'une approche statique du système, prenant en compte la sécrétion et l'absorption de liquide cérébrospinal (LCS). En parallèle, des mesures de pression intracrânienne invasives sont réalisées sur des chats, afin de comparer les résultats du modèle avec les mesures expérimentales.

Le schéma électrique représentatif du système cérébrospinal est présenté Fig. 2.1a. Le site de formation correspond aux plexus choroïdes (dans les ventricules cérébraux), le site d'absorption correspond aux granulations de Pacchioni (dans les espaces subarachnoïdiens cérébraux). Le site de stockage (la capacité) correspond à l'élasticité globale du système cérébrospinal, sa compliance, sa capacité à augmenter (respectivement diminuer) son volume sous l'effet d'une augmentation (respectivement d'une diminution) de pression.

Le principal résultat des études de Marmarou est la relation entre la pression et le volume dans l'environnement clos du système cérébrospinal, Fig. 2.1b. Les résultats numériques et les mesures expérimentales mettent en évidence le fait qu'à partir de la pression intracrânienne initiale



(a) **Modèle du système cérébrospinal.** Modélisation du processus de sécrétion/absorption du LCS.



(b) **Résultat principal.** Courbe pression-volume.

FIGURE 2.1 – **Études de Marmarou *et al.*** Extrait de : *A nonlinear analysis of the cerebrospinal fluid system and intracranial pressure dynamics* [50].

(P_0), un ajout de volume va provoquer une montée de la pression intracrânienne exponentielle. Inversement chaque retrait de volume provoque une diminution exponentielle de la pression intracrânienne. Ce comportement se traduit par l'équation :

$$P = P_0 e^{EV} \quad (2.1)$$

avec P la pression intracrânienne, E l'élasticité du système et V le volume.

Un modèle plus complet a été introduit par Ursino *et al.* [78], avec un modèle prenant en compte le système cérébrospinal en interaction avec le système artério-veineux, Fig. 2.2. Ce modèle ne considère encore que le système d'un point de vue statique (sécrétion + absorption) mais a l'avantage de considérer le compartiment sanguin dans le comportement du LCS.

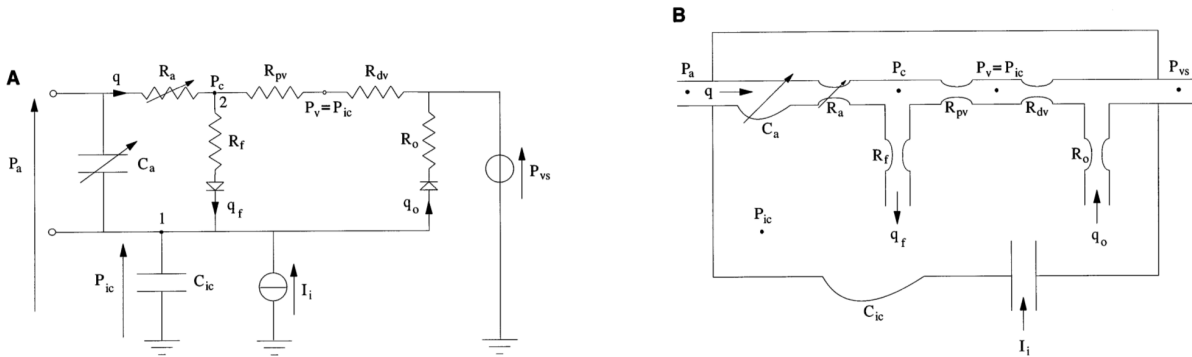


FIGURE 2.2 – **Étude d'Ursino *et al.*** Modèle du système cérébrospinal. Le premier schéma (à gauche) montre le modèle électrique utilisé, le second schéma (à droite) montre l'interprétation physique du modèle électrique. Extrait de : *A simple mathematical model of the interaction between intracranial pressure and cerebral hemodynamics* [78].

Le réseau sanguin est modélisé par un réseau électrique de résistances et de capacités (représentant la résistance à l'écoulement et l'élasticité des vaisseaux, respectivement) : C_a , R_a , R_{pv} et R_{dv} ; alimenté par un flux sanguin q . Les sites de formation et d'absorption du LCS sont modélisés par un transfert entre le compartiment sanguin et le compartiment de LCS paramétrés par R_f et R_o , respectivement. La compliance cérébrale est modélisée par la capacité C_{ic} tandis que l'injection de fluide à l'intérieur du compartiment de LCS (acte chirurgical invasif) est représenté par I_i .

Ce modèle permet d'étudier l'interaction entre le compartiment sanguin (artères et veines) et le compartiment de LCS, et son impact sur la pression intracrânienne. Toutefois, la principale

difficulté d'un tel modèle est le nombre de paramètres physiologiques à régler afin de correspondre au mieux au comportement réel du système.

Un exemple d'utilisation du modèle d'Ursino *et al.* est montré Fig. 2.3 lors d'un test d'infusion (injection de liquide dans le compartiment de LCS, voir Sec. 9.2).

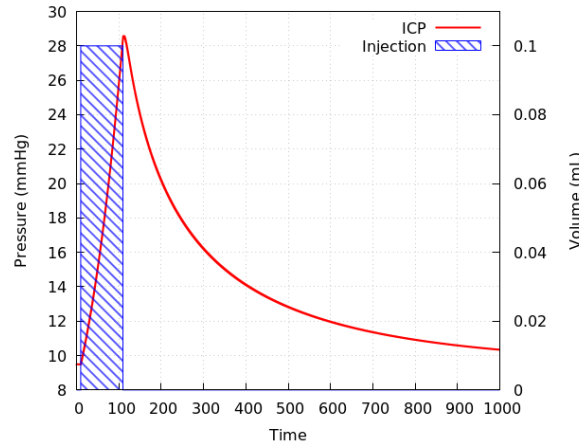
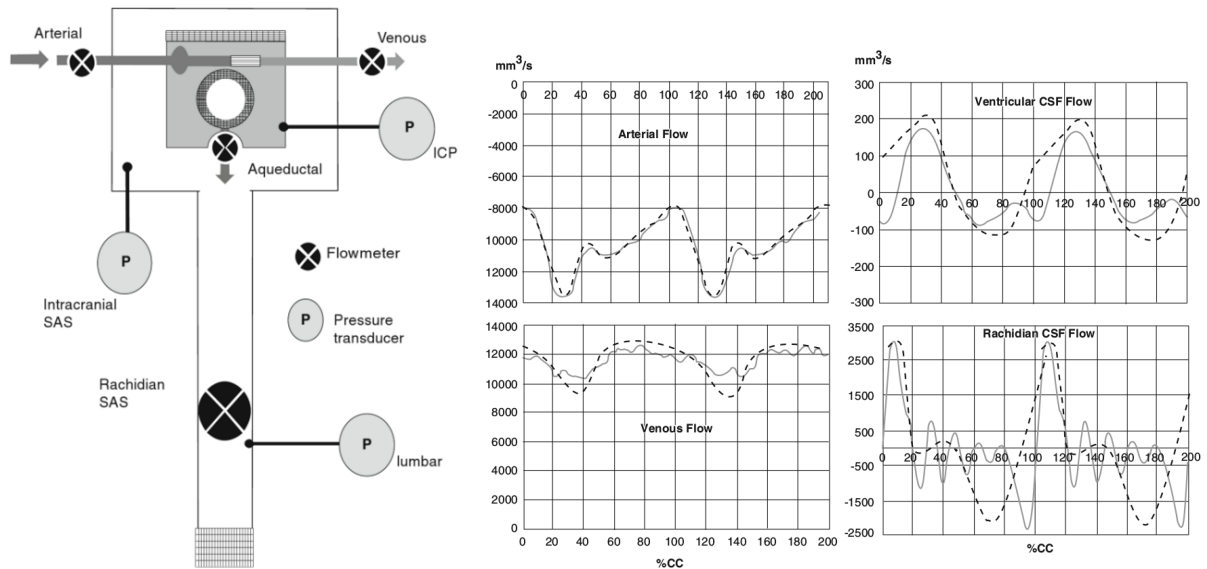


FIGURE 2.3 – Étude d'Ursino *et al.* Résultat dans le cas d'un test d'infusion. Lors de la phase d'injection, la pression augmente fortement, avant de retrouver un état d'équilibre : c'est ce que l'on appelle la régulation de la pression intracrânienne.

2.1.2 Modèles physiques

Une autre approche consiste à construire un système physique, appelé fantôme, visant à reproduire le comportement du système cérébrospinal qui, idéalement, est compatible avec l'imagerie par résonance magnétique (IRM). Un tel fantôme a été créé par Bouzerar *et al.* [13], Fig. 2.4a, afin de recréer les conditions de circulation du LCS en interaction avec le compartiment sanguin.



(a) Fantôme physique du système cérébrospinal. (b) Entrées et sorties. Ligne continue : mesures sur le fantôme. Ligne pointillée : mesures par IRM de flux.

FIGURE 2.4 – Étude de Bouzerar *et al.* Extrait de : *Physical phantom of craniospinal hydrodynamics* [13].

L'unique entrée du système est le flux artériel, qui actionne la mécanique du système cérébrospinal. Les flux et pressions sont mesurés aux endroits essentiels du système pour effectuer une comparaison avec les mesures par IRM de flux. Les entrées et sorties du modèle sont illustrées Fig. 2.4b.

Ce fantôme permet de reproduire fidèlement la sortie veineuse du système cérébrospinal et également d'obtenir une courbe de pression intracrânienne attendue. Néanmoins, ce type de modélisation ne permet pas de faire varier les paramètres facilement, mis à part ceux du flux artériel : l'élasticité ainsi que les dimensions du fantôme ne peuvent pas être modifiées.

Dernièrement, Daners *et al.* [23] ont recensé les fantômes physiques conçus dans leur laboratoire, en précisant les avantages et inconvénients de chacun. Il ont également créé un nouveau fantôme, Fig. 2.5a, permettant de reproduire une courbe de pression-volume donnée grâce à un système d'asservissement qui contrôle la pression de l'eau ou de l'air présent dans le fantôme. Des tests en régime oscillatoire ont été réalisés afin d'observer la courbe de pression intracrânienne au cours du temps dans différentes conditions de pression moyenne.

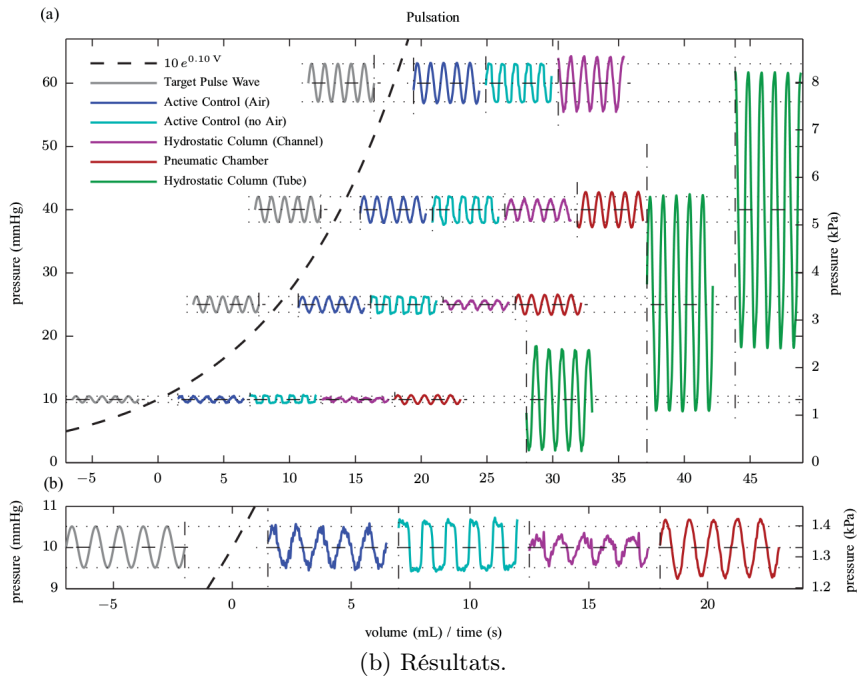
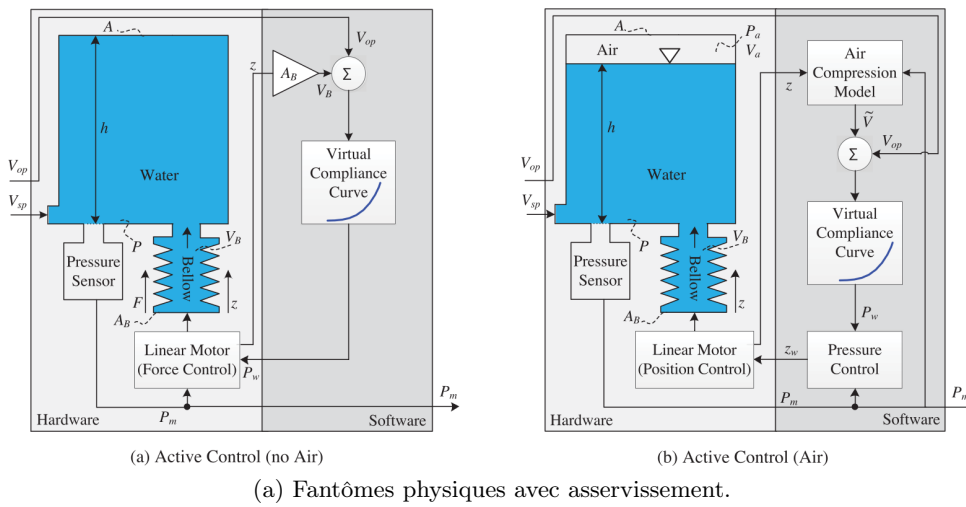


FIGURE 2.5 – Étude de Daners *et al.* Extrait de : *Craniospinal pressure-volume dynamics in phantom models* [23].

Les résultats obtenus, Fig. 2.5b, montrent l'évolution de l'amplitude de la courbe de pression en fonction de la courbe pression-volume imposée pour différents types d'asservissements. On observe bien une augmentation de l'amplitude de la courbe de pression avec l'augmentation du volume.

2.1.3 Modèles numériques

Les approches numériques permettent, pour leur part, d'avoir un contrôle total des paramètres du système, mais requièrent beaucoup de ressources de calcul. La plupart de ces études modélisent l'écoulement du LCS en utilisant les équations de Stokes ou de Navier-Stokes, optionnellement en interaction fluide-structure ou couplées à des modèles de milieux poreux.

Les études d'Alperin *et al.* [3, 4] sont les plus proches de ce travail de thèse car elles présentent une méthodologie permettant de déterminer la pression intracrânienne à partir de données de flux obtenues par IRM, donc de manière non-invasive.

Le modèle utilisé est un tube concentrique représentant le compartiment spinal du LCS (le canal spinal). L'écoulement du fluide est simulé dans ce domaine en résolvant les équations de Stokes avec, en condition limite d'entrée, le flux de LCS mesuré par IRM de flux et en condition limite de sortie, une pression nulle (sortie libre). Une courbe de pression est obtenue en sortie du modèle.

Les flux sanguins mesurés par IRM de flux permettent le calcul de changement de volume intracrânien (intracranial volume change - IVC).

La pression absolue est finalement obtenue en utilisant la relation pression-volume de Marmarou, Eq. (2.2), en choisissant arbitrairement le paramètre élastique E_1 :

$$\frac{dP}{dV} = E_1 \cdot P \quad (2.2)$$

Remarque 2.1. *Le modèle présenté dans ces études est extrêmement simplifié et ne prend pas en compte les différents compartiments du système cérébrospinal; seule une partie du compartiment spinal est étudiée sans prendre en compte son extrémité inférieure (fluide en sortie libre). La variabilité du paramètre d'élasticité n'est pas non plus prise en compte; elle est identique pour chaque cas afin de déterminer la pression absolue.*

Remarque 2.2. *Cette technique faisant l'objet d'un brevet aux États-Unis, peu de résultats sont présentés.*

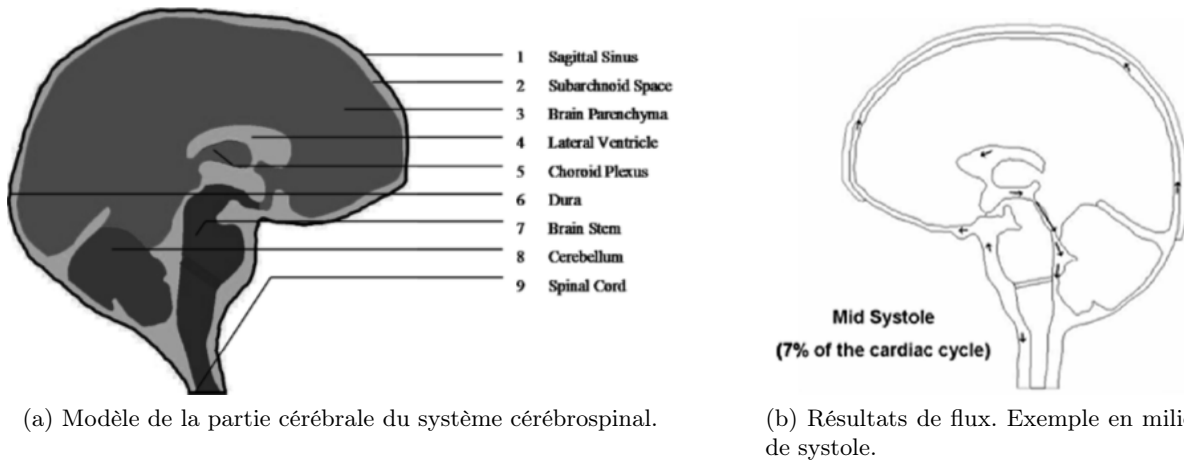
Linninger *et al.* [47] ont proposé une modélisation de la partie cérébrale du système cérébrospinal en 2 dimensions, Fig. 2.6, utilisant les milieux poreux.

Le modèle est utilisé dans un cas sain et dans un cas pathologique, afin de déterminer de manière non invasive les caractéristiques de l'écoulement de LCS et la pression intracrânienne. L'amplitude de pression obtenue est correcte mais les auteurs décrivent une impossibilité à déterminer la pression absolue qui n'apparaît pas dans les équations de Navier-Stokes (ou de Stokes). Une amplitude plus importante est observée dans le cas pathologique par rapport au cas sain.

Remarque 2.3. *Contrairement aux études d'Alperin et al. [3, 4], l'étude de Linninger et al. [47] évoque l'impossibilité de déterminer la pression absolue à partir uniquement du gradient de pression.*

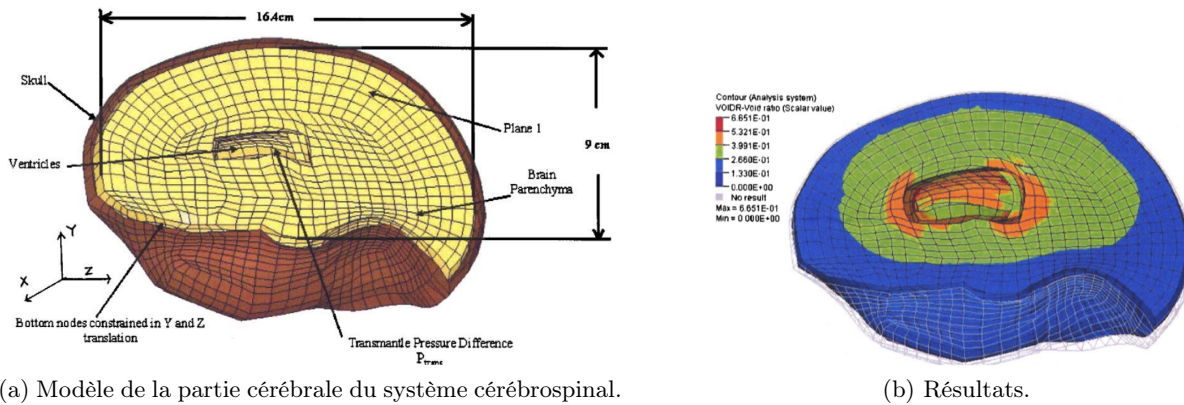
Dutta-Roy *et al.* [24] ont proposé une modélisation de la partie cérébrale du système cérébrospinal en interaction fluide structure en 3 dimensions, Fig. 2.7a.

Cette modélisation, réalisée à l'aide du logiciel ABAQUS [40], a pour but de reproduire les conditions de l'hydrocéphalie à pression normale, notamment d'observer la saturation en LCS des



(a) Modèle de la partie cérébrale du système cérébrospinal.

(b) Résultats de flux. Exemple en milieu de systole.

FIGURE 2.6 – Étude de Linninger *et al.* Extrait de : *Cerebrospinal fluid flow in the normal and hydrocephalic human brain* [47].

(a) Modèle de la partie cérébrale du système cérébrospinal.

(b) Résultats.

FIGURE 2.7 – Étude de Dutta-Roy *et al.* Extrait de : *Biomechanical modelling of normal pressure hydrocephalus* [24].

tissus avoisinants le compartiment liquidien, Fig. 2.7b. Les modifications du volume ventriculaires sont observées afin de déterminer les mécanismes de mise en place de l'hydrocéphalie. Cette étude fournit également une synthèse bibliographique concernant les paramètres mécaniques du cerveau. Nous utiliserons une partie de ces paramètres, présentés Tab. 2.2, dans le chapitre 7.

TABLE 2.2 – Paramètres mécaniques du cerveau. Dutta-Roy *et al.* [24].

Coefficient de Poisson ν	Module d'Young E (Pa)
0.35	420.58

2.2 Études expérimentales

2.2.1 Monitoring de pression intracrânienne

Les études les plus avancées sur le monitoring de la pression intracrânienne ont été menées par Czosnyka *et al.* [22] à l'aide d'un logiciel développé au sein de leur laboratoire de l'Université de Cambridge : ICM+ [77].

Ce logiciel, utilisé en pratique clinique, permet, entre autres, de surveiller et d'enregistrer (de manière invasive) la pression intracrânienne d'un patient. Il permet par exemple de mesurer la pression intracrânienne durant un test d'infusion, Fig. 2.8a ou durant de longues périodes (une nuit), Fig. 2.8b.

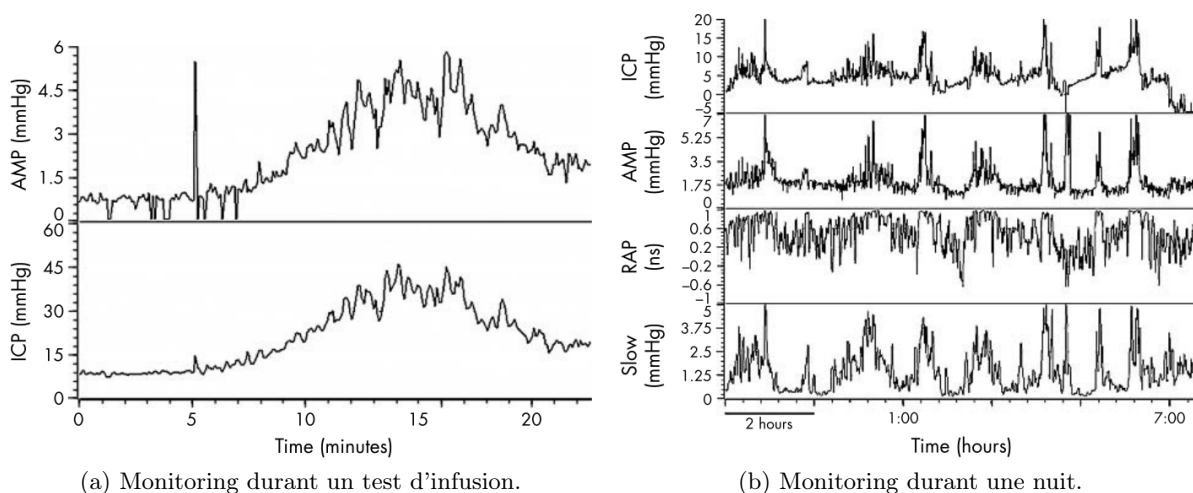


FIGURE 2.8 – Études de Czosnyka *et al.* Extrait de : *Monitoring and interpretation of intracranial pressure* [22].

Ce logiciel permet d'exporter la courbe de pression intracrânienne que nous utiliserons afin de valider les résultats obtenus grâce à nos modèles numériques.

2.2.2 Forme de la courbe de pression intracrânienne

Cardoso *et al.* [17] ont étudié des patients atteints de différentes pathologies afin d'observer la forme de la courbe de pression intracrânienne. Ils mettent en évidence une forme de courbe composée de trois pics (P_1 , P_2 , P_3) et de trois vallées, Fig. 2.9. Ils attribuent le premier pic (P_1) à la pulsation artérielle du sang arrivant dans le cerveau et le troisième pic (P_3) à fermeture de la valve aortique, au niveau du cœur. Le second pic (P_2) reste quant à lui inexpliqué ; les auteurs supposent un rôle de la compliance du système cérébrospinal dans la production de ce pic.

Remarque 2.4. *Le pic P_3 est dû à la fermeture de la valve aortique qui provoque une onde de pression dans le réseau veineux ; cette onde de pression se répercute sur la pression intracrânienne.*



FIGURE 2.9 – Étude de Cardoso *et al.* Forme de la pression intracrânienne dans un cas physiologique. Extrait de : *Analysis of the cerebrospinal fluid pulse wave in intracranial pressure* [17].

Hamilton *et al.* [38] ont développé un logiciel permettant de détecter les différents paramètres qui caractérisent la forme de la courbe de pression intracrânienne (au nombre de 128), Fig. 2.10, dans

le but de les étudier sur différents patients. Cette étude conclut à la relation entre la circulation du LCS et le second et le troisième pic (P_2 et P_3), contrairement aux études précédentes qui mettait le second pic (P_2) en relation avec la compliance du système cérébrospinal et le troisième pic (P_3) en relation avec le compartiment veineux. Le second pic (P_2) serait produit soit par l'arrivée de sang jusqu'au centre du cerveau, au niveau des ventricules, avec un retard par rapport au drainage périphérique ; soit par l'arrivée de LCS dans l'aqueduc, lui aussi avec un retard. Le troisième pic (P_3) serait dû au retour du LCS du compartiment spinal vers le compartiment cérébral.

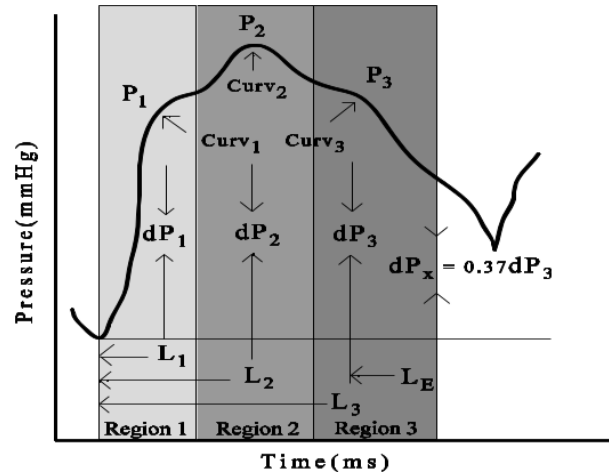


FIGURE 2.10 – Étude de Hamilton *et al.* Analyse de la forme de la pression intracrânienne dans un cas pathologique. Extrait de : *Subpeak regional analysis of intracranial pressure waveform morphology based on cerebrospinal fluid hydrodynamics in the cerebral aqueduct and prepontine cistern* [38].

2.3 Études cliniques

L'étude de Wilson [81] propose différents scénarii de modification délétère de la pression intracrânienne ($PIC > 20$ mmHg) avec une attention particulière sur le compartiment veineux. Parmi les propositions détaillées, on retrouve notamment :

- compression veineuse localisée ;
- sténose du sinus veineux ;
- obstruction veineuse localisée ;
- compression diffuse du système veineux ;
- mauvaise posture de la tête ;
- port d'une minerve ;
- augmentation de la pression intra-thoracique ou intra-abdominale.

Ces propositions sont issues d'observations expérimentales en pratique clinique.

Chapitre 3

Modèles utilisés

Les deux principaux modèles que nous allons utiliser par la suite sont présentés dans ce chapitre, ainsi que les modèles physiques à prendre en compte. Une présentation plus détaillée sera faite dans les chapitres relatifs à ces différents modèles.

3.1 Bifurcation

Dans un premier temps, et afin d'étudier la distribution du liquide cébrospinal (LCS) entre les différents compartiments du système cérébrospinal, un modèle très simple est considéré : une bifurcation, Fig. 3.1a. Ce modèle a pour but de déterminer la distribution du LCS entre les ventricules cérébraux, les espaces subarachnoïdiens (ESA) cérébraux et les ESA spinaux.

Ce modèle en forme de bifurcation, provient de l'observation du système cérébrospinal en imagerie par résonance magnétique (IRM) morphologique. Lorsque l'on représente schématiquement les différents compartiments du système cérébrospinal, on peut réduire leurs liens à une bifurcation, Fig. 3.1b.

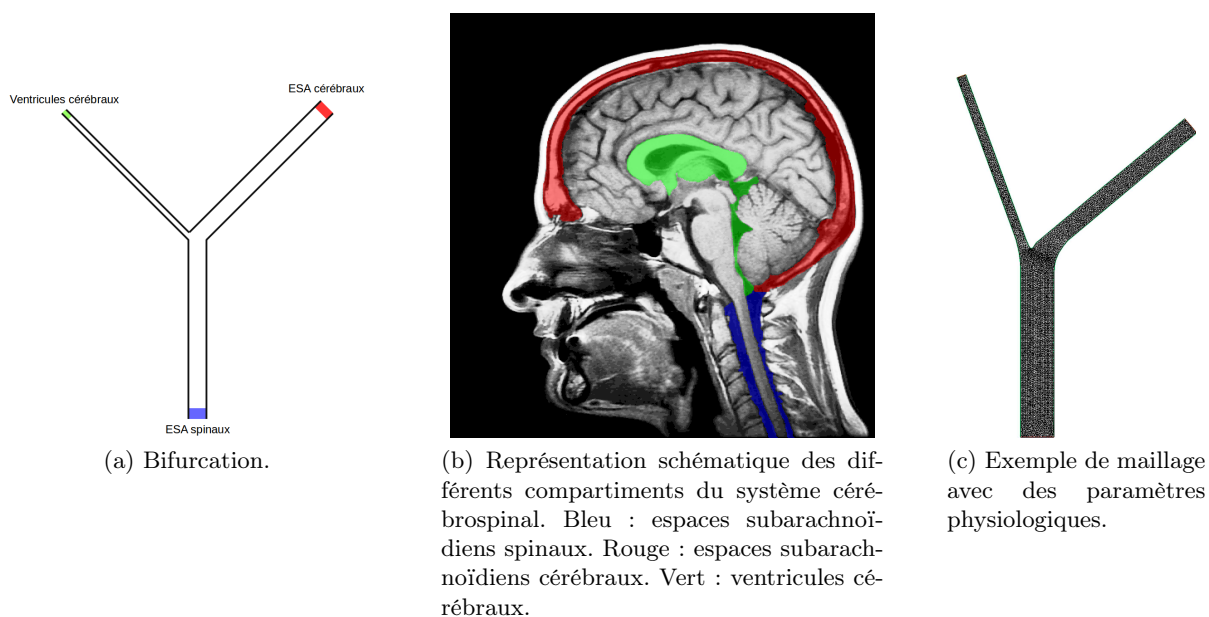


FIGURE 3.1 – **Bifurcation.** Modèle en bifurcation et observation IRM.

Ce modèle fait intervenir uniquement le LCS ; on aura donc à résoudre les équations de la mécanique des fluides : les équations de Stokes ou les équations de Navier-Stokes incompressibles instationnaires, il n'y a donc pas d'interaction fluide-structure dans ce modèle.

Néanmoins, il faut tout de même prendre en compte l'interaction entre les ventricules cérébraux et les ESA cérébraux, tous deux contenus dans la boîte crânienne rigide, ainsi que l'interaction du cerveau (qui reçoit du sang à chaque pulsation cardiaque) avec le compartiment de LCS. La modélisation de ces interactions fera intervenir des conditions aux limites spécifiques à notre modèle, les modèles de Windkessel.

3.2 Système cérébrospinal simplifié

Afin de rendre notre modèle plus réaliste, on représente désormais le système cérébrospinal par un modèle simplifié. Il fait intervenir le LCS ainsi que le cerveau, en 2 dimensions, Fig. 3.2a, ou en 3 dimensions, Fig. 3.2b.

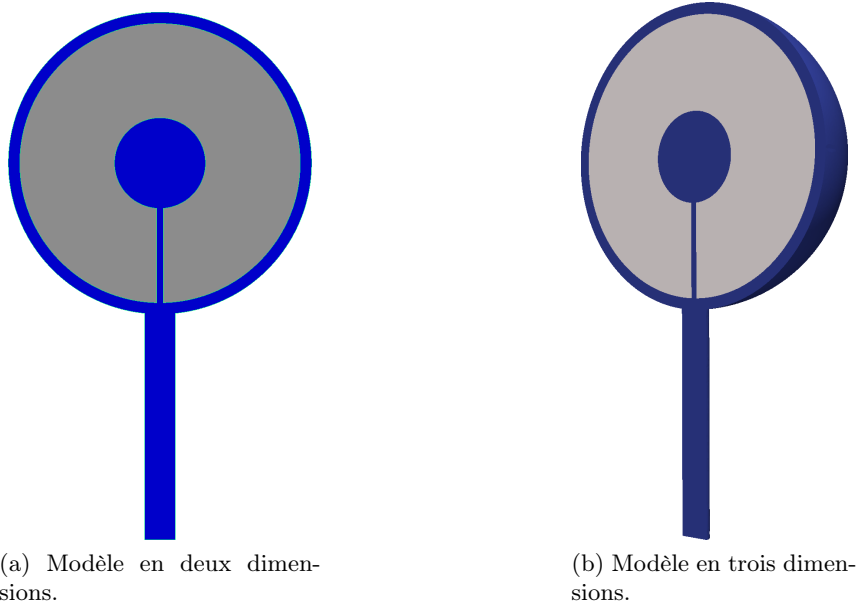


FIGURE 3.2 – **Système cérébrospinal simplifié.** Représentation 2D et 3D.

Il y a désormais une partie fluide et une partie structure ; on aura donc à résoudre un problème d'interaction fluide-structure faisant intervenir les équations de Stokes / Navier-Stokes et les équations de l'élasticité linéaire.

Première partie

Modélisation numérique

Chapitre 4

Méthodes numériques

La détermination de la pression intracrânienne (PIC), qui est la pression du liquide cébrospinal (LCS), se fait à travers la simulation numérique de la circulation du LCS. Nous allons donc avoir à résoudre les équations de Navier-Stokes incompressibles instationnaires.

Ce fluide est confiné dans un environnement clos, la boîte crânienne, et est en interaction avec l'encéphale et la moelle spinale. Nous allons étudier différentes techniques afin de représenter fidèlement le comportement du système à l'aide d'interactions fluide-structure ou non.

Parmi les différentes techniques de simulation numérique existantes, nous avons choisi la méthode des éléments finis. De nombreuses bibliothèques proposent des solutions de codage de cette méthode, parmi lesquels `FREEFEM++` [39] et `FEEL++` [66] que l'on utilisera dans cette partie car, elles ont l'avantage d'être libres et très performantes.

4.1 Simulation de la dynamique des fluides

La modélisation numérique d'un fluide visqueux newtonien incompressible fait intervenir les équations de Navier-Stokes incompressibles. Comme le fluide étudié, le LCS, présente une pulsativité, on considère les équations de Navier-Stokes incompressibles instationnaires Eq. (4.1).

Soit un domaine Ω de dimension d de frontière $\Gamma = \Gamma_D \cup \Gamma_N$, où Γ_D et Γ_N sont les portions de la frontière où on impose une condition de Dirichlet et une condition de Neumann, respectivement. Soit T le temps total de la simulation ; alors les équations de Navier-Stokes s'écrivent :

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p - \mu \Delta \mathbf{u} = \mathbf{f} \quad \text{dans } [0, T] \times \Omega \quad (4.1a)$$

$$\operatorname{div}(\mathbf{u}) = 0 \quad \text{dans } [0, T] \times \Omega \quad (4.1b)$$

$$\mathbf{u} = \mathbf{u}_D \quad \text{dans } [0, T] \times \Gamma_D \quad (4.1c)$$

$$\mu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} = \mathbf{g} \quad \text{sur } [0, T] \times \Gamma_N \quad (4.1d)$$

$$\mathbf{u} = \mathbf{u}^0 \quad \text{dans } \{0\} \times \Omega \quad (4.1e)$$

ρ est la masse volumique, μ la viscosité dynamique, \mathbf{u} est la vitesse (vecteur de dimension d), p la pression (scalaire), \mathbf{f} la force volumique extérieure (généralement prise à 0 car on néglige la gravité).

L'équation (4.1a) est appelée équation de conservation de la quantité de mouvement. L'équation (4.1b) est appelée condition d'incompressibilité. \mathbf{u}_0 est la condition initiale, et les équations (4.1c) et (4.1d) sont les conditions limites au bord du domaine (Dirichlet et Neumann, respectivement).

L'équation de conservation de la quantité de mouvement, Eq. (4.1a), est scindée en plusieurs termes :

- $\rho \frac{\partial \mathbf{u}}{\partial t}$, le terme instationnaire ;
- $\rho(\mathbf{u} \cdot \nabla) \mathbf{u}$, le terme de convection ;
- ∇p , le terme de gradient de pression ;
- $\mu \Delta \mathbf{u}$, le terme visqueux.

Ces équations seront résolues numériquement à l'aide de la méthode des éléments finis, avec le langage de programmation FREEFEM++ [39].

Remarque 4.1. *Dans la condition de Neumann, un tenseur non-symétrique $\mu \nabla \mathbf{u} - p \mathbf{I}$ est utilisé. Cette forme est plus adaptée aux applications physiologiques que son homologue symétrique $\mu \left(\frac{\nabla \mathbf{u} + \nabla \mathbf{u}^t}{2} \right) - p \mathbf{I}$ car elle est plus représentative du comportement du fluide dans un domaine tronqué, au niveau de la sortie notamment. Une comparaison entre ces deux formes est donnée dans une étude de J. Fouchet et al. [27].*

Une simplification des équations de Navier-Stokes consiste à ne pas prendre en compte le terme de convection, non-linéaire. On obtient donc le système d'équations de Stokes suivant :

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \nabla p - \mu \Delta \mathbf{u} = \mathbf{f} \quad \text{dans } [0, T] \times \Omega \quad (4.2a)$$

$$\operatorname{div}(\mathbf{u}) = 0 \quad \text{dans } [0, T] \times \Omega \quad (4.2b)$$

$$\mathbf{u} = \mathbf{u}_D \quad \text{dans } [0, T] \times \Gamma_D \quad (4.2c)$$

$$\mu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} = \mathbf{g} \quad \text{sur } [0, T] \times \Gamma_N \quad (4.2d)$$

$$\mathbf{u} = \mathbf{u}^0 \quad \text{dans } \{0\} \times \Omega \quad (4.2e)$$

Le problème de Stokes convient à la modélisation de la circulation de fluides dans le cas très visqueux ou à très basse vitesse.

4.1.1 Préliminaires

Lemme 4.1. *Formules de Green [36]*

Soient $p \in H^1(\Omega)$, $\mathbf{v} \in H(\operatorname{div})$, alors :

$$\int_{\Omega} \nabla p \cdot \mathbf{v} = - \int_{\Omega} p \operatorname{div}(\mathbf{v}) + \int_{\partial \Omega} p \mathbf{v} \cdot \mathbf{n} \quad (4.3)$$

Soient $\mathbf{u} \in H^2(\Omega)$, $\mathbf{v} \in H^1(\Omega)$, alors :

$$\int_{\Omega} \Delta \mathbf{u} \cdot \mathbf{v} = - \int_{\Omega} \nabla \mathbf{u} \cdot \nabla \mathbf{v} + \int_{\partial \Omega} \frac{\partial \mathbf{u}}{\partial \mathbf{n}} \cdot \mathbf{v} \quad (4.4)$$

Soit h un paramètre de discrétisation, appelé pas du maillage. On note \mathcal{T}_h une collection de simplexes non vides et disjoints formant une partition de Ω . On appelle Ω_h le domaine discrétisé par \mathcal{T}_h .

On note H^1 l'espace de Sobolev, L^2 l'espace de Lebesgue et $\mathbf{v}|_{\Gamma_D}$ la trace de la fonction \mathbf{v} sur le bord Γ_D .

Soient les espaces fonctionnels suivants :

$$\begin{aligned}
V^0 &= \{ \mathbf{v} \in H^1(\Omega), \mathbf{v}|_{\Gamma_D} = 0 \} & V^D &= \{ \mathbf{v} \in H^1(\Omega), \mathbf{v}|_{\Gamma_D} = \mathbf{u}_D \} \\
V_t^0 &= L^2([0, T]; V^0) & V_t^D &= L^2([0, T]; V^D) \\
V_h^0 &= \{ \mathbf{v}_h \in H^1(\Omega_h), \mathbf{v}_h|_{\Gamma_D} = 0 \} & V_h^D &= \{ \mathbf{v}_h \in H^1(\Omega_h), \mathbf{v}_h|_{\Gamma_D} = \mathbf{u}_D \} \\
V_{t,h}^0 &= L^2([0, T]; V_h^0) & V_{t,h}^D &= L^2([0, T]; V_h^D) \\
Q &= L^2(\Omega) \\
Q_h &= L^2(\Omega_h) \\
Q_t &= L^2([0, T]; Q) \\
Q_{t,h} &= L^2([0, T]; Q_h)
\end{aligned}$$

4.1.2 Grandeurs caractéristiques

La circulation d'un fluide est caractérisée par différents nombres sans dimensions ; le nombre de Reynolds qui détermine si le flux est turbulent ou non ; le nombre de Strouhal qui caractérise l'instantanéité du fluide ; et le nombre de Womersley qui caractérise les effets visqueux dans un flux pulsé.

Nombre de Reynolds

Le nombre de Reynolds [61] est un nombre adimensionné ; il caractérise le régime de l'écoulement : laminaire, transitoire, turbulent. Il est défini comme le ratio entre les forces d'inertie et les forces visqueuses.

$$Re = \frac{\rho \mathbf{v} L}{\mu} \quad (4.6)$$

avec ρ la masse volumique du fluide, \mathbf{v} la vitesse moyenne du fluide, L la longueur caractéristique et μ la viscosité dynamique du fluide.

Dans le cas particulier d'un tube de diamètre constant, le nombre de Reynolds s'écrit :

$$Re = \frac{\rho \mathbf{v} R}{\mu} \quad (4.7)$$

avec \mathbf{v} la vitesse moyenne du fluide et R le rayon du tube.

Dans ce cas particulier [64], le flux est laminaire pour $Re < 2300$, transitoire pour $2300 \leq Re \leq 4000$ et turbulent pour $Re > 4000$.

Nombre de Strouhal

Le nombre de Strouhal [44] est un nombre sans dimension ; il quantifie l'importance des termes inertiels.

$$St = \frac{fL}{\mathbf{v}} \quad (4.8)$$

avec f la fréquence, L la longueur caractéristique et \mathbf{v} la vitesse moyenne.

Dans le cas d'un tube circulaire, on a :

$$St = \frac{fR}{\mathbf{v}} \quad (4.9)$$

avec R le rayon et \mathbf{v} la vitesse moyenne.

Nombre de Womersley

Le nombre de Womersley est un nombre sans dimension ; il met en relation l'inertie totale de l'écoulement avec les effets visqueux dans les artères [82].

$$Wo = R\sqrt{\frac{\rho\omega}{\mu}} \quad (4.10)$$

avec R le rayon de l'artère, ρ la masse volumique du fluide, ω la fréquence angulaire et μ la viscosité dynamique fluide.

Remarque 4.2. *L'écoulement de Womersley est obtenu analytiquement en considérant les équations de Stokes instationnaires.*

Relation

Dans le cas d'un tube circulaire, ces grandeurs caractéristiques peuvent être mises en relation par :

$$Wo = \sqrt{2\pi Re.St} \quad (4.11)$$

En effet :

$$Re.St = \frac{\rho \mathbf{v} R}{\mu} \frac{f R}{\mathbf{v}} = R^2 \frac{\rho f}{\mu}$$

Donc :

$$2\pi Re.St = R^2 \frac{\rho 2\pi f}{\mu} = R^2 \frac{\rho \omega}{\mu}$$

Finalement :

$$\sqrt{2\pi Re.St} = R\sqrt{\frac{\rho\omega}{\mu}} = Wo$$

4.1.3 Formulation variationnelle des équations de Stokes

Soient $\mathbf{v} \in V_t^0$, $q \in Q_t$. La formulation variationnelle mixte [58] des équations de Stokes Eq. (4.2) s'écrit :

Trouver $\mathbf{u} \in V_t^D$, $p \in Q_t$ tels que $\forall \mathbf{v} \in V_t^0$, $q \in Q_t$:

$$\rho \int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} + \int_{\Omega} \nabla p \cdot \mathbf{v} - \mu \int_{\Omega} \Delta \mathbf{u} \cdot \mathbf{v} = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \quad (4.12a)$$

$$\int_{\Omega} \operatorname{div}(\mathbf{u})q = 0 \quad (4.12b)$$

Remarque 4.3. *La formulation variationnelle mixte nous impose une condition de compatibilité entre les espaces éléments finis utilisés pour la vitesse et pour la pression ; on a donc une condition Inf-Sup [7, 14] à respecter.*

En utilisant le Lemme 4.1, on obtient :

$$\rho \int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} - \int_{\Omega} p \operatorname{div}(\mathbf{v}) + \mu \int_{\Omega} \nabla \mathbf{u} \cdot \nabla \mathbf{v} - \int_{\Gamma_N} \left(\mu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} \right) \cdot \mathbf{v} = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \quad (4.13a)$$

$$\int_{\Omega} \operatorname{div}(\mathbf{u})q = 0 \quad (4.13b)$$

Discrétisation en espace

Soient $\mathbf{v}_h \in V_{t,h}^0$, $q_h \in Q_{t,h}$. La formulation variationnelle mixte discrète s'écrit :

Trouver $\mathbf{u}_h \in V_{t,h}^D$, $p_h \in Q_{t,h}$ tels que $\forall \mathbf{v}_h \in V_{t,h}^0$, $q_h \in Q_{t,h}$:

$$\begin{aligned} \rho \int_{\Omega} \frac{\partial \mathbf{u}_h}{\partial t} \cdot \mathbf{v}_h - \int_{\Omega} p_h \operatorname{div}(\mathbf{v}_h) + \mu \int_{\Omega} \nabla \mathbf{u}_h \cdot \nabla \mathbf{v}_h \\ - \int_{\Gamma_N} \left(\mu \frac{\partial \mathbf{u}_h}{\partial \mathbf{n}} - p_h \mathbf{n} \right) \cdot \mathbf{v}_h = \int_{\Omega} \mathbf{f} \cdot \mathbf{v}_h \end{aligned} \quad (4.14a)$$

$$\int_{\Omega} \operatorname{div}(\mathbf{u}_h) q = 0 \quad (4.14b)$$

Discrétisation en temps

La discrétisation en temps fait intervenir un schéma aux différences finies à l'ordre 1 :

$$\frac{\partial \mathbf{u}}{\partial t} = \frac{\mathbf{u}^{n-1} - \mathbf{u}^n}{\Delta t} \quad (4.15)$$

Soit $t^k = k\Delta t$, $0 \leq k \leq N$ avec $\Delta t = \frac{T}{N}$ le pas de temps.

En notant (\mathbf{u}^k, p^k) la solution au temps t^k , la formulation variationnelle mixte discrète s'écrit :

Trouver $\mathbf{u}_h^{k+1} \in V_{t,h}^D$, $p_h^{k+1} \in Q_{t,h}$ tels que $\forall \mathbf{v}_h \in V_{t,h}^0$, $q_h \in Q_{t,h}$:

$$\begin{aligned} \rho \int_{\Omega} \frac{\mathbf{u}_h^{k+1} - \mathbf{u}_h^k}{\Delta t} \cdot \mathbf{v}_h - \int_{\Omega} p_h^{k+1} \operatorname{div}(\mathbf{v}_h) + \mu \int_{\Omega} \nabla \mathbf{u}_h^{k+1} \cdot \nabla \mathbf{v}_h \\ - \int_{\Gamma_N} \left(\mu \frac{\partial \mathbf{u}_h^{k+1}}{\partial \mathbf{n}} - p_h^{k+1} \mathbf{n} \right) \cdot \mathbf{v}_h = \int_{\Omega} \mathbf{f}^{k+1} \cdot \mathbf{v}_h \end{aligned} \quad (4.16a)$$

$$\int_{\Omega} \operatorname{div}(\mathbf{u}_h^{k+1}) q = 0 \quad (4.16b)$$

L'algorithme FREEFEM++ et les détails de programmation sont donnés dans l'Annexe C.1.

4.1.4 Formulation variationnelle des équations de Navier-Stokes

Soient $\mathbf{v} \in V_t^0$, $q \in Q_t$. La formulation variationnelle mixte [58] des équations de Navier-Stokes Eq. (4.1) s'écrit :

Trouver $\mathbf{u} \in V_t^D$, $p \in Q_t$ tels que $\forall \mathbf{v} \in V_t^0$, $q \in Q_t$:

$$\rho \int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} + \rho \int_{\Omega} (\mathbf{u} \cdot \nabla) \mathbf{u} \cdot \mathbf{v} + \int_{\Omega} \nabla p \cdot \mathbf{v} - \mu \int_{\Omega} \Delta \mathbf{u} \cdot \mathbf{v} = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \quad (4.17a)$$

$$\int_{\Omega} \operatorname{div}(\mathbf{u}) q = 0 \quad (4.17b)$$

Remarque 4.4. La formulation variationnelle mixte nous impose une condition de compatibilité entre les espaces éléments finis utilisés pour la vitesse et pour la pression ; on a donc une condition Inf-Sup à respecter.

En utilisant le Lemme 4.1, on obtient :

Trouver $\mathbf{u} \in V_t^D$, $p \in Q_t$ tels que $\forall \mathbf{v} \in V_t^0$, $q \in Q_t$:

$$\begin{aligned} \rho \int_{\Omega} \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) \cdot \mathbf{v} - \int_{\Omega} p \operatorname{div}(\mathbf{v}) + \int_{\Gamma_N} p \mathbf{n} \cdot \mathbf{v} \\ + \mu \int_{\Omega} \nabla \mathbf{u} \cdot \nabla \mathbf{v} - \mu \int_{\Gamma_N} \frac{\partial \mathbf{u}}{\partial \mathbf{n}} \cdot \mathbf{v} = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \end{aligned} \quad (4.18a)$$

$$\int_{\Omega} \operatorname{div}(\mathbf{u}) q = 0 \quad (4.18b)$$

On a finalement :

Trouver $\mathbf{u} \in V_t^D$, $p \in Q_t$ tels que $\forall \mathbf{v} \in V_t^0$, $q \in Q_t$:

$$\begin{aligned} \rho \int_{\Omega} \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) \cdot \mathbf{v} - \int_{\Omega} p \operatorname{div}(\mathbf{v}) + \mu \int_{\Omega} \nabla \mathbf{u} \cdot \nabla \mathbf{v} \\ - \int_{\Gamma_N} \left(\mu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} \right) \cdot \mathbf{v} = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \end{aligned} \quad (4.19a)$$

$$\int_{\Omega} \operatorname{div}(\mathbf{u}) q = 0 \quad (4.19b)$$

Discrétisation en espace

Soient $\mathbf{v}_h \in V_{t,h}^0$, $q_h \in Q_{t,h}$. La formulation variationnelle mixte discrète s'écrit :

Trouver $\mathbf{u}_h \in V_{t,h}^D$, $p_h \in Q_{t,h}$ tels que $\forall \mathbf{v}_h \in V_{t,h}^0$, $q_h \in Q_{t,h}$:

$$\begin{aligned} \rho \int_{\Omega} \left(\frac{\partial \mathbf{u}_h}{\partial t} + (\mathbf{u}_h \cdot \nabla) \mathbf{u}_h \right) \cdot \mathbf{v}_h - \int_{\Omega} p_h \operatorname{div}(\mathbf{v}_h) \\ + \mu \int_{\Omega} \nabla \mathbf{u}_h \cdot \nabla \mathbf{v}_h - \int_{\Gamma_N} \left(\mu \frac{\partial \mathbf{u}_h}{\partial \mathbf{n}} - p_h \mathbf{n} \right) \cdot \mathbf{v}_h = \int_{\Omega} \mathbf{f} \cdot \mathbf{v}_h \end{aligned} \quad (4.20a)$$

$$\int_{\Omega} \operatorname{div}(\mathbf{u}_h) q = 0 \quad (4.20b)$$

Discrétisation en temps

Soit $t^k = k\Delta t$, $0 \leq k \leq N$ avec $\Delta t = \frac{T}{N}$ le pas de temps.

En notant (\mathbf{u}^k, p^k) la solution au temps t^k , la formulation variationnelle mixte discrète s'écrit :

Trouver $\mathbf{u}_h^{k+1} \in V_{t,h}^D$, $p_h^{k+1} \in Q_{t,h}$ tels que $\forall \mathbf{v}_h \in V_{t,h}^0$, $q_h \in Q_{t,h}$:

$$\begin{aligned} \rho \int_{\Omega} \left(\frac{\partial \mathbf{u}_h^{k+1}}{\partial t} \Big|_{t=t_k} + (\mathbf{u}_h^{k+1} \cdot \nabla) \mathbf{u}_h^{k+1} \right) \cdot \mathbf{v}_h - \int_{\Omega} p_h^{k+1} \operatorname{div}(\mathbf{v}_h) \\ + \mu \int_{\Omega} \nabla \mathbf{u}_h^{k+1} \cdot \nabla \mathbf{v}_h - \int_{\Gamma_N} \left(\mu \frac{\partial \mathbf{u}_h^{k+1}}{\partial \mathbf{n}} - p_h^{k+1} \mathbf{n} \right) \cdot \mathbf{v}_h = \int_{\Omega} \mathbf{f}^{k+1} \cdot \mathbf{v}_h \end{aligned} \quad (4.21a)$$

$$\int_{\Omega} \operatorname{div}(\mathbf{u}_h^{k+1}) q = 0 \quad (4.21b)$$

4.1.5 Méthode des caractéristiques

Afin d'obtenir une équation linéaire, on peut utiliser la méthode des caractéristiques [12] pour linéariser le terme de convection.

Le terme instationnaire et le terme de convection $\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u}$ peuvent être vus comme une dérivée particulière. On peut alors écrire une trajectoire pour chaque particule :

$$\frac{d\mathbf{X}}{dt}(\mathbf{x}, s; t) = \mathbf{u}(t, \mathbf{X}(\mathbf{x}, s; t)) \quad (4.22a)$$

$$\mathbf{X}(\mathbf{x}, s; s) = \mathbf{x} \quad (4.22b)$$

$\mathbf{X}(\mathbf{x}, s; t)$ est la position d'une particule à l'instant t qui était en \mathbf{x} à l'instant s .

On peut alors approximer la dérivée particulière par :

$$\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) (t^{k+1}, \mathbf{x}) \approx \frac{\mathbf{u}(t^{k+1}, \mathbf{x}) - \mathbf{u}(t^k, \mathbf{X}^k(\mathbf{x}))}{\Delta t} + \mathcal{O}(\Delta t) \quad (4.23)$$

Avec $\mathbf{X}^k(\mathbf{x}) = \mathbf{x} - \mathbf{u}(t^k, \mathbf{x})\Delta t + \mathcal{O}(\Delta t^2)$.

On obtient finalement la formulation variationnelle mixte discrète :

$$\begin{aligned} \frac{\rho}{\Delta t} \int_{\Omega} \mathbf{u}_h^{k+1} \cdot \mathbf{v}_h - \frac{\rho}{\Delta t} \int_{\Omega} (\mathbf{u}_h^k \circ \mathbf{X}_h^k) \cdot \mathbf{v}_h - \int_{\Omega} p_h^{k+1} \operatorname{div}(\mathbf{v}_h) \\ + \mu \int_{\Omega} \nabla \mathbf{u}_h^{k+1} \cdot \nabla \mathbf{v}_h - \int_{\Gamma_N} \left(\mu \frac{\partial \mathbf{u}_h^{k+1}}{\partial \mathbf{n}} - p_h^{k+1} \mathbf{n} \right) \cdot \mathbf{v}_h = \int_{\Omega} \mathbf{f}^{k+1} \cdot \mathbf{v}_h \end{aligned} \quad (4.24a)$$

$$\int_{\Omega} \operatorname{div}(\mathbf{u}_h^{k+1}) q = 0 \quad (4.24b)$$

La méthode des caractéristiques peut être considérée comme une approximation implicite, et le schéma est inconditionnellement stable [58]. Les ordres de convergence pour la méthode des caractéristiques à l'ordre 1, en utilisant les espaces éléments finis de Taylor-Hood $\mathbb{P}_2\mathbb{P}_1$ (\mathbb{P}_2 pour la vitesse et \mathbb{P}_1 pour la pression, afin de respecter la condition Inf-Sup), sont donnés Tab. 4.1.

TABLE 4.1 – **Ordres de convergence.** Méthode des caractéristiques à l'ordre 1, espaces éléments finis de Taylor-Hood $\mathbb{P}_2\mathbb{P}_1$.

Variable	Convergence en espace	Convergence en temps
Vitesse	3	1
Pression	2	1

L'algorithme FREEFEM++ et les détails de programmation sont donnés dans l'Annexe C.2.

Algorithme d'Uzawa

L'algorithme d'Uzawa permet de remplacer la résolution directe du système donné par la méthode des caractéristiques par une méthode itérative ; ainsi les temps de calcul peuvent être divisés par 5 (cf. Annexe C.3).

Remarque 4.5. *L'algorithme d'Uzawa permet de résoudre, de manière itérative, des problèmes du type :*

$$\mathbf{A}\mathbf{u} + B^t p = \tilde{F} \quad (4.25a)$$

$$B\mathbf{u} = 0 \quad (4.25b)$$

Le terme provenant de la méthode des caractéristiques est traité de manière explicite dans \tilde{F} .

On veut résoudre le problème :

$$\begin{aligned} \frac{\rho}{\Delta t} \int_{\Omega} \mathbf{u}_h^{k+1} \cdot \mathbf{v}_h - \frac{\rho}{\Delta t} \int_{\Omega} (\mathbf{u}_h^k \circ \mathbf{X}_h^k) \cdot \mathbf{v}_h - \int_{\Omega} p_h^{k+1} \operatorname{div}(\mathbf{v}_h) \\ + \mu \int_{\Omega} \nabla \mathbf{u}_h^{k+1} \cdot \nabla \mathbf{v}_h - \int_{\Gamma_N} \left(\mu \frac{\partial \mathbf{u}_h^{k+1}}{\partial \mathbf{n}} - p_h^{k+1} \mathbf{n} \right) \cdot \mathbf{v}_h = \int_{\Omega} \mathbf{f}^{k+1} \cdot \mathbf{v}_h \end{aligned} \quad (4.26a)$$

$$\int_{\Omega} \operatorname{div}(\mathbf{u}_h^{k+1}) q = 0 \quad (4.26b)$$

Ce problème est équivalent à la forme matricielle :

$$\begin{pmatrix} A^{k+1} & (B^{k+1})^t \\ B^{k+1} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}_h^{k+1} \\ p_h^{k+1} \end{pmatrix} = \begin{pmatrix} F^{k+1} \\ 0 \end{pmatrix} \quad (4.27)$$

où A^{k+1} et B^{k+1} sont respectivement les matrices représentatives des formes bilinéaires :

$$a(\mathbf{u}^{k+1}, \mathbf{v}) = \frac{\rho}{\Delta t} \int_{\Omega} \mathbf{u}^{k+1} \mathbf{v} + \mu \int_{\Omega} \nabla \mathbf{u}^{k+1} \cdot \nabla \mathbf{v} \quad (4.28)$$

$$b(p^{k+1}, \mathbf{v}) = \int_{\Omega} p^{k+1} \operatorname{div}(\mathbf{v}) \quad (4.29)$$

et où F^{k+1} est la matrice représentative de la forme linéaire :

$$f(\mathbf{u}^{k+1}, \mathbf{v}) = \int_{\Omega} \mathbf{f}^{k+1} \mathbf{v} + \frac{\rho}{\Delta t} \int_{\Omega} (\mathbf{u}^k \circ \mathbf{X}) \mathbf{v} \quad (4.30)$$

On a donc à minimiser la fonctionnelle :

$$J(\mathbf{u}) = \frac{1}{2} (A\mathbf{u}, \mathbf{u}) - (F, \mathbf{u}) \quad (4.31)$$

En utilisant l'algorithme d'Uzawa gradient-conjugué [63] avec un préconditionnement de Cahouet-Chabard [16], cela revient, pour un pas de temps, à l'algorithme :

Initialisation

Soit p_h^1 connu
 $n = 1$.

Étape 1

Trouver \mathbf{u}_h^n tel que :
 $A\mathbf{u}_h^n + B^t p_h^n = F$
 Ce qui revient à calculer (p_h^n est connue) :
 $\mathbf{u}_h^n = A^{-1}(F - B^t p_h^n)$

Étape 1 bis

Calcul de ρ_n (gradient conjugué)

Étape 2

Calculer p_h^{n+1} :
 $p_h^{n+1} = p_h^n + \rho_n B\mathbf{u}_h^n$

Étape 3

$n = n + 1$.
 Retour à l'étape 1 jusqu'à convergence.

L'algorithme d'Uzawa étant une optimisation de la méthode des caractéristiques, la stabilité et les ordres de convergence sont identiques à cette dernière.

L'algorithme FREEFEM++ et les détails de programmation sont donnés dans l'Annexe C.3.

4.1.6 Algorithme de splitting

Une autre manière de résoudre les équations de Navier-Stokes, Eq. (4.1), est d'utiliser des techniques dites de «splitting» : la résolution est scindée en deux parties, un problème pour la pression, puis un problème pour la vitesse.

Le premier avantage de cette algorithme est d'être facilement parallélisable, contrairement à l'algorithme des caractéristiques ou à celui d'Uzawa qui utilise la méthode des caractéristiques ; un autre avantage est de n'être pas contraint ici par une condition Inf-Sup, le problème étant découplé.

Un tel algorithme, faisant partie de la classe des schémas consistants [37], a récemment été proposé dans la littérature [70, 71].

Le point de départ de cet algorithme est de considérer la forme équivalente des équations de Navier-Stokes, Eq. (4.1), donnée par :

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p - \mu \Delta \mathbf{u} = \mathbf{f} \quad \text{dans } \Omega \quad (4.32a)$$

$$\mathbf{u} = \mathbf{u}_D \quad \text{sur } \Gamma_D \quad (4.32b)$$

$$\mu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} = \mathbf{g} \quad \text{sur } \Gamma_N \quad (4.32c)$$

$$\Delta p = -\rho \nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) + \nabla \cdot \mathbf{f} - \rho \frac{\partial (\nabla \cdot \mathbf{u})}{\partial t} \quad \text{dans } \Omega \quad (4.32d)$$

$$\nabla p \cdot \mathbf{n} = \left(\mathbf{f} - \rho \frac{\partial \mathbf{u}}{\partial t} - \rho (\mathbf{u} \cdot \nabla) \mathbf{u} - \mu \nabla \times \nabla \times \mathbf{u} \right) \cdot \mathbf{n} \quad \text{sur } \Gamma_D \quad (4.32e)$$

$$p = \mu \mathbf{n} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - \alpha \mu \nabla \cdot \mathbf{u} - \mathbf{g} \cdot \mathbf{n} \quad \text{sur } \Gamma_N \quad (4.32f)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{dans } \Omega \quad (4.32g)$$

$$\mathbf{u}(0) = \mathbf{u}_0 \quad \text{dans } \Omega \quad (4.32h)$$

où α est une constante telle que $0 < \alpha < 1$.

Remarque 4.6. L'opérateur $\text{div}()$ est ici noté $\nabla \cdot$ pour simplifier les équations.

Consistance

La consistance de la forme équivalente, notamment la condition d'incompressibilité, est prouvée par le raisonnement suivant.

En multipliant l'équation (4.32a) par \mathbf{n} , et en restreignant le résultat à Γ_D , on obtient :

$$\nabla p \cdot \mathbf{n} = \left(\mathbf{f} - \rho \frac{\partial \mathbf{u}}{\partial t} - \rho \mathbf{u} \cdot \nabla \mathbf{u} + \mu \Delta \mathbf{u} \right) \cdot \mathbf{n} \quad (4.33)$$

En utilisant l'identité $\Delta \mathbf{u} = -\nabla \times \nabla \times \mathbf{u} + \nabla(\nabla \cdot \mathbf{u})$ et en soustrayant Eq. (4.32e) à Eq. (4.33), on a :

$$\mu \nabla(\nabla \cdot \mathbf{u}) \mathbf{n} = 0 \quad \text{sur } \Gamma_D \quad (4.34)$$

En multipliant la condition de Neumann des équations de Navier-Stokes en forme «primaire» Eq. (4.1d) par \mathbf{n} , on obtient :

$$p = \mu \mathbf{n} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - \mathbf{g} \cdot \mathbf{n} \quad \text{sur } \Gamma_N \quad (4.35)$$

En soustrayant Eq. (4.32f) à Eq. (4.35), on a :

$$\alpha \mu \nabla \cdot \mathbf{u} = 0 \quad \text{sur } \Gamma_N \quad (4.36)$$

TABLE 4.2 – **Algorithme de splitting.** Coefficients de discrétisation.

β_j^k	$j = 0$	$j = 1$	$j = 2$	$j = 3$
$k = 1$	1	-1	0	0
$k = 2$	3/2	-2	1/2	0
$k = 3$	11/6	-3	3/2	-1/3

γ_j^m	$j = 0$	$j = 1$	$j = 2$	$j = 3$
$k = 1$	1	0	0	0
$k = 2$	2	-1	0	0
$k = 3$	3	-3	1	0

Finalement, en prenant la divergence de Eq. (4.1a) et en soustrayant Eq. (4.32d), on obtient :

$$\mu \Delta(\nabla \cdot \mathbf{u}) = 0 \text{ dans } \Omega \quad (4.37)$$

Les équations (4.34), (4.36) et (4.37) impliquent la condition $\nabla \cdot \mathbf{u} = 0$ dans Ω .

Notations

Les schémas BDF [43] (Backward Differential Formula) utilisés dans la discrétisation des équations (4.32) sont de la forme :

$$\mathcal{D}_k \phi^{n+1} = \frac{1}{\Delta t} \left(\beta_0^k \phi^{n+1} + \sum_{j=1}^k \beta_j^k \phi^{n+1-j} \right) \quad (4.38)$$

$$\mathcal{E}_m = \sum_{j=0}^{m-1} \gamma_j^m \phi^{n-j} \quad (4.39)$$

Les coefficients β_j^k et γ_j^m sont détaillés Tab. 4.2.

Forme discrète

Si l'on considère les variables $(\mathbf{u}^j, p^j) (j \leq n)$ connues, on commence par résoudre le problème en pression à l'aide des équations suivantes :

$$\Delta p^{n+1} = \nabla \cdot \mathbf{f}^{n+1} - \rho \nabla \cdot \mathcal{E}_m(\mathbf{u} \cdot \nabla \mathbf{u})^{n+1} - \rho \mathcal{D}_k(\nabla \cdot \mathbf{u}^{n+1}) \quad \text{dans } \Omega \quad (4.40a)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \quad \text{dans } \Omega \quad (4.40b)$$

$$\nabla p^{n+1} \cdot \mathbf{n} = (\mathbf{f}^{n+1} - \rho \mathcal{D}_k \mathbf{u}^{n+1} - \rho \mathcal{E}_m(\mathbf{u} \cdot \nabla \mathbf{u})^{n+1} - \mu \nabla \times \nabla \times \mathcal{E}_m \mathbf{u}^{n+1}) \cdot \mathbf{n} \quad \text{sur } \Gamma_D \quad (4.40c)$$

$$p^{n+1} = \mu \mathbf{n}(\nabla \mathcal{E}_m \mathbf{u}^{n+1} \cdot \mathbf{n}) - \alpha \mu \nabla \cdot \mathcal{E}_m \mathbf{u}^{n+1} - \mathbf{g}^{n+1} \cdot \mathbf{n} \quad \text{sur } \Gamma_N \quad (4.40d)$$

Comme $\nabla \cdot \mathbf{u}^{n+1} = 0$, l'équation (4.40a) ne dépend pas de \mathbf{u}^{n+1} . On peut donc résoudre le système de manière découplée pour la pression p^{n+1} et la vitesse \mathbf{u}^{n+1} .

On résout désormais le problème en vitesse à l'aide des équations suivantes :

$$\rho \mathcal{D}_k \mathbf{u}^{n+1} + \rho \mathcal{E}_m(\mathbf{u} \cdot \nabla \mathbf{u})^{n+1} - \mu \Delta \mathbf{u}^{n+1} + \nabla p^{n+1} = \mathbf{f}^{n+1} \quad \text{dans } \Omega \quad (4.41a)$$

$$\mathbf{u}^{n+1} = \mathbf{u}_D^{n+1} \quad \text{sur } \Gamma_D \quad (4.41b)$$

$$\mu \nabla \mathbf{u}^{n+1} \mathbf{n} = \mathbf{g}^{n+1} + p^{n+1} \mathbf{n} \quad \text{sur } \Gamma_N \quad (4.41c)$$

Comme la pression p^{n+1} est connue, on doit résoudre un système d'équations du type «équations de la chaleur» pour chaque composante de \mathbf{u}^{n+1} .

En écrivant désormais les équations sous forme variationnelle, on a pour le problème en pression :

$$\int_{\Omega} \Delta p^{n+1} q = \int_{\Omega} (\nabla \cdot \mathbf{f}^{n+1} - \rho \nabla \cdot \mathcal{E}_m(\mathbf{u} \cdot \nabla \mathbf{u})^{n+1} - \rho \mathcal{D}_k(\nabla \cdot \mathbf{u}^{n+1})) q \quad (4.42)$$

avec les conditions limites Eq. (4.40c) et Eq. (4.40d) et la condition d'incompressibilité Eq. (4.40b).

En utilisant des intégrations par parties (Lemme 4.1), on obtient :

$$\begin{aligned} - \int_{\Omega} \nabla p^{n+1} \cdot \nabla q &= - \int_{\Omega} \rho \mathcal{D}_k(\nabla \cdot \mathbf{u}^{n+1}) + \int_{\Gamma_D} \rho q \mathcal{D}_k \mathbf{u}_D^{n+1} \cdot \mathbf{n} \\ &\quad - \int_{\Omega} (\mathbf{f}^{n+1} - \rho \mathcal{E}_m(\mathbf{u} \cdot \nabla \mathbf{u})^{n+1}) \cdot \nabla q + \int_{\Gamma_D} \mu q \nabla \times \nabla \times \mathcal{E}_m \mathbf{u}^{n+1} \cdot \mathbf{n} \end{aligned} \quad (4.43)$$

avec la condition limite Eq. (4.40d) et la condition d'incompressibilité Eq. (4.40b).

Pour éviter le calcul explicite de $\nabla \times \nabla \times \mathbf{u}^{n+1}$ sur le bord Γ_D , on se sert de l'identité suivante :

$$\int_{\Gamma_D} q \nabla \times \mathbf{u} \cdot \mathbf{n} = - \int_{\Gamma_D} \nabla \times \mathbf{u} \cdot \mathbf{n} \times \nabla q \quad (4.44)$$

On a donc :

$$\begin{aligned} - \int_{\Omega} \nabla p^{n+1} \cdot \nabla q &= - \int_{\Omega} \rho \mathcal{D}_k(\nabla \cdot \mathbf{u}^{n+1}) + \int_{\Gamma_D} \rho q \mathcal{D}_k \mathbf{u}_D^{n+1} \cdot \mathbf{n} \\ &\quad - \int_{\Omega} (\mathbf{f}^{n+1} - \rho \mathcal{E}_m(\mathbf{u} \cdot \nabla \mathbf{u})^{n+1}) \cdot \nabla q + \int_{\Gamma_D} \mu \nabla \times \mathcal{E}_m \mathbf{u}^{n+1} \cdot \mathbf{n} \times \nabla q \end{aligned} \quad (4.45)$$

avec la condition limite Eq. (4.40d) et la condition d'incompressibilité Eq. (4.40b).

Algorithme (m, k)

Dans un premier temps, on résout le problème en pression.

Trouver p_h^{n+1} :

$$\begin{aligned} - (\nabla p_h^{n+1}, \nabla q_h) &= - (\rho \mathcal{D}_k(\nabla \cdot \mathbf{u}_h^{n+1}), q_h) + (\rho \mathcal{D}_k \mathbf{u}_{D,h}^{n+1} \cdot \mathbf{n}, q_h)_{\Gamma_D} \\ &\quad - (\mathbf{f}_h^{n+1} - \rho \mathcal{E}_m(\mathbf{u} \cdot \nabla \mathbf{u})_h^{n+1}, \nabla q_h) - \mu (\nabla \times \mathcal{E}_m \mathbf{u}_h^{n+1}, \mathbf{n} \times \nabla q_h)_{\Gamma_D} \end{aligned} \quad (4.46)$$

avec $p_h^{n+1} = \mu \mathbf{n} \cdot (\nabla \mathcal{E}_m \mathbf{u}_h^{n+1} \mathbf{n}) - \alpha \mu \nabla \cdot \mathcal{E}_m \mathbf{u}_h^{n+1} - \mathbf{n} \cdot \mathbf{g}_h^{n+1}$ sur Γ_N et $\nabla \cdot \mathbf{u}_h^{n+1} = 0$ dans Ω .

On résout ensuite le problème en vitesse.

Trouver \mathbf{u}_h^{n+1} :

$$\begin{aligned} (\rho \mathcal{D}_k \mathbf{u}_h^{n+1}, \mathbf{v}_h) + (\rho \mathcal{E}_m(\mathbf{u} \cdot \nabla \mathbf{u})_h^{n+1}, \mathbf{v}_h) + \mu (\nabla \mathbf{u}_h^{n+1}, \nabla \mathbf{v}_h) - (p_h^{n+1}, \nabla \cdot \mathbf{v}_h) \\ = (\mathbf{f}_h^{n+1}, \mathbf{v}_h) + (\mathbf{g}_h^{n+1}, \mathbf{v}_h)_{\Gamma_N} \end{aligned} \quad (4.47)$$

avec $\mathbf{u}_h^{n+1} = \mathbf{u}_{D,h}^{n+1}$ sur Γ_D .

Exemple de l'algorithme (2, 2)

Trouver p_h^{n+1} :

$$-(\nabla p_h^{n+1}, \nabla q_h) = \frac{\rho}{\Delta t} \left(2\nabla \cdot \mathbf{u}^n - \frac{1}{2}\nabla \cdot \mathbf{u}^{n+1}, q_h \right) \quad (4.48)$$

$$+ \frac{\rho}{\Delta t} \left(\left(\frac{3}{2}\mathbf{u}_{D,h}^{n+1} - 2\mathbf{u}_{D,h}^n + \frac{1}{2}\mathbf{u}_{D,h}^{n+1} \right) \cdot \mathbf{n}, q_h \right)_{\Gamma_D} \quad (4.49)$$

$$- \left(\mathbf{f}_h^{n+1} - \rho \left(2(\mathbf{u} \cdot \nabla \mathbf{u})_h^n - (\mathbf{u} \cdot \nabla \mathbf{u})_h^{n-1} \right), \nabla q_h \right) \quad (4.50)$$

$$- \mu \left(\nabla \times \mathcal{E}_m \mathbf{u}_h^{n+1}, \mathbf{n} \times \nabla q_h \right)_{\Gamma_D} \quad (4.51)$$

avec $p_h^{n+1} = \mu \mathbf{n} \cdot (\nabla (2\mathbf{u}_h^n - \mathbf{u}_h^{n-1}) \mathbf{n}) - \alpha \mu \nabla \cdot (2\mathbf{u}_h^n - \mathbf{u}_h^{n-1}) - \mathbf{n} \cdot \mathbf{g}_h^{n+1}$ sur Γ_N et $\nabla \cdot \mathbf{u}_h^{n+1} = 0$ dans Ω .

Trouver \mathbf{u}_h^{n+1} :

$$\begin{aligned} \frac{\rho}{\Delta t} \left(\frac{3}{2}\mathbf{u}_h^{n+1} - 2\mathbf{u}_h^n + \frac{1}{2}\mathbf{u}_h^{n-1}, \mathbf{v}_h \right) + \rho \left(2(\mathbf{u} \cdot \nabla \mathbf{u})_h^n - (\mathbf{u} \cdot \nabla \mathbf{u})_h^{n-1}, \mathbf{v}_h \right) + \mu (\nabla \mathbf{u}_h^{n+1}, \nabla \mathbf{v}_h) \\ - (p_h^{n+1}, \nabla \cdot \mathbf{v}_h) = (\mathbf{f}_h^{n+1}, \mathbf{v}_h) + (\mathbf{g}_h^{n+1}, \mathbf{v}_h)_{\Gamma_N} \end{aligned} \quad (4.52)$$

avec $\mathbf{u}_h^{n+1} = \mathbf{u}_{D,h}^{n+1}$ sur Γ_D .

Remarque 4.7. *La résolution du problème étant découplée, le respect de la condition Inf-Sup ne s'applique plus ici, contrairement à la méthode des caractéristiques (et à l'algorithme d'Uzawa). Le choix des espaces éléments finis est donc moins contraint.*

L'algorithme FREEFEM++ et les détails de programmation sont donnés dans l'Annexe C.4.

4.2 Modèles de Windkessel

Les modèles de Windkessel, ou modèles 0D, introduits par O. Frank [29], sont fréquemment utilisés dans la simulation d'écoulements sanguins [80]. Ils permettent de prendre en compte les parties non représentées dans le modèle (dans le maillage) lors de la simulation numérique.

Ils se présentent sous la forme de modèles électriques que l'on vient imposer en condition limite non locale du problème de Stokes ou de Navier-Stokes. L'analogie électrique / mécanique utilisée est décrite Tab. 4.3.

TABLE 4.3 – **Modèles de Windkessel.** Analogie Électrique / Mécanique

Électrique		Mécanique	
Courant	i	Flux	Q
Tension	U	Pression	p
Résistance	R	Résistance	R
Capacité	C	Compliance	C

Imposer un modèle de Windkessel en condition limite revient donc à imposer une pression ; c'est pourquoi on utilise la condition de Neumann pour le mettre en place.

Les modèles de Windkessel couramment utilisés sont les modèles à 2 ou 3 éléments, Fig. 4.1. Dans le cas des écoulements sanguins, ils permettent de représenter la partie du réseau en dehors



(a) Modèles de Windkessel à 2 éléments.

(b) Modèle de Windkessel à 3 éléments.

FIGURE 4.1 – Modèles de Windkessel standards. Modèles à 2 et 3 éléments.

de la zone de calcul. Ils représentent la résistance à l'écoulement du fluide ainsi que l'élasticité dans les artères / veines grâce, respectivement, à la résistance et à la capacité.

Dans notre cas, on souhaite non pas représenter un réseau artériel ou veineux mais uniquement un compartiment déformable ; on va donc utiliser une capacité uniquement. Comme nos compartiments sont en interaction entre eux et avec l'extérieur (compartiment sanguin), on utilise plus précisément une capacité variable, Fig. 4.2.



(a) Capacité constante.

(b) Capacité variable.

FIGURE 4.2 – Modèles de Windkessel pour la modélisation du système cérébrospinal. Modèle constant et modèle variable.

Cela revient à écrire une condition de pression, que l'on imposera via une condition de Neumann, du type :

$$P = \frac{1}{C(t)} \int_0^T Q(s) ds \tag{4.53}$$

Avec $C(t)$ la capacité variable, $Q(t)$ le débit à l'instant t et T le temps total.

Sous forme discrète, en utilisant un couplage explicite, on a :

$$P^{k+1} = \frac{1}{C^k} \sum_{i=0}^k Q_i \tag{4.54}$$

Avec C^k la capacité au temps t^k et Q_k le débit au temps t^k .

Remarque 4.8. Ici, le modèle de Windkessel représente un compartiment clos (montage en «série»), contrairement aux modèles appliqués aux réseaux artériels ou veineux (montage en «parallèle»).

4.3 Stabilité

Les techniques utilisés dans l'étude de stabilité des équations de Stokes couplées à des modèles 0D sont détaillées dans la littérature récente [27, 28].

4.3.1 Étude théorique – Conditions de Neumann

Dans un premier temps, et dans un souci de simplicité, on s'intéresse au problème de Stokes instationnaire semi-discrétisé en temps sur un domaine Ω , en 2 dimensions, comprenant une entrée Γ_{in} , sur laquelle une condition de Neumann p_{in} est imposée, et deux sorties Γ_{out_1} et Γ_{out_2} , sur lesquelles des conditions de Neumann p_{out_1} et p_{out_2} sont imposées, dans un intervalle de temps $[0, T]$:

$$\rho \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \nabla p^{n+1} - \mu \Delta \mathbf{u}^{n+1} = \mathbf{0} \quad \text{sur } [0, T] \times \Omega \quad (4.55a)$$

$$\operatorname{div}(\mathbf{u}^{n+1}) = 0 \quad \text{sur } [0, T] \times \Omega \quad (4.55b)$$

$$\mathbf{u}^{n+1} = \mathbf{0} \quad \text{sur } [0, T] \times \Gamma_{wall} \quad (4.55c)$$

$$\mu \frac{\partial \mathbf{u}^{n+1}}{\partial \mathbf{n}} - p^{n+1} \mathbf{n} = -p_{in}^{n+1} \mathbf{n} \quad \text{sur } [0, T] \times \Gamma_{in} \quad (4.55d)$$

$$\mu \frac{\partial \mathbf{u}^{n+1}}{\partial \mathbf{n}} - p^{n+1} \mathbf{n} = -p_{out_1}^{n+1} \mathbf{n} \quad \text{sur } [0, T] \times \Gamma_{out_1} \quad (4.55e)$$

$$\mu \frac{\partial \mathbf{u}^{n+1}}{\partial \mathbf{n}} - p^{n+1} \mathbf{n} = -p_{out_2}^{n+1} \mathbf{n} \quad \text{sur } [0, T] \times \Gamma_{out_2} \quad (4.55f)$$

Des modèles 0D explicites sont utilisés, ils s'écrivent :

$$p_{out_1}^{n+1} = \frac{\Delta t}{C_1^n} \sum_{k=0}^n \int_{\Gamma_{out_1}} \mathbf{u}^k \cdot \mathbf{n} \quad (4.56a)$$

$$p_{out_2}^{n+1} = \frac{\Delta t}{C_2^n} \sum_{k=0}^n \int_{\Gamma_{out_2}} \mathbf{u}^k \cdot \mathbf{n} \quad (4.56b)$$

Théorème 4.1. *Ce schéma est conditionnellement stable sous la condition Courant-Friedrichs-Lewy (CFL) suivante :*

$$\Delta t < \frac{\tilde{c}\rho}{T} \left(\frac{C_{1,min} C_{2,min}}{C_{1,min} + C_{2,min}} \right) \quad (4.57)$$

Démonstration. La formulation variationnelle du problème est donnée par :

$$\frac{\rho}{\Delta t} \int_{\Omega} \mathbf{u}^{n+1} \cdot \mathbf{v} - \frac{\rho}{\Delta t} \int_{\Omega} \mathbf{u}^n \cdot \mathbf{v} + \int_{\Omega} \nabla p^{n+1} \cdot \mathbf{v} - \mu \int_{\Omega} \Delta \mathbf{u}^{n+1} \cdot \mathbf{v} = 0 \quad (4.58)$$

En utilisant les formules de Green (Lemme 4.1), on obtient :

$$\frac{\rho}{\Delta t} \int_{\Omega} \mathbf{u}^{n+1} \cdot \mathbf{v} - \frac{\rho}{\Delta t} \int_{\Omega} \mathbf{u}^n \cdot \mathbf{v} + \mu \int_{\Omega} \nabla \mathbf{u}^{n+1} : \nabla \mathbf{v} + \int_{\partial\Omega} (p^{n+1} \mathbf{n} - \mu \nabla \mathbf{u}^{n+1} \cdot \mathbf{n}) \cdot \mathbf{v} = 0 \quad (4.59)$$

En utilisant les conditions aux bords définies dans le problème, on obtient :

$$\begin{aligned} \frac{\rho}{\Delta t} \int_{\Omega} \mathbf{u}^{n+1} \cdot \mathbf{v} - \frac{\rho}{\Delta t} \int_{\Omega} \mathbf{u}^n \cdot \mathbf{v} + \mu \int_{\Omega} \nabla \mathbf{u}^{n+1} : \nabla \mathbf{v} \\ + \int_{\Gamma_{in}} p_{in}^{n+1} \mathbf{n} \cdot \mathbf{v} + \int_{\Gamma_{out_1}} p_{out_1}^{n+1} \mathbf{n} \cdot \mathbf{v} + \int_{\Gamma_{out_2}} p_{out_2}^{n+1} \mathbf{n} \cdot \mathbf{v} = 0 \end{aligned} \quad (4.60)$$

Pour étudier la stabilité, on prend comme fonction test $\mathbf{v} = \mathbf{u}^{n+1}$ afin d'écrire le bilan d'énergie :

$$\begin{aligned} \frac{\rho}{\Delta t} \int_{\Omega} \mathbf{u}^{n+1} \cdot \mathbf{u}^{n+1} - \frac{\rho}{\Delta t} \int_{\Omega} \mathbf{u}^n \cdot \mathbf{u}^{n+1} + \mu \int_{\Omega} \nabla \mathbf{u}^{n+1} : \nabla \mathbf{u}^{n+1} \\ + \int_{\Gamma_{in}} p_{in}^{n+1} \mathbf{n} \cdot \mathbf{u}^{n+1} + \int_{\Gamma_{out_1}} p_{out_1}^{n+1} \mathbf{n} \cdot \mathbf{u}^{n+1} + \int_{\Gamma_{out_2}} p_{out_2}^{n+1} \mathbf{n} \cdot \mathbf{u}^{n+1} = 0 \end{aligned} \quad (4.61)$$

En utilisant la relation $(\mathbf{u}^{n+1} - \mathbf{u}^n) \mathbf{u}^{n+1} = \frac{1}{2} (|\mathbf{u}^{n+1}|^2 - |\mathbf{u}^n|^2 + |\mathbf{u}^{n+1} - \mathbf{u}^n|^2)$, on a :

$$\begin{aligned} \frac{\rho}{2\Delta t} \int_{\Omega} |\mathbf{u}^{n+1}|^2 - \frac{\rho}{2\Delta t} \int_{\Omega} |\mathbf{u}^n|^2 + \frac{\rho}{2\Delta t} \int_{\Omega} |\mathbf{u}^{n+1} - \mathbf{u}^n|^2 + \mu \int_{\Omega} |\nabla \mathbf{u}^{n+1}|^2 \\ + \int_{\Gamma_{in}} p_{in}^{n+1} \mathbf{n} \cdot \mathbf{u}^{n+1} + \int_{\Gamma_{out_1}} p_{out_1}^{n+1} \mathbf{n} \cdot \mathbf{u}^{n+1} + \int_{\Gamma_{out_2}} p_{out_2}^{n+1} \mathbf{n} \cdot \mathbf{u}^{n+1} = 0 \end{aligned} \quad (4.62)$$

En utilisant la définition des modèles 0D et en multipliant l'équation précédente par Δt , on obtient :

$$\begin{aligned} \frac{\rho}{2} \|\mathbf{u}^{n+1}\|_{L^2(\Omega)}^2 - \frac{\rho}{2} \|\mathbf{u}^n\|_{L^2(\Omega)}^2 + \frac{\rho}{2} \|\mathbf{u}^{n+1} - \mathbf{u}^n\|_{L^2(\Omega)}^2 + \Delta t \mu \|\nabla \mathbf{u}^{n+1}\|_{L^2(\Omega)}^2 \\ + \int_{\Gamma_{in}} p_{in}^{n+1} \mathbf{n} \cdot \mathbf{u}^{n+1} \\ + \frac{\Delta t^2}{C_1^n} \left(\int_{\Gamma_{out_1}} \mathbf{u}^{n+1} \cdot \mathbf{n} \right) \left(\sum_{k=0}^n \int_{\Gamma_{out_1}} \mathbf{u}^k \cdot \mathbf{n} \right) \\ + \frac{\Delta t^2}{C_2^n} \left(\int_{\Gamma_{out_2}} \mathbf{u}^{n+1} \cdot \mathbf{n} \right) \left(\sum_{k=0}^n \int_{\Gamma_{out_2}} \mathbf{u}^k \cdot \mathbf{n} \right) \\ = 0 \end{aligned} \quad (4.63)$$

En utilisant une inégalité de trace de $H^1(\Omega)$ dans $H^{1/2}(\Gamma_{out_i})$, de Cauchy-Schwarz et de Young, pour $i \in \{1, 2\}$, on a :

$$\begin{aligned} \left| \frac{\Delta t^2}{C_i^n} \left(\int_{\Gamma_{out_i}} \mathbf{u}^{n+1} \cdot \mathbf{n} \right) \left(\sum_{k=0}^n \int_{\Gamma_{out_i}} \mathbf{u}^k \cdot \mathbf{n} \right) \right| \leq \tilde{c} \frac{\Delta t}{C_i^n} \sum_{k=0}^n \left(\Delta t \|\mathbf{u}^{n+1}\|_{L^2(\Omega)}^2 \|\mathbf{u}^k\|_{L^2(\Omega)}^2 \right) \\ \leq \tilde{c} \frac{T \Delta t}{2C_i^n} \|\mathbf{u}^{n+1}\|_{L^2(\Omega)}^2 + \tilde{c} \frac{\Delta t^2}{2C_i^n} \sum_{k=0}^n \|\mathbf{u}^k\|_{L^2(\Omega)}^2 \end{aligned} \quad (4.64)$$

En utilisant une inégalité de trace de $H^1(\Omega)$ dans $H^{1/2}(\Gamma_{in})$, on a :

$$\left| \int_{\Gamma_{in}} p_{in}^{n+1} \mathbf{n} \cdot \mathbf{u}^{n+1} \right| \leq \tilde{c} \|p_{in}^{n+1}\|_{L^2(\Gamma_{in})} \|\mathbf{u} \cdot \mathbf{n}\|_{L^2(\Gamma_{in})} \leq \tilde{c} \|p_{in}^{n+1}\|_{L^2(\Gamma_{in})} \|\nabla \mathbf{u}\|_{L^2(\Omega)}^2 \quad (4.65)$$

En utilisant l'inégalité de Young, on aboutit à :

$$\left| \int_{\Gamma_{in}} p_{in}^{n+1} \mathbf{n} \cdot \mathbf{u}^{n+1} \right| \leq \frac{\mu}{2} \|\nabla \mathbf{u}^{n+1}\|_{L^2(\Omega)}^2 + \frac{\tilde{c}}{2\mu} \|p_{in}\|_{L^2(\Gamma_{in})}^2 \quad (4.66)$$

En utilisant les inégalités précédentes, on obtient :

$$\begin{aligned} \frac{\rho}{2} \|\mathbf{u}^{n+1}\|_{L^2(\Omega)}^2 + \frac{\Delta t \mu}{2} \|\nabla \mathbf{u}^{n+1}\|_{L^2(\Omega)}^2 \leq \frac{\rho}{2} \|\mathbf{u}^n\|_{L^2(\Omega)}^2 + \frac{\tilde{c} \Delta t}{2\mu} \|p_{in}\|_{L^2(\Gamma_{in})}^2 \\ + \tilde{c} \frac{T \Delta t}{2C_1^n} \|\mathbf{u}^{n+1}\|_{L^2(\Omega)}^2 + \tilde{c} \frac{\Delta t^2}{2C_1^n} \sum_{k=0}^n \|\mathbf{u}^k\|_{L^2(\Omega)}^2 + \tilde{c} \frac{T \Delta t}{2C_2^n} \|\mathbf{u}^{n+1}\|_{L^2(\Omega)}^2 + \tilde{c} \frac{\Delta t^2}{2C_2^n} \sum_{k=0}^n \|\mathbf{u}^k\|_{L^2(\Omega)}^2 \end{aligned} \quad (4.67)$$

$$\begin{aligned} \frac{\rho}{2} \|\mathbf{u}^{n+1}\|_{L^2(\Omega)}^2 + \frac{\Delta t \mu}{2} \|\nabla \mathbf{u}^{n+1}\|_{L^2(\Omega)}^2 &\leq \frac{\rho}{2} \|\mathbf{u}^n\|_{L^2(\Omega)}^2 + \frac{\tilde{c} \Delta t}{2\mu} \|p_{in}\|_{L^2(\Gamma_{in})}^2 \\ &+ \tilde{c} \frac{T \Delta t}{2} \left(\frac{1}{C_1^n} + \frac{1}{C_2^n} \right) \|\mathbf{u}^{n+1}\|_{L^2(\Omega)}^2 + \tilde{c} \frac{\Delta t^2}{2} \left(\frac{1}{C_1^n} + \frac{1}{C_2^n} \right) \sum_{k=0}^n \|\mathbf{u}^k\|_{L^2(\Omega)}^2 \end{aligned} \quad (4.68)$$

Par sommation sur $\{0, \dots, N-1\}$, on a :

$$\begin{aligned} \frac{\rho}{2} \|\mathbf{u}^N\|_{L^2(\Omega)}^2 + \frac{\mu}{2} \sum_{k=1}^N \Delta t \|\nabla \mathbf{u}^k\|_{L^2(\Omega)}^2 &\leq \frac{\rho}{2} \|\mathbf{u}^0\|_{L^2(\Omega)}^2 + \frac{\tilde{c}}{2\mu} \sum_{k=1}^N \Delta t \|p_{in}\|_{L^2(\Gamma_{in})}^2 \\ &+ \tilde{c} \frac{T}{2} \sum_{k=1}^N \left(\frac{1}{C_1^k} + \frac{1}{C_2^k} \right) \Delta t \|\mathbf{u}^k\|_{L^2(\Omega)}^2 + \tilde{c} \sum_{k=0}^{N-1} \frac{\Delta t^2}{2} \left(\frac{1}{C_1^k} + \frac{1}{C_2^k} \right) \sum_{\ell=0}^k \|\mathbf{u}^\ell\|_{L^2(\Omega)}^2 \end{aligned} \quad (4.69)$$

En considérant que les capacités atteignent un minimum global, on a :

$$\tilde{c} \frac{T}{2} \sum_{k=1}^N \left(\frac{1}{C_1^k} + \frac{1}{C_2^k} \right) \Delta t \|\mathbf{u}^k\|_{L^2(\Omega)}^2 \leq \tilde{c} \frac{T}{2} \left(\frac{1}{C_{1,min}} + \frac{1}{C_{2,min}} \right) \sum_{k=1}^N \Delta t \|\mathbf{u}^k\|_{L^2(\Omega)}^2 \quad (4.70)$$

En développant les sommes, on a :

$$\begin{aligned} \sum_{k=0}^{N-1} \frac{\Delta t^2}{2C_i^k} \sum_{\ell=0}^k \|\mathbf{u}^\ell\|_{L^2(\Omega)}^2 &\leq N \frac{\Delta t^2}{2C_{i,min}} \sum_{k=0}^{N-1} \|\mathbf{u}^k\|_{L^2(\Omega)}^2 \\ &\leq \frac{T}{2C_{i,min}} \sum_{k=0}^{N-1} \Delta t \|\mathbf{u}^k\|_{L^2(\Omega)}^2, \quad i \in \{1, 2\} \end{aligned} \quad (4.71)$$

En utilisant l'inégalité précédente, on obtient :

$$\begin{aligned} \frac{\rho}{2} \left(1 - \tilde{c} \Delta t \frac{T}{\rho} \left(\frac{1}{C_{1,min}} + \frac{1}{C_{2,min}} \right) \right) \|\mathbf{u}^N\|_{L^2(\Omega)}^2 &+ \frac{\mu}{2} \sum_{k=1}^N \Delta t \|\nabla \mathbf{u}^k\|_{L^2(\Omega)}^2 \\ &\leq \frac{\rho}{2} \|\mathbf{u}^0\|_{L^2(\Omega)}^2 + \frac{\tilde{c}}{2\mu} \sum_{k=1}^N \Delta t \|p_{in}\|_{L^2(\Gamma_{in})}^2 + \tilde{c} T \left(\frac{1}{C_{1,min}} + \frac{1}{C_{2,min}} \right) \sum_{k=0}^{N-1} \Delta t \|\mathbf{u}^k\|_{L^2(\Omega)}^2 \end{aligned} \quad (4.72)$$

Finalement, le lemme de Gronwall discret nous donne la condition CFL :

$$\Delta t < \frac{\tilde{c} \rho}{T} \left(\frac{C_{1,min} C_{2,min}}{C_{1,min} + C_{2,min}} \right) \quad (4.73)$$

□

Remarque 4.9. La condition CFL trouvée ici est similaire à celle obtenue dans la littérature [27, 28] pour un modèle RC série, en posant $R = 0$.

Remarque 4.10. La condition CFL que l'on obtient dépend de l'inverse du temps, ce qui n'est pas courant. De plus elle est très contraignante puisqu'une augmentation du temps total de la simulation impose une diminution du pas de temps qui peut amener à des temps de calculs très élevés.

Remarque 4.11. *L'hypothèse des capacités atteignant un minimum global est justifiée dans notre étude, puisque le flux de LCS est périodique et à moyenne nulle. Le minimum des capacités (ou des compliances) sera atteint à chaque cycle cardiaque lors de l'arrivée de LCS vers le compartiment cérébral.*

Comme nous utiliserons un couplage explicite des modèles de Windkessel, seule la stabilité pour ce couplage est exposée ici. Néanmoins, nous avons vérifié q'un couplage implicite nous permet d'avoir un schéma inconditionnellement stable, comme présenté dans la littérature [27, 28].

4.3.2 Étude théorique – Conditions mêlées

On s'intéresse maintenant au problème de Stokes instationnaire sur un domaine Ω comprenant une entrée Γ_{in} , sur laquelle une condition de Dirichlet \mathbf{u}_{in} est imposée, et deux sorties Γ_{out_1} et Γ_{out_2} , sur lesquelles des conditions de Neumann p_{out_1} et p_{out_2} sont imposées, dans un intervalle de temps $[0, T]$:

$$\rho \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \nabla p^{n+1} - \mu \Delta \mathbf{u}^{n+1} = \mathbf{0} \quad \text{dans } [0, T] \times \Omega \quad (4.74a)$$

$$\operatorname{div}(\mathbf{u}^{n+1}) = 0 \quad \text{dans } [0, T] \times \Omega \quad (4.74b)$$

$$\mathbf{u}^{n+1} = \mathbf{0} \quad \text{sur } [0, T] \times \Gamma_{wall} \quad (4.74c)$$

$$\mathbf{u}^{n+1} = \mathbf{u}_{in} \quad \text{sur } [0, T] \times \Gamma_{in} \quad (4.74d)$$

$$\mu \frac{\partial \mathbf{u}^{n+1}}{\partial \mathbf{n}} - p^{n+1} \mathbf{n} = -p_{out_1}^{n+1} \mathbf{n} \quad \text{sur } [0, T] \times \Gamma_{out_1} \quad (4.74e)$$

$$\mu \frac{\partial \mathbf{u}^{n+1}}{\partial \mathbf{n}} - p^{n+1} \mathbf{n} = -p_{out_2}^{n+1} \mathbf{n} \quad \text{sur } [0, T] \times \Gamma_{out_2} \quad (4.74f)$$

Les mêmes modèles 0D que pour la section précédente sont utilisés.

On souhaite démontrer une condition CFL du type de celle que nous avons déjà montrée dans la sous-section 4.3.1.

Néanmoins, la stabilité est bien plus complexe à démontrer dans ce cas et la tentative présentée par la suite n'est pas concluante.

Début de preuve. *On cherche ici à homogénéiser le problème, afin de mettre la condition de Dirichlet en entrée à 0, pour procéder de la même façon que pour la partie précédente.*

Soit φ solution du problème :

$$\Delta \varphi = 0 \quad \text{dans } \Omega \quad (4.75a)$$

$$\varphi = \mathbf{u}_{in} \quad \text{sur } \Gamma_{in} \quad (4.75b)$$

$$\varphi = \mathbf{0} \quad \text{sur } \Gamma_{wall} \quad (4.75c)$$

$$\frac{\partial \varphi}{\partial \mathbf{n}} = 0 \quad \text{sur } \Gamma_{out_1} \cup \Gamma_{out_2} \quad (4.75d)$$

Remarque 4.12. *Ce type de problème avec conditions aux limites mêlées n'admet pas, à priori, de solution dans $H^2(\Omega)$ mais dans $H^{1+\varepsilon}(\Omega)$, avec ε dépendant de la géométrie ($\varepsilon = 1$ pour un domaine rectangulaire).*

Remarque 4.13. *En considérant que la vitesse d'entrée \mathbf{u}_{in} est un profil de Poiseuille (nulle sur les bords), une solution à ce problème existe et est unique [41].*

On prend $\tilde{\mathbf{u}} = \mathbf{u} - \boldsymbol{\varphi} \Leftrightarrow \mathbf{u} = \tilde{\mathbf{u}} + \boldsymbol{\varphi}$; on réduit notre problème a :

$$\rho \frac{(\tilde{\mathbf{u}}^{n+1} + \boldsymbol{\varphi}^{n+1}) - (\tilde{\mathbf{u}}^n + \boldsymbol{\varphi}^n)}{\Delta t} + \nabla p^{n+1} - \mu \Delta (\tilde{\mathbf{u}}^{n+1} + \boldsymbol{\varphi}^{n+1}) = \mathbf{0} \quad \text{dans } [0, T] \times \Omega \quad (4.76a)$$

$$\operatorname{div}(\tilde{\mathbf{u}}^{n+1} + \boldsymbol{\varphi}^{n+1}) = 0 \quad \text{dans } [0, T] \times \Omega \quad (4.76b)$$

$$\tilde{\mathbf{u}}^{n+1} + \boldsymbol{\varphi}^{n+1} = \mathbf{0} \quad \text{sur } [0, T] \times \Gamma_{wall} \quad (4.76c)$$

$$\tilde{\mathbf{u}}^{n+1} + \boldsymbol{\varphi}^{n+1} = \mathbf{u}_{in} \quad \text{sur } [0, T] \times \Gamma_{in} \quad (4.76d)$$

$$\mu \frac{\partial(\tilde{\mathbf{u}}^{n+1} + \boldsymbol{\varphi}^{n+1})}{\partial \mathbf{n}} - p^{n+1} \mathbf{n} = -p_{out_1}^{n+1} \mathbf{n} \quad \text{sur } [0, T] \times \Gamma_{out_1} \quad (4.76e)$$

$$\mu \frac{\partial(\tilde{\mathbf{u}}^{n+1} + \boldsymbol{\varphi}^{n+1})}{\partial \mathbf{n}} - p^{n+1} \mathbf{n} = -p_{out_2}^{n+1} \mathbf{n} \quad \text{sur } [0, T] \times \Gamma_{out_2} \quad (4.76f)$$

En prenant en compte les propriétés de la solution $\boldsymbol{\varphi}$, on obtient :

$$\rho \frac{(\tilde{\mathbf{u}}^{n+1} + \boldsymbol{\varphi}^{n+1}) - (\tilde{\mathbf{u}}^n + \boldsymbol{\varphi}^n)}{\Delta t} + \nabla p^{n+1} - \mu \Delta \tilde{\mathbf{u}}^{n+1} = \mathbf{0} \quad \text{dans } [0, T] \times \Omega \quad (4.77a)$$

$$\operatorname{div}(\tilde{\mathbf{u}}^{n+1} + \boldsymbol{\varphi}^{n+1}) = 0 \quad \text{dans } [0, T] \times \Omega \quad (4.77b)$$

$$\tilde{\mathbf{u}}^{n+1} = \mathbf{0} \quad \text{sur } [0, T] \times \Gamma_{wall} \quad (4.77c)$$

$$\tilde{\mathbf{u}}^{n+1} = \mathbf{0} \quad \text{sur } [0, T] \times \Gamma_{in} \quad (4.77d)$$

$$\mu \frac{\partial \tilde{\mathbf{u}}^{n+1}}{\partial \mathbf{n}} - p^{n+1} \mathbf{n} = -p_{out_1}^{n+1} \mathbf{n} \quad \text{sur } [0, T] \times \Gamma_{out_1} \quad (4.77e)$$

$$\mu \frac{\partial \tilde{\mathbf{u}}^{n+1}}{\partial \mathbf{n}} - p^{n+1} \mathbf{n} = -p_{out_2}^{n+1} \mathbf{n} \quad \text{sur } [0, T] \times \Gamma_{out_2} \quad (4.77f)$$

Considérons, pour simplifier le problème, \mathbf{u}_{in} constant (en temps), on a :

$$\rho \frac{\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}^n}{\Delta t} + \nabla p^{n+1} - \mu \Delta \tilde{\mathbf{u}}^{n+1} = \mathbf{0} \quad \text{dans } [0, T] \times \Omega \quad (4.78a)$$

$$\operatorname{div}(\tilde{\mathbf{u}}^{n+1} + \boldsymbol{\varphi}^{n+1}) = 0 \quad \text{dans } [0, T] \times \Omega \quad (4.78b)$$

$$\tilde{\mathbf{u}}^{n+1} = \mathbf{0} \quad \text{sur } [0, T] \times \Gamma_{wall} \quad (4.78c)$$

$$\tilde{\mathbf{u}}^{n+1} = \mathbf{0} \quad \text{sur } [0, T] \times \Gamma_{in} \quad (4.78d)$$

$$\mu \frac{\partial \tilde{\mathbf{u}}^{n+1}}{\partial \mathbf{n}} - p^{n+1} \mathbf{n} = -p_{out_1}^{n+1} \mathbf{n} \quad \text{sur } [0, T] \times \Gamma_{out_1} \quad (4.78e)$$

$$\mu \frac{\partial \tilde{\mathbf{u}}^{n+1}}{\partial \mathbf{n}} - p^{n+1} \mathbf{n} = -p_{out_2}^{n+1} \mathbf{n} \quad \text{sur } [0, T] \times \Gamma_{out_2} \quad (4.78f)$$

On a donc la formulation variationnelle :

$$\frac{\rho}{\Delta t} \int_{\Omega} \tilde{\mathbf{u}}^{n+1} \cdot \mathbf{v} - \frac{\rho}{\Delta t} \int_{\Omega} \tilde{\mathbf{u}}^n \cdot \mathbf{v} + \int_{\Omega} \nabla p^{n+1} \cdot \mathbf{v} - \mu \int_{\Omega} \Delta \tilde{\mathbf{u}}^{n+1} \cdot \mathbf{v} = 0 \quad (4.79)$$

En utilisant les formules de Green (Lemme 4.1), on obtient :

$$\begin{aligned} \frac{\rho}{\Delta t} \int_{\Omega} \tilde{\mathbf{u}}^{n+1} \cdot \mathbf{v} - \frac{\rho}{\Delta t} \int_{\Omega} \tilde{\mathbf{u}}^n \cdot \mathbf{v} - \int_{\Omega} p^{n+1} \operatorname{div}(\mathbf{v}) + \mu \int_{\Omega} \nabla \tilde{\mathbf{u}}^{n+1} : \nabla \mathbf{v} \\ + \int_{\Gamma_{out_1} \cup \Gamma_{out_2}} (p^{n+1} \mathbf{n} - \mu \nabla \tilde{\mathbf{u}}^{n+1} \cdot \mathbf{n}) \cdot \mathbf{v} = 0 \end{aligned} \quad (4.80)$$

Comme pour l'étude précédente, on prend comme fonction test $\tilde{\mathbf{u}}^{n+1}$:

$$\begin{aligned} \frac{\rho}{\Delta t} \int_{\Omega} \tilde{\mathbf{u}}^{n+1} \cdot \tilde{\mathbf{u}}^{n+1} - \frac{\rho}{\Delta t} \int_{\Omega} \tilde{\mathbf{u}}^n \cdot \tilde{\mathbf{u}}^{n+1} - \int_{\Omega} p^{n+1} \operatorname{div}(\tilde{\mathbf{u}}^{n+1}) + \mu \int_{\Omega} \nabla \tilde{\mathbf{u}}^{n+1} : \nabla \tilde{\mathbf{u}}^{n+1} \\ + \int_{\Gamma_{out_1} \cup \Gamma_{out_2}} (p^{n+1} \mathbf{n} - \mu \nabla \tilde{\mathbf{u}}^{n+1} \cdot \mathbf{n}) \cdot \tilde{\mathbf{u}}^{n+1} = 0 \end{aligned} \quad (4.81)$$

En utilisant les propriétés du problème, on a :

$$\begin{aligned} \frac{\rho}{\Delta t} \int_{\Omega} \tilde{\mathbf{u}}^{n+1} \cdot \tilde{\mathbf{u}}^{n+1} - \frac{\rho}{\Delta t} \int_{\Omega} \tilde{\mathbf{u}}^n \cdot \tilde{\mathbf{u}}^{n+1} + \int_{\Omega} p^{n+1} \operatorname{div}(\varphi^{n+1}) + \mu \int_{\Omega} \nabla \tilde{\mathbf{u}}^{n+1} : \nabla \tilde{\mathbf{u}}^{n+1} \\ + \int_{\Gamma_{out_1}} p_{out_1}^{n+1} \mathbf{n} \cdot \mathbf{u}^{n+1} + \int_{\Gamma_{out_2}} p_{out_2}^{n+1} \mathbf{n} \cdot \mathbf{u}^{n+1} = 0 \end{aligned} \quad (4.82)$$

Afin de revenir à l'étude de stabilité du problème avec conditions de Neumann uniquement, il convient de majorer le terme :

$$\int_{\Omega} p^{n+1} \operatorname{div}(\varphi^{n+1}) \quad (4.83)$$

Remarque 4.14. On a, en 2 dimensions, à l'aide de l'inégalité de Young :

$$\int_{\Omega} p^{n+1} \operatorname{div}(\varphi^{n+1}) \leq \frac{1}{2} \|p^{n+1}\|_{L^2(\Omega)}^2 + \frac{1}{2} \|\varphi^{n+1}\|_{L^2(\Omega)}^2 \leq \frac{1}{2} \|p^{n+1}\|_{L^2(\Omega)}^2 + \|\mathbf{u}_{in}\|_{L^2(\Omega)}^2 \quad (4.84)$$

Il devient alors nécessaire de contrôler la pression afin d'obtenir un résultat similaire au cas avec condition de Neumann.

D'un point de vue physique, comme notre compartiment est clos, si la vitesse d'entrée \mathbf{u}_{in} est constante, le volume contenu dans le compartiment va augmenter linéairement et la pression va être modifiée en conséquence. Il nous sera donc impossible de majorer la pression, et le schéma sera donc instable.

Il faut donc utiliser une vitesse d'entrée à moyenne nulle en temps, comme un sinus, afin de ne pas observer ce genre de phénomène. Un terme supplémentaire vient donc s'ajouter :

$$\frac{\rho}{\Delta t} \int_{\Omega} (\varphi^{n+1} - \varphi^n) \tilde{\mathbf{u}}^{n+1} \quad (4.85)$$

La stabilité est donc bien plus complexe à démontrer dans ce cas (faisant probablement intervenir la théorie des solutions périodiques des équations de Stokes), c'est pourquoi nous ne la démontrerons pas théoriquement mais numériquement.

Concernant les équations de Navier-Stokes, nous n'étudierons que numériquement la stabilité.

4.3.3 Étude numérique

L'étude numérique de stabilité est faite avec l'algorithme de résolution des équations de Stokes, Sec. 4.1.3, et l'algorithme de résolution des équations de Navier-Stokes utilisant la méthode des caractéristiques, Sec. 4.1.5.

La géométrie considérée est une bifurcation symétrique, Fig. 4.3, avec un diamètre de 10 mm et une longueur de 100 mm pour chaque branche. Les branches de sorties ont un angle de $\pm \frac{\pi}{4}$ avec l'axe principal de la branche d'entrée.

On considère que le fluide est de l'eau, avec une masse volumique $\rho = 1e^{-3} \text{ g.mm}^{-3}$ et une viscosité dynamique $\mu = 1e^{-3} \text{ Pa.s}^{-1}$. Le temps total de l'étude est fixé à $T = 10 \text{ s}$. Les capacités prises en compte sont égales et constantes pour les deux sorties et valent $5e^{-2} \text{ mm}^3.\text{Pa}^{-1}$ chacune.

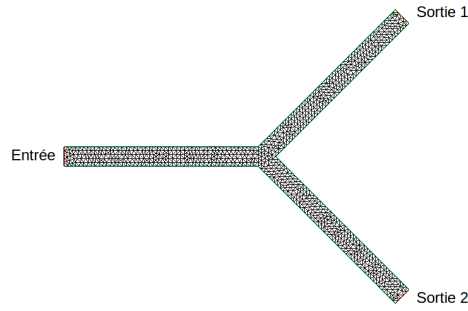


FIGURE 4.3 – **Étude numérique de stabilité.** Maillage utilisé pour l'étude numérique de stabilité.

Comme l'étude théorique de la stabilité nous apporte un résultat pour une condition de Neumann en entrée uniquement, la condition imposée à l'entrée est une pression sinusoïdale de valeur maximum 1000 Pa et de période 1 s, Eq. (4.86).

$$p_{in}(t) = 1000 \sin(2\pi t) \quad (4.86)$$

On a, par ailleurs, la condition CFL théorique suivante :

$$\Delta t < \tilde{c} \times 2.5e^{-6} \quad (4.87)$$

Comme on ne connaît pas *a priori* le coefficient de Poincaré, on étudie un large intervalle de pas de temps : $1e^{-6} \leq \Delta t \leq 1e^{-1}$.

L'allure des courbes de flux et de pression, dans le cas stable, est donnée Fig. 4.4. Comme les capacités de sorties sont identiques, la répartition des flux entre les deux entrées l'est également.

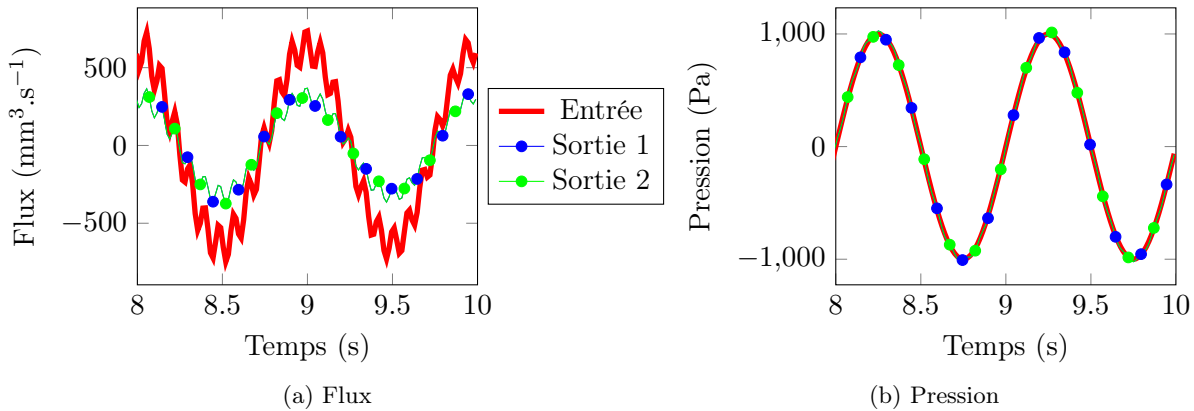


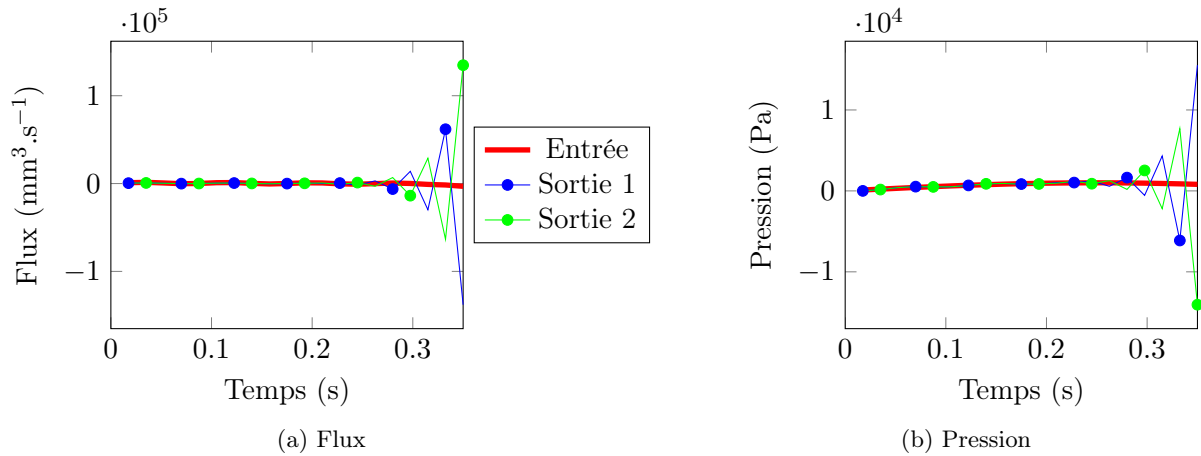
FIGURE 4.4 – **Stabilité.** Comportement du système dans le cas stable. Condition de Neumann.

Comme détaillé Tab. 4.4a, la stabilité est perdue pour un pas de temps compris entre $dt = 1.5e^{-2}$ et $dt = 1.75e^{-2}$, et on retrouve le même comportement de stabilité pour le problème de Stokes et pour le problème de Navier-Stokes. On peut donc en conclure que le problème de Navier-Stokes se comporte identiquement au problème de Stokes en termes de stabilité.

L'allure des courbes de flux et de pression, dans le cas instable, est donnée Fig. 4.5.

Grâce à ce premier résultat, si notre condition CFL est optimale, on peut en déduire le coefficient de Poincaré \tilde{c} . On a :

$$\begin{aligned} 1.5e^{-2} &\leq CFL \leq 1.75e^{-2} \\ 1.5e^{-2} &\leq \tilde{c} \times 2.5e^{-6} \leq 1.75e^{-2} \\ \frac{1.5e^{-2}}{2.5e^{-6}} &\leq \tilde{c} \leq \frac{1.75e^{-2}}{2.5e^{-6}} \\ 6000 &\leq \tilde{c} \leq 7000 \end{aligned} \quad (4.88)$$

FIGURE 4.5 – **Stabilité.** Comportement du système dans le cas instable. Condition de Neumann.

Modification des capacités

Considérons maintenant les mêmes paramètres que pour les simulations précédentes, mais avec des capacités aux sorties égales et constantes, valant $1e^{-1} \text{ mm}^3 \cdot \text{Pa}^{-1}$ chacune. On a maintenant la condition CFL suivante :

$$\Delta t \leq \tilde{c} \times 5e^{-7} \quad (4.89)$$

En utilisant le coefficient de Poincaré calculé, la perte de stabilité doit se produire pour :

$$3e^{-3} \leq \Delta t \leq 3.5e^{-3} \quad (4.90)$$

Comme détaillé Tab. 4.4b, la stabilité persiste au delà de notre condition CFL théorique, elle est perdue pour un pas de temps compris entre $dt = 7e^{-3}$ et $dt = 8e^{-3}$.

Modification du temps final

Considérons maintenant les mêmes paramètres que pour les simulations précédentes (capacités à $1.e^{-1} \text{ mm}^3 \cdot \text{Pa}^{-1}$), mais avec un temps final fixé à $T = 1 \text{ s}$. On a maintenant la condition CFL suivante :

$$3e^{-2} \leq \Delta t \leq 3.5e^{-2} \quad (4.91)$$

Comme détaillé Tab. 4.4c, la stabilité est identique au cas $T = 10 \text{ s}$. La condition CFL n'est donc pas optimale et ne dépend *a priori* pas du temps final de la simulation.

Conditions mêlées

Afin de correspondre au problème que nous allons résoudre, nous allons étudier numériquement la stabilité en conditions mêlées.

La condition imposée à l'entrée est un profil de Poiseuille de vitesse moyenne $10 \text{ mm} \cdot \text{s}^{-1}$ multiplié par un sinus de période 1 s, Eq. (4.92), afin d'avoir une amplitude similaire au cas précédent. On a donc un nombre de Reynolds $Re = 100$.

$$\mathbf{u}_{in}(t) = 10 \sin(2\pi t) \times \text{Poiseuille} \quad (4.92)$$

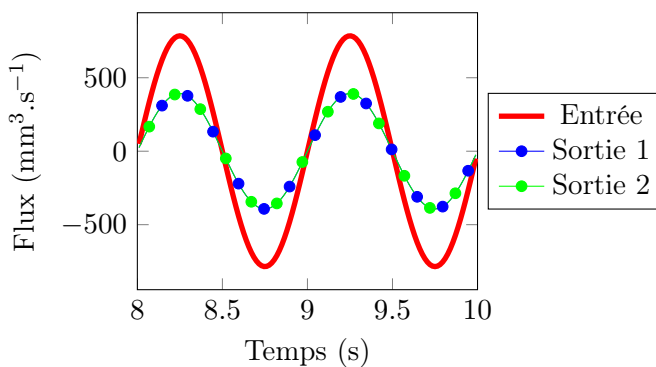
L'allure des courbes de flux et de pression, dans le cas stable, est donnée Fig. 4.6. Comme les capacités de sorties sont identiques, la répartition des flux l'est également.

TABLE 4.4 – Tests de stabilité ✓ : stable, ✗ : non stable.

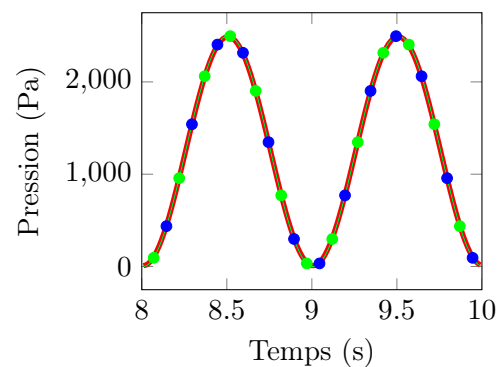
(a) Première partie.			(b) Deuxième partie. Capacités diminuées.		
Δt	Stokes	Navier-Stokes	Δt	Stokes	Navier-Stokes
$1e^{-1}$	✗	✗	$9e^{-3}$	✗	✗
$1.75e^{-2}$	✗	✗	$8e^{-3}$	✗	✗
$1.5e^{-2}$	✓	✓	$7e^{-3}$	✓	✓
$1e^{-2}$	✓	✓	$6e^{-3}$	✓	✓
$1e^{-3}$	✓	✓	$5e^{-3}$	✓	✓
$1e^{-4}$	✓	✓	$4e^{-3}$	✓	✓
$1e^{-5}$	✓	✓	$3e^{-3}$	✓	✓
$1e^{-6}$	✓	✓	$2e^{-3}$	✓	✓

(c) Troisième partie. Temps total diminué.

Δt	Stokes	Navier-Stokes
$9e^{-3}$	✗	✗
$8e^{-3}$	✗	✗
$7e^{-3}$	✓	✓
$6e^{-3}$	✓	✓
$5e^{-3}$	✓	✓
$4e^{-3}$	✓	✓
$3e^{-3}$	✓	✓
$2e^{-3}$	✓	✓



(a) Flux



(b) Pression

FIGURE 4.6 – **Stabilité.** Comportement du système dans le cas stable. Condition de Dirichlet.

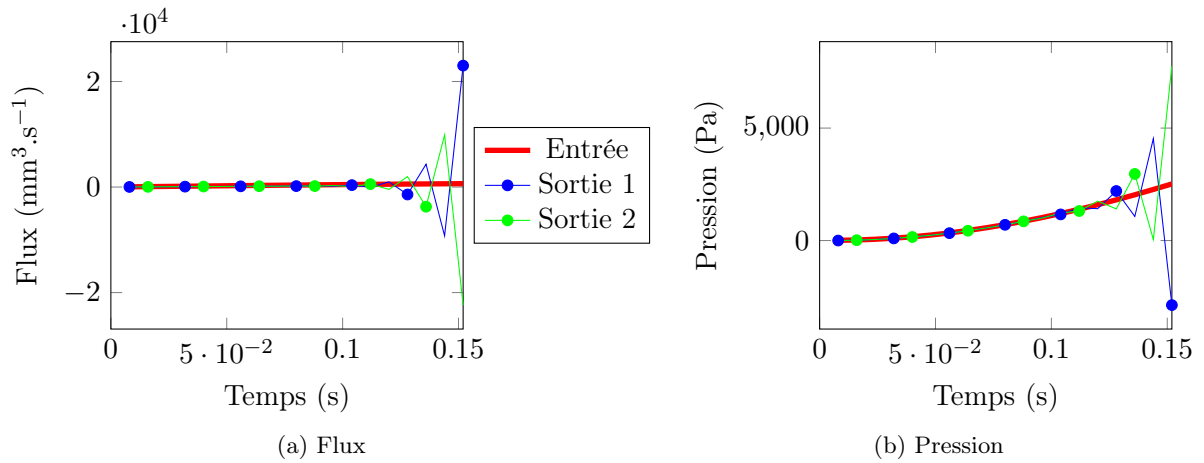


FIGURE 4.7 – **Stabilité.** Comportement du système dans le cas instable. Condition de Dirichlet.

L'allure des courbes de flux et de pression, dans le cas instable, est donnée Fig. 4.7.

Les résultats de stabilité sont identiques aux trois précédents cas avec condition de Neumann en entrée. On a donc une condition de stabilité identique si l'on a une entrée avec une condition de Neumann ou avec une condition de Dirichlet, même si cette dernière n'est pas explicitée théoriquement.

Remarque 4.15. Une étude plus complète prenant en compte des capacités variables se doit d'être réalisée.

4.4 Simulation de l'interaction fluide-structure

4.4.1 FREEFEM++

Tous les détails de cet algorithme sont donnés par S. Sy et C.M. Murea [73].

Soit un domaine Ω décomposé en deux domaines Ω^F pour la partie fluide et Ω^S pour la partie structure, Fig. 4.8. On note $\Sigma_1 = [AF]$, $\Sigma_2 = [AB]$, $\Sigma_3 = [BC]$, $\Gamma_0 = [CF]$, $\Gamma_D = [CD] \cup [EF]$, $\Gamma_N = [DE]$.

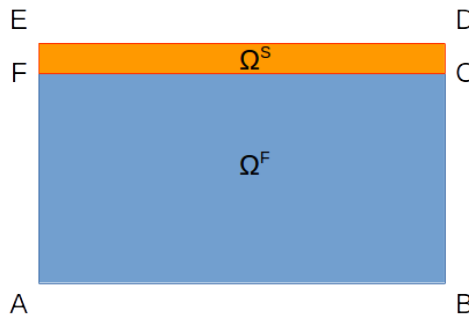


FIGURE 4.8 – **Interaction fluide-structure.** Domaines fluide et structure.

Σ_1 désignera la section d'entrée du fluide, Σ_2 un mur rigide et Σ_3 la section de sortie. Γ_0 est l'interface entre le fluide et la structure, Γ_D et Γ_N sont respectivement les bords où l'on imposera une condition de Dirichlet et une condition de Neumann.

Le modèle d'interaction fluide-structure est donné par les équations suivantes :

Modèle fluide :

$$\rho^F \frac{\partial \mathbf{v}^F}{\partial t} + \rho^F ((\mathbf{v}^F - \mathcal{V}) \cdot \nabla) \mathbf{v}^F - 2\mu^F \nabla \cdot \boldsymbol{\varepsilon}(\mathbf{v}^F) + \nabla p^F = 0 \quad \text{dans } \Omega^F \times [0, T] \quad (4.93a)$$

$$\nabla \cdot \mathbf{v}^F = 0 \quad \text{dans } \Omega^F \times [0, T] \quad (4.93b)$$

$$\mathbf{v}^F = \mathbf{v}_{in} \quad \text{sur } \Sigma_1 \times [0, T] \quad (4.93c)$$

$$\mathbf{v}^F = \mathbf{0} \quad \text{sur } \Sigma_2 \times [0, T] \quad (4.93d)$$

$$\boldsymbol{\sigma}^F(\mathbf{v}^F, p^F) = \mathbf{h}_{out} \quad \text{sur } \Sigma_3 \times [0, T] \quad (4.93e)$$

$$\mathbf{v}^F = \mathbf{v}^0 \quad \text{dans } \Omega^F \times \{0\} \quad (4.93f)$$

$$(4.93g)$$

Modèle structure :

$$\rho^S \frac{\partial^2 \mathbf{u}^S}{\partial t^2} - \nabla \cdot \boldsymbol{\sigma}^S(\mathbf{u}^S) = 0 \quad \text{dans } \Omega^S \times [0, T] \quad (4.94a)$$

$$\mathbf{u}^S = 0 \quad \text{sur } \Gamma_D \times [0, T] \quad (4.94b)$$

$$\boldsymbol{\sigma}^S(\mathbf{u}^S) = 0 \quad \text{sur } \Gamma_N \times [0, T] \quad (4.94c)$$

$$\mathbf{u}^S = \mathbf{u}^0 \quad \text{sur } \Omega^S \times \{0\} \quad (4.94d)$$

$$(4.94e)$$

Couplage :

$$\mathbf{v}^F = \frac{\partial \mathbf{u}^S}{\partial t} \quad \text{sur } \Gamma_0 \times [0, T] \quad (4.95a)$$

$$\boldsymbol{\sigma}^F \mathbf{n}^F = -\boldsymbol{\sigma}^S \mathbf{n}^S \quad \text{sur } \Gamma_0 \times [0, T] \quad (4.95b)$$

$$(4.95c)$$

Avec :

$$\boldsymbol{\varepsilon}(\mathbf{v}) = \frac{1}{2}(\nabla \mathbf{v} + (\nabla \mathbf{v})^t) \quad (4.96)$$

$$\boldsymbol{\sigma}^F(\mathbf{v}^F, p^F) = -p^F \mathbf{I} + 2\mu^F \boldsymbol{\varepsilon}(\mathbf{v}^F) \quad (4.97)$$

$$\boldsymbol{\sigma}^S(\mathbf{u}^S) = \lambda^S (\nabla \cdot \mathbf{u}^S) \mathbf{I} + 2\mu^S \boldsymbol{\varepsilon}(\mathbf{u}^S) \quad (4.98)$$

On note \mathcal{V} la vitesse du maillage, \mathbf{v}^F (\mathbf{v} pour simplifier) la vitesse du fluide, p la pression du fluide et \mathbf{u}^S (\mathbf{u} pour simplifier) le déplacement de la structure. ρ^F est la masse volumique, μ^F la viscosité pour le fluide. ρ^S est la masse volumique, λ^S et μ^S sont les coefficients de Lamé pour la structure.

\mathbf{v}_{in} est la vitesse d'entrée (Dirichlet) du fluide, \mathbf{h}_{out} est la condition de sortie (Neumann).

Le domaine Ω au temps t_n sera noté Ω_n .

Soit $W_n = \{\mathbf{w} \in (H^1(\Omega_n))^2, \mathbf{w} = 0 \text{ sur } \Gamma_D \cup \Sigma_2\}$. La vitesse du maillage est définie par l'équation suivante :

Trouver $\mathcal{V}^n \in W_n$ tel que :

$$-\Delta \mathcal{V}^n = 0 \quad \text{dans } \Omega_n \quad (4.99a)$$

$$\mathcal{V}^n = 0 \quad \text{sur } \partial\Omega_n \setminus \Gamma_0 \quad (4.99b)$$

$$\mathcal{V}^n = \mathbf{v}^n \quad \text{sur } \Gamma_0 \quad (4.99c)$$

On note respectivement $\chi_{\Omega_n^F}$ et $\chi_{\Omega_n^S}$ les fonctions caractéristiques des domaines fluide et structure :

$$\chi_{\Omega_n^S} = \begin{cases} 1 & \text{sur } \overline{\Omega_n^S} \\ 0 & \text{sinon} \end{cases} \quad \chi_{\Omega_n^F} = 1 - \chi_{\Omega_n^S} \quad (4.100)$$

On note par un chapeau, par exemple $\hat{\mathbf{v}}$, la fonction correspondante en coordonnées arbitrairement lagrangiennes eulériennes (ALE, *Arbitrary Lagrangian Eulerian* en anglais), par exemple ici la vitesse du fluide en coordonnées ALE.

Soit $\hat{W}_n^F = \{\hat{\mathbf{w}}^f \in (H^1(\Omega_n^F))^2, \hat{\mathbf{w}}^f = 0 \text{ sur } \Sigma_2\}$ et $\hat{Q}_n^F = L^2(\Omega_n^F)$, le problème d'interaction fluide-structure est défini par les équations suivantes :

Soit $\hat{\mathbf{w}} \in \hat{W}_n^F$ et $\hat{q} \in \hat{Q}_n^F$; trouver $\hat{\mathbf{v}}^{n+1} \in \hat{W}_n^F$ et $p^{n+1} \in \hat{Q}_n^F$ tels que :

$$\begin{aligned} & \int_{\Omega_n} \chi_{\Omega_n^F} \frac{\rho^F}{\Delta t} \hat{\mathbf{v}}^{n+1} \cdot \hat{\mathbf{w}} + \int_{\Omega_n} \chi_{\Omega_n^F} \rho^F ((\mathbf{v}^n - \mathcal{V}^n) \hat{\mathbf{v}}^{n+1}) \cdot \hat{\mathbf{w}} \\ & - \int_{\Omega_n} \chi_{\Omega_n^F} (\nabla \cdot \hat{\mathbf{w}}) \hat{p}^{n+1} + \int_{\Omega_n} \chi_{\Omega_n^F} 2\mu^F \varepsilon(\hat{\mathbf{v}}^{n+1}) : \varepsilon(\hat{\mathbf{w}}) \end{aligned} \quad (4.101a)$$

$$+ \int_{\Omega_n} \chi_{\Omega_n^S} \frac{2\rho^S}{\Delta t} \hat{\mathbf{v}}^{n+1} \cdot \hat{\mathbf{w}} + 2\theta \Delta t \tilde{a}_S(\hat{\mathbf{v}}^{n+1}, \hat{\mathbf{w}}) = \mathcal{L}_F(\hat{\mathbf{w}}) + \mathcal{L}_S(\hat{\mathbf{w}}) \quad (4.101b)$$

$$\int_{\Omega_n} \chi_{\Omega_n^F} \hat{q} (\nabla \cdot \hat{\mathbf{v}}^{n+1}) = 0 \quad (4.101c)$$

Avec :

$$\mathcal{L}_F(\hat{\mathbf{w}}) = \int_{\Omega_n} \chi_{\Omega_n^F} \frac{\rho^F}{\Delta t} \mathbf{v}^n \cdot \hat{\mathbf{w}} + \int_{\Sigma_1} \hat{\mathbf{v}}_{in}^{n+1} \cdot \hat{\mathbf{w}} + \int_{\Sigma_3} \hat{\mathbf{h}}_{out}^{n+1} \cdot \hat{\mathbf{w}} \quad (4.102)$$

$$\begin{aligned} \mathcal{L}_S(\hat{\mathbf{w}}) &= \int_{\Omega_n} \chi_{\Omega_n^S} \frac{2\rho^S}{\Delta t} \mathbf{v}^n \cdot \hat{\mathbf{w}} + \int_{\Omega_n} \chi_{\Omega_n^S} \rho^S \ddot{\mathbf{u}}^n \cdot \hat{\mathbf{w}} - \tilde{a}_S(\mathbf{u}^n, \hat{\mathbf{w}}) \\ & - \Delta t (1 - 2\theta) \tilde{a}_S(\mathbf{v}^n, \hat{\mathbf{w}}) - (\Delta t)^2 \left(\frac{1}{2} - 2\theta \right) \tilde{a}_S(\ddot{\mathbf{u}}^n, \hat{\mathbf{w}}) \end{aligned} \quad (4.103)$$

$$\tilde{a}_S(\mathbf{u}, \mathbf{w}) = \int_{\Omega_n} \chi_{\Omega_n^S} [\lambda^S (\nabla \cdot \mathbf{u}) (\nabla \cdot \mathbf{w}) + 2\mu^S \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{w})] \quad (4.104)$$

On calcule ensuite le déplacement fluide-structure :

$$\hat{\mathbf{u}}^{n+1} = \mathbf{u}^n + (\Delta t)^2 \left(\frac{1}{2} - 2\theta \right) \ddot{\mathbf{u}}^n + \Delta t (1 - 2\theta) \mathbf{v}^n + 2\theta \Delta t \hat{\mathbf{v}}^{n+1} \quad (4.105)$$

L'accélération fluide-structure est donnée par la formule suivante :

$$\hat{\mathbf{u}}^{n+1} = \frac{2}{\Delta t} (\hat{\mathbf{v}}^{n+1} - \mathbf{v}^n) - \ddot{\mathbf{u}}^n \quad (4.106)$$

On définit ensuite le mouvement du domaine par :

$$\mathbb{T}_n = \hat{\mathbf{x}} + \Delta t \mathcal{V}^n \chi_{\Omega_n^F} + (\hat{\mathbf{u}}^{n+1} - \mathbf{u}^n) \chi_{\Omega_n^S} \quad (4.107)$$

Le nouveau domaine est alors $\Omega_{n+1} = \mathbb{T}_n(\Omega_n)$.

On calcule maintenant la vitesse et la pression fluide-structure :

$$\mathbf{v}^{n+1}(\mathbf{x}) = \hat{\mathbf{v}}^{n+1}(\hat{\mathbf{x}}) \quad p^{n+1}(\mathbf{x}) = \hat{p}^{n+1}(\hat{\mathbf{x}}) \quad (4.108)$$

De façon équivalente, on calcule le déplacement et l'accélération fluide-structure :

$$\mathbf{u}^{n+1}(\mathbf{x}) = \hat{\mathbf{u}}^{n+1}(\hat{\mathbf{x}}) \quad \ddot{\mathbf{u}}^{n+1}(\mathbf{x}) = \hat{\ddot{\mathbf{u}}}^{n+1}(\hat{\mathbf{x}}) \quad (4.109)$$

Le système, sous sa forme matricielle, s'écrit :

$$\begin{pmatrix} A & B^T & 0 \\ B & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} V \\ P^F \\ P^S \end{pmatrix} = \begin{pmatrix} \mathcal{L} \\ 0 \\ 0 \end{pmatrix} \quad (4.110)$$

avec $\mathcal{L} = \mathcal{L}_F(\phi_i) + \mathcal{L}_S(\phi_i)$ et les matrices A et B définies par :

$$\begin{aligned} A_{i,j} = & \int_{\Omega_n} \chi_{\Omega_n^F} \rho^F \frac{\phi_j}{\Delta t} \phi_i + \int_{\Omega_n} \chi_{\Omega_n^F} \rho^F ((\mathbf{v}^n - \boldsymbol{\gamma}^n) \cdot \nabla) \phi_j \cdot \phi_i + \int_{\Omega_n} \chi_{\Omega_n^F} 2\nu^F \varepsilon(\phi_j) : \varepsilon(\phi_i) \\ & + \int_{\Omega_n} \chi_{\Omega_n^S} \frac{2\rho^S}{\Delta t} \phi_j \cdot \phi_i + 2\theta \Delta t \tilde{a}_S(\phi_j, \phi_i) \end{aligned} \quad (4.111)$$

$$B_{i,j} = - \int_{\Omega_n} \chi_{\Omega_n^F} (\nabla \cdot \phi_j) \pi_i \quad (4.112)$$

On remarque que P^S peut prendre n'importe quelle valeur, donc le système n'a pas une solution unique. On utilise un terme de stabilisation pour remédier à ce problème et le système matriciel devient :

$$\begin{pmatrix} A & B^T & 0 \\ B & \varepsilon M^F & 0 \\ 0 & 0 & \varepsilon M^F \end{pmatrix} \begin{pmatrix} V \\ P^F \\ P^S \end{pmatrix} = \begin{pmatrix} \mathcal{L} \\ 0 \\ 0 \end{pmatrix} \quad (4.113)$$

avec ε petit et :

$$\begin{pmatrix} \varepsilon M^F & 0 \\ 0 & \varepsilon M^F \end{pmatrix} = \left(\varepsilon \int_{\Omega_n} \pi_j \pi_i \right)_{1 \leq i, j \leq nv} \quad (4.114)$$

nv étant le nombre de nœuds du maillage fluide-structure.

Remarque 4.16. *Cet algorithme, déjà existant et suffisamment robuste pour notre application, a été choisi car il est simple à développer et à tester, et également pour la formulation monolithique qui assure un couplage fort entre le fluide et la structure.*

L'algorithme FREEFEM++ et les détails de programmation sont donnés dans l'Annexe C.5.

4.4.2 FEEL++

Le modèle d'interaction fluide-structure est une combinaison du modèle fluide et du modèle solide suivants [66].

Modèle fluide : équations de Navier-Stokes

$$\rho_f \frac{\partial \mathbf{u}_f}{\partial t} + \rho_f (\mathbf{u}_f \cdot \nabla) \mathbf{u}_f - \operatorname{div}(\boldsymbol{\sigma}_f) = \mathbf{f}_f^t \quad \text{dans } \Omega_f^t \times [t_i, t_f] \quad (4.115a)$$

$$\operatorname{div}(\mathbf{u}_f) = 0 \quad \text{dans } \Omega_f^t \times [t_i, t_f] \quad (4.115b)$$

avec :

$$\boldsymbol{\sigma}_f = -p_f \mathbf{I} + \mu_f (\nabla \mathbf{u}_f + (\nabla \mathbf{u}_f)^T) \quad (4.116)$$

où ρ_f est la masse volumique, μ_f la viscosité dynamique, $\boldsymbol{\sigma}_f$ le tenseur des contraintes, \mathbf{f}_f^t le terme source, \mathbf{u}_f le vecteur vitesse et p_f le champ de pression.

Modèle solide : équation de Saint-Venant-Kirchhoff compressible

$$\boldsymbol{\Sigma}_s = \lambda_s \text{tr}(\mathbf{E}_s) \mathbf{I} + 2\mu_s \mathbf{E}_s \quad (4.117)$$

où λ_s est le premier coefficient de Lamé, μ_s le second coefficient de Lamé, \mathbf{E}_s le tenseur de Green-Lagrange et $\boldsymbol{\Sigma}_s$ le second tenseur de Piola-Kirchhoff.

Ces deux modèles sont couplés par les conditions de couplage suivantes :

$$\frac{\partial \boldsymbol{\eta}_s}{\partial t} - \mathbf{u}_f \circ \mathcal{A}_f^t = \mathbf{0} \quad \text{sur } \Gamma_{fsi}^* \times [t_i, t_f] \quad (4.118a)$$

$$\mathbf{F}_s \boldsymbol{\Sigma}_s \mathbf{n}_s^* + J_{\mathcal{A}_f^t} \hat{\boldsymbol{\sigma}}_f \mathbf{F}_{\mathcal{A}_f^t}^{-T} \mathbf{n}_f^* = \mathbf{0} \quad \text{sur } \Gamma_{fsi}^* \times [t_i, t_f] \quad (4.118b)$$

$$\boldsymbol{\varphi}_s^t - \mathcal{A}_f^t = \mathbf{0} \quad \text{sur } \Gamma_{fsi}^* \times [t_i, t_f] \quad (4.118c)$$

avec $\boldsymbol{\eta}_s$ le déplacement du solide, F_s le gradient de déformation, $\boldsymbol{\Sigma}_s$ le second tenseur de Piola-Kirchhoff, \mathcal{A}_f^t la carte ALE, et avec :

$$\mathbf{F}_{\mathcal{A}_f^t} = \mathbf{x}^* + \nabla \mathcal{A}_f^t \quad (4.119a)$$

$$J_{\mathcal{A}_f^t} = \det(\mathbf{F}_{\mathcal{A}_f^t}) \quad (4.119b)$$

Remarque 4.17. *La mise en place de la simulation à l'aide de FEEL++ est bien plus aisée grâce aux «modèles» prédéfinis dans cette librairie, par contre le temps de simulation est bien plus long et l'algorithme paraît moins robuste pour nos applications. De plus, le côté «boîte noire» rend la modification de l'algorithme en profondeur plus complexe. Néanmoins, la gestion de la mémoire plus efficace que dans FREEFEM++ permet de résoudre des problèmes de plus grande taille.*

Chapitre 5

Validation

Afin de s'assurer du bon fonctionnement des algorithmes utilisés, il est nécessaire de procéder à la vérification et à la validation de ces derniers. Pour les algorithmes fluide, où des solutions exactes sont connues, il convient d'étudier leur convergence. On observera également le comportement de nos algorithmes dans des conditions bien connues de circulation d'un fluide (écoulement de Poiseuille, effet annulaire de Richardson).

L'utilité des modèles de Windkessel sera également mise en évidence dans la distribution d'un fluide entre plusieurs compartiments. Pour l'algorithme fluide-structure, les résultats obtenus seront comparés aux résultats obtenus précédemment dans la littérature.

Finalement, des comparaisons entre différentes bibliothèques de calcul par éléments finis et entre des mesures par imagerie par résonance magnétique (IRM) seront effectuées, afin de s'assurer de la bonne reproduction des phénomènes réels par nos algorithmes.

5.1 Cas tests – Équations de Navier-Stokes

Afin de vérifier les algorithmes fluide développés, on utilise des cas test : des solutions théoriques des équations de Stokes et de Navier-Stokes dans des cas particuliers. Pour la vérification en dimension 2, on utilisera le cas test de Taylor-Green. Pour la vérification en dimension 3, on utilisera le cas test de Ethier-Steinman. On validera également les algorithmes en reproduisant des comportements connus comme l'écoulement de Poiseuille et l'effet annulaire de Richardson.

Remarque 5.1. *D'autres vérifications et validations sur des cas test différents ont été réalisées en collaboration avec Olivia Miraucourt. Les résultats sont disponibles sur le site internet <http://numtourcfd.univ-reims.fr/> [54].*

5.1.1 Vortex de Taylor-Green

G.I. Taylor et A.E. Green [35] ont proposé une solution aux équations de Navier-Stokes incompressibles instationnaires en dimension 2.

On considère le domaine $\Omega = [0, 1]^2$:

$$u_1 = -A \cos(\pi x) \sin(\pi y) e^{-2\nu\pi^2 t} \quad (5.1a)$$

$$u_2 = A \sin(\pi x) \cos(\pi y) e^{-2\nu\pi^2 t} \quad (5.1b)$$

$$p = \frac{-A^2}{4} \rho (\cos(2\pi x) + \cos(2\pi y)) e^{-4\nu\pi^2 t} \quad (5.1c)$$

Avec A un réel et ν la viscosité cinématique :

$$\nu = \frac{\mu}{\rho} \quad (5.2)$$

Une illustration de la solution de Taylor-Green au temps 0 est donnée Fig. 5.1.

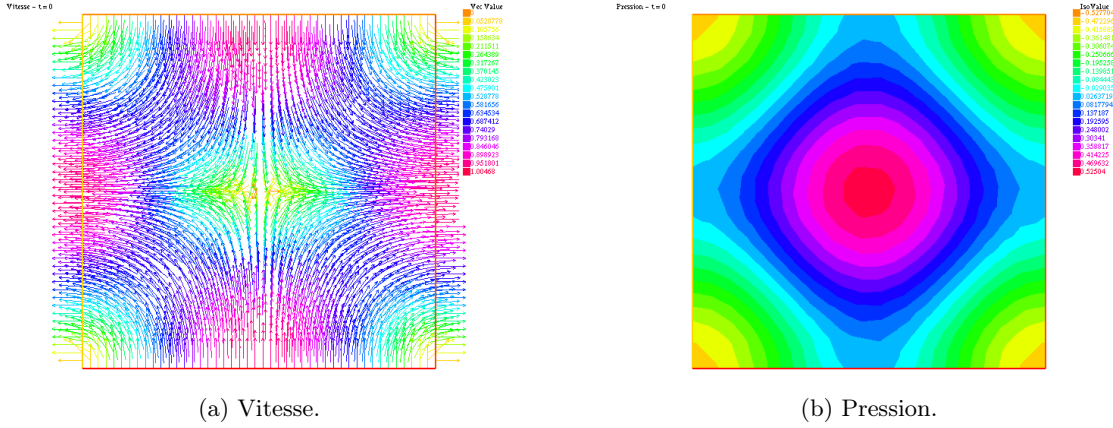


FIGURE 5.1 – Solution de Taylor-Green. Temps 0.

5.1.2 Solution de Ethier-Steinman

Une solution des équations de Navier-Stokes incompressibles instationnaires est donnée en 3D par C.R. Ethier et D.A. Steinman [26].

On considère le domaine $[-1, 1]^3$:

$$u_1 = -a (e^{ax} \sin(ay \pm bz) + e^{az} \cos(ax \pm by)) e^{-b^2 t} \quad (5.3a)$$

$$u_2 = -a (e^{ay} \sin(az \pm bx) + e^{ax} \cos(ay \pm bz)) e^{-b^2 t} \quad (5.3b)$$

$$u_3 = -a (e^{az} \sin(ax \pm by) + e^{ay} \cos(az \pm bx)) e^{-b^2 t} \quad (5.3c)$$

$$p = -\frac{a^2}{2} \left[e^{2ax} + e^{2ay} + e^{2az} + 2 \sin(ax \pm by) \cos(az \pm bx) e^{a(y+z)} + 2 \sin(ay \pm bz) \cos(ax \pm by) e^{a(z+x)} + 2 \sin(az \pm bx) \cos(ay \pm bz) e^{a(x+y)} \right] e^{-2b^2 t} \quad (5.4)$$

avec $a, b \in \mathbb{R}$.

Une illustration de la solution de Ethier-Steinman au temps 0 est donné Fig. 5.2.

5.1.3 Écoulement de Poiseuille

L'écoulement de Poiseuille [56] décrit l'écoulement d'un fluide visqueux dans un tube cylindrique de rayon R constant et de longueur L , Fig. 5.3.

En dimension 3, le profil de vitesse, de forme parabolique, est décrit par la formule :

$$\mathbf{u}(r) = \mathbf{u}_{max} \left(1 - \frac{r^2}{R^2} \right) \quad (5.5)$$

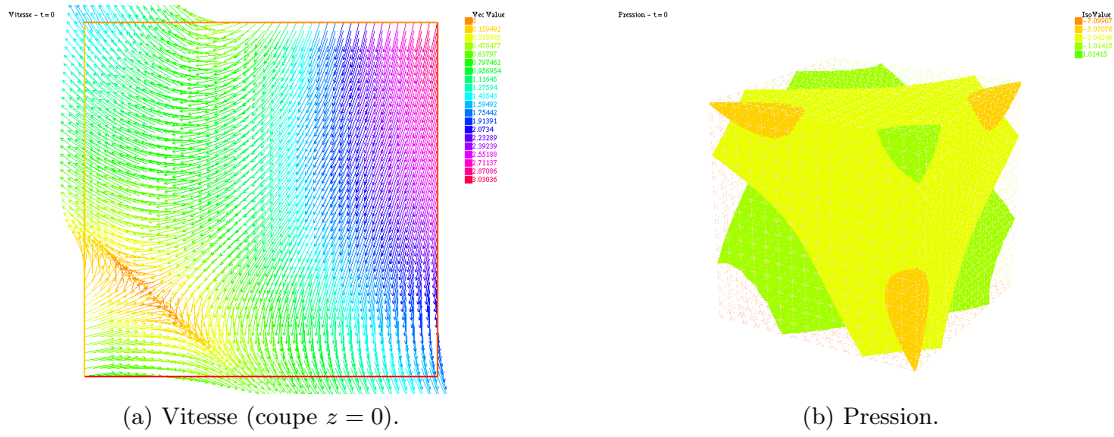


FIGURE 5.2 – Solution de Ethier-Steinman. Temps 0.

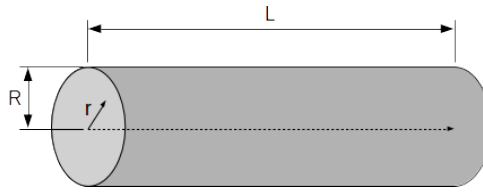


FIGURE 5.3 – Écoulement de Poiseuille. Description du tube.

où \mathbf{u}_{max} est la vitesse maximale, atteinte au centre du tube et r le rayon courant.

On a alors une différence de pression entre l'entrée et la sortie du tube, décrite par la formule :

$$\Delta P = P_{in} - P_{out} = \frac{8\mu L Q}{\pi R^4} \quad (5.6)$$

avec Q le débit de fluide.

En dimension 2, la différence de pression n'est pas identique et est décrite par la formule :

$$\Delta P = \frac{3\mu L Q}{2R^3} \quad (5.7)$$

5.1.4 Écoulement de Richardson

L'effet annulaire de Richardson [62, 76] est un comportement particulier d'écoulements de fluides.

Lorsqu'un fluide est soumis à une forte pulsativité dans un tube, le profil de vitesse diffère d'un profil de Poiseuille parabolique : les vitesses sont plus élevées sur le bord du tube, Fig. 5.4.

5.2 Vérification et validation des algorithmes fluide

Solutions exactes

La convergence est étudiée dans le cas de conditions aux bords mélangées. On impose à la fois des conditions de Dirichlet et de Neumann sur les bords, en utilisant la solution théorique de Taylor-Green (en 2D) ou de Ethier-Steinman (en 3D). La condition de Neumann est définie grâce à la solution théorique par :

$$\mathbf{g}_N = - \int_{\Gamma_N} \mu \frac{\partial \mathbf{u}^E}{\partial \mathbf{n}} - p^E \mathbf{n} \quad (5.8)$$

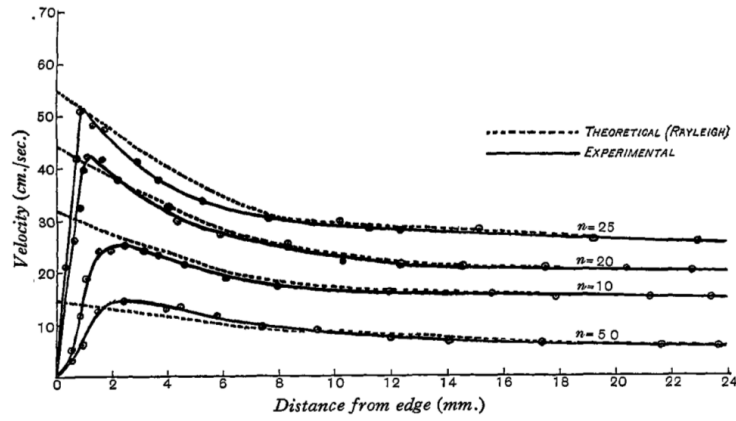


FIGURE 5.4 – **Effet annulaire de Richardson.** Vitesse entre le bord et le centre du tube. Extrait de : *The transverse velocity gradient near the mouths of pipes in which an alternating or continuous flow of air is established* [62].

où \mathbf{u}^E et p^E sont les solutions exactes de vitesse et de pression, respectivement.

Les courbes de convergence sont détaillées Fig. 5.5 pour la convergence en maillage, et Fig. 5.6 pour la convergence en temps dans le cas du vortex de Taylor-Green. Les ordres de convergence calculés, comparés aux ordres théoriques, sont fournis Tab.5.1.

TABLE 5.1 – **Ordres de convergence pour le vortex de Taylor-Green (2D).** Comparaison des ordres théoriques et des ordres calculés.

Variable	Convergence en espace		Convergence en temps	
	Théorique	Calculée	Théorique	Calculée
Caractéristiques				
\mathbf{u}	3	2.66	1	0.95
p	2	2.04	1	0.96
Uzawa				
\mathbf{u}	3	2.66	1	0.95
p	2	2.04	1	0.96
Splitting				
\mathbf{u}	2	2.32	2	1.10
p	2	2.15	2	1.23

Les ordres de convergence calculés sont proches des valeurs théoriques. On observe également que la convergence de la méthode des caractéristiques est identique à la convergence de l'algorithme d'Uzawa.

Les courbes de convergence sont détaillées Fig. 5.7 pour la convergence en maillage, et Fig. 5.8 pour la convergence en temps dans le cas de la solution de Ethier-Steinman. Les ordres de convergence calculés, comparés aux ordres théoriques, sont fournis Tab.5.2.

Les ordres de convergence calculés sont proches des valeurs théoriques, excepté pour l'algorithme de splitting qui a un ordre de convergence très inférieur à la valeur théorique pour la convergence en temps de la vitesse, pour une raison actuellement inexpliquée. On observe également que la convergence de la méthode des caractéristiques est identique à la convergence de l'algorithme d'Uzawa.

Le ralentissement de la convergence en temps est due à un pas de maillage trop grossier pour observer la convergence. Idéalement, il faudrait choisir un pas de maillage plus fin, mais les temps

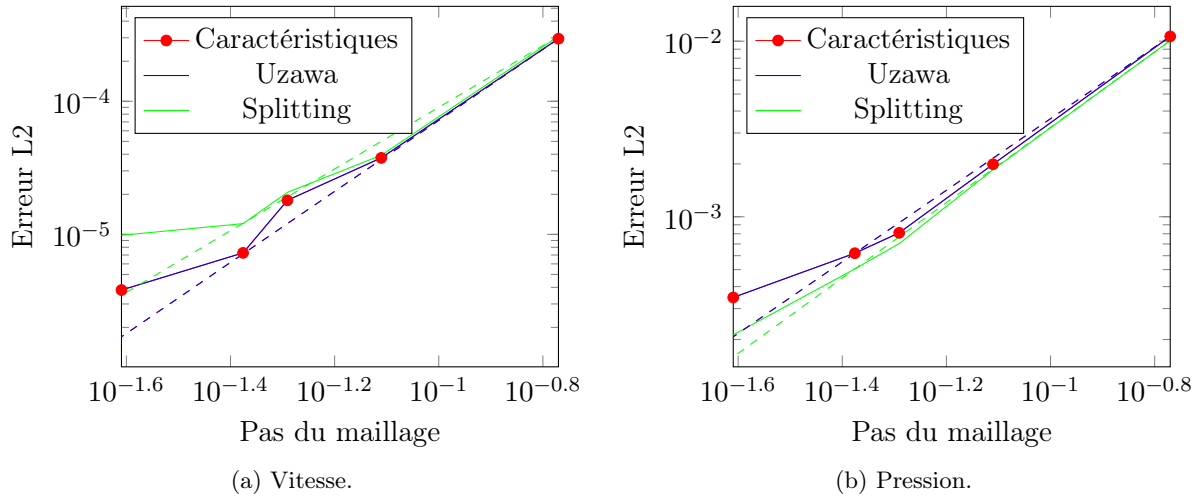


FIGURE 5.5 – **Convergence en maillage.** Solution de Taylor-Green (2D).

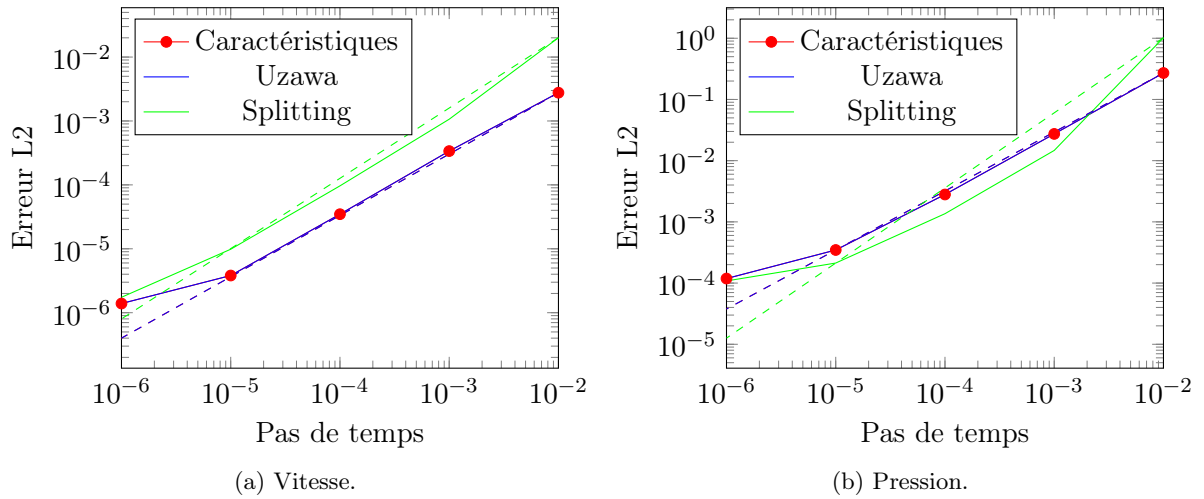
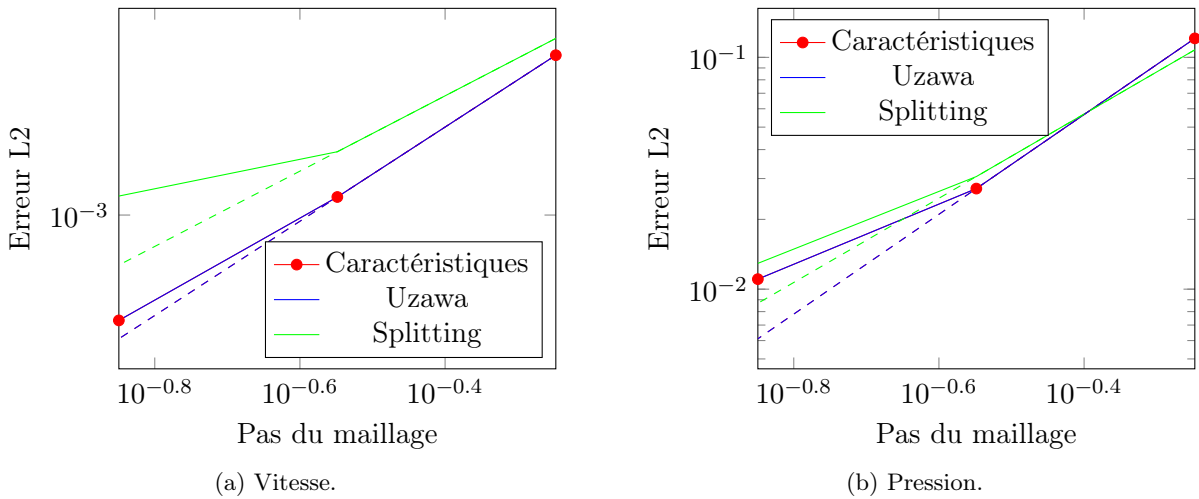
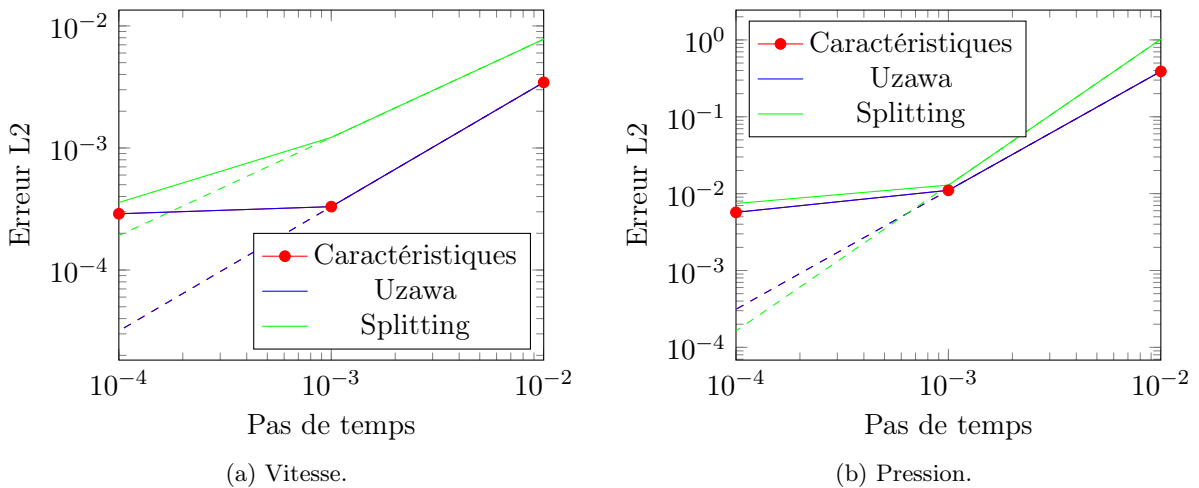


FIGURE 5.6 – **Convergence en temps.** Solution de Taylor-Green (2D).

TABLE 5.2 – **Ordres de convergence pour le vortex de Ethier-Steinman (3D).** Comparaison des ordres théoriques et des ordres calculés.

Variable	Convergence en espace		Convergence en temps	
	Théorique	Calculée	Théorique	Calculée
Caractéristiques				
\mathbf{u}	3	2.15	1	1.02
p	2	2.15	1	1.55
Uzawa				
\mathbf{u}	3	2.15	1	1.02
p	2	2.15	1	1.55
Splitting				
\mathbf{u}	2	1.72	2	0.80
p	2	1.82	2	1.90

FIGURE 5.7 – **Convergence en maillage.** Solution de Ethier-Steinman (3D).FIGURE 5.8 – **Convergence en temps.** Solution de Ethier-Steinman (3D).

de calculs et les quantités de mémoires nécessaires augmentent considérablement. Ce phénomène est également visible dans la convergence en espace, mais dans une moindre mesure ; en effet diminuer le pas de temps provoque exclusivement une augmentation du temps de calcul qui est moins contraignante qu'une augmentation des quantités de mémoires.

Écoulement de Poiseuille

On considère un tube de longueur $L = 10$ mm et de rayon $R = 0.5$ mm. On impose à l'une des deux extrémités un profil de Poiseuille avec $\mathbf{u}_{max} = 10$ mm.s $^{-1}$.

Le profil de Poiseuille est retrouvé à l'aide des différents algorithmes précédents (Stokes, Navier-Stokes caractéristiques, Navier-Stokes Uzawa, Navier-Stokes splitting).

Une illustration est donnée en 2 dimensions, Fig. 5.9. Une illustration est donnée en 3 dimensions, Fig. 5.10.

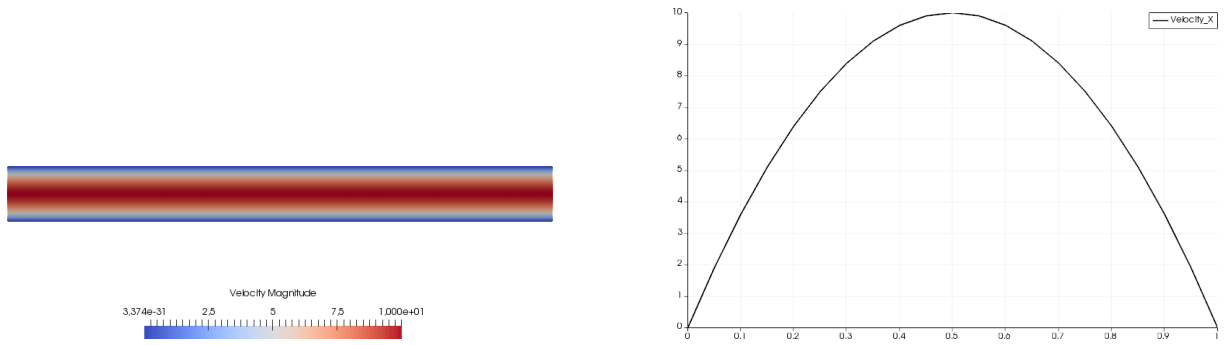


FIGURE 5.9 – Écoulement de Poiseuille 2D. Profil de vitesse à la sortie.

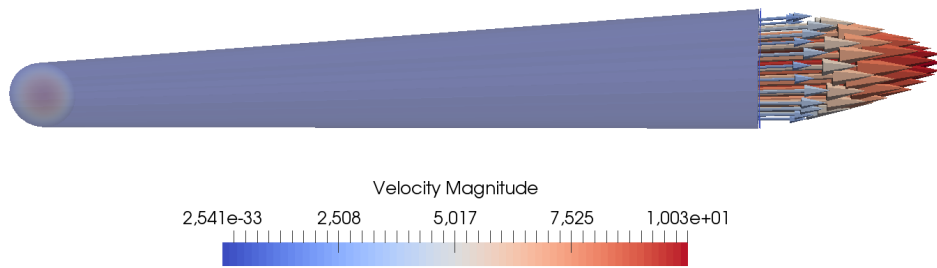


FIGURE 5.10 – Écoulement de Poiseuille 3D. Vecteur vitesse à la sortie.

Effet annulaire de Richardson

On considère un tube de longueur $L = 10$ mm et de rayon $R = 0.5$ mm. On impose à l'une des deux extrémités un profil de Poiseuille que l'on multiplie par un sinus :

$$\mathbf{u}_{entree} = \mathbf{u}_{max} \left(1 - \frac{r^2}{R^2}\right) \sin\left(\frac{2\pi t}{p}\right) \quad (5.9)$$

avec \mathbf{u}_{max} la vitesse maximale, choisie ici à $10 \text{ mm}\cdot\text{s}^{-1}$, et p la période, choisie ici égale à 0.1 s. Les grandeurs caractéristiques de l'écoulement du fluide (Reynolds, Strouhal et Womersley) sont les suivants :

$$Re = 3.33 \quad St = 0.75 \quad Wo = 3.96$$

L'effet annulaire de Richardson est retrouvé à l'aide des différents algorithmes précédents (Stokes, Navier-Stokes caractéristiques, Navier-Stokes Uzawa, Navier-Stokes splitting).

Une illustration est donnée en 2 dimensions, Fig. 5.11. Une illustration est donnée en 3 dimensions, Fig. 5.12.

Pour une période $p = 0.5$ s, avec $Re = 3.33$, $St = 0.15$, $Wo = 1.77$, on observe une atténuation de cet effet vers un profil parabolique aplati, appelé profil de Womersley, Fig. 5.13a. Pour une période $p = 1$ s, avec $Re = 3.333$, $St = 0.075$, $Wo = 1.253$, on constate que l'on retrouve le profil parabolique de Poiseuille, Fig. 5.13b.

5.3 Modèles de Windkessel

Afin de valider l'utilisation des modèles de Windkessel, deux simulations sont réalisées sur une géométrie simple, l'une sans modèle de Windkessel, l'autre avec.

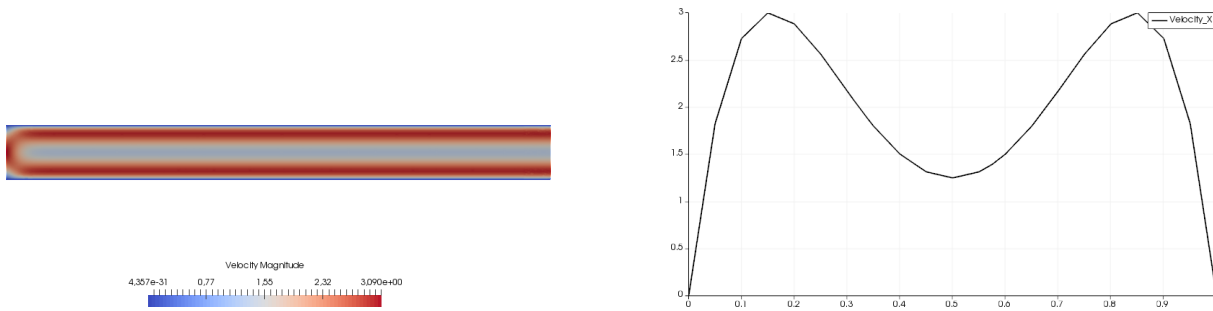


FIGURE 5.11 – Effet annulaire de Richardson 2D. Profil de vitesse à un instant t .

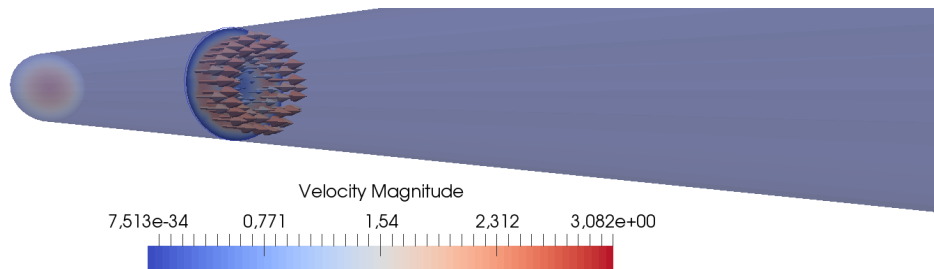


FIGURE 5.12 – Effet annulaire de Richardson 3D. Vecteur vitesse à un instant t .

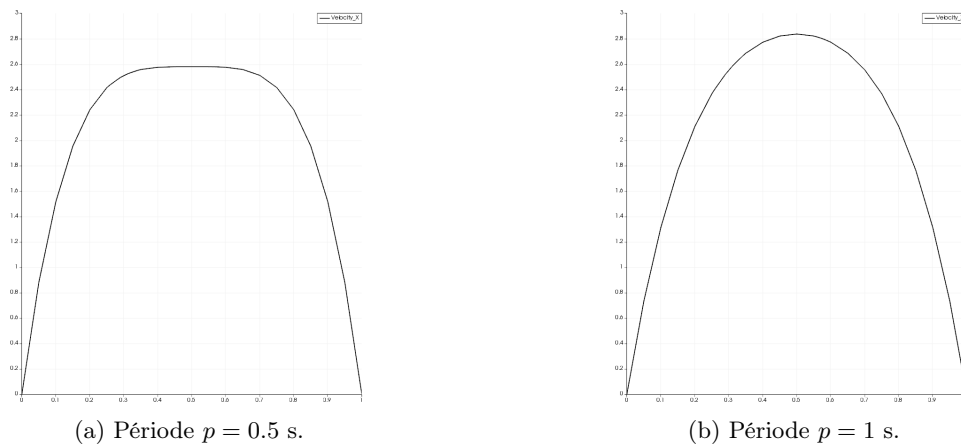


FIGURE 5.13 – Effet annulaire de Richardson. Profil pour différentes périodes.

Le maillage considéré est une bifurcation symétrique identique à celle considérée Sec. 4.3 (branches de 100 mm de longueur et de 10 mm de diamètre, angles de $\pm \frac{\pi}{4}$ pour les deux branches de sorties).

La vitesse en entrée est imposée par un profil de Poiseuille avec une vitesse maximale de la forme :

$$\mathbf{u}_{in} = \sin\left(\frac{2\pi t}{5}\right) \quad (5.10)$$

Dans le premier cas, les sorties sont «libres» ; on y impose une pression nulle via des conditions de Neumann.

Dans le second cas, on impose en sortie des conditions de Windkessel capacitives avec comme valeurs de capacité :

$$\begin{aligned} C_1 &= 1 \\ C_2 &= 0.01 \end{aligned} \quad (5.11)$$

Le temps total de la simulation est $T = 100$ s, avec un pas de temps $\Delta t = 0.01$ s. Les résultats affichés Fig. 5.14 correspondent aux 10 dernières secondes.

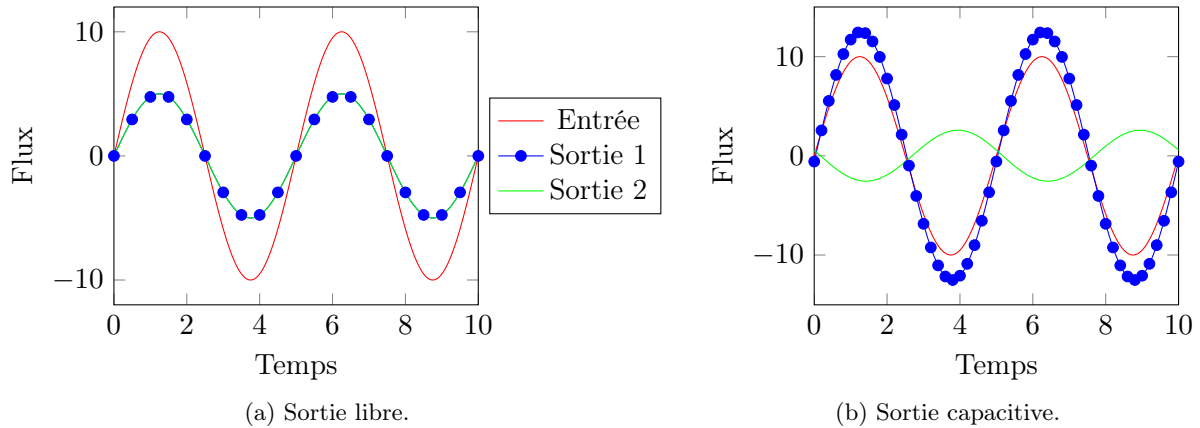


FIGURE 5.14 – **Validation du modèle de Windkessel.** (a) Résultat dans le cas de sorties libres. (b) Résultat dans le cas de sorties capacitives.

Dans le cas de sorties libres, on voit clairement que la distribution des flux est identique dans les deux branches de sortie du système, ce qui est le résultat attendu dans une telle géométrie.

En revanche, dans le cas des sorties capacitives, la distribution des flux est modifiée par les modèles de Windkessel. Il n'y a plus égalité des flux entre les deux sorties.

On a bien un impact des modèles de Windkessel dans le système qui nous permet de contrôler la direction principale des flux.

Remarque 5.2. Dans le cas de sorties capacitives symétriques, c'est à dire $C_1 = C_2$, les flux de sorties sont identiques dans les deux branches.

5.4 Cas test – Interaction fluide–structure

La géométrie prise en compte est un tube de longueur $L = 3$ cm et de hauteur $H = 1$ cm pour la partie fluide.

La viscosité dynamique du fluide est fixée à $\mu = 0.035$ g.cm⁻¹.s⁻¹, sa densité à $\rho^F = 1$ g.cm⁻³. Les forces volumiques sur le fluide sont nulles.

En entrée, on impose une pression de la forme :

$$\mathbf{h}_{in}(\mathbf{x}, t) = \begin{cases} 10^3(1 - \cos(2\pi t/0.025), 0)^t & \text{si } 0 \leq t \leq 0.025 \\ (0, 0)^t & \text{si } 0.025 \leq t \leq T \end{cases} \quad (5.12)$$

La partie structure a une longueur identique à la partie fluide et une épaisseur $e = 0.1$ cm. Le module d'Young est pris égal à $E = 3e^6$ g.cm⁻¹.s⁻², le module de Poisson égal à $\nu = 0.3$, la masse volumique $\rho^S = 1.1$ g.cm⁻³. Les forces volumiques sur la structure sont nulles.

Les coefficients de Lamé sont donnés par les formules suivantes :

$$\lambda^S = \frac{\nu E}{(1 - 2\nu)(1 + \nu)}, \quad \mu^S = \frac{E}{2(1 + \nu)} \quad (5.13)$$

Le paramètre θ du schéma de Newmark est égal à 0.3. On prend un pas de temps $\Delta t = 0.001$ s et on réalise $N = 100$ itérations en temps.

5.5 Validation des algorithmes d'interaction fluide-structure

5.5.1 FREEFEM++

Les résultats de la simulation du cas-test pour l'algorithme d'interaction fluide-structure sont donnés Fig. 5.15 pour le vecteur vitesse, et Fig. 5.16 pour le champ de pression.

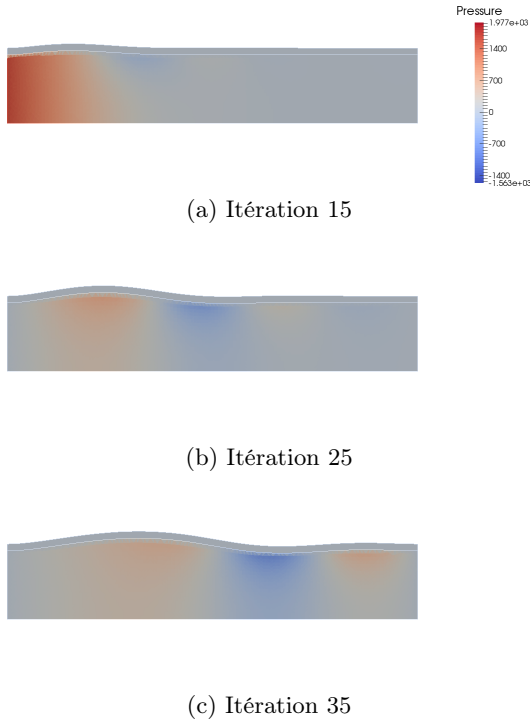


FIGURE 5.15 – Pression

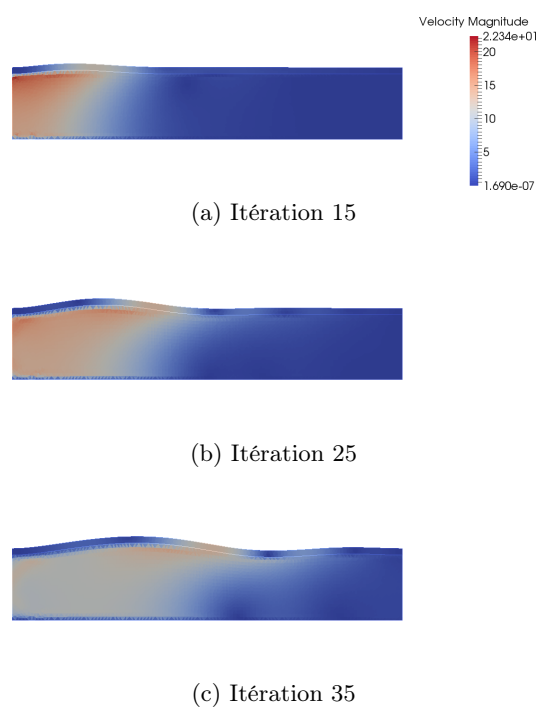


FIGURE 5.16 – Vitesse

La solution trouvée aux différents pas de temps correspond aux résultats obtenus par S. Sy et C. Murea [73].

5.5.2 FEEL++

Les algorithmes FEEL++ sont testés par les équipes qui les développent. Dans le cas particulier de l'algorithme d'interaction fluide-structure, des informations peuvent être trouvées dans [19].

5.6 Validation croisée Simulation – IRM

5.6.1 Conditions expérimentales

L'une des principales difficultés dans la comparaison entre les simulations numériques et les mesures IRM est la mise en place du fantôme physique dans l'IRM.

En effet, comme tout objet ferromagnétique ne peut être à proximité de l'IRM, de fortes contraintes en termes de longueur de tuyaux entre la pompe et le fantôme et de fixation du fantôme apparaissent.

Il est donc difficile de reproduire les conditions exactes des mesures IRM numériquement. La figure 5.17 illustre les problèmes rencontrés et les différences de résultats obtenus lorsque la position du fantôme est légèrement modifiée, Fig. 5.17c, où le flux est fortement perturbé, par rapport à la position idéale, Fig. 5.17b.

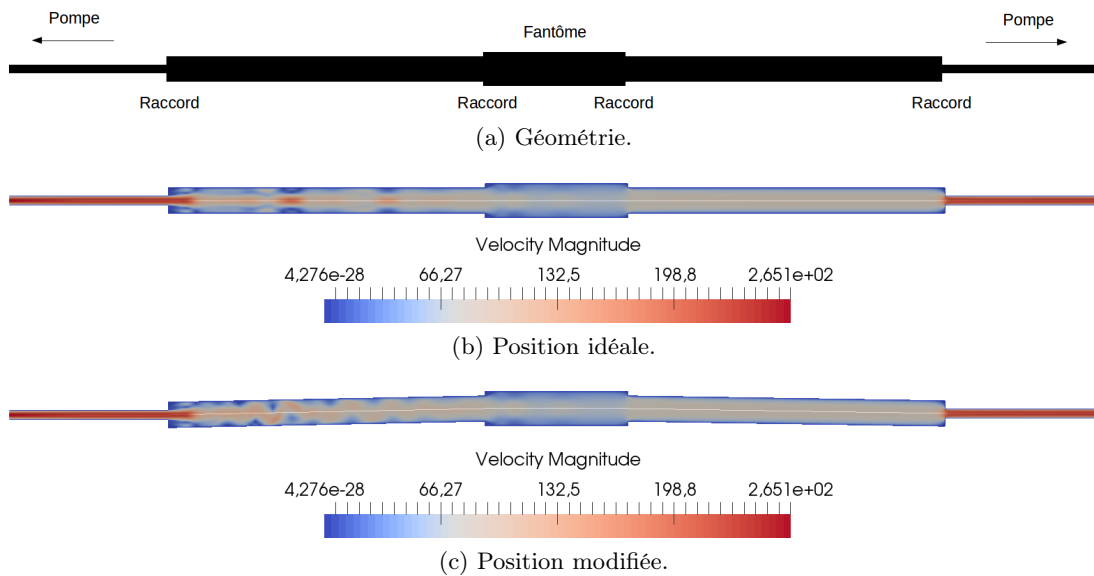


FIGURE 5.17 – **Simulation numérique du fantôme physique.** Schéma et différentes positions possibles.

Le tube le plus fin, celui qui est connecté à la pompe, a une longueur d'environ 5 mètres des deux cotés du fantôme. Il traverse la salle où est installée l'IRM.

Étant donnée la perturbation créée par le raccord à la sortie de ce tube, on utilise un tube d'un diamètre intermédiaire d'une longueur suffisante pour obtenir la stabilisation du flux avant de le connecter au fantôme, pour que son entrée ne soit pas perturbée elle aussi.

La comparaison utilisant un fantôme souple, dans le but de valider l'algorithme d'interaction fluide-structure, présente une difficulté supplémentaire : la gravité ; en effet, la déformation du tube a tendance à être plus importante sur la partie inférieure.

5.6.2 Validation en fluide seul

De nombreux calculs ont été réalisés sur le calculateur ROMEO [75].

Une validation croisée Simulation – IRM peut être trouvée dans l'Annexe E. Ce rapport présente aussi une validation entre FREEFEM++ et FEEL++ et plus largement une validation de la boucle VIVABRAIN faisant intervenir des acquisitions IRM, de la segmentation / du filtrage d'images, de la simulation numérique des écoulements et de la simulation numérique d'IRM angiographiques.

Des mesures par IRM de flux sur un fantôme IRM, Fig. 5.18, sont comparées à des résultats de simulation numérique FREEFEM++ et FEEL++ sur la géométrie reconstruite du fantôme.

Différents plans de coupe sont définis afin de comparer les mesures par IRM de flux et les résultats de simulation, Fig. 5.19.

Des résultats similaires sont trouvés entre les deux bibliothèques FREEFEM++ et FEEL++ et avec les mesures par IRM de flux.

5.6.3 Validation en interaction fluide-structure

Partie en collaboration avec Gwenaël Pagé, doctorant en «Physique – Biophysique» au laboratoire BioFlowImage, Amiens.

Un fantôme souple, Fig. 5.20, de diamètre intérieur $D = 10.8$ mm et de longueur $L = 45$ mm avec une paroi d'épaisseur $E = 1.7$ mm est utilisé afin de comparer les résultats obtenus par IRM



FIGURE 5.18 – Validation en fluide seul. Fantôme IRM.

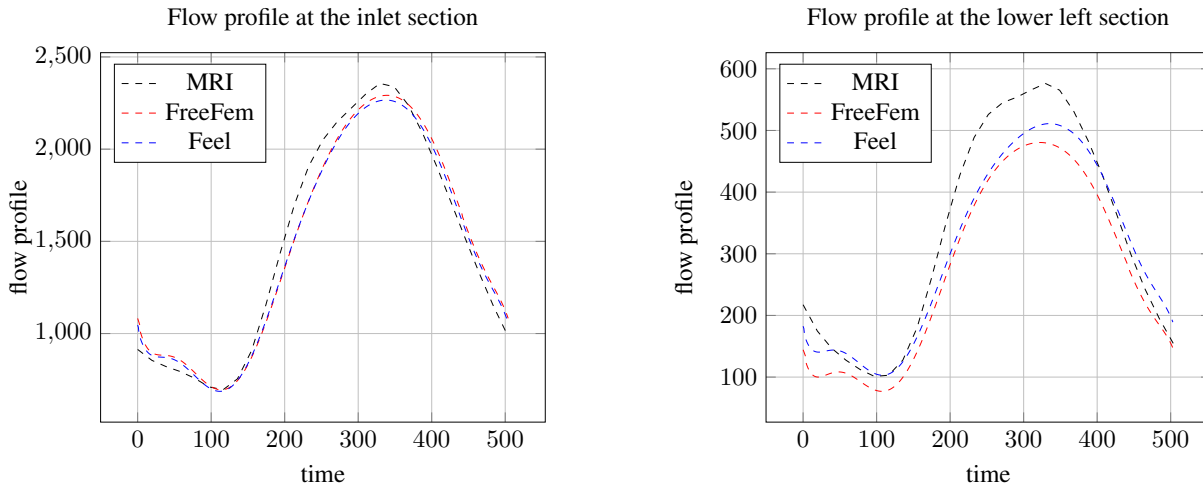


FIGURE 5.19 – Validation en fluide seul. Exemple de résultats.

morphologique et par IRM de flux avec les résultats de simulation obtenus grâce à l'algorithme d'interaction fluide-structure FREEFEM++, Sec. 4.4.1.

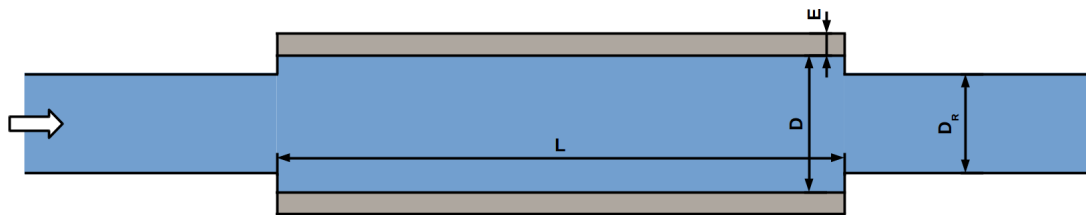


FIGURE 5.20 – Fantôme fluide-structure. Géométrie du fantôme IRM.

Le matériau utilisé pour la construction du fantôme a une masse volumique $\rho^S = 1.1e^{-3} \text{ g.mm}^{-3}$, un module d'Young $E = 70 \text{ kPa}$ et un coefficient de Poisson $\nu = 0.5$.

Remarque 5.3. L'algorithme fluide-structure développé ici étant défini pour une structure compressible, on prendra un coefficient de Poisson proche de l'incompressibilité : $\nu = 0.45$ (proche de 0.5).

Le fantôme est fixé à ses deux extrémités par un tube rigide de diamètre $D_R = 8 \text{ mm}$.

Des mesures IRM de flux sont réalisées en cinq coupes du fantôme, Fig. 5.21, espacées de 7 mm à partir de l'entrée du tube souple.

Il est ainsi possible de comparer les vitesses et surfaces obtenues en ces cinq plans de coupe.

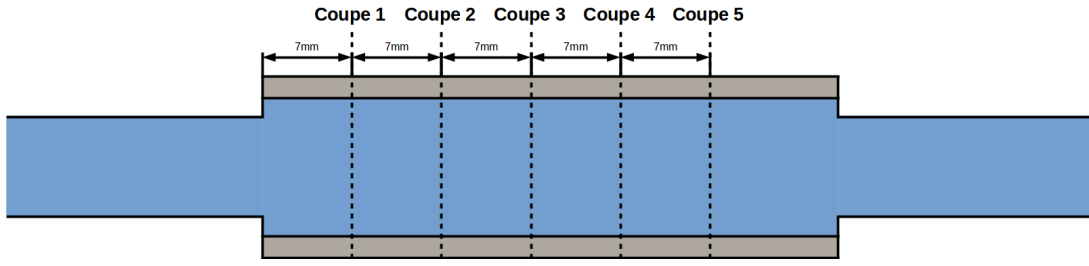


FIGURE 5.21 – **Fantôme fluide-structure.** Positions des coupes IRM.

Le flux d'entrée imposé à l'algorithme fluide-structure est défini par une mesure dans le raccord d'entrée. La simulation, en 2 dimensions, est réalisée sur 60 cycles avec un pas de temps $\Delta t = 1e^{-3}$ s. Seul le dernier cycle est analysé, après avoir vérifié la stabilité de la solution obtenue.

Les vitesses moyennes au cours du temps aux différentes coupes sont données Fig. 5.22.

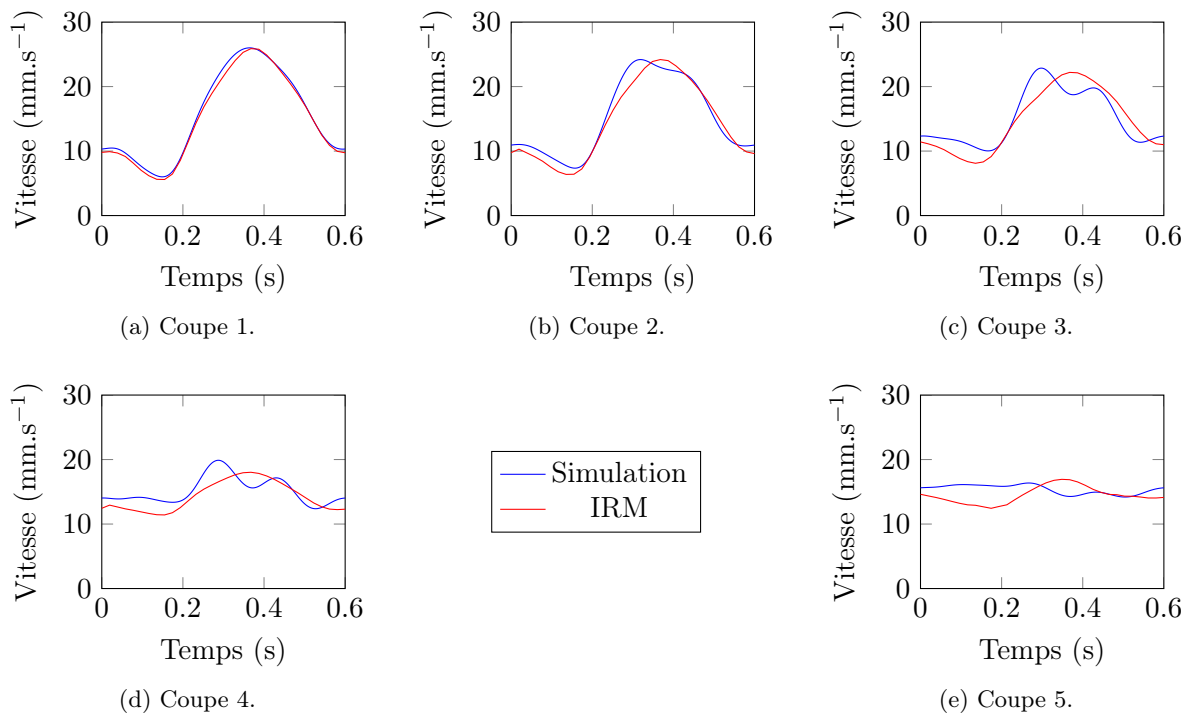


FIGURE 5.22 – **Fantôme fluide-structure.** Comparaison des vitesses moyennes simulées et mesurées.

On observe que les vitesses moyennes simulées et mesurées sont très proches. Seule la dernière coupe présente un déphasage entre vitesse simulée et vitesse mesurée. Les courbes de vitesse des coupes 3 et 4 présentent également des «bosses» sur les résultats numériques qui ne sont pas présentes sur les mesures par IRM de flux qui atténue les hautes fréquences du signal.

La décroissance de l'amplitude des vitesses moyennes aux différentes coupes est linéaire, pour les résultats calculés et mesurés, Fig. 5.23.

Les courbes de décroissance de l'amplitude sont très proches.

Les surfaces des différentes sections de coupe au cours du temps sont données Fig. 5.24. En ce qui concerne les simulations numériques, la surface est obtenue par la formule suivante :

$$S = \pi \left(\frac{D}{2} + d \right)^2 \tag{5.14}$$

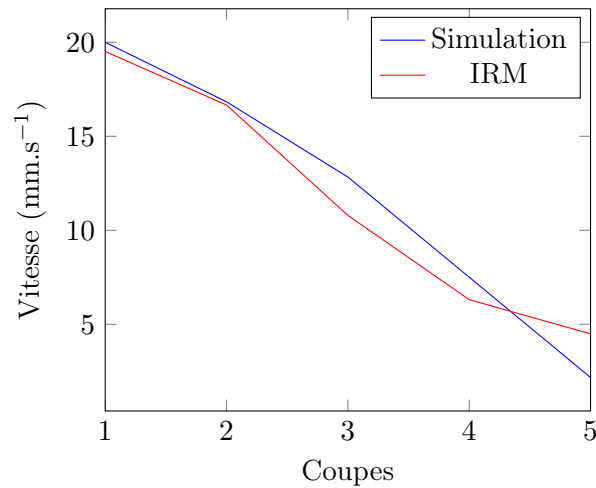


FIGURE 5.23 – **Fantôme fluide-structure.** Comparaison des amplitudes de vitesse.

avec d le déplacement de la structure sur la coupe considérée.

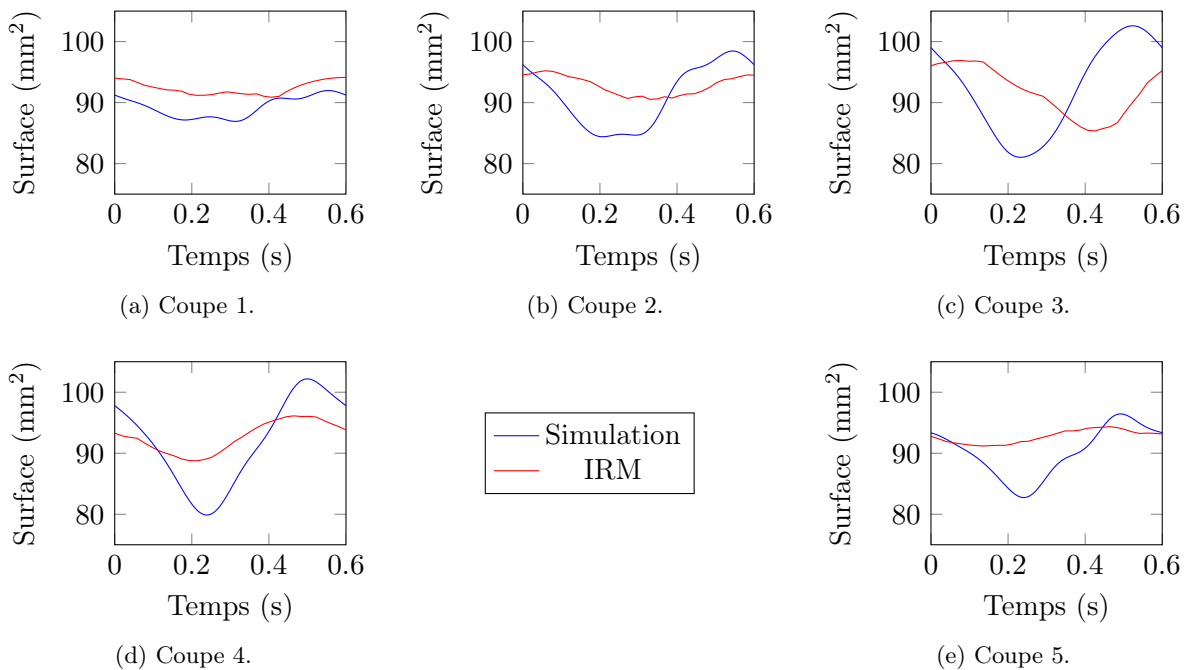


FIGURE 5.24 – **Fantôme fluide-structure.** Comparaison des surfaces simulées et mesurées.

On note que la modification de la surface est largement supérieure dans les simulation numériques (erreur supérieure à 50%). Ceci s'explique par la précision des mesures par IRM de flux depuis lesquelles sont observées les surfaces. En effet, la segmentation des sections se fait à partir des valeurs de vitesses ; si celles-ci sont trop faibles, elles sont tout simplement invisibles. Le section n'est donc pas complètement segmentée. Une solution serait d'utiliser d'autres types de séquences IRM pour parvenir à une meilleure segmentation de la section au cours du temps ; ou d'améliorer les algorithmes de segmentation.

On observe également un déphasage entre les courbes de surface des résultats de simulation numérique et les surfaces mesurées par IRM de flux sur les différentes coupes.

Les résultats, dans l'ensemble, sont relativement concordants. L'erreur commise par l'IRM lors des mesures de vitesse étant quantifiée à 10% [55], nos résultats de simulation présentent une

erreur inférieure. Malgré les difficultés de mise en œuvre expérimentale et les quelques différences entre la simulation et la mesure IRM (la gravité par exemple), les résultats obtenus sont proches. Cette étude permet de valider la cohérence de l'algorithme fluide-structure en conditions réelles.

Pour compléter cette étude de la même manière que l'étude précédente, les simulations ont également été réalisées avec l'algorithme d'interaction fluide-structure FEEL++, Sec. 4.4.2. Des résultats similaires ont été obtenus (le maillage étant construit différemment avec Gmsh [34]). Ceci permet, de plus, de valider l'algorithme d'interaction fluide-structure FREEFEM++.

Une étude plus poussée sur ce sujet est prévue afin de déterminer l'origine exacte des différences obtenues entre les mesures par IRM de flux et les résultats de simulation numérique.

Chapitre 6

Premier modèle : Bifurcation

Sur ce modèle simple, nous allons étudier l'impact des paramètres sur la distribution du liquide cébrospinal (LCS) entre les compartiments intracrâniens : espaces subarachnoïdiens (ESA) cérébraux et ventricules cérébraux. Nous allons également chercher à vérifier l'hypothèse posée par O. Balédent concernant l'influence du cycle cardiaque sur la distribution du LCS.

La construction du premier modèle, en bifurcation, la mise en place de modèles de Windkessel spécifiques à notre problème, ainsi que les paramètres considérés, sont détaillés dans ce chapitre. Les résultats obtenus, en parties à l'aide du calculateur MeCS [74], sont également présentés.

6.1 Modèle

On souhaite représenter le système cérébrospinal à l'aide d'un modèle simple en bifurcation, Fig. 6.1.

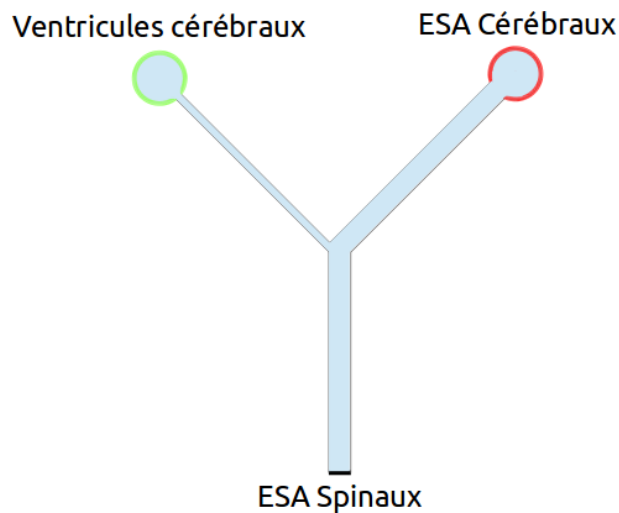


FIGURE 6.1 – Représentation du système cérébrospinal avec une bifurcation.

La donnée concernant le liquide cébrospinal la plus simple à obtenir en IRM de flux est la vitesse au niveau cervical (vertèbres C2–C3), on considérera donc les espaces subarachnoïdiens spinaux comme notre entrée.

Pour modéliser les ESA cérébraux et les ventricules cérébraux, nous avons recours à des modèles de Windkessel de type capacitif. En première approche, on considère des modèles indépendants. Mais les ventricules cérébraux sont en interaction avec les ESA cérébraux et le cerveau, à l'intérieur de la boîte crânienne rigide, il faut donc améliorer notre modèle pour prendre en compte

cette inter-dépendance entre les deux compartiments cérébraux de LCS ainsi qu'avec le compartiment sanguin.

Les différents paramètres pour la création du maillage de la bifurcation sont détaillés Fig. 6.2.

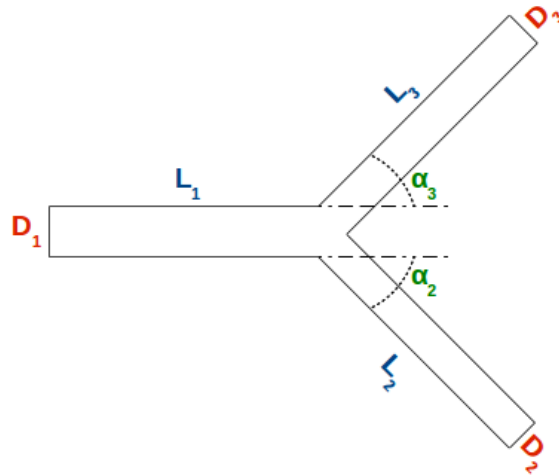


FIGURE 6.2 – **Bifurcation.** Paramètres géométriques.

Le paramètre L_i correspond à la longueur de la branche i ; le paramètre D_i correspond au diamètre de la branche i ; et le paramètre α_i correspond à l'angle de la branche i par rapport à l'axe principal de la branche 1.

6.1.1 Modèles capacitifs constants

En première approche, on définit donc nos modèles de Windkessel de manière indépendante, Fig. 6.3.

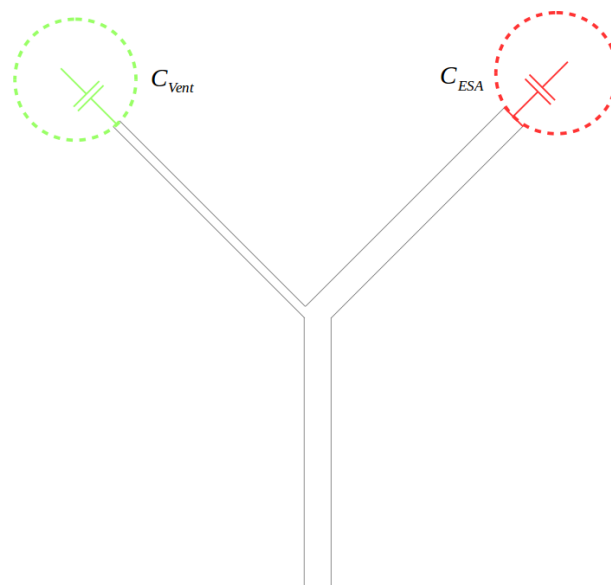


FIGURE 6.3 – **Modèles de Windkessel constants.** Capacités indépendantes.

Les capacités sont définies comme des constantes, et nos conditions aux bords (de type Neumann) sont de la forme :

$$p_{ESA}(t) = \frac{1}{C_{ESA}} V_{ESA}(t) \quad (6.1a)$$

$$p_{Vent}(t) = \frac{1}{C_{Vent}} V_{Vent}(t) \quad (6.1b)$$

Où C_{Vent} et C_{ESA} sont les capacités des ESA cérébraux et des ventricules cérébraux, respectivement, avec V_{ESA} et V_{Vent} les volumes des ESA cérébraux et des ventricules cérébraux, respectivement, obtenus numériquement suivant la formule :

$$V_{\mathbf{X}}(t) = \sum_{k=0}^n Q_{\mathbf{X}}(t) \quad (6.2)$$

avec $Q_{\mathbf{X}}$ le débit, et $\mathbf{X} \in \{ESA, Vent\}$.

Remarque 6.1. Dans notre application, le terme capacité sera remplacé par le terme compliance correspondant à l'appellation utilisée dans la pratique clinique.

6.1.2 Modèles capacitifs variables

Afin de prendre en compte l'interaction entre les ventricules cérébraux et les ESA cérébraux ainsi que l'impact du flux artério-veineux dans notre modèle, on modifie les modèles de Windkessel pour les rendre variables.

La variabilité des modèles de Windkessel se fait par l'intermédiaire d'une pression : «la pression du cerveau», Fig. 6.4.

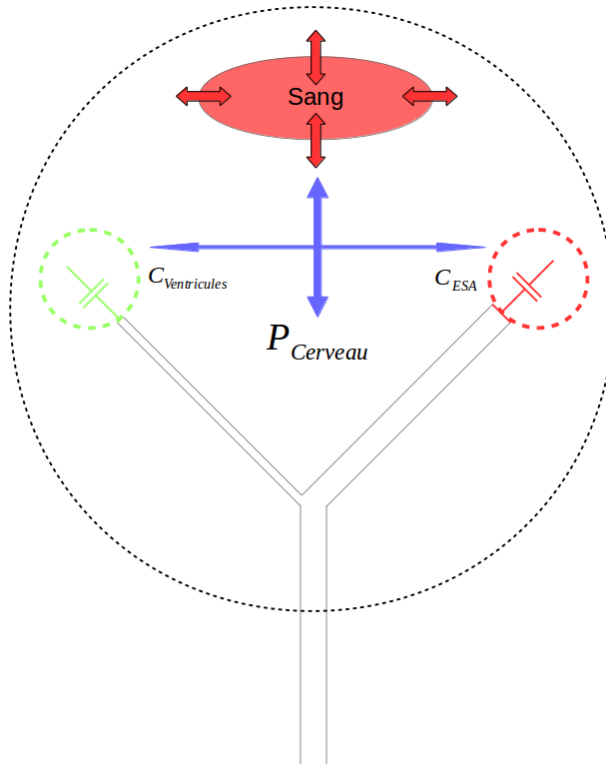


FIGURE 6.4 – Modèles de Windkessel variables. Capacités dépendantes de la pression du cerveau.

On commence par définir une pression du cerveau dépendante du temps $p_{Cerveau}(t)$ afin de rendre nos compliances variables selon les formules :

$$C_{ESA}(t) = \frac{C_{ESA}^0}{p_{Cerveau}(t)} \quad (6.3a)$$

$$C_{Vent}(t) = \frac{C_{Vent}^0}{p_{Cerveau}(t)} \quad (6.3b)$$

avec C_{ESA}^0 et C_{Vent}^0 respectivement les compliances initiales pour les ESA cérébraux et pour les ventricules cérébraux.

Nos conditions aux bords deviennent donc :

$$p_{ESA}(t) = \frac{1}{C_{ESA}(t)} V_{ESA}(t) \quad (6.4a)$$

$$p_{Vent}(t) = \frac{1}{C_{Vent}(t)} V_{Vent}(t) \quad (6.4b)$$

On prendra comme pression du cerveau la fonction :

$$P_{Cerveau}(t) = P_0 e^{kV(t)} \quad (6.5)$$

avec P_0 la pression initiale, k une constante et $V(t) = V_{ESA}(t) + V_{Vent}(t) + V_{AV}(t)$ la somme du volume des ESA cérébraux, du volume des ventricules cérébraux et du volume artério-veineux.

Remarque 6.2. Cette fonction de pression est définie à l'aide des études de Marmarou sur la pression intracrânienne, Chap. 2.

Ainsi, plus le volume contenu dans le compartiment intracrânien augmente, plus la pression est importante. De même pour les compliances, plus le volume contenu est important, plus il est difficile d'y faire rentrer du fluide.

6.2 Résultats – Windkessel constant

6.2.1 Stroke volume

Une quantité très utilisée pour quantifier les échanges de flux dans le système cérébrospinal est le stroke volume (ou volume oscillatoire). Il représente, pour un flux oscillatoire, la quantité de volume déplacé au cours d'un cycle cardiaque. Pour l'obtenir numériquement, on utilisera la formule suivante (en utilisant la formule des trapèzes) :

$$SV = \sum_{k=0}^{N-1} \frac{|Q^{k+1} + Q^k|}{2} (t^{k+1} - t^k) \quad (6.6)$$

où $[t^0, t^N]$ représente le temps du cycle cardiaque et Q^k le flux au temps t^k .

6.2.2 Paramètres

Dans les premiers tests concernant l'influence des différents paramètres physiologiques du système cérébrospinal sur la distribution de LCS, les paramètres détaillés Tab. 6.1 sont utilisés.

Le maillage considéré, Fig. 6.5, est une bifurcation. Chaque branche mesure 100 mm de longueur. Les branches représentatives des ESA spinaux et des ESA cérébraux ont un diamètre de 10 mm ;

TABLE 6.1 – Paramètres de simulation. Étude de l’impact des paramètres physiologiques.

Paramètre	Symbole	Valeur	Unité
Masse volumique	ρ	$1.e^{-3}$	g.mm^{-3}
Viscosité dynamique	μ	$1.e^{-3}$	Pa.s
Compliance des ESA cérébraux	C_{ESA}	1	$\text{mm}^3.\text{Pa}^{-1}$
Compliance des ventricules cérébraux	C_{Vent}	2	$\text{mm}^3.\text{Pa}^{-1}$
Amplitude du flux d’entrée	A	10	$\text{mm}^3.\text{s}^{-1}$
Période du flux d’entrée	p	1	s

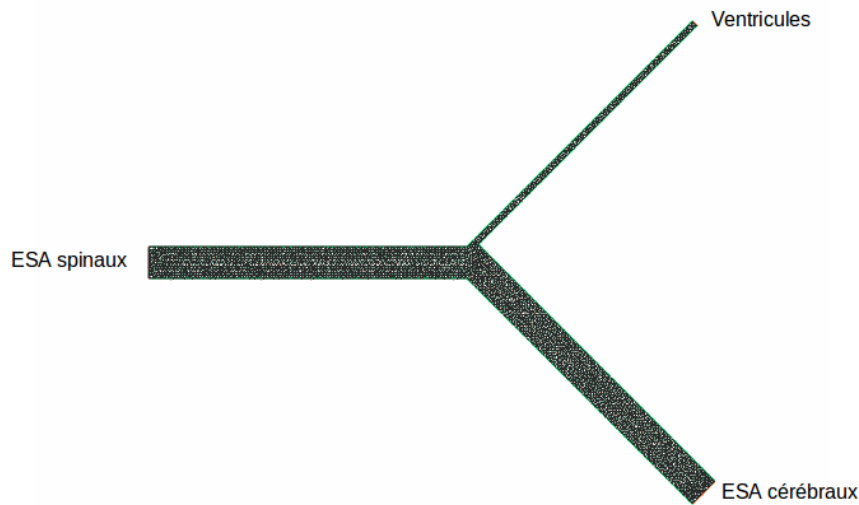


FIGURE 6.5 – Modèle en bifurcation.

la branche représentative des ventricules cérébraux a un diamètre de 2 mm. Les branches représentatives des ventricules et des ESA cérébraux ont un angle de $+\frac{\pi}{4}$ et $-\frac{\pi}{4}$ respectivement par rapport à l’axe de la branche représentative des ESA spinaux ($L_1 = L_2 = L_3 = 100$ mm, $D_1 = D_2 = 10$ mm, $D_3 = 2$ mm et $\alpha_2 = -\frac{\pi}{4}$, $\alpha_3 = \frac{\pi}{4}$). Le pas du maillage est de 1 mm.

Le flux d’entrée, au niveau des ESA spinaux, sera pris égal à :

$$F(t) = A \sin\left(\frac{2\pi t}{p}\right) \quad (6.7)$$

avec A l’amplitude et p la période du flux d’entrée.

Les valeurs de compliances sont choisies arbitrairement.

Les résultats présentés sont obtenus avec l’algorithme d’Uzawa avec un temps total de simulation $T = 20$ s et un pas de temps $\Delta t = \frac{p}{100}$ s.

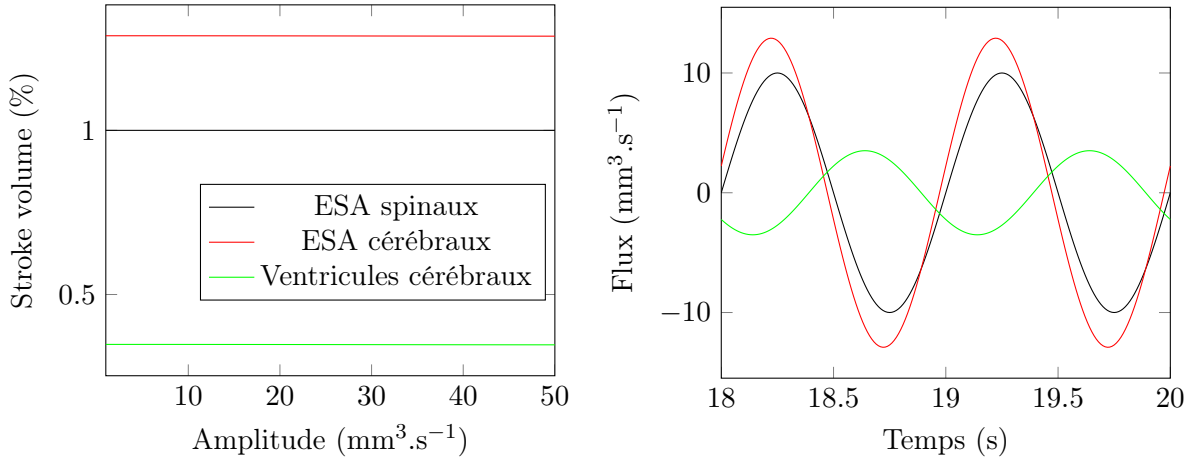
Dans la section traitant des cas réels, les paramètres, ainsi que les maillages, seront spécifiques à la géométrie observée par IRM morphologiques et aux vitesses mesurées par IRM de flux.

Les résultats présentés dans les cas réels sont obtenus avec l’algorithme d’Uzawa avec un temps total de simulation $T = 10$ s et un pas de temps $\Delta t = 1e^{-2}$ s.

6.2.3 Impact de l'amplitude

Afin d'étudier l'impact de l'amplitude du flux d'entrée, nous prenons en compte les paramètres détaillés Tab. 6.1 et nous faisons varier l'amplitude d'entrée dans l'intervalle $[1, 50]$, qui représente une large gamme de valeurs physiologiques observées sur témoins ou sur patients.

Les résultats de simulation sont illustrés Fig. 6.6a par le stroke volume dans les différents compartiments normalisés par le stroke volume des ESA spinaux.



(a) Stroke volume.

(b) Exemple de flux dans les différents compartiments. On peut observer le déphasage entre les flux dans les ventricules cérébraux et dans les ESA cérébraux.

FIGURE 6.6 – **Windkessel constant.** Premières observations.

On constate que l'amplitude du flux d'entrée n'a pas d'impact sur la distribution du LCS dans les deux compartiments cérébraux. La majorité du flux va en direction des ESA cérébraux (tube de faible résistance) ; le reste du flux va dans les ventricules cérébraux (forte résistance).

Le stroke volume dans les ESA cérébraux est supérieur à 1 à cause d'un déphasage avec le flux dans les ventricules cérébraux, Fig. 6.6b. On a ainsi un stroke volume supérieur au stroke volume d'entrée dans les ESA cérébraux, dû à un échange entre les ventricules cérébraux et les ESA cérébraux.

On s'intéressera donc plutôt aux courbes de stroke volume normalisées, Fig. 6.7. La normalisation du stroke volume est réalisée par rapport à l'entrée : les ESA spinaux, puis par rapport à la somme entre les valeurs des ESA cérébraux et des ventricules cérébraux. On ajoute aux résultats la pression moyenne en temps et le gradient de pression (maximum moins minimum pendant un cycle cardiaque) en fonction de l'amplitude.

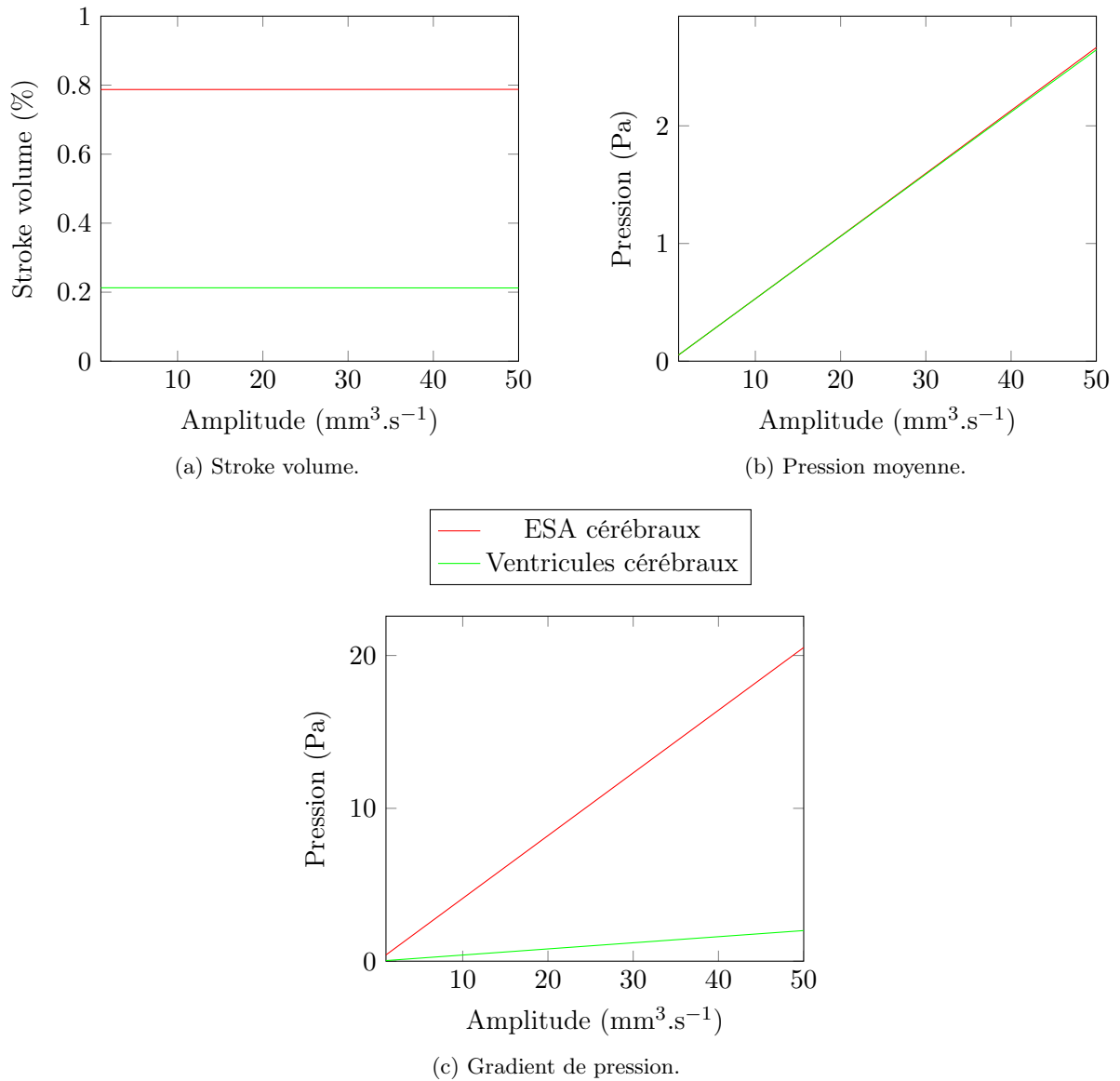
On a une répartition des stroke volume de 20% pour les ventricules cérébraux et de 80% pour les ESA cérébraux, ce qui est proche du comportement physiologique.

La pression moyenne croît linéairement avec l'augmentation de l'amplitude ; ceci est dû au caractère constant des compliances.

Le gradient de pression croît également linéairement, avec une pression plus élevée dans les ESA cérébraux, là où il y a le plus de changements de volume (c'est à dire, là où le stroke volume est le plus important).

6.2.4 Impact du cycle cardiaque

Deux tests sont réalisés en conservant ou non le stroke volume en entrée. Dans le premier cas, le stroke volume n'est pas conservé ; on ne modifie donc que la période du flux d'entrée dans


 FIGURE 6.7 – **Windkessel constant.** Effet de l'amplitude.

l'intervalle $[0.2, 2]$ qui correspond à une variation de la fréquence cardiaque de 30 battements par minutes (bpm) à 300 bpm (intervalle physiologique). Les autres paramètres sont ceux détaillées Tab. 6.1.

Dans le second cas, on modifie la période du flux d'entrée, mais également l'amplitude pour garder un stroke volume constant. L'amplitude sera donc fonction de la période selon la formule suivante :

$$A_{SV} = A \frac{\pi}{p} \quad (6.8)$$

Remarque 6.3. *On veut que le stroke volume soit identique selon la période. Comme notre entrée est symétrique, on ne regarde que le volume du lobe supérieur de l'entrée.*

$$SV_+ = \int_0^{p/2} A \sin\left(\frac{2\pi t}{p}\right) dt = \left[-\frac{p}{2\pi} \cos\left(\frac{2\pi t}{p}\right)\right]_0^{p/2} = \frac{p}{\pi}$$

On doit donc diviser l'amplitude du flux d'entrée par $\frac{p}{\pi}$.

Ce second cas est plus représentatif des conditions physiologiques du système cérébrospinal. En effet, lors de modifications de la fréquence cardiaque, le stroke volume de LCS observé sur patient / témoin reste constant.

Non-conservation du stroke volume

Les résultats de simulation exposés Fig. 6.8 mettent en évidence un très grand impact du cycle cardiaque sur la distribution du LCS entre les ESA cérébraux et les ventricules cérébraux.

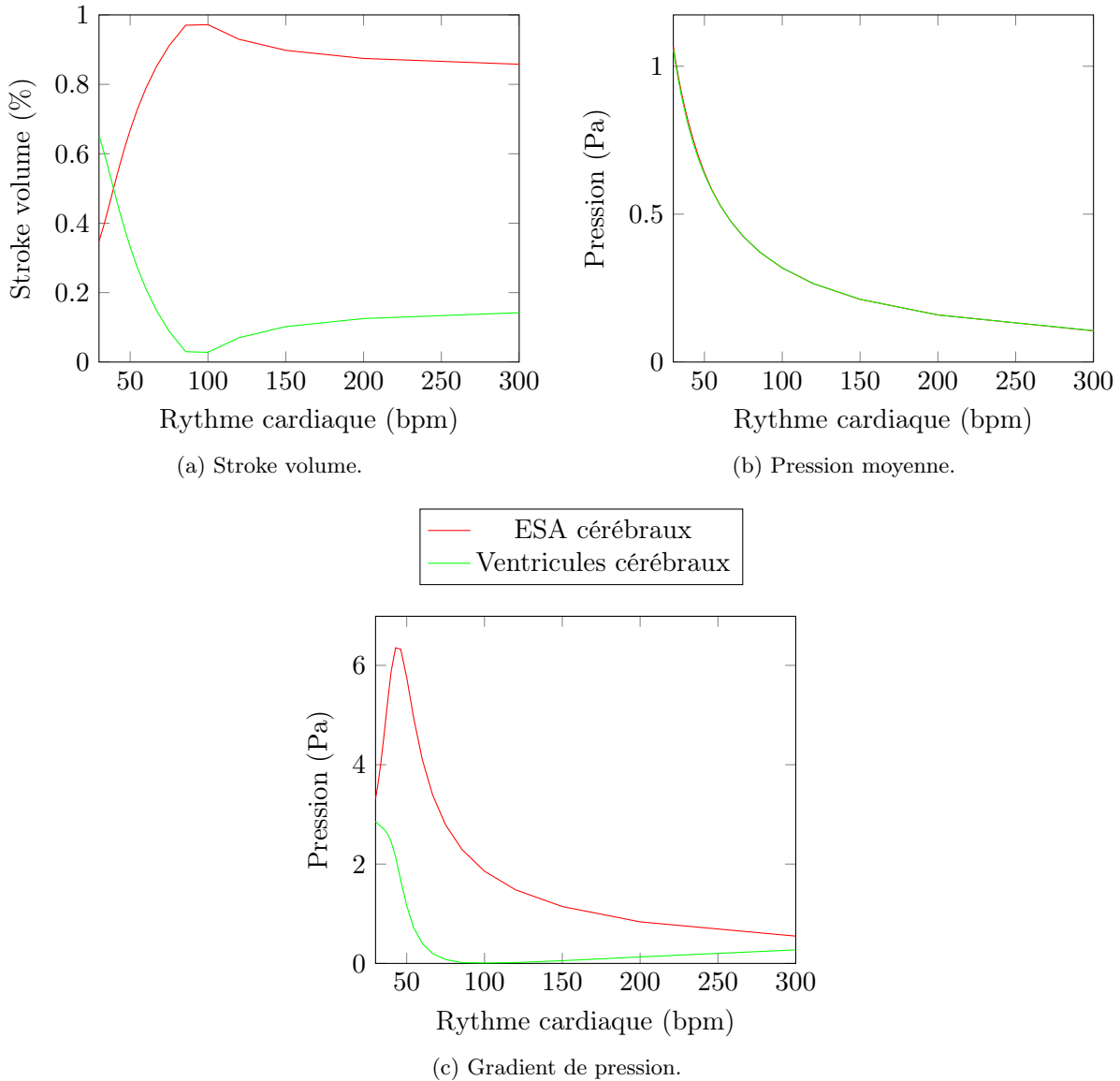


FIGURE 6.8 – **Windkessel constant.** Effet de la période.

Les rythmes cardiaques lents (au repos, par exemple) privilégient les ventricules cérébraux alors que les rythmes cardiaques rapides (à l'exercice, par exemple) privilégient les ESA cérébraux.

La pression moyenne décroît exponentiellement car il n'y a pas de conservation du stroke volume ici : plus le rythme cardiaque est élevé, moins il y a de volume entrant dans les compliances.

Le gradient de pression commence par augmenter durant l'équilibrage des flux, avant de diminuer exponentiellement comme la pression moyenne. Le gradient de pression dans les ventricules cérébraux remonte légèrement pour les rythmes cardiaques élevés, tout comme le stroke volume.

Conservation du stroke volume

Dans le cas où on conserve le stroke volume, Fig. 6.9, on observe des résultats identiques au cas de non-conservation du stroke volume. En effet comme l'amplitude du flux n'a pas d'effet sur la distribution du LCS entre les deux compartiments, il est normal de retrouver des résultats similaires.

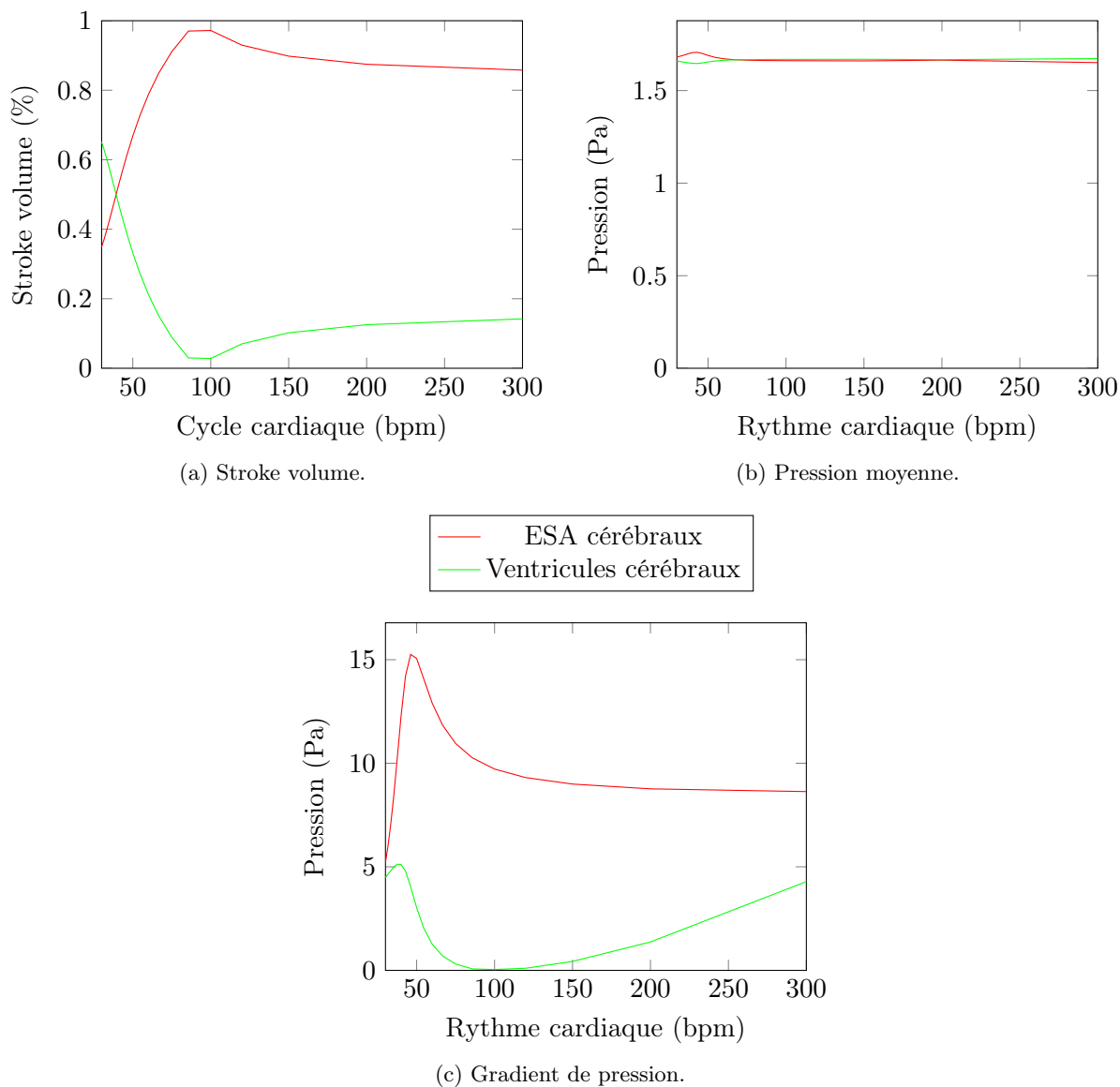


FIGURE 6.9 – **Windkessel constant.** Effet de la période avec conservation du stroke volume.

La pression moyenne ne décroît plus exponentiellement ici, comme le stroke volume est conservé. La valeur de pression est pratiquement constante, avec quelques fluctuations lors de l'équilibrage des flux.

Le gradient de pression a une forme similaire au cas de non-conservation du stroke volume. Il est plus élevé et la séparation est plus franche entre les ESA cérébraux et les ventricules cérébraux.

On a donc des résultats similaires en conservant ou non le stroke volume. Dans la suite, on exposera uniquement les résultats avec conservation du stroke volume, puisqu'ils représentent de façon plus fidèle le comportement physiologique du système cérébrospinal, et qu'ils diffèrent peu des résultats avec non conservation du stroke volume.

6.2.5 Impact de la compliance

Finalement, on modifie la valeur de la compliance des ventricules dans l'intervalle $[0.1, 5]$, en conservant les autres paramètres détaillés Tab. 6.1.

Les résultats de simulation sont exposés Fig. 6.10. On y observe un comportement assez particulier du système. Dans un premier temps, plus la compliance des ventricules cérébraux est grande, plus le stroke volume augmente dans ce compartiment ; néanmoins, ce comportement s'inverse à partir de la valeur $C_{Vent} = 0.4$.

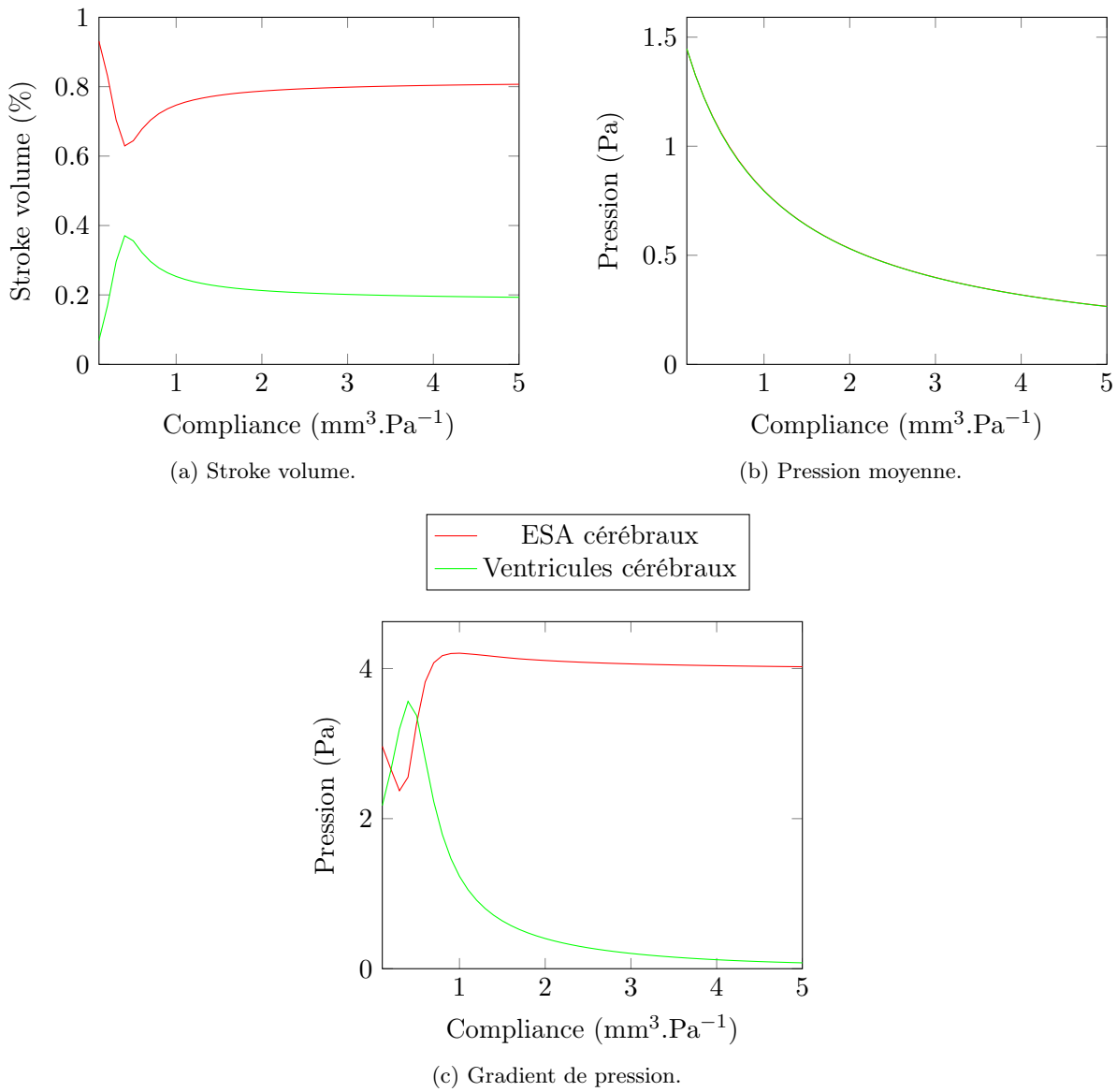


FIGURE 6.10 – **Windkessel constant.** Effet de la compliance.

La pression moyenne diminue exponentiellement avec l'augmentation de la compliance globale du système (somme de la compliance des ESA cérébraux et de la compliance des ventricules cérébraux).

Le gradient de pression suit la courbe de stroke volume. Le gradient de pression dans les ventricules cérébraux augmente avant de décroître et de se stabiliser. Le gradient de pression dans les ESA cérébraux diminue avant d'augmenter et de se stabiliser.

Ce comportement, qui n'est pas celui attendu (on s'attend à observer une distribution des flux en fonction des compliances), est dû à un volume entrant insuffisant à l'intérieur des compliances que nous allons mettre en évidence dans un second test avec une période différente.

On réalise une nouvelle fois le test de l'influence des compliances, en considérant cette fois-ci une période plus grande, de 2 s, afin de solliciter d'avantage les compliances et d'en observer plus précisément l'impact. Les résultats de simulation sont exposés Fig. 6.11.

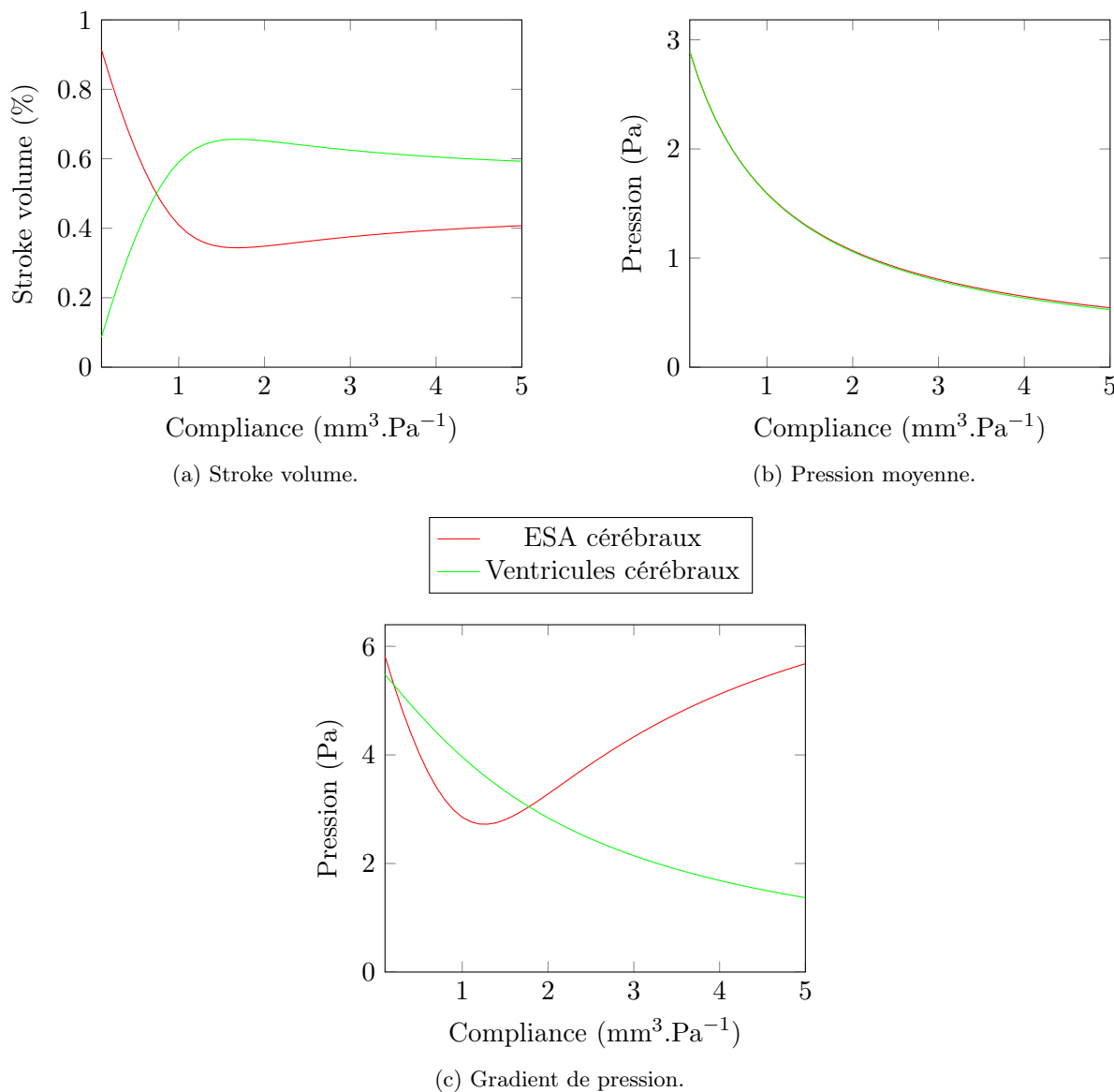


FIGURE 6.11 – **Windkessel constant.** Effet de la compliance (période de 2 s).

On obtient bien une répartition des stroke volumes en fonction des compliances des différents compartiments.

La pression moyenne diminue toujours de manière exponentielle avec l'augmentation de la compliance globale du système.

Le gradient de pression dans les ventricules cérébraux décroît avec l'augmentation de la compliance dans ce compartiment. Le gradient de pression dans les ESA cérébraux décroît avant d'augmenter en suivant la courbe de stroke volume.

6.2.6 Cas réels

Dans cette partie, les paramètres géométriques du maillage seront déterminés grâce à des mesures directes sur des IRM morphologiques, Fig. 6.12, à l'aide des logiciels MIPAV [53] et ImageJ [69].

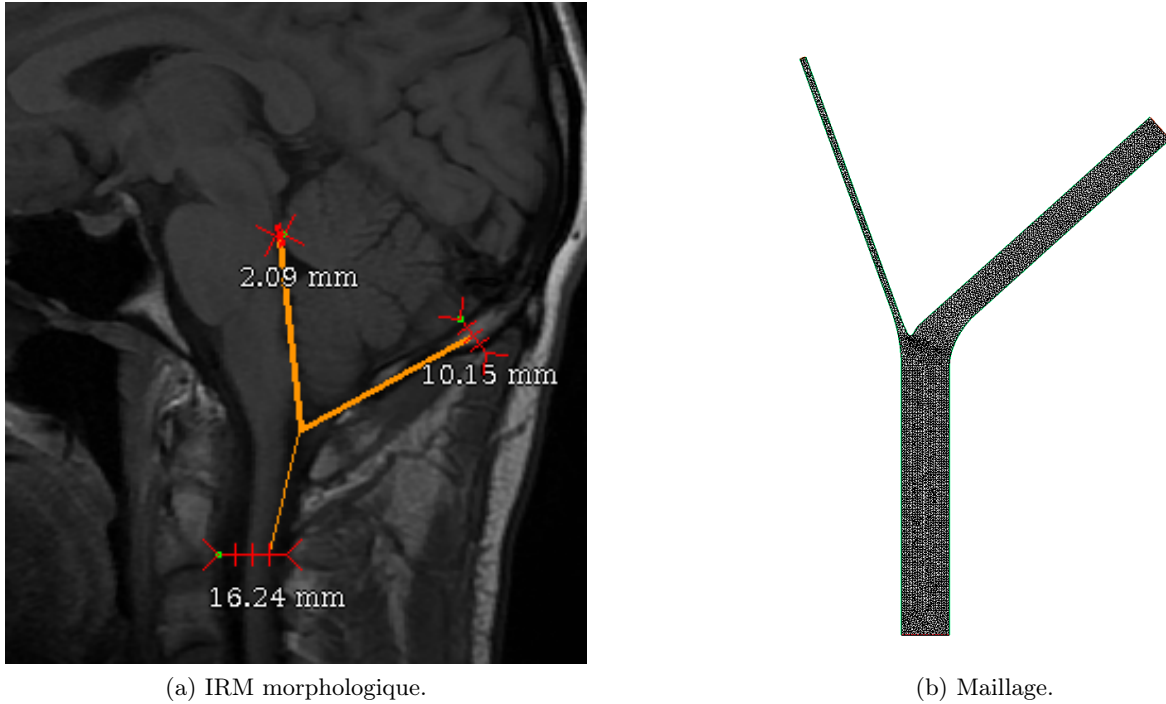


FIGURE 6.12 – **Cas réel.** Mesures des paramètres géométriques sur une IRM morphologique pour la création d'un maillage physiologique.

La vitesse imposée à l'entrée du modèle est déterminée par des mesures par IRM de flux, voir Chap. 12.

Cas pathologique

Dans ce cas particulier, il s'agit d'un patient ayant subi un test d'injection (projet PROLIPHYC, voir Sec. 9.2). Nous avons donc à notre disposition une mesure de la pression intracrânienne pendant différentes phases de stress du système cérébrospinal. Des mesures par IRM morphologiques et de flux ont également été réalisées.

Les images issues de l'IRM morphologique nous permettent de définir les paramètres géométriques patient-dépendants. On obtient le maillage présenté Fig. 6.13, avec les dimensions détaillées Tab. 6.2.

Les mesures par IRM de flux nous permettent d'imposer la vitesse d'entrée au niveau des ESA spinaux. Les nombres de Reynolds et de Strouhal sont calculés :

$$Re = 7.97366 ; St = 8.29316 \quad (6.9)$$

Remarque 6.4. *Le nombre de Reynolds étant petit, nous aurions très bien pu nous contenter d'un algorithme de résolution des équations de Stokes. En revanche, le nombre de Strouhal étant élevé, il est bien nécessaire de considérer un algorithme instationnaire.*

Finalement, les mesures de pression durant l'état basal du test d'injection (état cohérent avec les mesures IRM) nous permettent de déterminer la compliance du système (voir Section 10.3). On obtient une valeur de compliance $C_G = 1.42 \text{ mm}^3 \cdot \text{Pa}^{-1}$ pour le système dans sa globalité.

TABLE 6.2 – Cas pathologique. Dimensions.

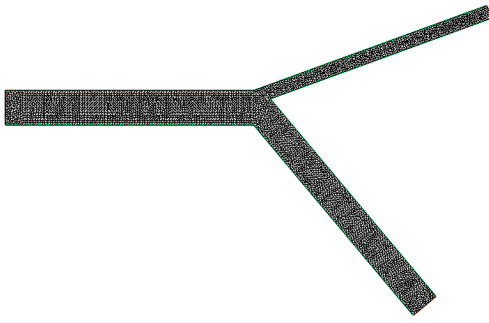


FIGURE 6.13 – Cas pathologique. Maillage.

Paramètre	Valeur
L_1	100 mm
L_2	100 mm
L_3	100 mm
D_1	14 mm
D_2	12 mm
D_3	5.5 mm
α_2	50 °
α_3	20 °

La répartition entre la compliabilité des ventricules cérébraux et la compliabilité des ESA cérébraux se fait de telle sorte que la répartition des flux dans le modèle numérique soit en adéquation avec les flux mesurés par IRM de flux. On a $C_{Vent} = 1/4 C_G$ et $C_{ESA} = 3/4 C_G$.

Les résultats de simulation sont détaillés Fig. 6.14.

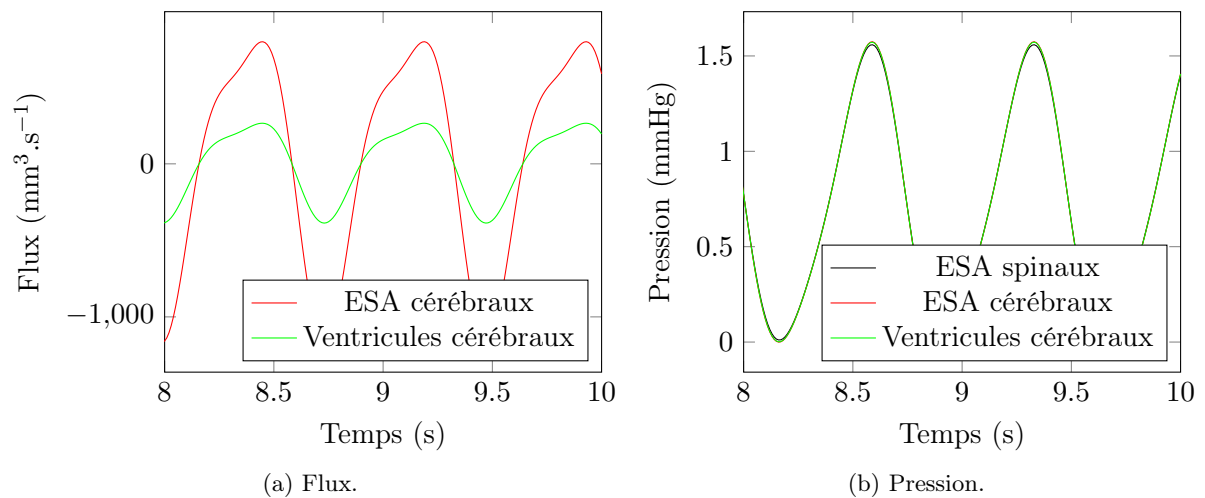


FIGURE 6.14 – Cas pathologique. Flux et pressions.

Grâce aux données mesurées, on peut comparer nos résultats de simulation aux mesures de flux et de pression, Fig. 6.15.

On observe que les courbes de flux sont équivalentes en amplitude. Le flux des ESA spinaux correspond à notre condition d'entrée ; elle diffère uniquement à cause de l'approximation par une série de Fourier, voir Chap. 12. Par contre, le flux dans les ventricules cérébraux présente un déphasage avec le flux des ESA spinaux dans les mesures, que l'on ne retrouve pas dans les simulations.

La courbe de pression simulée est quant à elle éloignée de la pression mesurée. L'amplitude de la pression calculée est plus petite que la pression mesurée et la courbe ne présente pas de pics et de vallées.

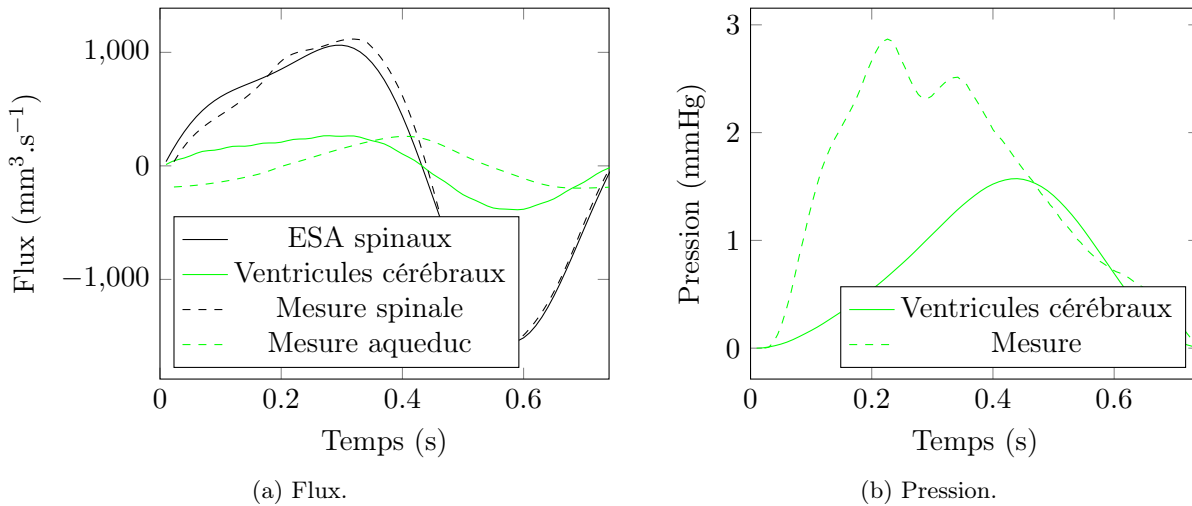


FIGURE 6.15 – **Cas pathologique.** Comparaison des résultats et des mesures.

Cas sain

Pour ce cas, seules des IRM morphologiques et de flux sont réalisées, la mesure de pression intracrânienne étant une technique invasive et comportant un risque, celle-ci ne peut être pratiquée sur un sujet sain.

Les images issues de l'IRM morphologique, comme dans le cas pathologique, permettent de définir les paramètres géométriques du maillage, Fig. 6.16.

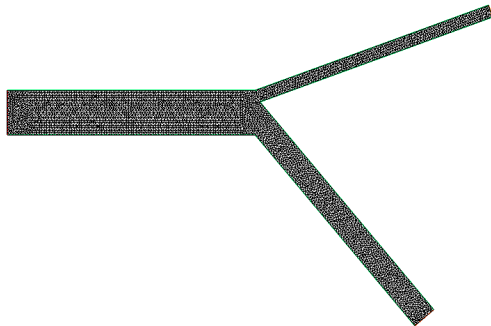


FIGURE 6.16 – **Cas sain.** Maillage.

TABLE 6.3 – **Cas sain.** Dimensions.

Paramètre	Valeur
L_1	100 mm
L_2	100 mm
L_3	100 mm
D_1	18 mm
D_2	10 mm
D_3	5 mm
α_2	50 °
α_3	20 °

Les mesures par IRM de flux nous permettent d'imposer la vitesse d'entrée au niveau des ESA spinaux. Les nombres de Reynolds et de Strouhal sont calculés :

$$Re = 23.2084 ; St = 4.09637 \quad (6.10)$$

Remarque 6.5. De même que pour le cas pathologique, un algorithme de Stokes instationnaire aurait suffi.

Contrairement au cas pathologique, on ne peut pas évaluer la compliance globale du système à l'aide de la pression mesurée de manière invasive. On la fixe arbitrairement à $C_G = 4 \text{ mm}^3 \cdot \text{Pa}^{-1}$ afin d'obtenir un ordre de grandeur physiologique pour l'amplitude de pression.

On répartit la compliance globale entre les ventricules cérébraux et les ESA cérébraux de sorte à obtenir des distributions cohérentes avec les mesures par IRM de flux ; on obtient $C_{Vent} = \frac{1}{20}C_G$ et $C_{ESA} = \frac{19}{20}C_G$.

Les résultats de simulation sont détaillés Fig. 6.17.

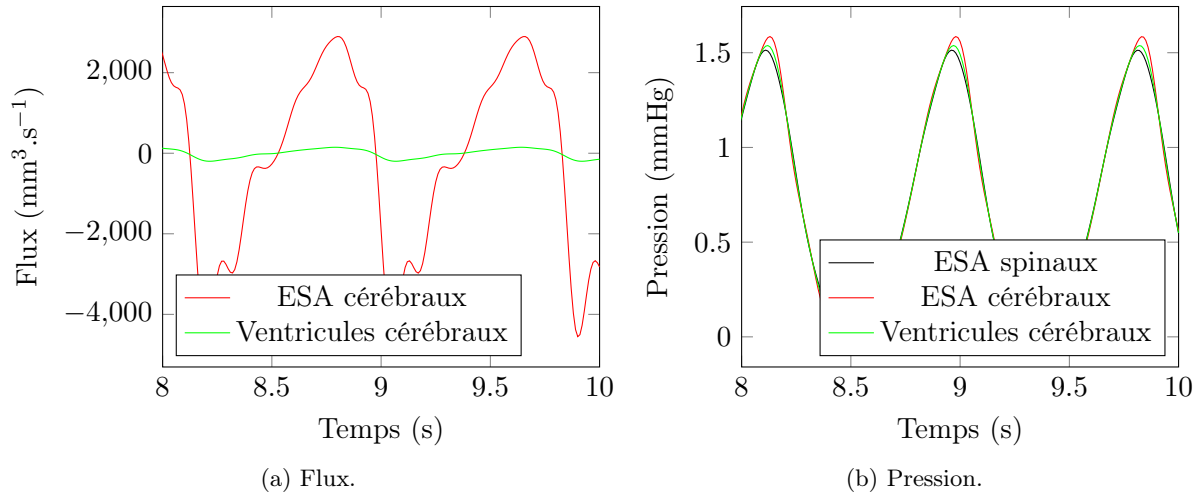


FIGURE 6.17 – Cas sain. Flux et pressions.

La courbe de pression obtenue, même si elle ne peut être comparée à des mesures, a une amplitude dans un ordre de grandeur physiologique.

Grâce aux données mesurées par IRM de flux, on peut comparer nos résultats de simulation aux mesures de flux, Fig. 6.18.

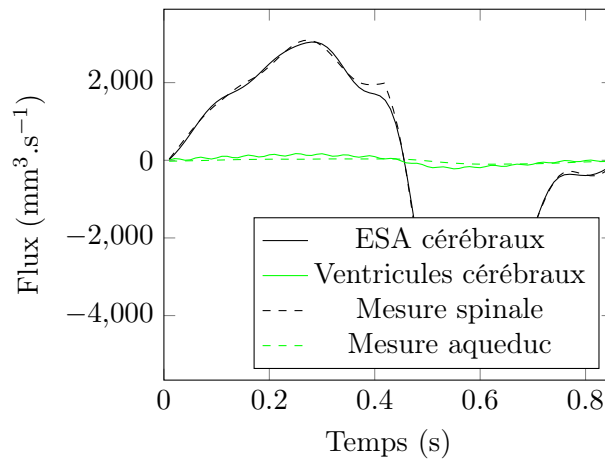


FIGURE 6.18 – Cas sain. Comparaison des résultats et des mesures.

On constate que les courbes de flux sont équivalentes. Comme pour le cas pathologique, la différence entre le flux des ESA spinaux simulé et mesuré provient de l'approximation par une décomposition de Fourier, voir Chap. 12. Les courbes de flux simulées et mesurées sont très proches.

Remarque 6.6. Ici, nous n'avons aucune indication sur la pression à obtenir. Celle-ci se trouve tout de même dans un intervalle physiologique : l'amplitude est comprise entre 1 et 3 mmHg [2, 59], mais elle ne présente pas de pics et de vallées [17].

6.2.7 Conclusion

Ce premier modèle, extrêmement simplifié, permet de mettre en évidence le rôle de certains paramètres dans la distribution du LCS entre les compartiments ventriculaires et subarachnoïdiens cérébraux, notamment le paramètre de la fréquence cardiaque.

La compliance de chaque compartiment joue également un rôle dans la distribution du LCS, mais ce paramètre est moins variable que la fréquence cardiaque *in vivo* et est directement en relation avec la condition physiologique du patient / témoin.

L'impact sur la pression moyenne et sur le gradient de pression est également décrit. Dans tous les cas, on constate que le gradient de pression est toujours plus élevé que la pression moyenne.

Les premiers tests sur des cas réels, pathologiques et sains, montrent une bonne adéquation des flux dans les différents compartiments, mais une mauvaise représentation de la pression intracrânienne.

On peut tout de même constater que la répartition des compliances est totalement différente dans le cas pathologique et dans le cas sain. En effet, il existe un taux de la compliance des ESA cérébraux sur la compliance des ventricules cérébraux de 3 dans le cas pathologique et de 20 dans le cas sain.

La valeur de la compliance globale est également différente dans le cas pathologique et dans le cas sain ; on a une compliance globale plus de 3 fois plus grande dans le cas sain par rapport au cas pathologique.

Il est d'ores et déjà possible de suspecter un rôle de la compliance du système cérébrospinal dans la mise en place de certaines pathologies.

Les modèles de Windkessel doivent donc être améliorés pour ajuster le comportement de notre modèle au comportement physiologique du système cérébrospinal. Nous allons donc désormais observer le comportement du système avec les modèles de Windkessel variables.

6.3 Résultats – Windkessel variable

6.3.1 Paramètres

Dans les premiers tests concernant l'influence des différents paramètres physiologiques du système cérébrospinal sur la distribution de LCS, les mêmes paramètres que dans la section «Windkessel constant», détaillés Tab. 6.1, seront utilisés, ainsi que les paramètres supplémentaires détaillés Tab. 6.4.

TABLE 6.4 – **Paramètres de simulation.** Étude de l'impact des paramètres physiologiques.

Paramètre	Symbole	Valeur	Unité
Compliance initiale ESA cérébraux	C_{ESA}^0	1	$\text{mm}^3 \cdot \text{Pa}^{-1}$
Compliance initiale ventricules cérébraux	C_{Vent}^0	2	$\text{mm}^3 \cdot \text{Pa}^{-1}$
Pression initiale	P_0	1	Pa
Constante élastique	k	$5e^{-2}$	$\text{mm}^{-3} \cdot \text{Pa}$

Le volume sanguin sera défini par :

$$V_{AV}(t) = \frac{1}{2} \left(1 - \sin \left(\frac{2\pi t}{p} \right) \right) \quad (6.11)$$

Le maillage considéré est le même que pour la partie «Windkessel constant», Fig. 6.5.

Dans la section traitant des cas réels, les paramètres, ainsi que les maillages, seront identiques à la partie «Windkessel constant».

On utilisera, comme dans la section précédente, l'algorithme d'Uzawa avec le temps total de simulation et le pas de temps précédemment défini.

6.3.2 Impact de l'amplitude

Afin d'étudier l'impact de l'amplitude du flux d'entrée, on prend en compte les paramètres des tableaux 6.1 et 6.4 et on fait varier l'amplitude d'entrée dans l'intervalle $[1, 50]$, qui couvre une large gamme de valeurs physiologiques observées sur témoins ou sur patients.

Les résultats de simulation sont illustrés Fig. 6.19.

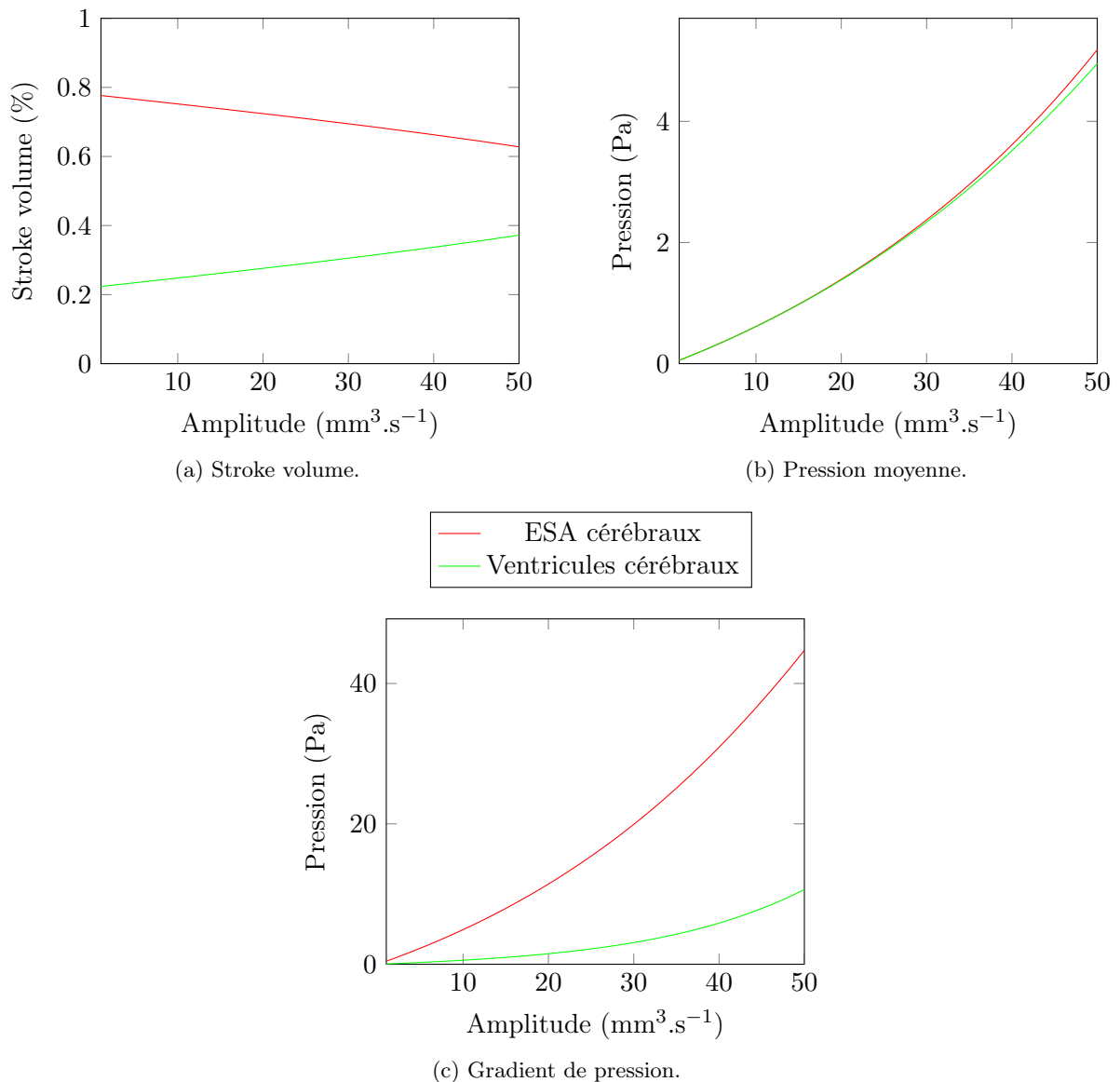


FIGURE 6.19 – **Windkessel variable.** Effet de l'amplitude.

Contrairement au cas du modèle «Windkessel constant», ici, l'amplitude du flux d'entrée a un impact sur la distribution du LCS entre les différents compartiments du système cérébrospinal. Plus le flux d'entrée est élevé, plus la distribution des flux tend à s'équilibrer.

La pression moyenne croît maintenant de manière exponentielle avec l'amplitude ; ceci est dû au caractère variable des compliances.

Le gradient de pression croît également de manière exponentielle avec une valeur plus élevée dans les ESA cérébraux, où le stroke volume est plus important.

6.3.3 Impact du cycle cardiaque

On fait varier ici la période du flux d'entrée dans l'intervalle $[0.2, 2]$ qui correspond à une variation de la fréquence cardiaque de 30 à 300 bpm, en conservant le stroke volume, comme dans le cas «Windkessel constant». Les autres paramètres sont ceux détaillées Tab. 6.1 et Tab. 6.4.

Les résultats de simulation exposés Fig. 6.20 mettent en évidence un très grand impact du cycle cardiaque sur la distribution du LCS entre les ESA cérébraux et les ventricules cérébraux, comme dans le cas «Windkessel constant».

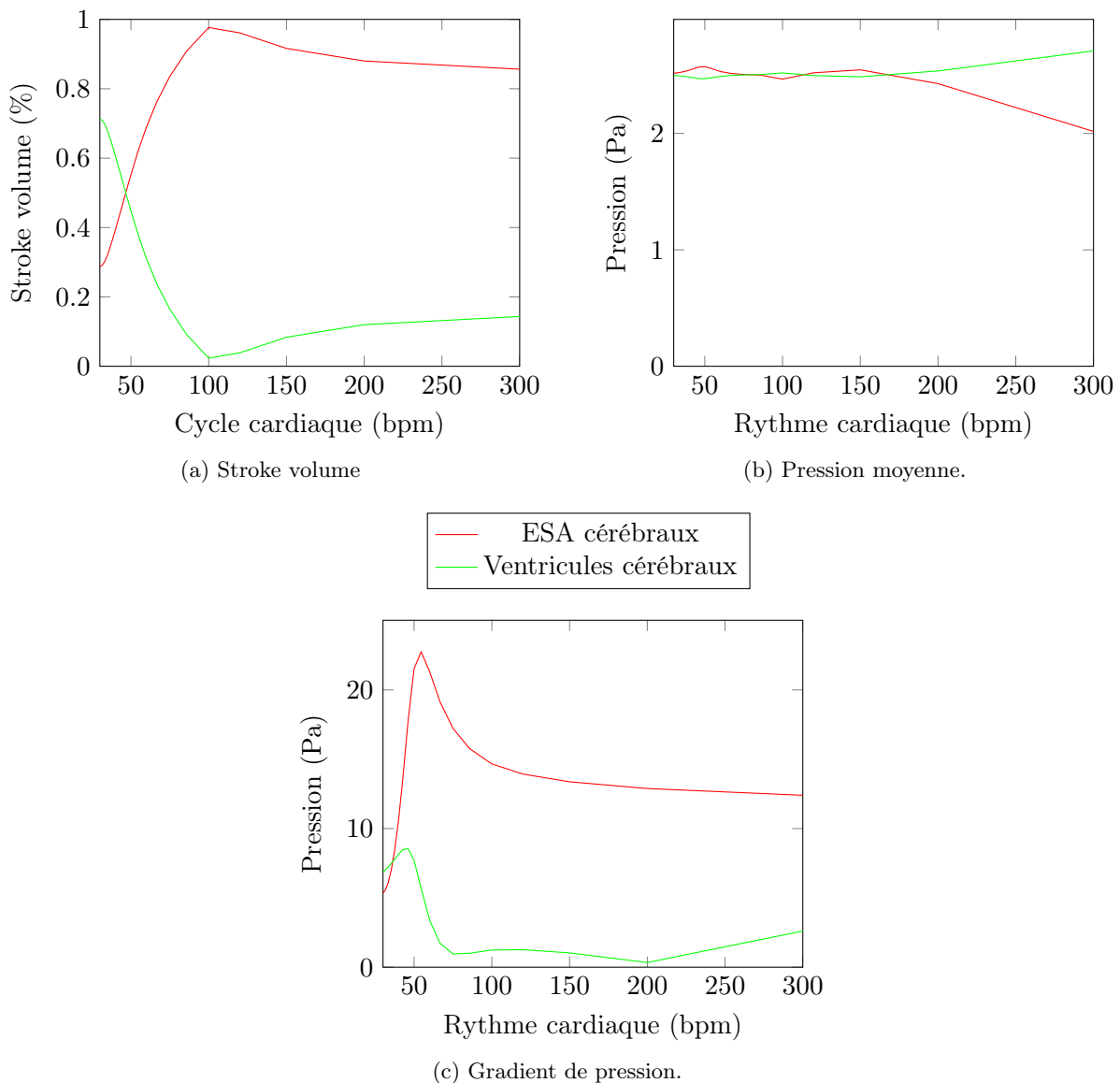


FIGURE 6.20 – **Windkessel variable.** Effet de la période avec conservation du stroke volume.

Les cycles cardiaques lents (au repos, par exemple) privilégient les ventricules cérébraux alors que les cycles cardiaques rapides (à l'exercice, par exemple) privilégient les ESA cérébraux.

La pression moyenne se comporte de la même façon que dans le cas «Windkessel constant», avec une valeur un peu plus élevée. Une séparation plus franche entre les ventricules cérébraux et les ESA cérébraux est observée.

Le gradient de pression est également équivalent à celui obtenu avec le précédent modèle avec lui aussi une valeur un peu plus élevée.

On a donc un comportement similaire aux cas du modèle «Windkessel constant».

Remarque 6.7. *Des résultats similaires sont obtenus en ne conservant pas le stroke volume. Seule la pression moyenne diffère et présente une décroissance exponentielle.*

6.3.4 Impact de la compliance

Comme dans la section précédente, on modifie la valeur de la compliance des ventricules cérébraux dans l'intervalle $[0.1, 5]$, en conservant les autres paramètres détaillés Tab. 6.1 et Tab. 6.4.

Les résultats de simulation sont exposés Fig. 6.21. On y observe un comportement assez particulier du système. Dans un premier temps, plus la compliance des ventricules cérébraux augmente, plus le stroke volume augmente dans ce compartiment, mais ce comportement s'inverse à partir de la valeur $C_{Vent} = 0.5$ (équivalent au cas «Windkessel constant»).

La pression moyenne décroît exponentiellement avec l'augmentation de la compliance globale, comme dans le modèle «Windkessel constant».

Ce comportement, qui n'est pas celui attendu, est dû à un volume entrant non suffisant à l'intérieur des compliances, comme dans le cas «Windkessel constant», que nous allons mettre en évidence par un second test avec une période différente.

On réalise de nouveau le test de l'influence des compliances, en prenant cette fois-ci une période plus grande, de 2 s, afin de solliciter d'avantage les compliances et d'en observer plus précisément l'impact. Les résultats de simulation sont exposés Fig. 6.22.

On a bien une distribution du LCS entre les compartiments cérébraux dirigée par les compliances.

La pression moyenne décroît également exponentiellement dans ce cas, même si la valeur est plus élevée, ce qui est dû à un plus grand volume entrant dans les compliances (période plus importante).

Le gradient de pression présente la même forme que dans le cas «Windkessel constant».

6.3.5 Impact de la constante élastique

Finalement on modifie la constante élastique dans l'intervalle $[1e^{-3}, 1]$, en conservant les autres paramètres détaillés Tab. 6.1 et Tab. 6.4.

Les résultats de simulation sont exposés Fig. 6.23.

La distribution de LCS entre les deux compartiments cérébraux est impactée par la constante élastique. Un comportement physiologique est observé pour de faibles valeurs, puis la distribution tend à s'équilibrer en s'inversant plusieurs fois.

La pression moyenne croît de façon exponentielle avec l'augmentation de la constante élastique. Ceci traduit la contrainte de plus en plus élevée imposée aux compliances.

Le gradient de pression augmente également de façon exponentielle, avec une vallée pour la courbe de gradient de pression dans les ESA cérébraux qui suit la courbe de stroke volume.

6.3.6 Cas réels

Comme dans le cas des compliances constantes, on utilisera les acquisitions IRM, et éventuellement les mesures de pression intracrânienne, pour déterminer les paramètres nécessaires.

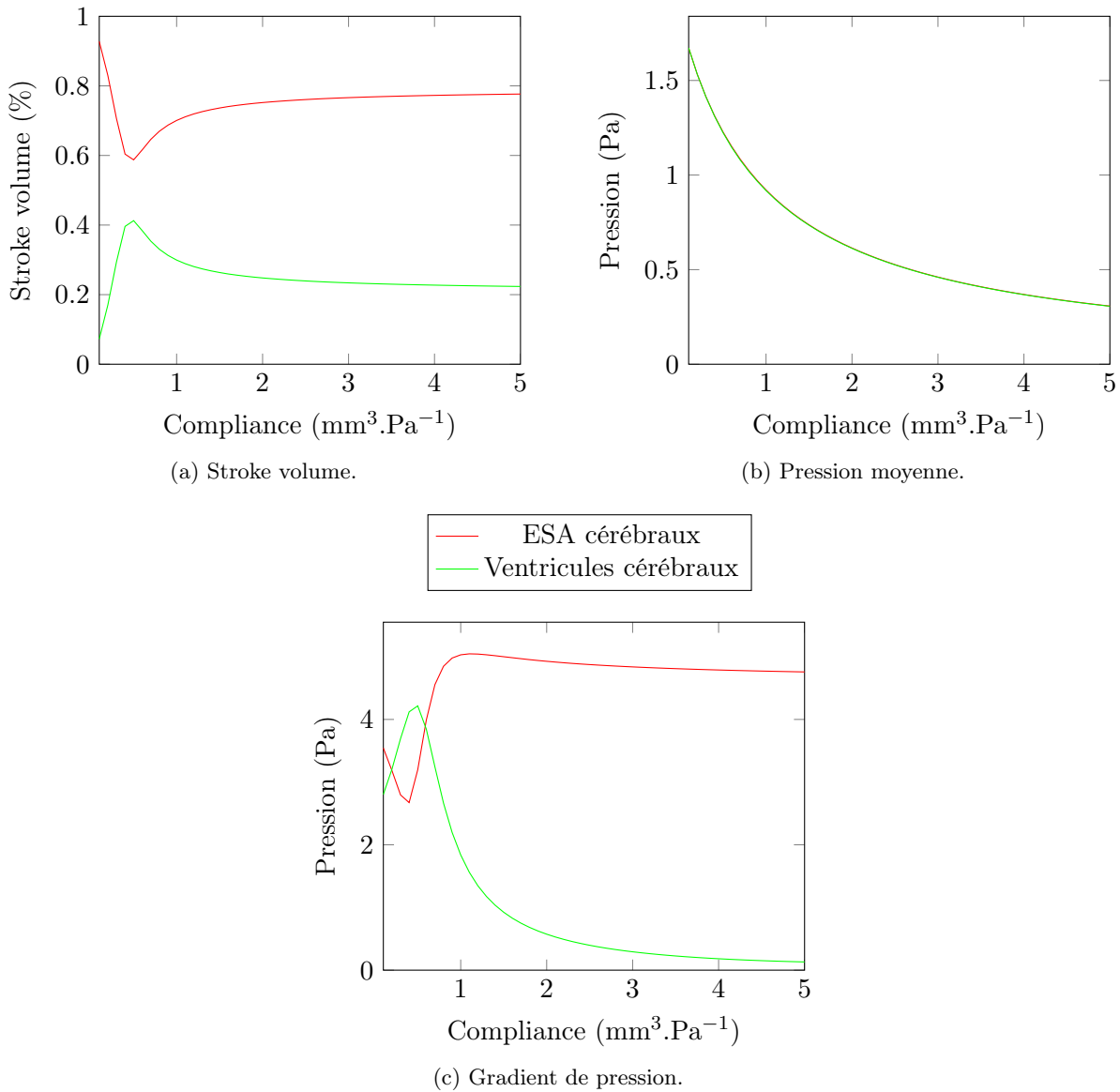


FIGURE 6.21 – **Windkessel variable.** Effet de la compliance.

Cas pathologique

On reprend le même maillage et les mêmes paramètres que dans le cas «Windkessel constant». Il y a deux paramètres supplémentaires du fait des compliances variables : P_0 et k .

Pour la pression initiale P_0 , on choisit la valeur physiologique de la pression intracrânienne, soit $P_0 = 12$ mmHg.

Le paramètre k est déterminé pour obtenir une amplitude de pression intracrânienne proche de celle mesurée. On a ainsi $k = \frac{2.6}{C_{Vent}^0 + C_{ESA}^0}$.

Les résultats de simulation sont détaillés Fig. 6.24.

Grâce aux données mesurées, nous pouvons comparer nos résultats de simulation aux mesures de flux et de pression, Fig. 6.25.

On observe que les courbes de flux sont équivalentes en amplitude. Le flux des ESA spinaux correspond à notre condition d'entrée ; elle diffère uniquement à cause de l'approximation par une série de Fourier, voir Chap. 12. En revanche, le flux dans les ventricules cérébraux présente

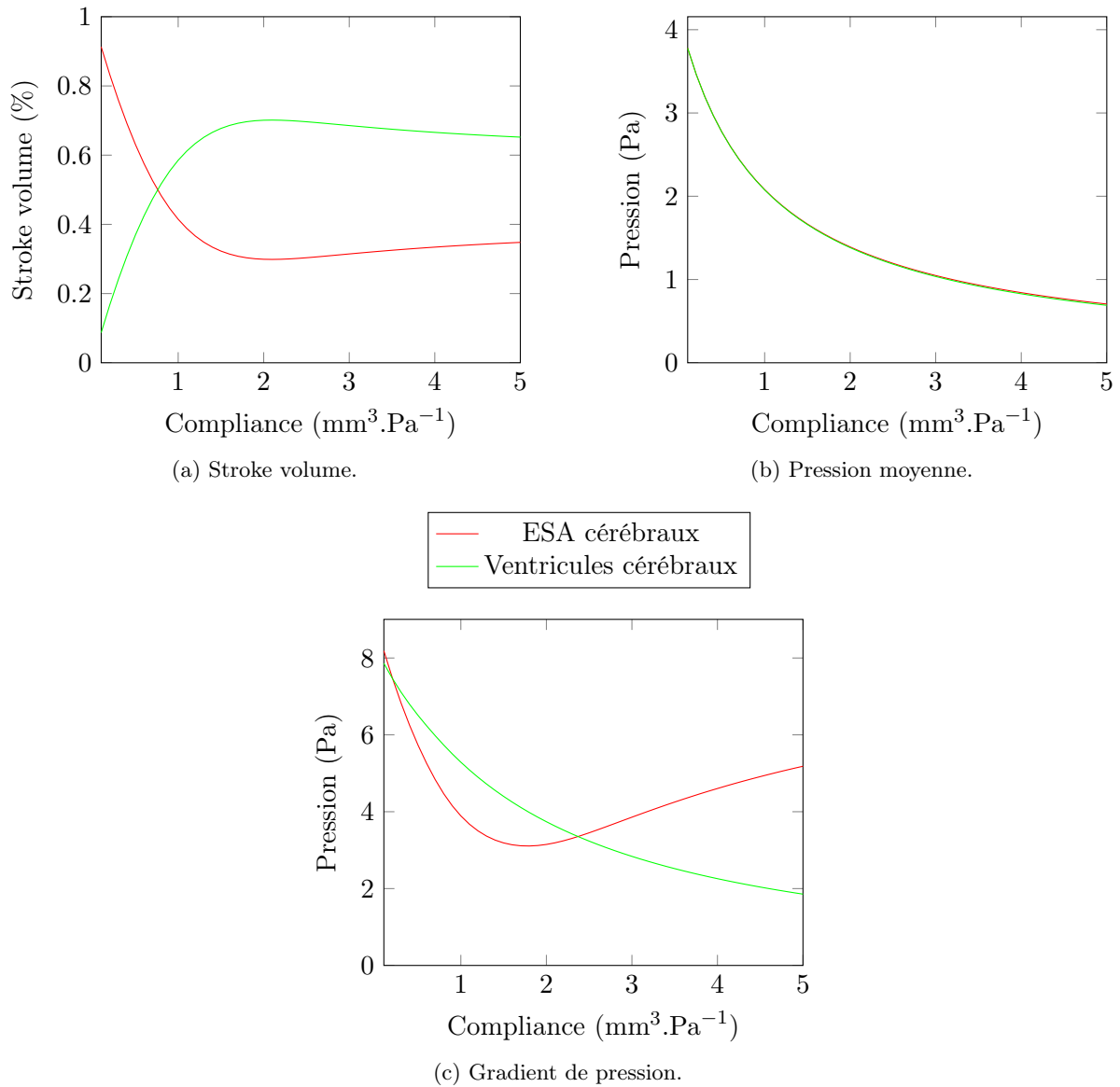


FIGURE 6.22 – **Windkessel variable.** Effet de la compliance (période de 2 s).

un déphasage avec le flux des ESA spinaux dans les mesures que l'on ne retrouve toujours pas dans les simulations.

La courbe de pression simulée est plus proche de la courbe de pression mesurée dans ce cas (contrairement au cas «Windkessel constant»). L'amplitude est similaire ; en revanche, la forme de courbe est assez différente.

Remarque 6.8. *Le paramètre le plus important dans la courbe de pression est l'amplitude, c'est ce paramètre qui peut nous permettre de déterminer la pression absolue et donc de définir correctement la pression intracrânienne réelle d'après les études de N. Alperin et al. [3, 4]. La forme de la courbe n'a que peu d'importance pour la détermination de la PIC absolue.*

Cas sain

Nous reprenons le même maillage et les mêmes paramètres que dans le cas «Windkessel constant». Les paramètres P_0 et k sont pris identiques au cas pathologique.

Les résultats de simulation sont détaillés Fig. 6.26.

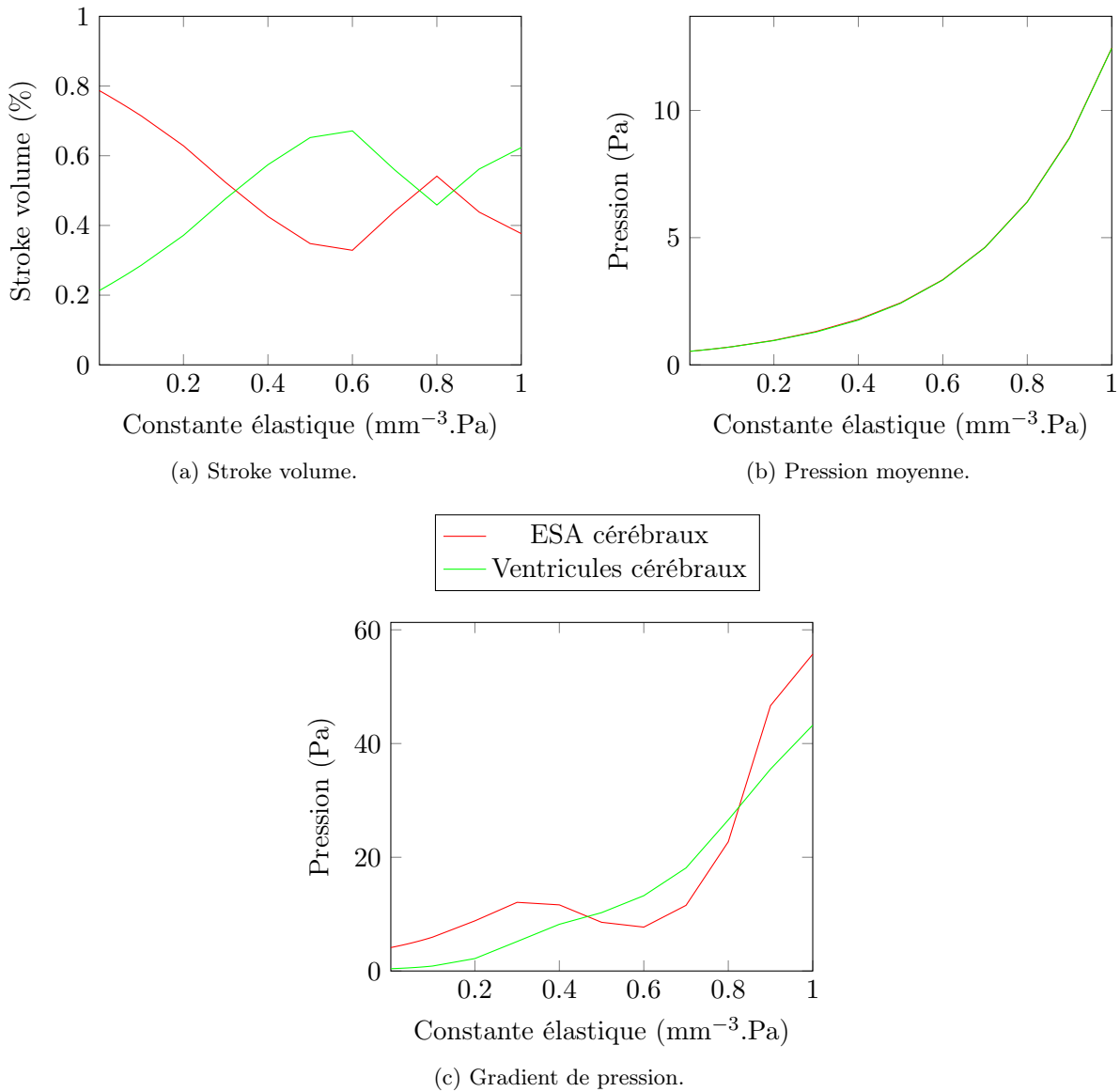


FIGURE 6.23 – **Windkessel variable.** Effet de la constante élastique.

La courbe de pression présente de nouveau une amplitude dans un ordre de grandeur physiologique.

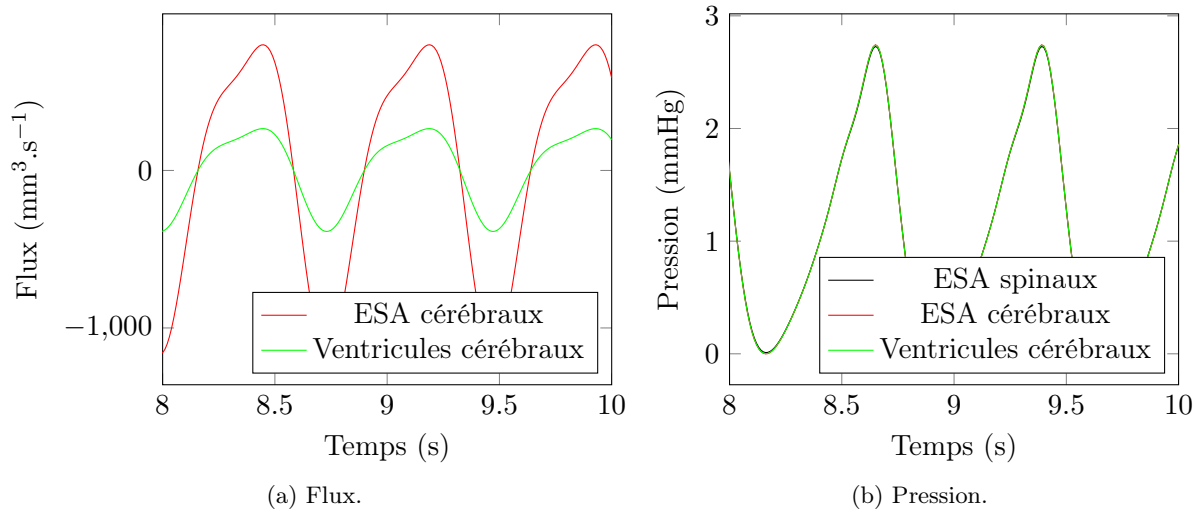
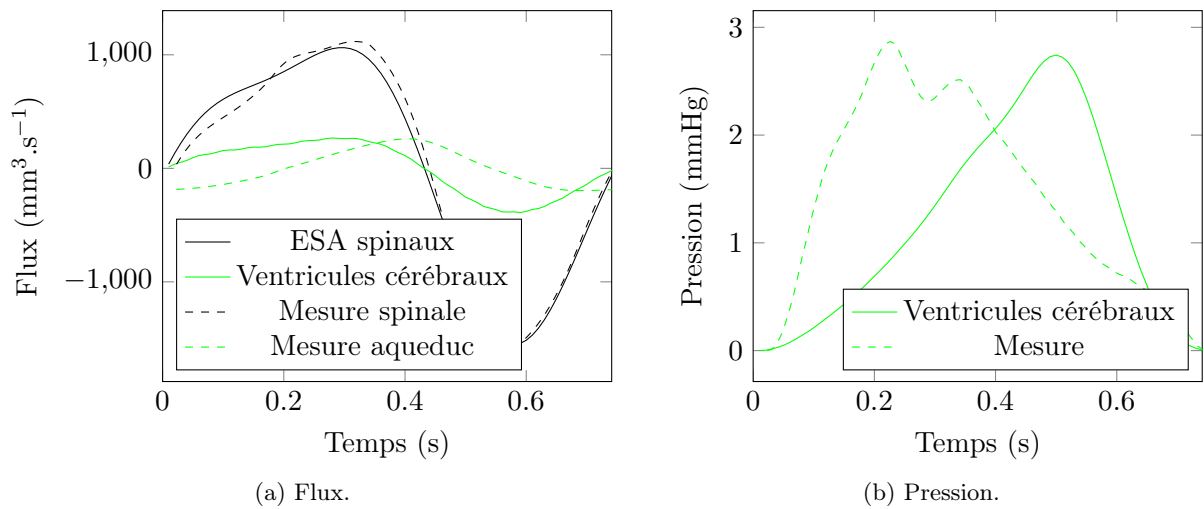
Grâce aux données mesurées par IRM de flux, nous pouvons comparer nos résultats de simulation aux mesures de flux, Fig. 6.27.

On constate que les courbes sont équivalentes. Comme pour le cas pathologique, la différence entre le flux des ESA spinaux simulé et mesuré repose sur l'approximation par une décomposition de Fourier, voir Chap. 12. Les courbes de flux simulées et mesurées sont très proches.

6.3.7 Conclusion

Contrairement au cas «Windkessel constant», ce modèle à compliances variables est plus en adéquation avec le comportement physiologique du système cérébrospinal dans l'étude des cas réels. Les pressions obtenues sont plus proches des mesures et/ou dans un intervalle physiologique.

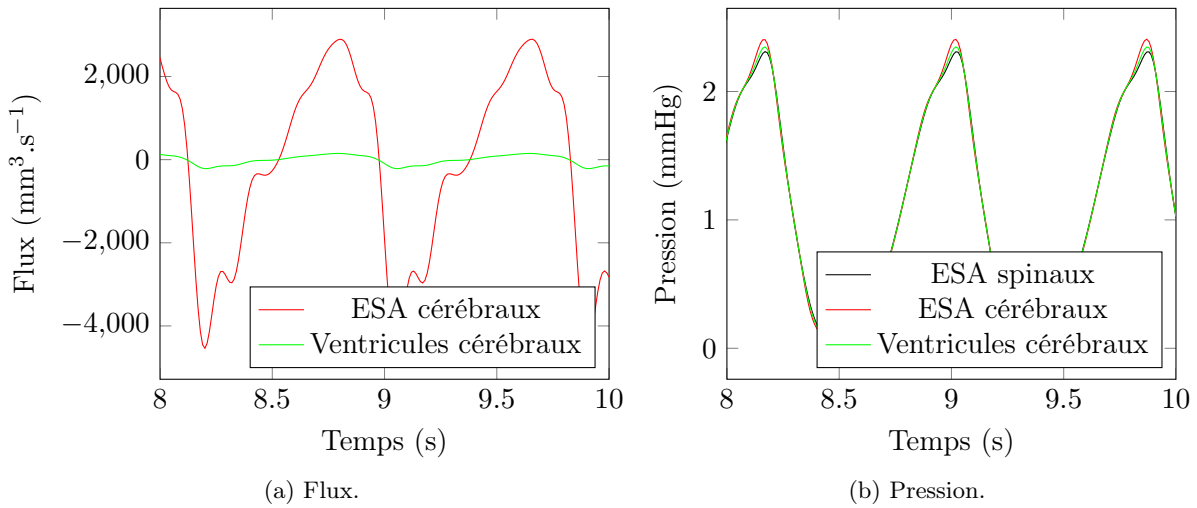
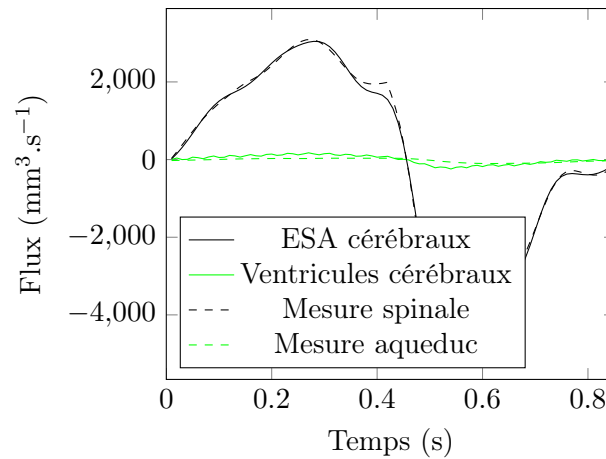
Malgré l'amélioration des modèles de Windkessel, quelques différences subsistent entre les mesures de flux et de pression et les résultats de simulation.


 FIGURE 6.24 – **Cas pathologique.** Flux et pressions.

 FIGURE 6.25 – **Cas pathologique.** Comparaison des résultats et des mesures.

Le principal point faible de ce modèle est le choix des paramètres P_0 et k , notamment dans l'étude de l'impact des paramètres, qui conditionnent le comportement du système.

6.4 Conclusion

Ce modèle très simplifié permet de reproduire la plupart des comportements attendus du système cérébrospinal. Il permet par ailleurs de déterminer l'impact d'un seul paramètre sur l'ensemble du système. En effet, il est très compliqué *in vivo* de faire varier, par exemple, uniquement la fréquence cardiaque d'un témoin / patient et d'en observer l'impact sur le système. En effet, le système s'adapte avec le temps aux contraintes qu'il subit et des paramètres tels que le diamètre de l'aqueduc, l'épaisseur des ESA cérébraux, la compliance ou le volume des ventricules cérébraux sont modifiés [18]. De plus, l'obtention d'un accord clinique pour une étude visant à modifier le rythme cardiaque d'un patient dans une IRM sans que le système ait le temps de s'adapter est difficile à obtenir (injection de drogues, ...)

FIGURE 6.26 – **Cas sain.** Flux et pressions.FIGURE 6.27 – **Cas sain.** Comparaison des résultats et des mesures.

Dans un comportement physiologique, les paramètres sont liés et doivent être modifiés en fonction de l'amplitude du flux d'entrée, du rythme cardiaque ou des autres paramètres du système cérébrospinal. Le choix des paramètres de couplage (constante élastique k et pression initiale P_0) entre les différents compartiments a donc un très grand impact dans le comportement du système, ceux-ci doivent donc être choisis avec précaution sous peine d'obtenir un comportement non représentatif de la réalité.

Un impact très important du paramètre de fréquence cardiaque est constaté, la distribution peut se retrouver inversée selon le rythme cardiaque du patient / témoin. Un impact des compliances est également mis en évidence, même si celles-ci ne varient pas rapidement dans des conditions physiologiques; une variation de la compliance du système est due, par exemple, à une modification du tissu cérébral en cas de troubles pathologiques ou encore de chocs.

Dans la simulation des cas réels, on observe que la compliance globale du système est très inférieure dans le cas pathologique, ce qui peut indiquer un désordre d'ordre structurel dans la mise en place de certaines pathologies. De plus, la répartition en les compliances au niveau des ventricules cérébraux et des ESA cérébraux est également très différente dans les cas sains et pathologiques. Les ventricules cérébraux ont une compliance plus élevée dans le cas pathologique étudié, par rapport à la compliance des ESA cérébraux. On peut donc suggérer une perte de

compliance au niveau des ESA cérébraux dans l'altération de l'hydrodynamique cérébrale de ce patient.

Le principal point faible de ce modèle est la détermination des paramètres en jeux dans les modèles de Windkessel. Ceux ci sont difficilement identifiables, même à l'aide de techniques invasives, telles que le monitoring de pression intracrânienne qui nous donne une information sur la compliance globale du système et non sur la répartition des compliances entre compartiments.

Une avancée vers un modèle plus réaliste semble indispensable pour parvenir à approcher numériquement la pression intracrânienne, prenant en compte une géométrie plus réaliste et les interactions complexes entre le liquide cébrospinal et les structures environnantes telles que le cerveau.

Chapitre 7

Second modèle : Système cérébrospinal simplifié

Ce modèle va nous permettre de mieux modéliser les interactions entre le compartiment de liquide cérébrospinal (LCS) et les structures environnantes (principalement le cerveau).

La construction du second modèle, le système cérébrospinal simplifié, les différents paramètres spécifiques à notre problème, ainsi que de possibles améliorations, sont détaillés dans ce chapitre. Les résultats obtenus, en parties à l'aide du calculateur MeCS [74], sont également présentés.

7.1 Modèle

Afin de représenter le système cérébrospinal de manière plus réaliste, on le modélise désormais comme présenté Fig. 7.1.

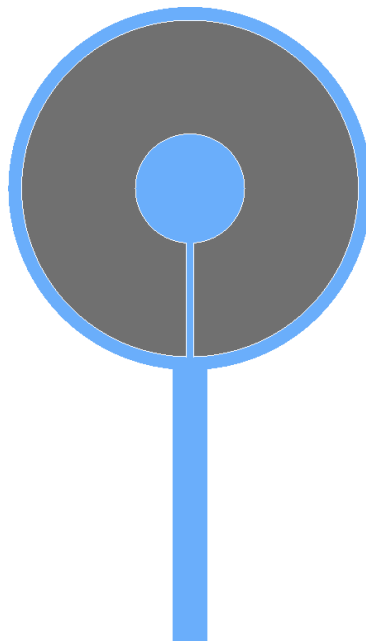


FIGURE 7.1 – Modèle cérébrospinal simplifié.

La partie bleue représente le compartiment de liquide cérébrospinal (LCS) et la partie grise représente l'encéphale (ici le cerveau).

Comme dans le modèle précédent, l'entrée de LCS est imposée au niveau des espaces subarachnoïdiens (ESA) spinaux. Le crâne et le canal spinal sont considérés comme rigides ; l'aqueduc

est considéré comme fixe. L'interaction fluide-structure prend place au niveau du cerveau, à l'interface des ventricules cérébraux et des ESA cérébraux.

Remarque 7.1. *Le problème est en quelque sorte pris «à l'envers» dans ce modèle, comme dans le modèle précédent, puisque dans la réalité, c'est l'afflux sanguin qui provoque la déformation du cerveau qui, à son tour, met en mouvement le LCS. Ici, on impose le mouvement de LCS pour provoquer la déformation du cerveau.*

7.1.1 Première amélioration

Une première amélioration simple du système peut être mise en place en différenciant la matière blanche et la matière grise, Fig. 7.2.

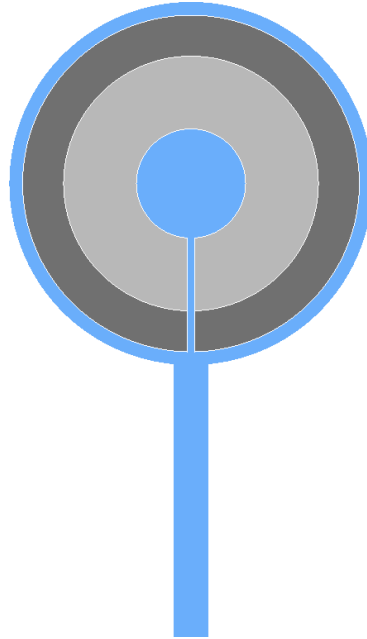


FIGURE 7.2 – **Modèle cérébrospinal simplifié.** Différenciation matière grise / matière blanche.

La partie gris clair représente la matière blanche, la partie gris foncé représente la matière grise. Des paramètres différents (module d'Young et coefficient de Poisson) sont choisis selon la localisation de la structure.

7.1.2 Deuxième amélioration

Finalement, on peut ajouter le compartiment sanguin à notre modèle pour obtenir un comportement plus proche de la réalité, Fig. 7.3.

La différenciation matière grise / matière blanche est également présente dans ce modèle (respectivement de couleur gris foncé, gris clair). On ajoute la partie rouge représentant l'ensemble du système artério-veineux du système cérébrospinal. Les deux disques en bas représentent les arrivées artérielles et le disque en haut représente la sortie veineuse.

Plusieurs configurations sont possibles grâce aux données acquises par IRM de flux :

- les entrées artérielles, la sortie veineuse et l'entrée de LCS au niveau des ESA spinaux sont toutes les trois imposées par des mesures par IRM de flux ;
- seules les entrées artérielles et la sortie veineuse sont imposées. Le LCS est libre de circuler selon la déformation du cerveau : la sortie est libre ou avec un modèle de Windkessel ;
- seules les entrées artérielles sont imposées. La sortie veineuse et l'entrée de LCS sont libres ou avec des modèles de Windkessel.

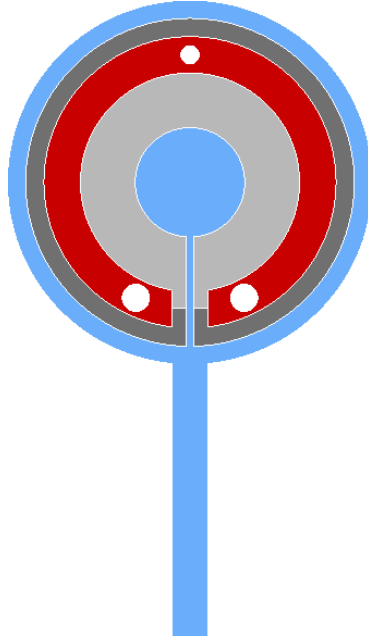


FIGURE 7.3 – **Modèle cérébrospinal simplifié.** Ajout du compartiment sanguin.

7.1.3 Construction géométrique

Les différents paramètres géométriques pour la création du maillage du système cérébrospinal simplifié sont détaillées Fig. 7.4.

R_V est le rayon des ventricules cérébraux, R_{ESA} est le rayon des ESA cérébraux, R_C est le rayon du crâne. D_S est le diamètre du canal spinal, D_A est le diamètre de l'aqueduc.

Pour le modèle avec compartiment sanguin, il convient d'ajouter $R_{AV_{ext}}$ et $R_{AV_{int}}$, respectivement les rayons du bord extérieur et du bord intérieur du compartiment sanguin, D_{Art} le diamètre des entrées artérielles, et D_V le diamètre de la sortie veineuse.

7.2 Résultats

7.2.1 Paramètres

Dans les premiers tests concernant l'influence des paramètres physiologiques du système cérébrospinal, les paramètres détaillés, Tab. 7.1, seront pris en compte.

Le premier modèle, sans améliorations, est utilisé.

Le maillage considéré, Fig. 7.5, prend en compte les paramètres physiologiques mesurés sur l'IRM morphologique d'un sujet sain. Le pas du maillage est de $\frac{1}{3}$ cm.

Le flux d'entrée, au niveau des ESA spinaux, est pris égal à :

$$F(t) = A \sin\left(\frac{2\pi t}{p}\right) \quad (7.1)$$

avec A l'amplitude et p la période.

Les résultats présentés sont obtenus avec l'algorithme d'interaction fluide-structure FREEFEM++, Sec. 4.4.1, pour un temps total de simulation $T = 20$ s et un pas de temps $\Delta t = \frac{p}{100}$ s.

Dans la section traitant des cas réels, les paramètres, ainsi que les maillages, seront spécifiques à la géométrie observée par IRM morphologique et aux vitesses mesurées par IRM de flux.

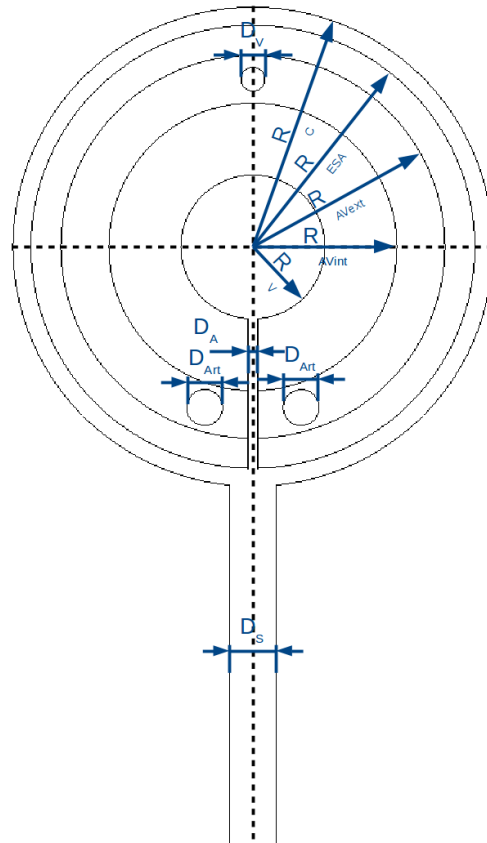


FIGURE 7.4 – **Modèle cérébrospinal simplifié.** Paramètres géométriques.

TABLE 7.1 – **Paramètres de simulation.** Impact des paramètres.

Paramètres	Symbole	Valeur	Unité
Fluide			
Masse volumique	ρ^F	1	g.cm^{-3}
Viscosité dynamique	μ^F	0.01	$\text{g.cm}^{-1}.\text{s}^{-1}$
Amplitude du flux d'entrée	A	$1e^{-2}$	$\text{cm}^3.\text{s}$
Période du flux d'entrée	p	1	s
Solide			
Masse volumique	ρ^S	1.1	g.cm^{-3}
Module d'Young	E^S	$5.e^3$	$\text{g.cm}^{-1}.\text{s}^{-2}$
Coefficient de Poisson	ν^S	0.35	–

Les résultats présentés dans les cas réels sont obtenus avec l'algorithme d'interaction fluide-structure FREEFEM++, Sec. 4.4.1, avec un temps total de simulation $T = 50$ s et un pas de temps $\Delta t = 1e^{-2}$ s.

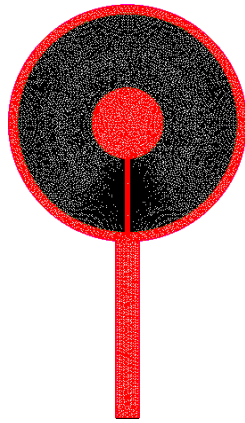


FIGURE 7.5 – Modèle cérébrospinal simplifié. Maillage.

TABLE 7.2 – Impact des paramètres. Dimensions.

Paramètre	Valeur
R_C	10 cm
R_{ESA}	9.25 cm
R_V	3 cm
D_A	0.20 cm
D_S	1 cm

7.2.2 Impact de l'amplitude

Afin d'étudier l'impact de l'amplitude du flux d'entrée sur la distribution de LCS, on prend en compte les paramètres détaillés Tab. 7.1 et on fait varier l'amplitude d'entrée dans l'intervalle $[1e^{-3}, 5e^{-2}]$, qui correspond à l'étude réalisée sur le modèle «bifurcation».

Les résultats de simulation sont illustrés Fig. 7.6.

Comme avec le modèle «bifurcation» (et les modèles de Winkdessel constants), l'amplitude du flux d'entrée de LCS n'a que peu d'impact sur la distribution du LCS entre les ventricules cérébraux et les ESA cérébraux. La répartition des flux est également dans l'ordre de grandeur physiologique, avec une répartition de 10% du stroke volume pour les ventricules cérébraux et du 90% pour les ESA cérébraux.

La pression augmente quasi-linéairement avec l'augmentation de l'amplitude.

Le gradient de pression augmente également de façon linéaire avec une pression plus élevée dans les ESA cérébraux, là où le stroke volume est le plus important.

7.2.3 Impact du cycle cardiaque

On modifie ici la période du flux d'entrée dans l'intervalle $[0.2, 2]$ qui correspond à la gamme physiologique et à l'étude réalisée sur le modèle «bifurcation», en conservant le stroke volume. Les autres paramètres sont ceux détaillés Tab. 7.1.

Les résultats de simulation sont exposés Fig. 7.7.

On voit que la période a un impact sur la distribution du LCS entre les ventricules cérébraux et les ESA cérébraux, même si celui-ci est moins prononcé que pour le modèle «bifurcation».

Les rythmes cardiaques lents délivrent une plus grande partie du stroke volume aux ventricules cérébraux avant de revenir dans un état physiologique lorsque le rythme cardiaque augmente (10% pour les ventricules cérébraux, 90% pour les ESA cérébraux).

La pression moyenne tend à diminuer avec l'augmentation du rythme cardiaque.

Le gradient de pression croît de manière exponentiel avec l'augmentation du rythme cardiaque.

Remarque 7.2. *Des résultats similaires sont obtenus en ne conservant pas le stroke volume.*

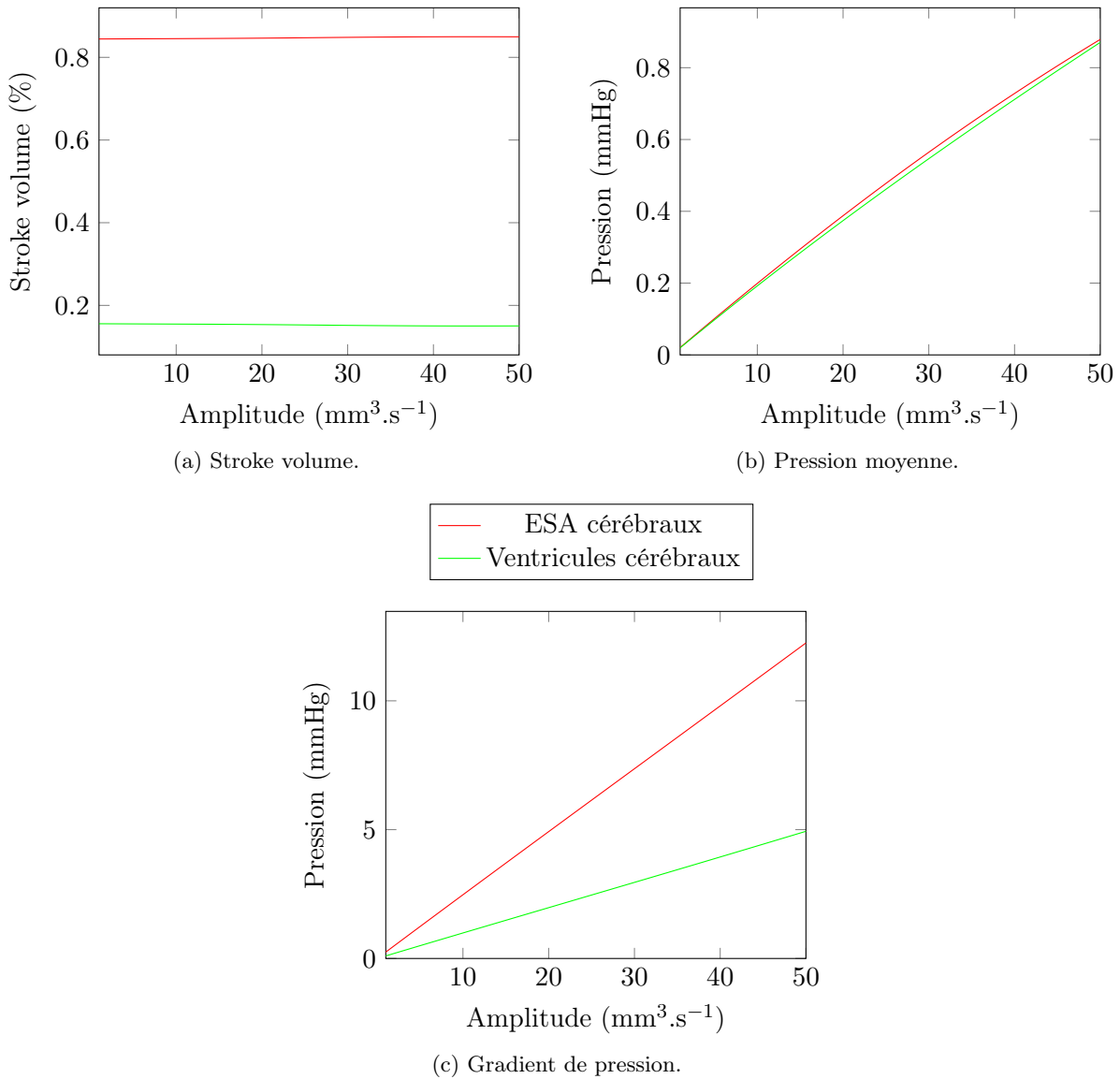


FIGURE 7.6 – **Impact des paramètres.** Effet de l'amplitude.

7.2.4 Impact du module d'Young

On fait varier la valeur du module d'Young de la structure dans l'intervalle $[100, 100000]$, en conservant les autres paramètres détaillés Tab. 7.1.

Les résultats de simulation sont exposés Fig. 7.8.

Le module d'Young a un impact sur la distribution du LCS dans les valeurs proches de la valeur physiologique (1000–10000). Passées ces valeurs, un phénomène de plateau s'installe et le module d'Young n'a plus qu'un impact très limité sur la distribution du LCS entre les différents compartiments cérébraux du système cérébrospinal.

La pression augmente quasi-linéairement avec l'augmentation du module d'Young. En effet, plus le module d'Young est élevé, plus le matériau est difficilement déformable.

Le gradient de pression est très variable dans les premières valeurs du module d'Young, tout comme le stroke volume. Un pic se produit aux alentours de 40000, puis le gradient de pression tend à se stabiliser.

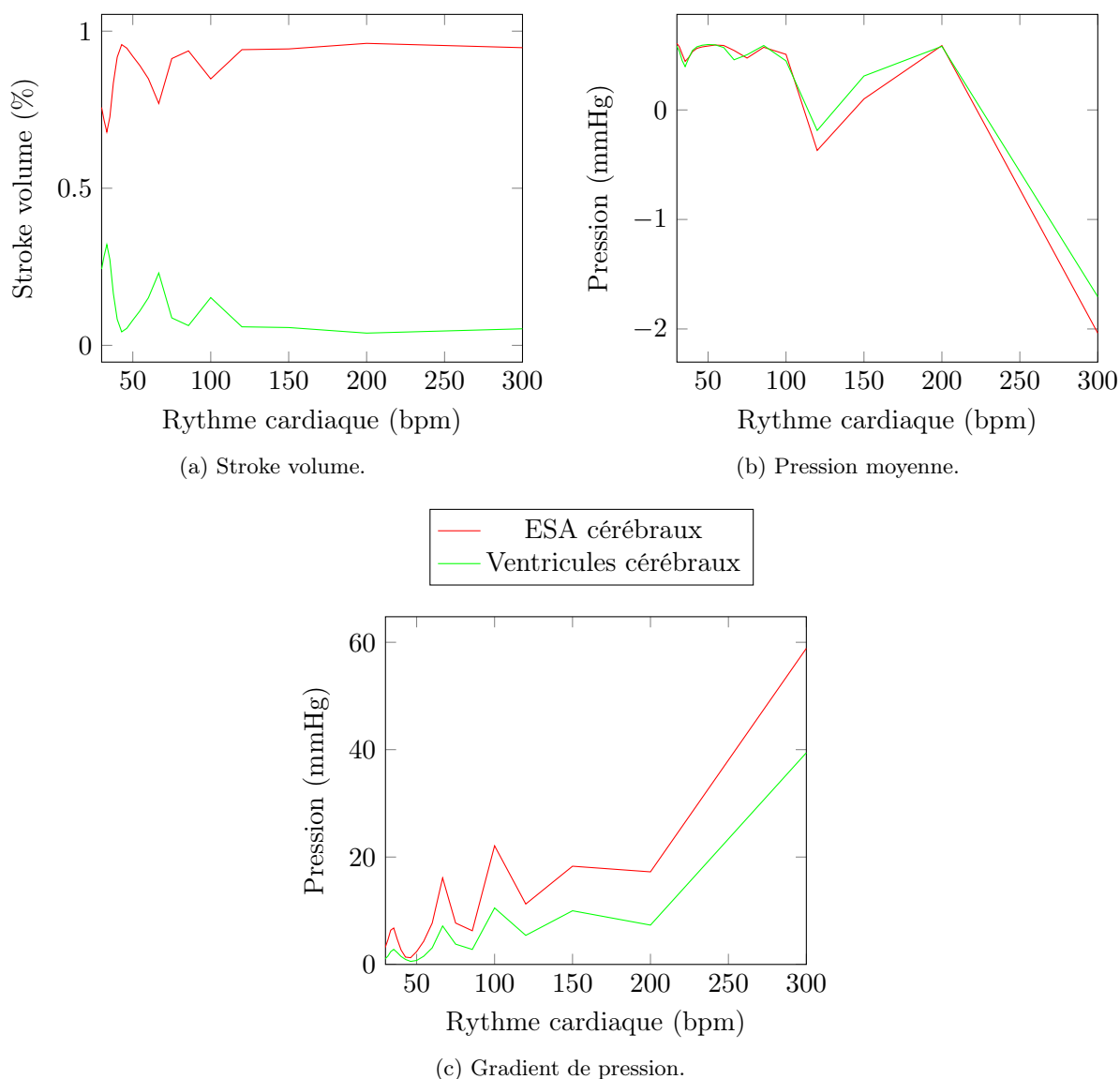


FIGURE 7.7 – **Impact des paramètres.** Effet de la période avec conservation du stroke volume.

7.2.5 Impact de la résistance à l'écoulement

Finalement, on modifie le maillage afin de créer des résistances à l'écoulement dans les ESA cérébraux, Fig. 7.9.

Les résultats de simulation sont exposés Fig. 7.10.

On observe que la résistance à l'écoulement dans les ESA cérébraux n'a que peu d'impact sur la distribution du LCS entre les ventricules cérébraux et les ESA cérébraux.

La pression moyenne, ainsi que le gradient de pression, ne sont pas modifiés par le changement de résistance à l'écoulement. Ceci est dû à une vitesse faible dans les ESA cérébraux et à une résistance déjà élevée dans ce compartiment, de par sa forme.

7.2.6 Cas réels

Dans cette partie, les paramètres géométriques du maillage sont déterminés à l'aide de mesures directes sur des IRM morphologiques, à l'aide du logiciel ImageJ [69].

Le flux d'entrée imposé au modèle est déterminé par des mesures par IRM de flux, voir Chap. 12.

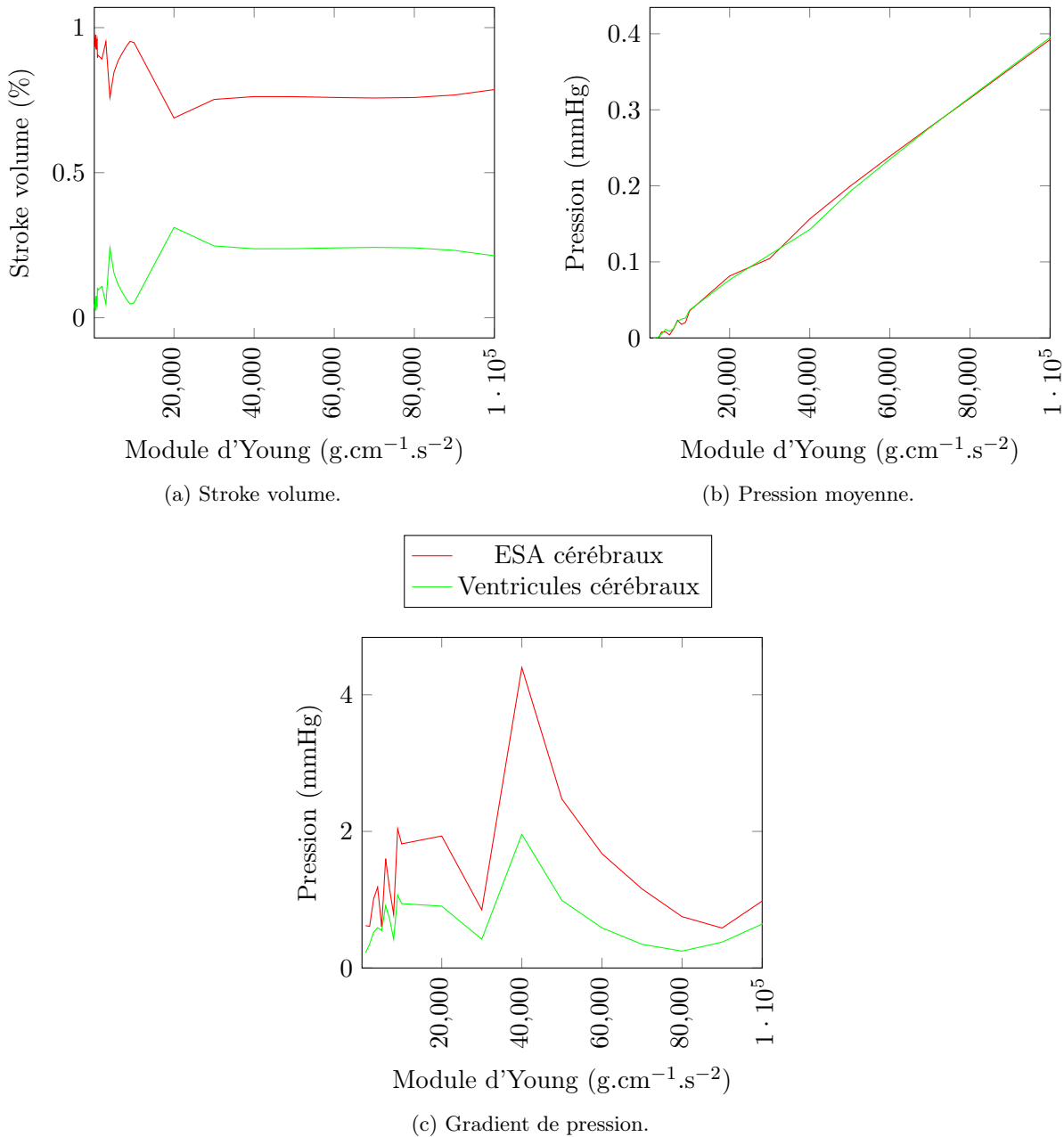


FIGURE 7.8 – **Impact des paramètres.** Effet du module d'Young.

Cas pathologique

Il s'agit du même patient que dans le cas du modèle «bifurcation», ayant subi un test d'injection et une IRM.

Les dimensions calculées à partir de l'IRM morphologique sont détaillées Tab. 7.3; elles nous permettent de créer le maillage représenté Fig. 7.11.

Les mesures par IRM de flux nous permettent d'imposer la vitesse d'entrée au niveau des ESA spinaux.

Pour cette simulation, le module d'Young est pris égal à $1.5e^4 \text{ g.cm}^{-1}.\text{s}^{-2}$; les autres paramètres sont inchangés.

Les résultats de simulation sont détaillés Fig. 7.12.

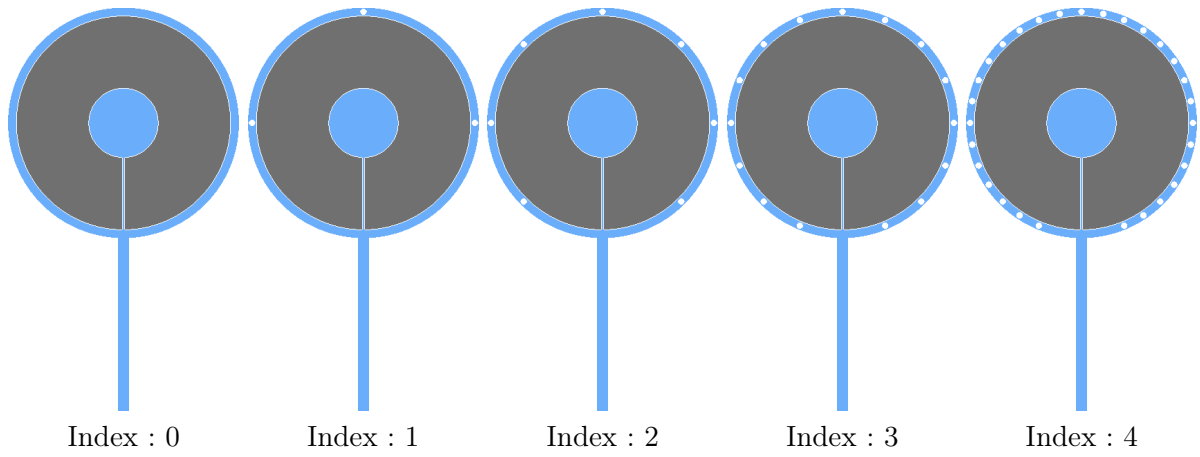


FIGURE 7.9 – **Impact des paramètres.** Création de résistances à l’écoulement.

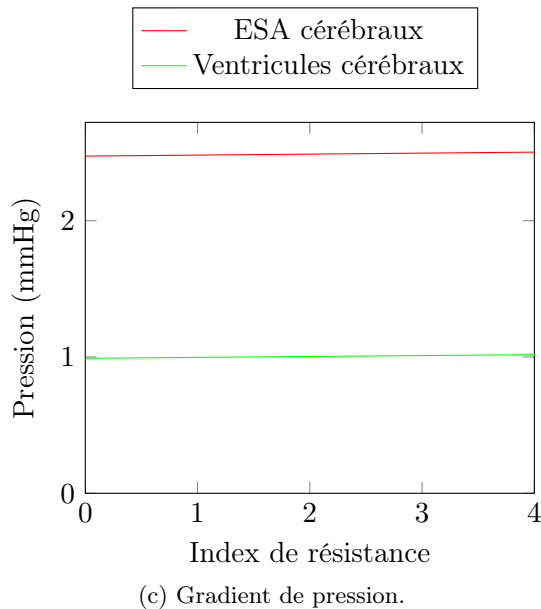
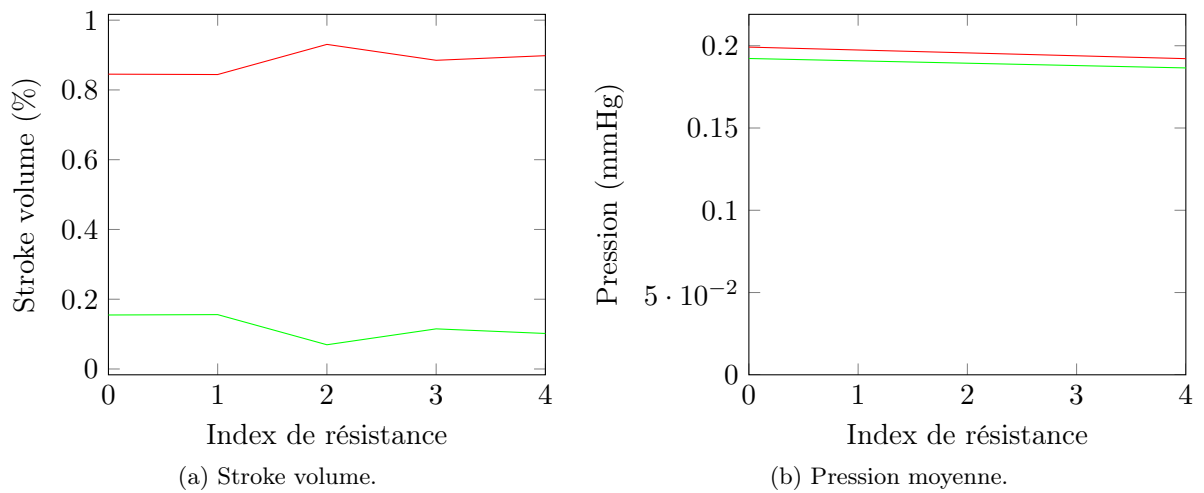
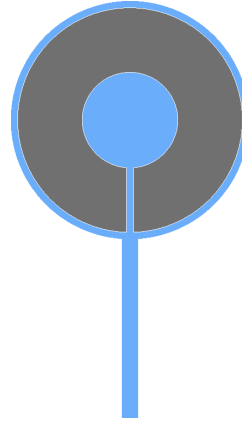
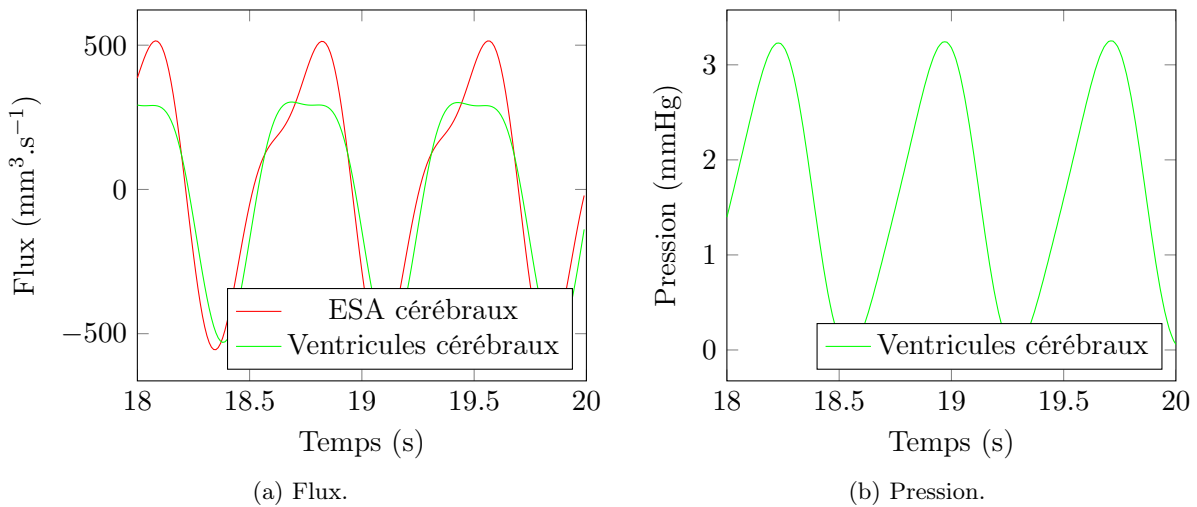


FIGURE 7.10 – **Impact des paramètres.** Effet de la résistance à l’écoulement.

On constate, dans cette simulation, un rapport du flux dans les ventricules cérébraux par rapport au flux dans les ESA cérébraux très élevé comparativement au résultat du modèle «bifurcation».

TABLE 7.3 – **Cas pathologique.** Dimensions.

Paramètre	Valeur
R_C	10 cm
R_{ESA}	9.4 cm
R_V	4 cm
D_A	0.55 cm
D_S	1.4 cm

FIGURE 7.11 – **Cas pathologique.** Maillage.FIGURE 7.12 – **Cas pathologique.** Flux et pression.

Ce flux non physiologique reflète le caractère pathologique de nos données, qui n'était pas mis en évidence avec le précédent modèle.

Grâce aux données mesurées, on peut comparer nos résultats de simulation aux mesures de flux et de pression, Fig. 7.13.

On constate que les courbes de flux sont équivalentes en amplitude, en revanche, comme dans le cas du modèle «bifurcation», le déphasage de la courbe de flux dans l'aqueduc n'est pas retrouvé.

La courbe de pression simulée est également d'amplitude similaire, bien que sa forme ne corresponde pas à la mesure effectuée (comme pour le modèle «bifurcation»).

Cas sain

Il s'agit du même sujet sain que dans le cas du modèle «bifurcation», ayant passé une IRM.

Les images issues de l'IRM morphologique permettent de définir les paramètres géométriques patient-dépendants du système cérébrospinal, Tab. 7.4, afin de créer un maillage, Fig. 7.14.

Les mesures par IRM de flux nous permettent d'imposer le flux d'entrée au niveau des ESA spinaux.

On prend ici un module d'Young égal à $5e^3 \text{ g.cm}^{-1}.\text{s}^{-2}$ et les mêmes paramètres que précédemment.

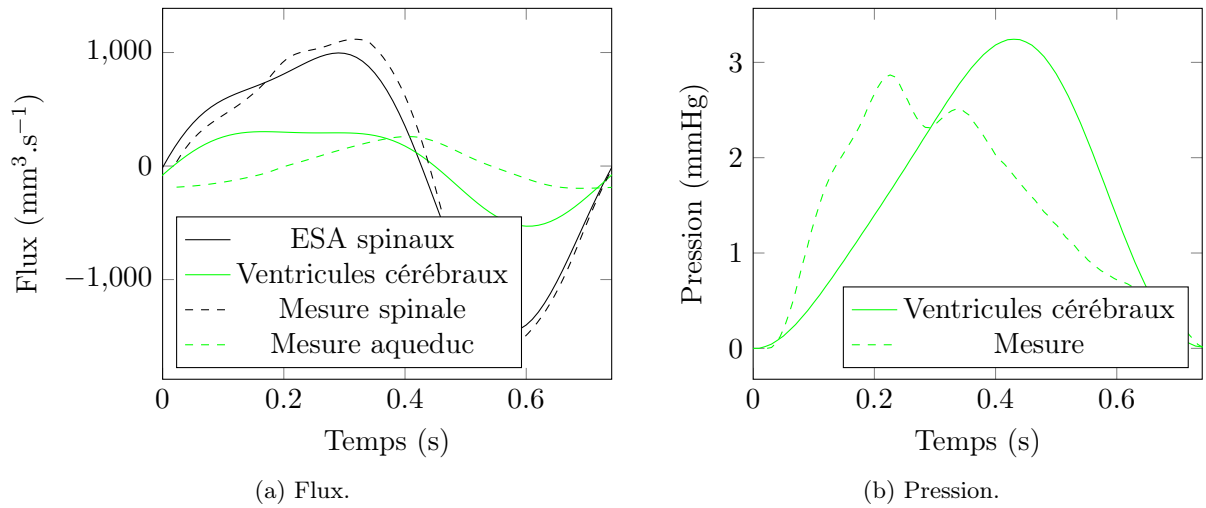


FIGURE 7.13 – **Cas pathologique.** Comparaison des résultats et des mesures.

TABLE 7.4 – **Cas sain.** Dimensions.

Paramètre	Valeur
R_C	9 cm
R_{ESA}	8.50 cm
R_V	3.50 cm
D_A	0.25 cm
D_S	1.80 cm

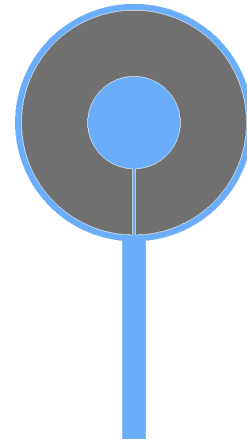


FIGURE 7.14 – **Cas sain.** Maillage.

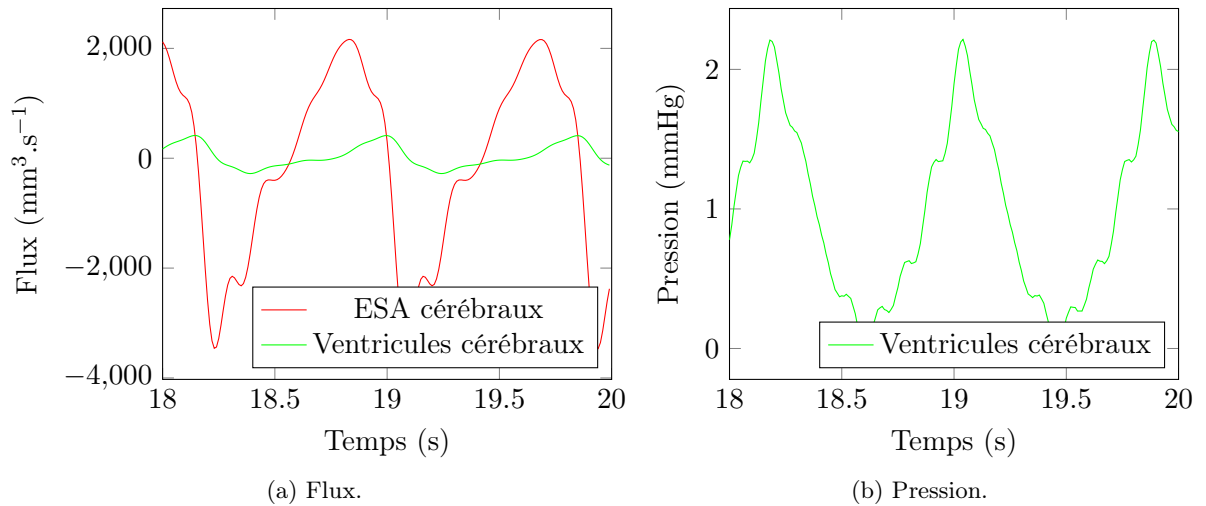


FIGURE 7.15 – **Cas sain.** Flux et pression.

Les résultats de simulation sont détaillés Fig. 7.15.

On obtient une répartition physiologique des flux dans les ESA cérébraux et dans les ventricules cérébraux. L'amplitude de la courbe de pression est dans l'intervalle physiologique et elle présente des pics et des vallées, comme décrit dans la littérature [17].

Grâce aux données mesurées, on peut comparer nos résultats de simulation aux mesures de flux, Fig. 7.16.

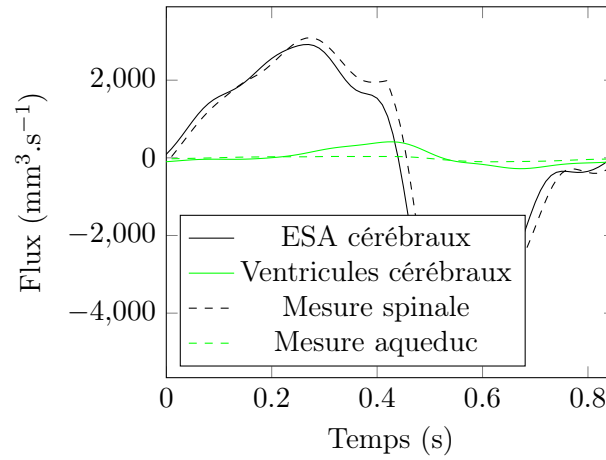


FIGURE 7.16 – **Cas sain.** Comparaison des résultats et des mesures.

On constate une sur-estimation du flux dans les ventricules cérébraux.

7.3 Impact des améliorations

7.3.1 Structures différenciées

Afin d'observer l'effet de la différenciation entre la matière grise et la matière blanche, nous allons reprendre la simulation dans le cas réel sain en imposant un module d'Young différent pour la matière blanche et pour la matière grise. La matière blanche étant plus souple que la matière grise, il convient de prendre $E_{\text{matière blanche}} = 3e^3 \text{ g.cm}^{-1}.\text{s}^{-2} < E_{\text{matière grise}} = 7e^3 \text{ g.cm}^{-1}.\text{s}^{-2}$.

Les résultats de simulation sont exposés Fig. 7.17.

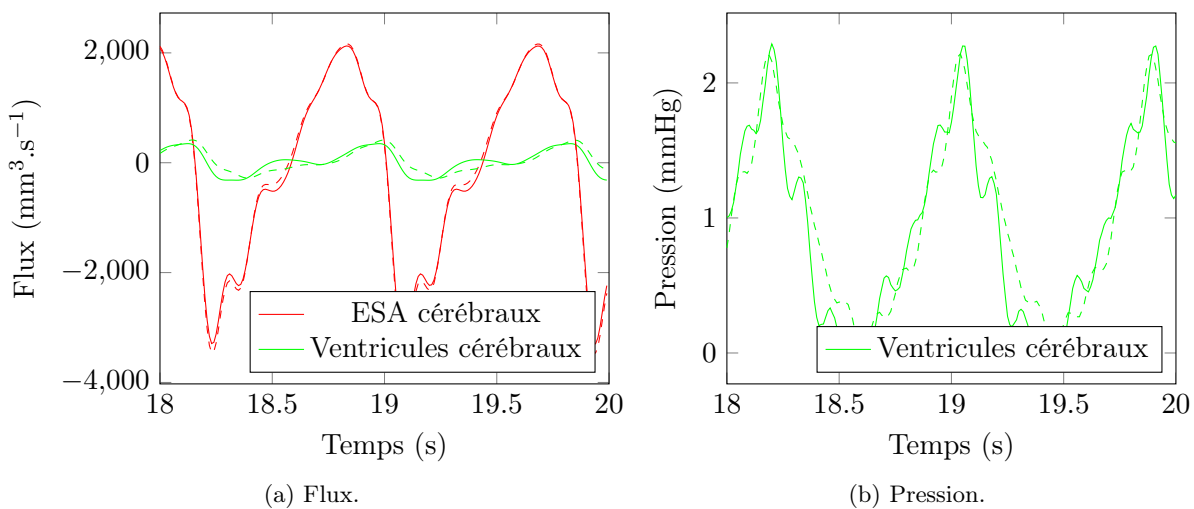


FIGURE 7.17 – **Matière blanche / matière grise.** Flux et pression. Données avant amélioration en pointillés.

On observe qu'il existe bien une différence lorsque l'on considère la différenciation de la structure en une partie «matière blanche» et une partie «matière grise». Cependant, cette différence est minimale pour les paramètres choisis.

7.3.2 Compartiment sanguin

On reprend le cas réel sain pour observer l'impact de l'ajout du compartiment sanguin. On utilise ici toutes les données apportées par l'IRM morphologique et de flux. La géométrie est définie par l'IRM morphologique et les flux sont mesurés par IRM de flux. L'entrée/sortie de LCS, l'entrée artérielle et la sortie veineuse sont appliquées en conditions aux limites du modèle.

Les résultats de simulation sont exposés Fig. 7.18.

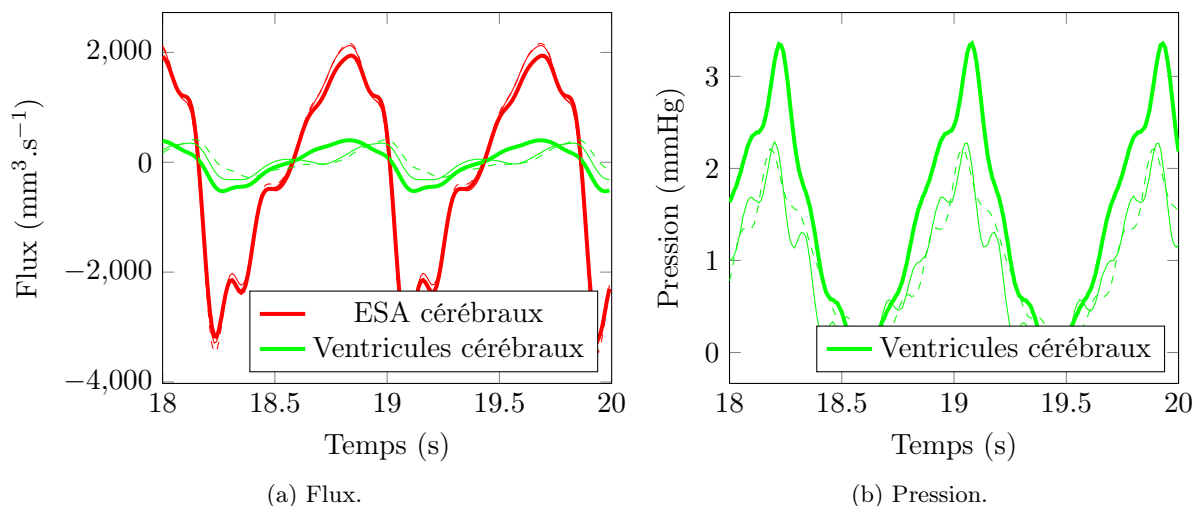


FIGURE 7.18 – **Compartiment sanguin.** Flux et pression. Données sans amélioration en pointillés, données avec différenciation matière grise / matière blanche en trait plein.

On constate que la répartition des flux est relativement similaire aux deux précédents cas. La pression est légèrement augmentée et présente toujours des pics et des vallées.

7.4 Conclusion

Ce modèle présente un comportement similaire au précédent modèle en «bifurcation». Il a, de plus, l'avantage de ne pas nécessiter d'estimation de paramètres tels que ceux requis par les modèles de Windkessel (difficiles à estimer). Les paramètres mécaniques du cerveau sont donnés dans la littérature ou sont possiblement accessibles par élastographie par résonance magnétique. L'étude des impacts des différents paramètres vient confirmer les résultats obtenus avec le modèle «bifurcation». L'impact de la période est moins important, mais va dans le même sens que pour le modèle «bifurcation».

L'impact des paramètres sur la pression vient confirmer les résultats obtenus avec le modèle «bifurcation». Le gradient de pression est bien plus important que la pression moyenne. Ici, comme on n'utilise pas de condition de Neumann (modèles de Windkessel), il est normal de calculer une pression à moyenne nulle (il n'y a que le gradient de pression dans les équations de Navier-Stokes, et pas la pression absolue).

Les simulations sur des cas réels produisent des résultats plus satisfaisants que pour le précédent modèle. On a encore un module d'Young plus élevé dans le cas pathologique (similaire au modèle «bifurcation» avec une compliance plus basse), qui tend à indiquer un problème de nature structurelle dans les désordres hydrodynamiques cérébraux.

Les améliorations réalisées grâce à l'ajout de la différenciation matière grise / matière blanche et de la prise en compte des flux artériels et veineux apportent encore une représentation plus fidèle du système ; le comportement observé tend vers un comportement physiologique.

Remarque 7.3. *Le modèle avec compartiment sanguin, où l'on impose uniquement l'entrée artérielle, et où l'on récupère le flux veineux et de LCS, représente la modélisation optimale du système cérébrospinal. Néanmoins, celui-ci n'est pas fonctionnel et nécessite une étude plus approfondie.*

Chapitre 8

Conclusion

Dans un premier temps, les méthodes numériques utilisées pour simuler la dynamique des fluides et les interactions fluide-structure ont été décrites. La stabilité des algorithmes de résolution des écoulements de fluides couplés à des modèles de Windkessel (0D) ont également été étudiée afin de s'assurer du bon fonctionnement de l'algorithme.

Ensuite, la vérification et la validation des différents algorithmes ont été réalisées en utilisant des cas test, des écoulements de fluides bien connus, et grâce à des comparaisons entre différentes bibliothèques de calcul utilisant la méthode des éléments finis et entre des mesures par IRM de flux. Cette étape de validation nous a permis d'observer des courbes de convergence obtenues numériquement concordant avec les valeurs théoriques, des écoulements de Poiseuille reproduits fidèlement et une correspondance entre les mesures par IRM de flux et les résultats de simulation. L'utilité des modèles de Windkessel a également été démontrée.

Le premier modèle, en «bifurcation», met en œuvre une représentation très simplifiée du système cérébrospinal, qui a l'avantage d'être très rapide en temps de calcul et très simple à développer.

Des paramètres clés de l'hydrodynamique du système cérébrospinal ont été mis en évidence, tels que la fréquence cardiaque et la répartition des compliances.

Ce modèle donne également des résultats physiologiques en termes de répartition des flux de LCS entre les deux compartiments cérébraux : les ventricules cérébraux et les espaces subarachnoïdiens cérébraux, mais une représentation de la pression éloignée de celle mesurée, physiologique ou pathologique.

En revanche, l'estimation des paramètres de Windkessel pose problème et les valeurs de ces paramètres sont difficilement identifiables aussi bien dans le cas pathologique que dans le cas physiologique. En particulier, les valeurs à imposer ne sont pas présentes dans la littérature. L'estimation de la compliance globale du système nécessite une mesure invasive de la pression intracrânienne, qui de plus, ne peut se faire que sur un sujet pathologique.

Le second modèle, plus représentatif du système physiologique, apporte une confirmation des résultats obtenus avec le premier modèle. Ce modèle a, de plus, l'avantage de supprimer l'identification des paramètres de Windkessel pour les remplacer par les paramètres mécaniques du cerveau accessibles dans la littérature.

Des comportements similaires du système sont retrouvés et les simulations dans des cas réels apportent des résultats plus satisfaisants au vu des mesures effectuées par IRM de flux et monitoring de pression intracrânienne.

Les principaux points mis en évidence dans cette partie sont :

- la fréquence cardiaque a un grand impact sur la distribution du LCS entre les ventricules cérébraux et les espaces subarachnoïdiens cérébraux ;

- la compliance / le module d'Young a également un impact sur la distribution du LCS ;
- la différence entre les cas sain et physiologique étudiés réside principalement dans la valeur de la compliance / du module d'Young. Ceci indique un défaut structurel à l'origine de la pathologie.

Deuxième partie

Analyse de données

Chapitre 9

Acquisition des données

Dans la pratique clinique, l'hydrodynamique cérébrale est évaluée par IRM de flux chez les patients suspectés d'hydrocéphalie. Un monitoring de pression intracrânienne est également réalisé dans certains cas. Dans ce chapitre, nous allons détailler les protocoles d'acquisition IRM et de mesure invasive de la pression intracrânienne.

9.1 Flux sanguins et de liquide cébrospinal

Les données de flux sanguins et de liquide cébrospinal (LCS) sont obtenues grâce à l'IRM de flux, plus précisément par la technique IRM par contraste de phase, détaillée dans la section 1.3. Plusieurs plans de coupe sont sélectionnés afin d'obtenir le flux dans les artères, veines et compartiments de LCS principaux, Fig. 9.1. Typiquement, des plans de coupe au niveau C2–C3 (entre la 2^{ème} et la 3^{ème} vertèbre cervicale) et au niveau intracrânien sont définis.



FIGURE 9.1 – **Plans de coupe usuels.** Un plan de coupe en C2–C3, en bas, et un plan de coupe intracrânien, en haut.

- Le plan de coupe au niveau C2-C3 sert à obtenir le flux :
- de LCS dans les espaces subarachnoïdiens spinaux ;
 - des carotides internes (niveau cervical), droite et gauche ;
 - des artères vertébrales, droite et gauche ;
 - des jugulaires, droite et gauche ;
 - des veines épidurales, droite et gauche ;
 - du plexus postérieur, droit et gauche.

Le plan de coupe intracrânien sert à obtenir le flux :

- de LCS dans l'aqueduc de Sylvius ;
- des carotides internes (niveau cérébral), droite et gauche ;
- du tronc basilaire ;
- du sinus sagittal supérieur ;
- du sinus droit.

Remarque 9.1. Les flux de LCS et les flux sanguins sont obtenus durant deux séquences différentes. En effet, comme la vitesse du LCS est faible par rapport à la vitesse du sang artériel et veineux, une vitesse d'encodage différente est considérée dans les deux séquences.

Typiquement, la vitesse d'encodage pour le LCS est de l'ordre de 50 mm/s alors que celle pour le sang est de 800 mm/s.

Les IRM de flux par contraste de phase sont traités avec le logiciel Flow Analysis [10], développé au laboratoire BioFlowImage, qui permet d'obtenir des courbes de flux, Fig. 9.2.

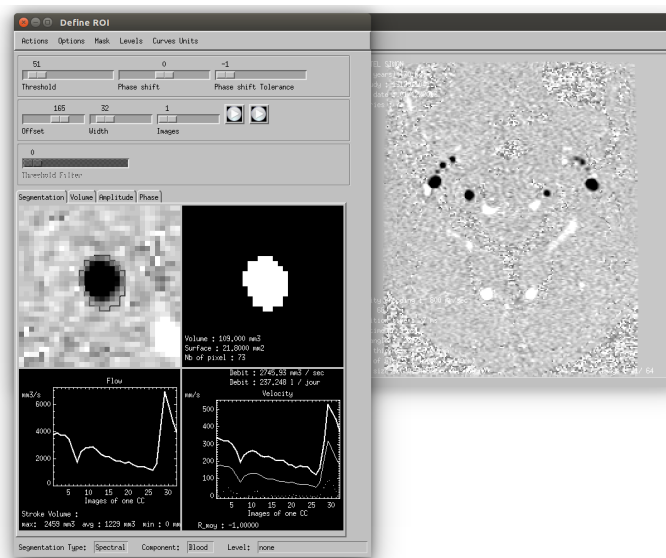


FIGURE 9.2 – Exemple de traitement de données IRM. On extrait ici le flux de l'artère vertébrale droite.

9.2 Pression intracrânienne

Le monitoring de la pression intracrânienne est une méthode invasive, qui nécessite de placer un capteur de pression dans le cerveau, dans le parenchyme, les ventricules cérébraux, l'espace épidural ou les espaces subarachnoïdiens intracrâniens ou spinaux, Fig. 9.3. Il constitue le «gold standard» dans le diagnostic de certaines pathologies telles que l'hydrocéphalie.

En complément du monitoring, il est également possible de faire varier de manière rapide et répétée le volume du LCS, afin de calculer la résistance à l'absorption (R_{out}) ou encore la compli-
ance (C) connaissant le volume injecté : il s'agit des tests d'infusion. Ils représentent une manière d'évaluer la dynamique du LCS *in vivo*.

En pratique, le test d'infusion consiste à injecter un débit constant de liquide physiologique. On commence par recueillir la PIC au repos pendant 10 à 20 minutes, puis on infuse ensuite le liquide physiologique jusqu'à atteindre un plateau ; à ce moment, on arrête l'injection et la pression diminue à nouveau, Fig. 9.4.

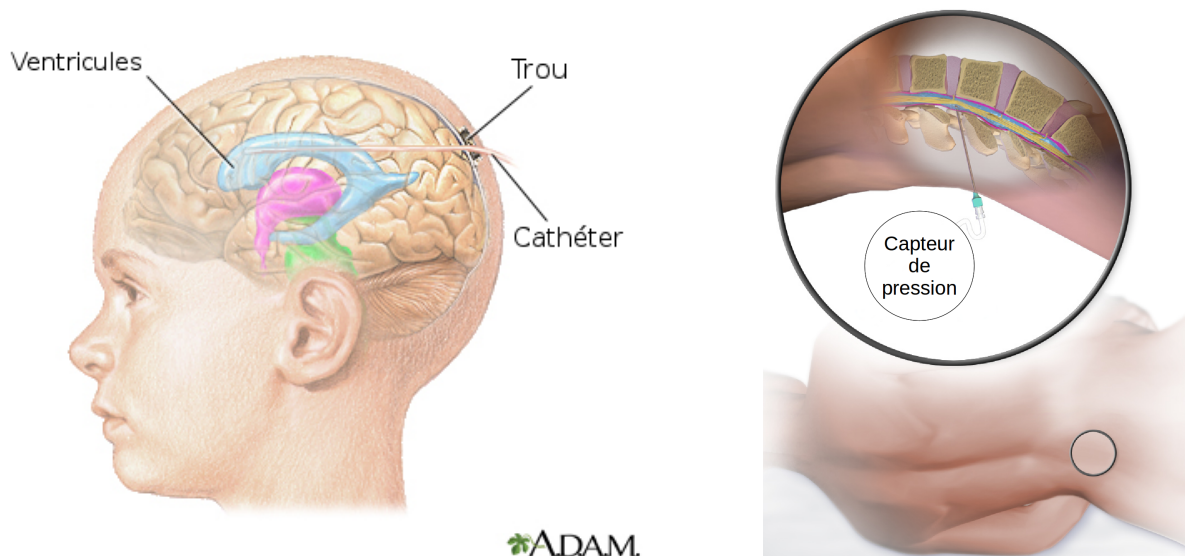


FIGURE 9.3 – **Monitoring de pression intracrânienne.** Exemples de monitoring dans les ventricules cérébraux (Auteur : A.D.A.M. Inc.) et dans le canal spinal (Auteur : Blausen Medical).

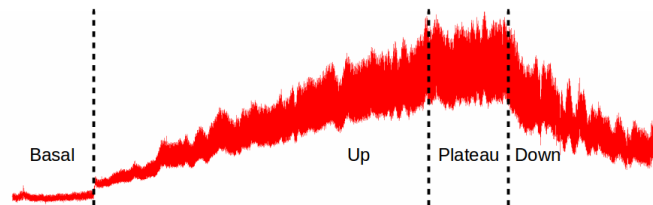


FIGURE 9.4 – **Monitoring de pression intracrânienne.** Courbe de pression. Basal : au repos, Up : pendant l'injection, Plateau : après l'injection, Down : retour à la normale.

Dans notre étude, les données de pression intracrânienne (PIC) sont obtenues via le projet PROLIPHYC [68].

Un test d'infusion à débit constant est réalisé sur des patients, positionnés en décubitus latéral droit ou gauche (allongés sur le côté droit ou le côté gauche).

Chapitre 10

Traitement des données

Un logiciel de traitement des données de flux provenant des images IRM, et des données de pression provenant des tests d'infusion, a été développé. Ce logiciel d'analyse Flux / Pression intracrânienne (PIC), baptisé *MRiCP* [33], est un complément aux logiciels déjà existants Flow Analysis (traitement des images IRM de flux) et ICM+ (traitement des données de monitoring de PIC).

Ce logiciel est développé en C++ [20], l'interface graphique est réalisée grâce à Qt [21]. Les bibliothèques annexes QCustomPlot [25] pour l'affichage des courbes et FFTW3 [31] pour le calcul des transformées de Fourier sont également utilisées.

La présentation globale du logiciel est présentée Fig. 10.1.

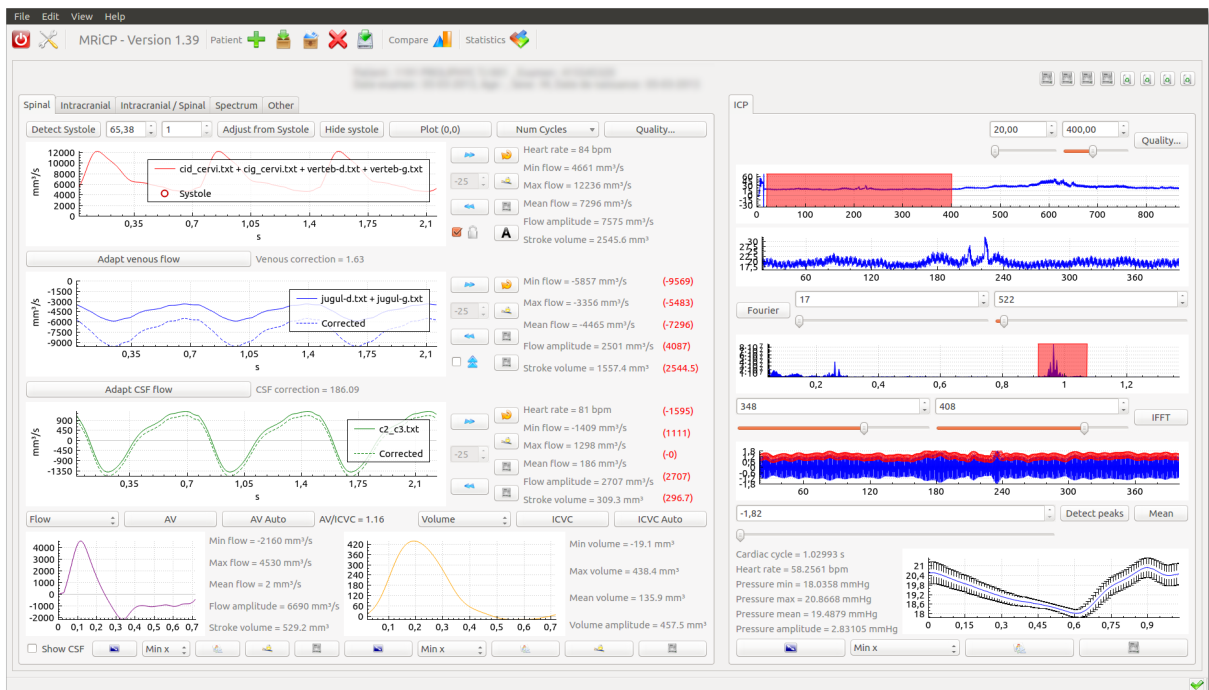


FIGURE 10.1 – Vue globale du logiciel d'analyse Flux/PIC. Les modules IRM 1 et PIC sont visibles.

L'intérêt de ce logiciel est de constituer une base de données des résultats extraits des IRM de flux et des tests d'infusion. Il permet également de mettre en relation flux et pression, afin de déterminer d'éventuelles corrélations entre ces deux grandeurs et de calculer la compliance du système cérébrospinal.

Le logiciel est composé de plusieurs modules :

- analyse de la pression provenant du test d'infusion ;
- analyse des flux provenant de l'IRM de flux ;
- calcul de la compliance à l'aide de la pression et des flux ;
- analyses statistiques.

10.1 Analyse de la pression

10.1.1 Principe

La pression mesurée durant un test d'infusion est généralement enregistrée durant plusieurs minutes ($\approx 20 - 30$ min) et pendant différentes phases de stress du système cérébrospinal, Fig. 9.4, basal (état normal), up (pendant l'infusion), plateau (juste après l'infusion) et down (état de retour à la normale).

Afin de comparer cette pression aux flux mesurés par IRM dans des conditions physiologiques, c'est-à-dire dans des conditions non stressées, et moyennés sur un cycle cardiaque à partir de quelques minutes (≈ 3 min) d'acquisition, on calcule un cycle de pression moyen par phase. La phase basale est la plus intéressante, dans la mesure où elle correspond à l'état physiologique, état dans lequel sont réalisées les IRM de flux.

L'interface graphique du module PIC est présentée Fig. 10.2.

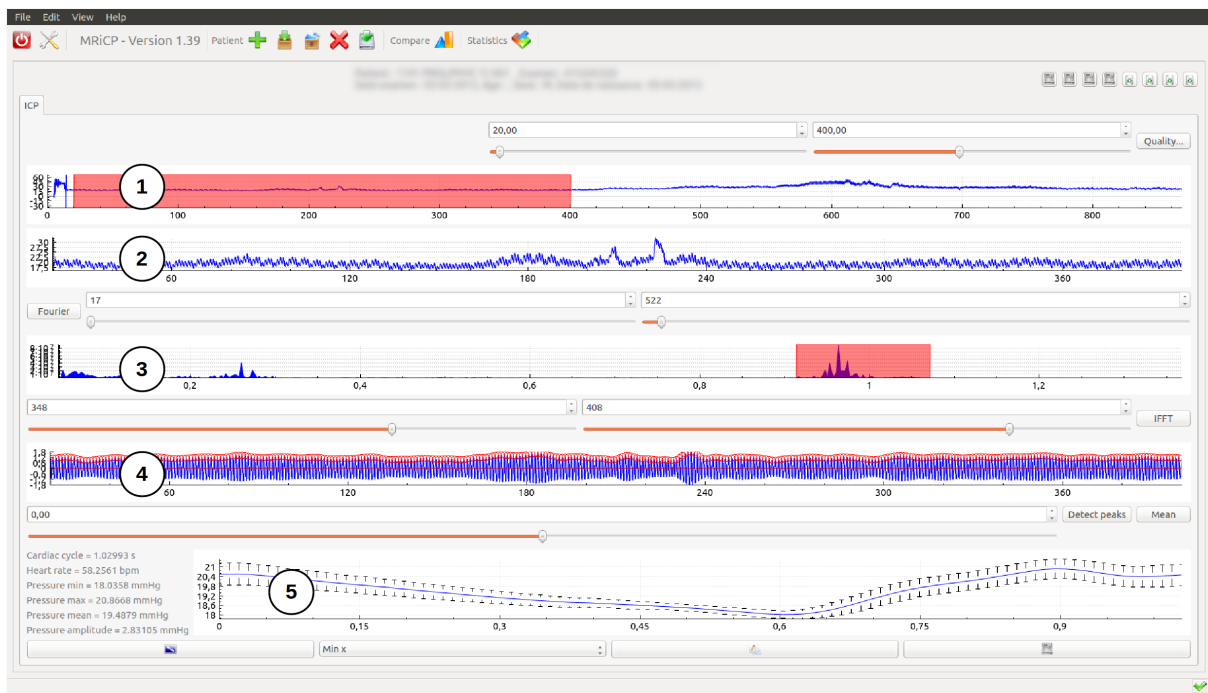


FIGURE 10.2 – Vue du module PIC.

10.1.2 Fonctionnement

Les différentes étapes sont numérotées sur la figure Fig. 10.2.

Étape 1 :

La courbe de pression correspondante à toute la mesure du test d'infusion est chargée dans le logiciel d'analyse.

Afin de correspondre au maximum avec les données IRM de flux, il convient de sélectionner une phase de la mesure de pression, de préférence la phase basale, pendant une durée proche de la durée d'acquisition IRM. Cette partie de la courbe sélectionnée sera appelée région d'intérêt (ROI, *region of interest* en anglais).

Étape 2 :

La sélection de l'étape 1 (ROI) est affichée, ce qui permet de déceler d'éventuels parasites sur la mesure de pression.

Une transformée de Fourier [31] est maintenant appliquée à la ROI.

Étape 3 :

La transformée de Fourier de la ROI est affichée. Il faut alors choisir la fréquence en relation avec la pulsation cardiaque. Pour une indication sur la fréquence à choisir, il est possible de s'aider du traitement des IRM de flux qui donne accès à cette information.

Une transformée de Fourier inverse est réalisée afin d'obtenir un signal filtré.

Étape 4 :

Le signal filtré est affiché. On réalise alors une détection de pics afin de déterminer tous les cycles de pression correspondant à la pulsation cardiaque.

La détection de pics est effectuée en cherchant le point maximum de la courbe supérieur à un seuil défini par l'utilisateur (typiquement zéro, quand le signal est correctement filtré, ou la moyenne du signal filtré).

En utilisant les pics détectés précédemment, on découpe la ROI en plusieurs cycles cardiaques et on moyenne le tout pour obtenir un cycle cardiaque moyen.

Étape 5 :

Le cycle moyen de la pression est affiché. Les informations d'amplitude, de maximum, de minimum et de moyenne du signal sont affichés. Des outils de recalage, de normalisation et de lissage sont disponibles pour modifier le signal.

Les informations sur le cycle moyen, ainsi que la courbe, sont sauvegardées pour une utilisation ultérieure dans le module de statistiques.

10.2 Analyse des flux

10.2.1 Principe

Les quantités mesurées par l'IRM de flux sont des vitesses de fluide, encodées grâce à des niveaux de gris. Le logiciel Flow Analysis [10] permet de convertir ces données en flux sur des zones bien définies, par exemple dans une artère carotide interne droite.

Les quantités qui nous intéressent ici sont les flux ou les volumes nets déplacés dans la boîte crânienne. Des opérations sur les flux sont réalisées afin de les convertir en flux nets et volumes nets au niveau cervical comme au niveau intracrânien.

Une partie de l'interface du module IRM est présentée Fig. 10.3

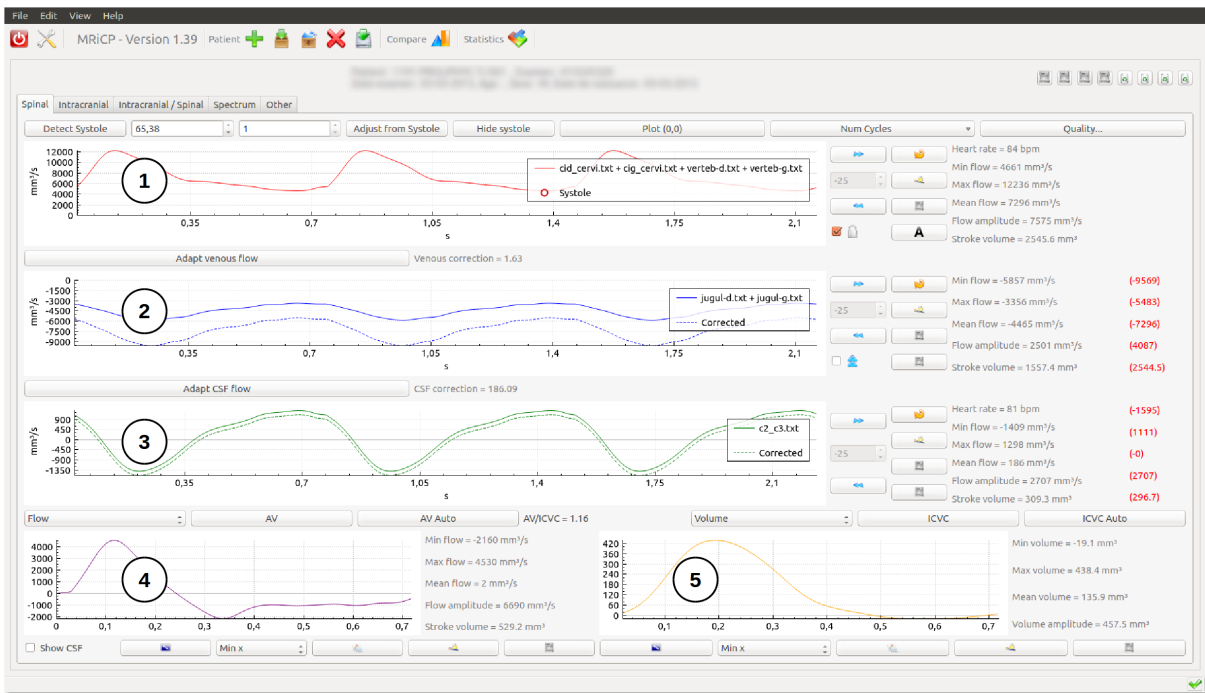


FIGURE 10.3 – Vue du module IRM 1. Flux cervicaux.

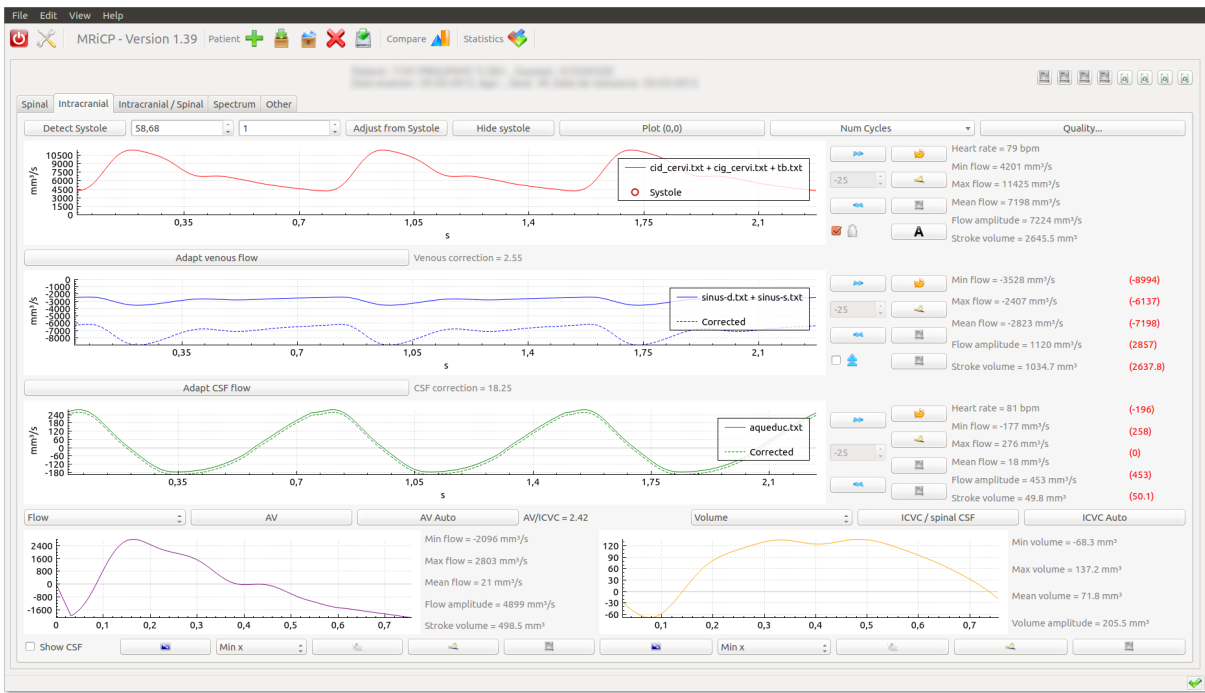


FIGURE 10.4 – Vue du module IRM 2. Flux cérébraux.

10.2.2 Fonctionnement

Les différentes étapes sont numérotées sur la figure 10.3. L'étude au niveau intracrânien, Fig. 10.4, est identique.

Étape 1 :

Les courbes de flux sont chargées depuis les données traitées du logiciel Flow Analysis. Les flux artériels cervicaux sont sommés pour obtenir une courbe de flux globale des artères cervicales. On

procède de même avec les flux veineux cervicaux, les flux artériels cérébraux et les flux veineux cérébraux.

Un recalage de la courbe artérielle sur le pic systolique est effectué en calculant sa dérivée et en sélectionnant les valeurs au dessus d'un seuil prédéfini automatiquement et modifiable par l'utilisateur afin de prendre en compte une plus grande partie du cycle.

Le recalage est appliqué aux courbes veineuses et de LCS, ainsi que sur toutes les courbes de la partie intracrânienne.

Étape 2 :

La courbe de flux veineux est corrigée pour correspondre à la courbe de flux artériel selon l'hypothèse suivante : tout ce qui rentre au cours du cycle cardiaque doit ressortir. La courbe de flux veineux est donc corrigée par un facteur α égal à la moyenne du flux artériel sur la moyenne du flux veineux.

Étape 3 :

La courbe de flux de LCS est corrigée selon la même hypothèse que le flux artério-veineux. On la corrige donc pour que le flux soit à moyenne nulle.

Étape 4 :

Le flux artério-veineux est calculé par la soustraction du flux veineux et du flux artériel. Le volume artério-veineux est calculé par intégration du flux artério-veineux.

Étape 5 :

Le flux intracrânien est calculé par la soustraction du flux artériel, du flux veineux et du flux de LCS. Le volume intracrânien est calculé par intégration du flux intracrânien.

Modules supplémentaires

Le premier module supplémentaire permet d'observer les déphasages entre les pics artériels, veineux et de LCS, Fig. 10.5a. Il peut permettre, dans certains cas, de détecter des anomalies de fonctionnement du système cérébrospinal, lorsque les déphasages sont très importants.

Le second module permet d'observer le spectre de Fourier des différentes courbes de flux, Fig. 10.5b.

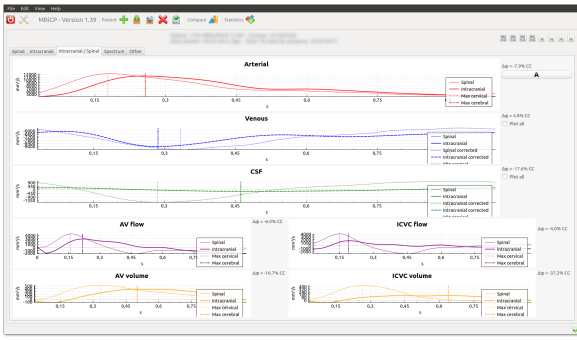
Le dernier module permet de visionner chaque courbe de flux indépendamment, Fig. 10.5c. Il sert principalement à détecter une anomalie de traitement des données au niveau du logiciel Flow Analysis.

Toutes les informations d'amplitude, de maximum, de minimum, de moyenne et de stroke volume des différentes courbes sont sauvegardées pour une utilisation ultérieure dans le module de statistiques.

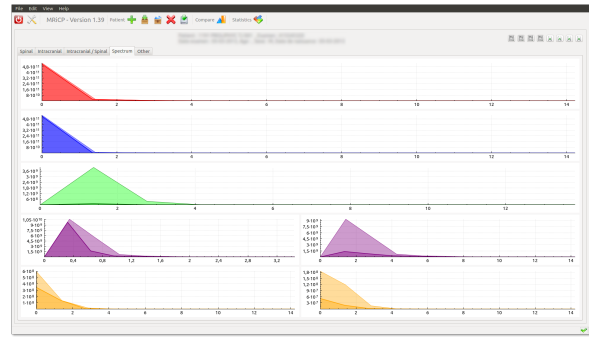
10.3 Calcul de la compliance

10.3.1 Principe

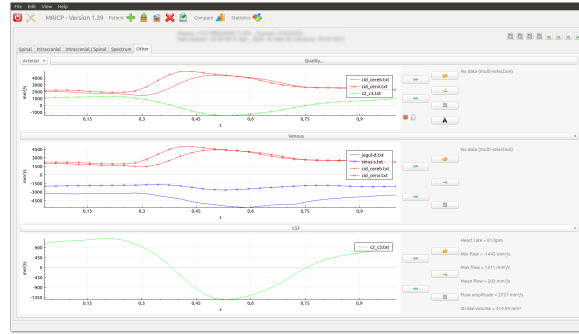
Une fois les flux et les pressions traités, il nous est possible d'obtenir une information sur la compliance : l'élasticité du système cérébrospinal, en mettant en relation le volume oscillatoire de LCS et la modification de pression qui en résulte, Fig. 10.6.



(a) Module 3.



(b) Module 4.



(c) Module 5.

FIGURE 10.5 – Vue du module IRM 3, 4 et 5. Déphasages, spectres et visualisation indépendante.



FIGURE 10.6 – Vue du module compliance. Compliance globale, par parties et instantannée.

10.3.2 Fonctionnement

La compliance est calculée à l'aide de la formule :

$$C = \frac{\Delta V}{\Delta P} \quad (10.1)$$

où C est la compliance, ΔV l'amplitude de volume et ΔP l'amplitude de pression.

Cette compliance peut être calculée de manière globale (sur le cycle cardiaque entier), par partie en sélectionnant des régions particulières, ou point par point pour obtenir une compliance variable au cours du cycle cardiaque.

Pour le calcul de la compliance instantanée, le choix du délai (nombre de points) considéré ainsi qu'un outil de filtrage sont mis en place pour éviter de trop grandes fluctuations dues à l'imprécision des mesures.

Les différentes compliances calculées sont sauvegardées pour une utilisation ultérieure dans le module de statistiques.

10.4 Outils statistiques

10.4.1 Principe

Une fois toutes les données extraites des courbes de flux et de pression, il est intéressant de pouvoir observer les relations entre les différents paramètres physiologiques de toute une population de patients, Fig. 10.7

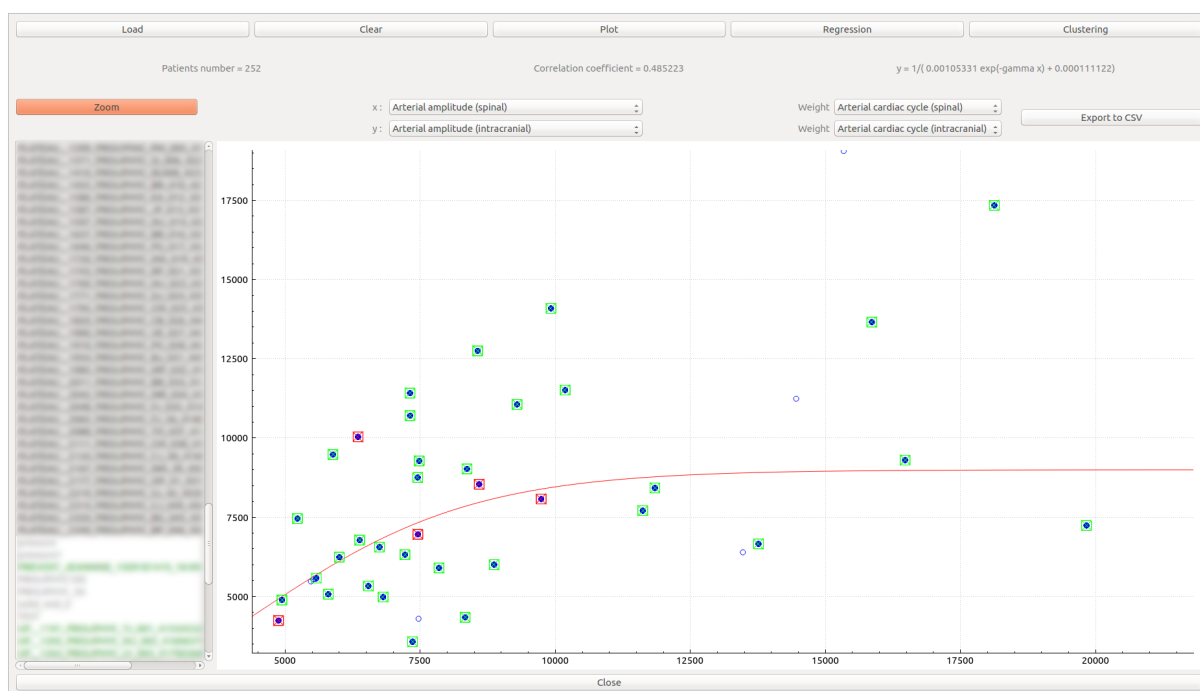


FIGURE 10.7 – Vue du module statistique. Données d'une population, régression logarithmique et clustering.

10.4.2 Fonctionnement

Toutes les données concernant chaque patient sont chargées (minimum, maximum, amplitude, moyenne et stroke volume des différentes courbes de flux et de volume et minimum, maximum, amplitude et moyenne des courbes de pression).

Il est possible de n'afficher qu'une partie de la population, afin de réaliser des régression linéaires, puissance ou logarithmiques ainsi que du clustering¹. Toutes les données et les traitements sont exportables vers un tableur (par exemple LibreOffice Calc ou Microsoft Excel[®]).

1. Partitionnement statistique de données, algorithme K-means.

Chapitre 11

Résultats

Partie en collaboration avec Armelle LOKOSSOU [48], stagiaire de Master 2 «Physiopathologie humaine, Exploration fonctionnelle et Imagerie» puis doctorante en «Biologie-Santé – Neurosciences-Traitement du Signal» au laboratoire BioFlowImage, Amiens.

Une population de patients suspectés d’hydrocéphalie, provenant du protocole clinique PROLI-PHYC, a constitué la première base de données provenant d’IRM de flux et de monitoring de pression intracrânienne. Les résultats présentés dans ce chapitre proviennent de l’analyse de cette base de données à l’aide du logiciel MRiCP.

11.1 Population

Une population de 100 patients suspectés d’hydrocéphalie chronique de l’adulte a été incluse dans le protocole PROLIPHYC [68]. Ces patients ont tous subi un test d’infusion au niveau lombaire et, le jour précédent, une IRM cérébrale comprenant des séquences de flux cervicales et cérébrales à l’hôpital Purpan de Toulouse.

Après traitement des données, 17 patients ont été exclus pour cause de monitoring de pression intracrânienne défaillant ou d’IRM de flux incomplète. Il reste donc 83 patients, 43 hommes et 40 femmes, âgés de 52 à 92 ans (74 ± 7 ans).

Le diagnostic posé par l’équipe de l’hôpital de Toulouse, dirigée par E. Schmidt, fait apparaître deux sous-groupes :

- 32 patients avec profil hydrodynamique «normal» ;
- 51 patients avec profil hydrodynamique «pathologique».

En première intention, les données seront analysées «en aveugle» ; les sous-groupes ne seront pas considérés. Ensuite, une comparaison des résultats obtenus sera faite entre nos résultats et ceux de l’équipe de Toulouse.

Remarque 11.1. *La classification des sous-groupes par l’équipe de Toulouse est réalisée grâce à l’étude de l’hydrodynamique cérébrale et de la pression intracrânienne mais également à l’aide d’analyses biochimiques du LCS, de bilan psychologiques, d’étude de la motricité et de nombreux autres paramètres.*

11.2 Méthodes

Les IRM de flux et les courbes de pression intracrânienne de chaque patient ont été traitées.

L’analyse des résultats obtenus après le traitement des IRM de flux et des pressions intracrâniennes (PIC) est réalisée avec le logiciel de statistiques R [60].

Une corrélation de Pearson est utilisée pour mettre en relation différentes données et un test statistique de Student est utilisé pour déterminer les différences significatives entre plusieurs jeux de données.

Les notations des différences significatives sont détaillées Tab. 11.1.

TABLE 11.1 – **Tests statistiques.** Notations.

Symbole	p -value
n.s.	$p > 0.05$
*	$p < 0.05$
***	$p < 0.001$

11.3 Reproductibilité des mesures

Dans un premier temps, et pour valider le travail d'analyse des données IRM et PIC, on réalise une étude de reproductibilité des mesures selon l'opérateur.

Deux opérateurs différents (A. LOKOSSOU et moi-même) ont analysé les IRM de flux et les mesures de PIC d'une partie des 100 patients (35 patients).

Pour la partie pression intracrânienne, l'amplitude de la pression moyennée est comparée pour les deux opérateurs, Fig. 11.1. Le coefficient de corrélation entre les deux traitements est de 0.96 ($p < 0.001$). On a donc une reproductibilité quasi-parfaite entre les deux opérateurs.

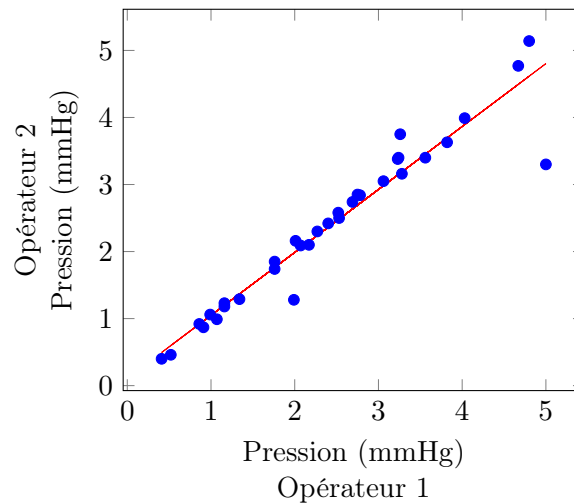
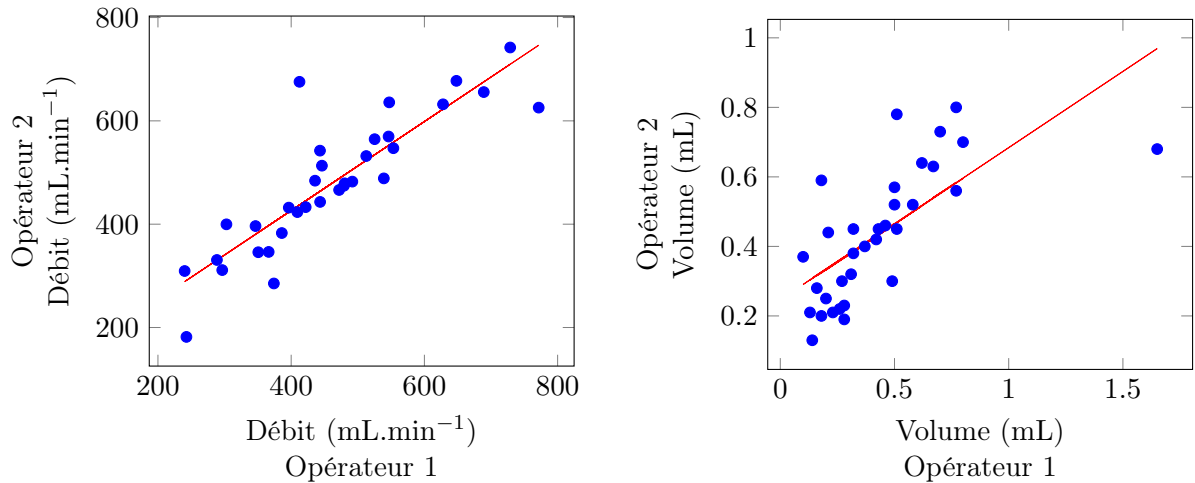


FIGURE 11.1 – **Reproductibilité des mesures.** Partie pression intracrânienne. Amplitude de la pression intracrânienne analysée par deux opérateurs. Coefficient de corrélation à 0.96 ($p < 0.001$).

Pour la partie IRM de flux, le débit artériel moyen cérébral et le changement de volume cérébral sont comparés pour les deux opérateurs, Fig. 11.2. Ce dernier paramètre est très représentatif du traitement effectué puisqu'il prend en compte l'ensemble des données de flux (artères + veines + LCS). Les coefficients de corrélation obtenus entre les deux traitements sont de 0.87 ($p < 0.001$) et 0.69 ($p < 0.001$) respectivement pour le débit artériel moyen cérébral et pour le changement de volume intracrânien. On a donc également une reproductibilité entre les deux opérateurs, même si celle-ci est moins forte que pour le traitement de la pression. Ceci est dû aux nombreuses manipulations nécessaires pour le traitement des images IRM, qui ne sont pas toujours d'une très bonne qualité.



(a) Débit artériel moyen. Coefficient de corrélation à 0.87 ($p < 0.001$). (b) Changement de volume intracrânien. Coefficient de corrélation à 0.69 ($p < 0.001$).

FIGURE 11.2 – **Reproductibilité des mesures.** Partie IRM de flux. Débit artériel moyen et amplitude du changement de volume intracrânien, analysés par deux opérateurs.

Au vu des résultats, la reproductibilité de la méthode dans son ensemble est avérée entre différents opérateurs.

11.4 Résultats

11.4.1 Description de la population

La pression intracrânienne moyenne au sein de la population étudiée est de 10.12 ± 3.48 mmHg, avec une valeur maximale de 20.05 mmHg et une valeur minimale de 2.81 mmHg. Le gradient de pression intracrânienne ΔP moyen est de 2.26 ± 1.25 mmHg avec une valeur maximale de 5.70 mmHg et une valeur minimale de 0.35 mmHg.

Le stroke volume du LCS au niveau cervical moyen est de 0.56 ± 0.21 mL. Cette valeur est proche de celles évoquées dans la littérature [10].

Il est détaillé, Fig. 11.3a, le débit artériel moyen aux niveaux cervical et cérébral. On constate que les valeurs sont significativement similaires. Ce résultat indique que les mesures effectuées par IRM de flux sur le réseau artériel sont cohérentes.

Les valeurs obtenues dans cette étude sont un peu inférieures aux valeurs présentes dans la littérature, variant de $748 \text{ mL}\cdot\text{min}^{-1}$ [15] chez le sujet jeune jusqu'à $509 \pm 103 \text{ mL}\cdot\text{min}^{-1}$ [15] $\sim 517 \pm 110 \text{ mL}\cdot\text{min}^{-1}$ [8] chez les sujets plus âgés. On peut donc suspecter une diminution du débit artériel dans les cas d'hydrocéphalie. Cette hypothèse est déjà présentée dans la littérature en argumentant sur les propriétés structurelles et mécaniques du réseau vasculaire cérébral [5, 8, 11, 42].

On observe, Fig. 11.3b, que le débit veineux cervical est significativement inférieur ($p < 0.005$) au débit veineux cérébral. Ce résultat montre que les mesures effectuées par IRM de flux sur le réseau veineux sont partielles.

Le débit retrouvé au niveau des sinus droit et sagittal est supérieur à celui présent dans les jugulaires. Ceci montre l'implication d'autres chemins de retour du sang veineux que les jugulaires, comme le plexus postérieur ou les veines épidurales [10, 45, 67, 79].

Remarque 11.2. *Le débit veineux que nous comparons ici n'est pas pondéré par le facteur de correction α introduit par O. Balédent et al. [10].*

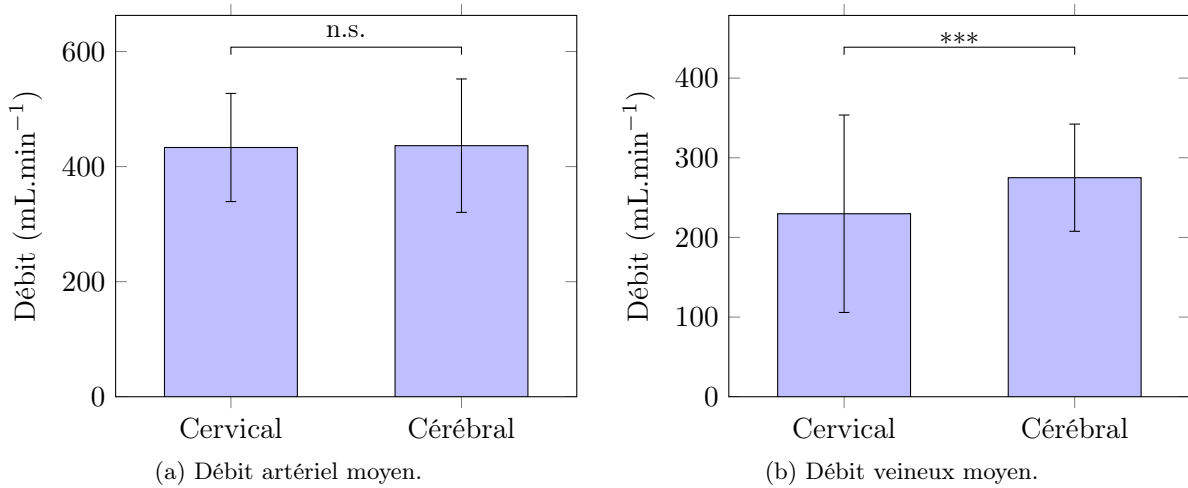


FIGURE 11.3 – Description de la population. Flux.

Le changement de volume sanguin au niveau cérébral $\Delta V_{\text{Cervical}}$ moyen est de 0.78 ± 0.36 mL. Il est significativement supérieur ($p < 0.001$) à celui calculé au niveau intracrânien $\Delta V_{\text{Cérébral}} = 0.38 \pm 0.23$ mL, Fig. 11.4.

La différence entre les valeurs au niveau cervical et au niveau cérébral s'explique par la différence observée sur les débits veineux.

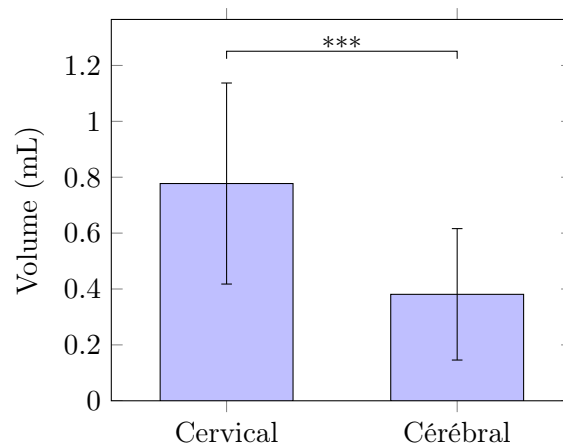


FIGURE 11.4 – Description de la population. Amplitude du changement de volume intracrânien.

Le compliance calculée au niveau cervical $C_{\text{Cervical}} = 0.48 \pm 0.42$ mL/mmHg est significativement supérieure ($p < 0.001$) à celle calculée au niveau cérébral $C_{\text{Cérébral}} = 0.22 \pm 0.17$ mL/mmHg, Fig. 11.5. Ce résultat signifie qu'une partie de la compliance est répartie dans tout le système cérébrospinal et pas seulement au niveau cérébral.

La pression intracrânienne étant identique au niveau cervical et cérébral [46], la différence de compliance entre le niveau cervical et le niveau cérébral est due à la différence de changement de volume intracrânien.

11.4.2 Catégorisation de la population

Nous utilisons maintenant le diagnostic établi par les équipes médicales pour scinder notre population en deux groupes :

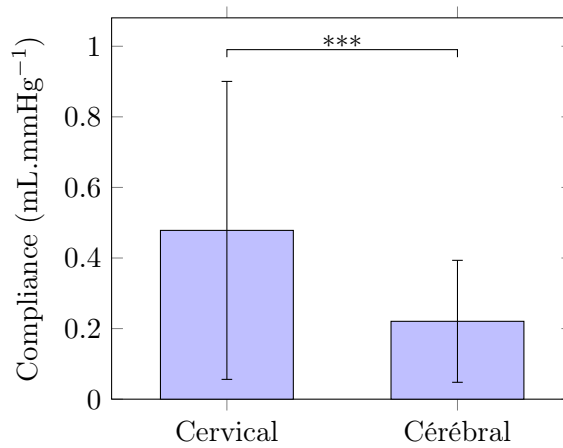


FIGURE 11.5 – Description de la population. Compliance.

— 1 groupe avec profil hydrodynamique «normal» : *HydroNorm* ;
 — 1 groupe avec profil hydrodynamique «pathologique» : *HydroPatho*.
 L'amplitude de la pression intracrânienne chez les sujets *HydroPatho* ($\Delta P = 2.46 \pm 1.34$ mmHg) est significativement supérieure ($p < 0.05$) à celle des sujets *HydroNorm* ($\Delta P = 1.92 \pm 1.03$ mmHg), Fig. 11.6.

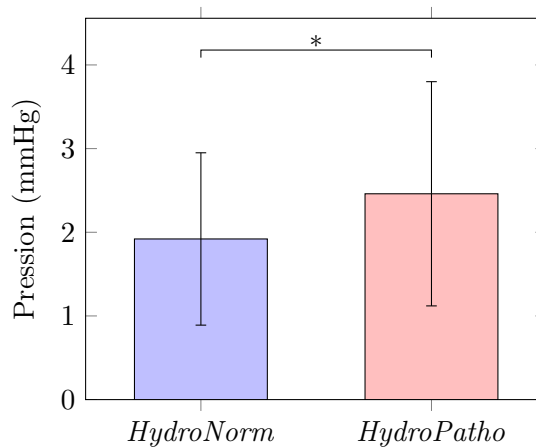


FIGURE 11.6 – Catégorisation de la population. Amplitude de la PIC.

Le volume oscillatoire de LCS (stroke volume) au niveau cervical des patients *HydroPatho* ($SV = 0.52 \pm 0.19$ mL) est significativement inférieur ($p < 0.05$) à celui des patients *HydroNorm* ($SV = 0.62 \pm 0.22$ mL), Fig. 11.7.

Au niveau cervical, le débit artériel moyen des patients *HydroPatho* (440.54 ± 99.20 mL.min⁻¹) est similaire à celui des patients *HydroNorm* (421.52 ± 85.21 mL.min⁻¹). Au niveau cérébral, le débit artériel moyen des patients *HydroPatho* (441.40 ± 125.96 mL.min⁻¹) est également similaire à celui des patients *HydroNorm* (428.32 ± 99.28 mL.min⁻¹), Fig. 11.8a.

Au niveau cervical, le débit veineux moyen des patients *HydroPatho* (230.78 ± 122.52 mL.min⁻¹) est similaire à celui des patients *HydroNorm* (228.06 ± 128.08 mL.min⁻¹). Au niveau cérébral, le débit veineux moyen des patients *HydroPatho* (279.19 ± 51.27 mL.min⁻¹) est également similaire à celui des patients *HydroNorm* (272.33 ± 76.02 mL.min⁻¹), Fig. 11.8b.

L'amplitude du changement de volume intracrânien au niveau cervical des patients *HydroPatho* (0.82 ± 0.40 mL) est similaire à celle des patients *HydroNorm* (0.72 ± 0.29 mL). Au niveau cérébral, l'amplitude du changement de volume intracrânien des patients *HydroPatho* (0.36 ± 0.21 mL) est similaire à celle des patients *HydroNorm* (0.41 ± 0.28 mL), Fig. 11.9a.

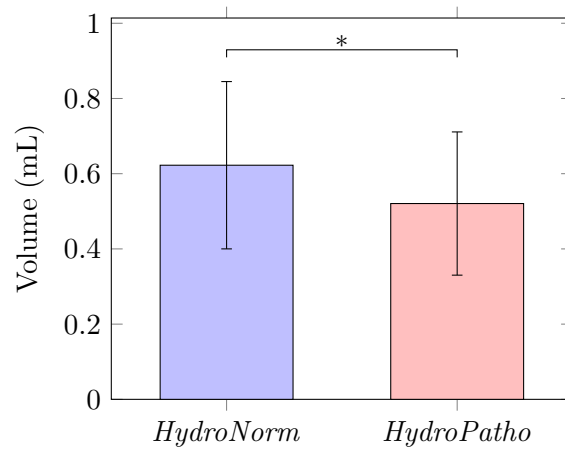


FIGURE 11.7 – **Catégorisation de la population.** Stroke volume de LCS.

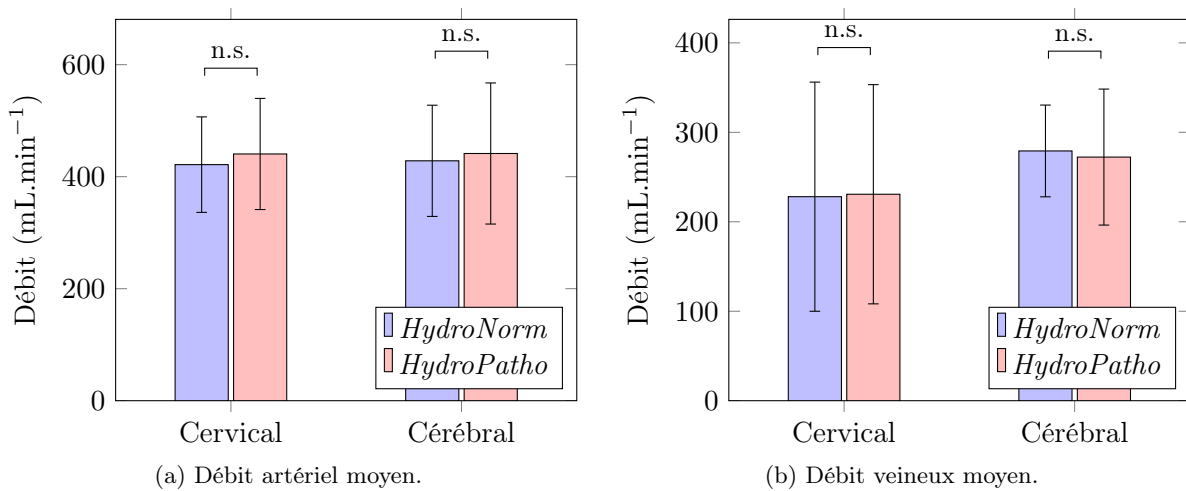


FIGURE 11.8 – **Catégorisation de la population.** Débits.

Le volume artérioveineux moyen au niveau cervical des patients *HydroPatho* (0.38 ± 0.23 mL) est similaire à celle des patients *HydroNorm* (0.47 ± 0.22 mL). Au niveau cérébral, le volume artérioveineux moyen des patients *HydroPatho* (0.43 ± 0.15 mL) est similaire à celle des patients *HydroNorm* (0.48 ± 0.17 mL), Fig. 11.9b.

Les compliances calculées au niveau cervical chez les patients *HydroPatho* (0.43 ± 0.31 mL) sont similaires à celles des patients *HydroNorm* (0.56 ± 0.55 mL). Par contre, au niveau cérébral, les compliances des patients *HydroPatho* (0.18 ± 0.13 mL) sont significativement inférieures ($p < 0.05$) à celles des patients *HydroNorm* (0.27 ± 0.21 mL), Fig. 11.10.

L'amplitude de la pression intracrânienne augmentée chez les sujets *HydroPatho* par rapport aux sujets *HydroNorm*, couplée à la diminution du stroke volume de LCS et aucune autre modification significative des autres paramètres du système cérébrospinal, suggèrent une diminution de la compliance chez les sujets *HydroPatho*. Ces résultats sont en accord avec ceux exposés dans la littérature [52, 51, 9] qui ont également mis en évidence une diminution de la compliance cérébrale dans le cas de l'hydrocéphalie et également en accord avec les résultats de simulation numérique exposés dans la première partie du manuscrit.

11.4.3 Récapitulatif

Différences significatives aux niveaux cervical et cérébral :

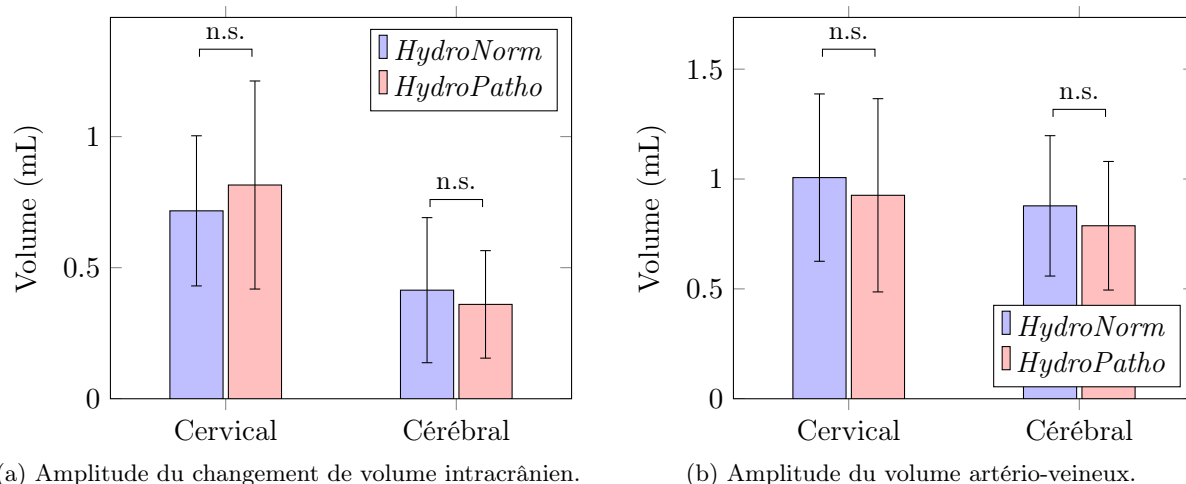


FIGURE 11.9 – Catégorisation de la population. Volumes.

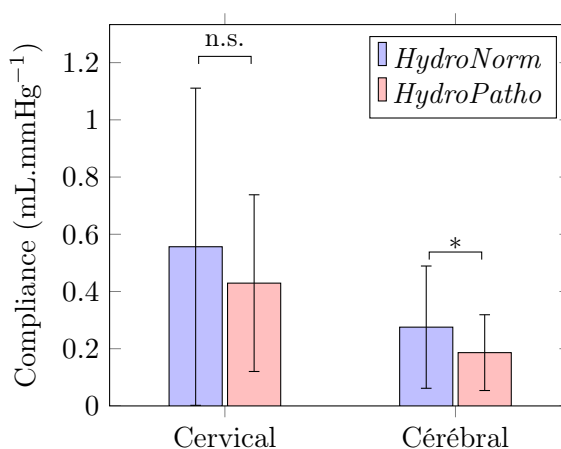


FIGURE 11.10 – Catégorisation de la population. Compliance.

	Débit veineux moyen	ICVC	Compliance
Cervical	–	+	+
Cérébral	+	–	–

Différences significatives entre les groupes *HydroNorm* et *HydroPatho* :

	Amplitude de la PIC	Stroke volume de LCS	Compliance
<i>HydroNorm</i>	–	+	+
<i>HydroPatho</i>	+	–	–

11.5 Résultats complémentaires

On remarque, Fig. 11.11, les corrélations entre les différents paramètres artériels (moyenne, stroke volume) aux niveaux cervical et cérébral.

La corrélation entre les mesures au niveau cervical et les mesures au niveau cérébral, que ce soit pour la valeur moyenne ou pour le stroke volume, montre que l'extraction des données depuis les IRM de flux est correctement réalisée : toutes les artères ont bien été analysées.

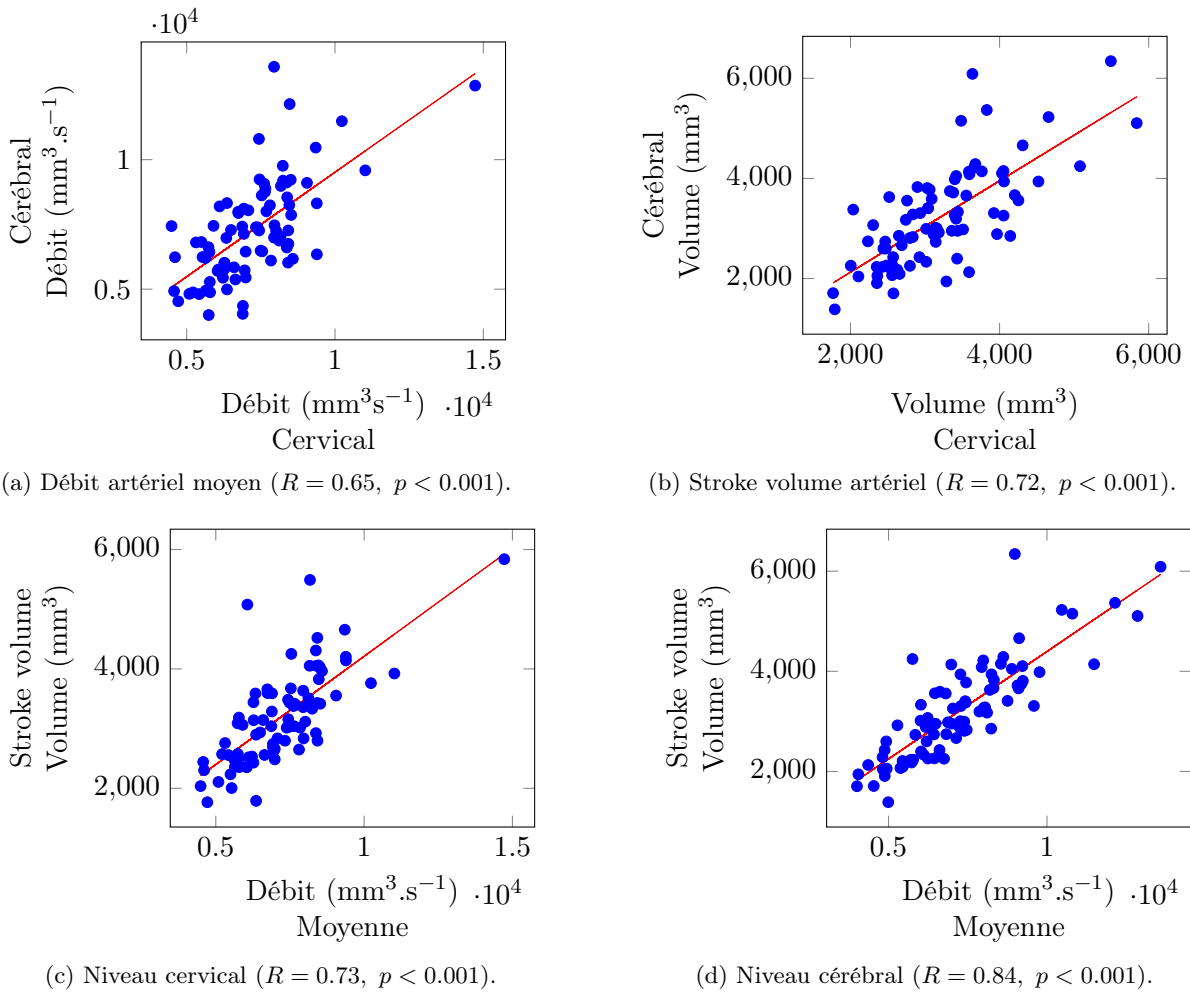


FIGURE 11.11 – **Flux artériel.** Relations entre les amplitudes et les stroke volumes aux niveaux cervical et cérébral.

La corrélation entre la valeur moyenne et le stroke volume, au niveau cervical comme au niveau cérébral, montre que le stroke volume est principalement dépendant de la valeur moyenne du flux.

On voit, Fig. 11.12, les corrélations entre les différents paramètres veineux (moyenne, stroke volume) aux niveaux cervical et cérébral.

Comme on observe le flux veineux corrigé, on s'attend à obtenir les mêmes valeurs que pour les flux artériels.

On retrouve une bonne corrélation entre le niveau cervical et le niveau cérébral (due à la correction) ainsi qu'entre les valeurs moyennes et les stroke volumes.

On observe, Fig. 11.13, les corrélations entre les différents paramètres du LCS (amplitude, stroke volume) aux niveaux cervical et cérébral. On s'intéresse ici à l'amplitude du flux de LCS, car la moyenne est nulle (ou proche de 0).

Les corrélations de l'amplitude et du stroke volume au niveau cervical et au niveau cérébral montrent une répartition quasi-identique selon l'amplitude du flux d'entrée : le flux dans l'aqueduc suit une relation linéaire par rapport au flux dans le compartiment spinal ; ce résultat vient en partie confirmer le comportement obtenu dans nos simulations numériques.

Les corrélations entre l'amplitude et le stroke volume montrent que le stroke volume est très dépendant de l'amplitude du flux de LCS. Ceci est dû au fait qu'ici, le flux moyen est nul.

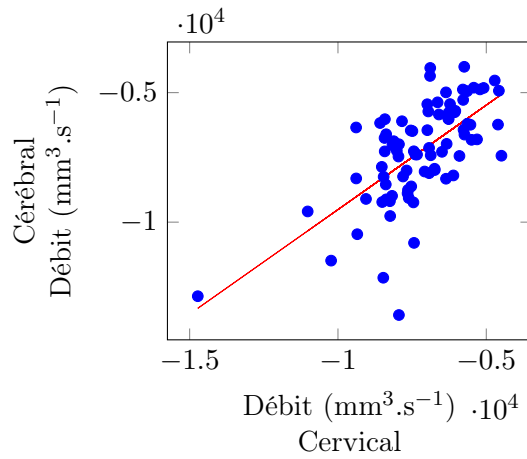
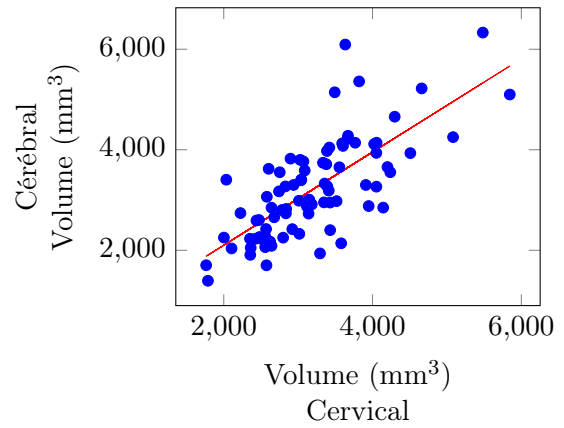
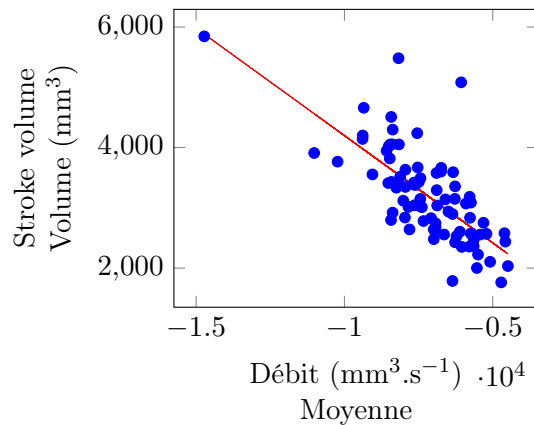
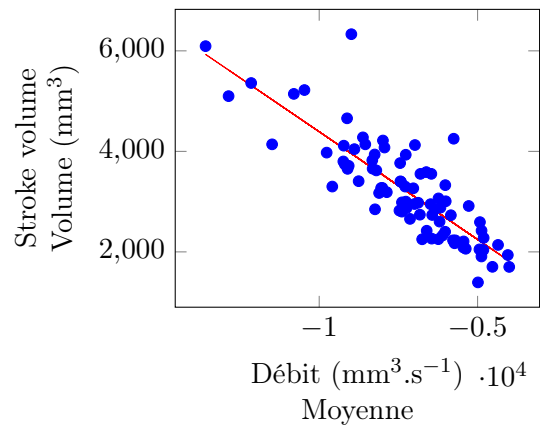
(a) Débit veineux moyen ($R = 0.65$, $p < 0.001$).(b) Stroke volume veineux ($R = 0.73$, $p < 0.001$).(c) Niveau cervical ($R = 0.72$, $p < 0.001$).(d) Niveau cérébral ($R = 0.84$, $p < 0.001$).

FIGURE 11.12 – **Flux veineux corrigé.** Relations entre les amplitudes et les stroke volumes aux niveaux cervical et cérébral.

On observe, Fig. 11.14, les corrélations entre l'amplitude du volume artério-veineux et du changement de volume intracrânien au niveau cervical et au niveau cérébral.

Dans les deux cas, la corrélation est très basse. Ceci montre qu'une partie du changement de volume du système cérébrospinal s'opère en dehors du niveau cérébral.

On constate, Fig. 11.15, les corrélations entre la pression intracrânienne et l'amplitude du flux de LCS au niveau cervical et au niveau cérébral.

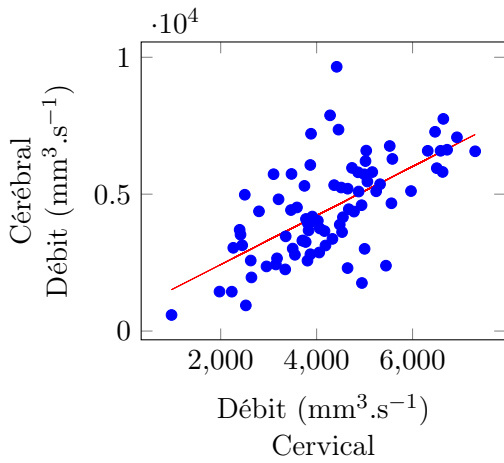
Les corrélations sont nulles dans les deux cas. Ceci montre qu'on ne peut pas, à partir simplement de l'amplitude du débit de LCS, retrouver la pression intracrânienne moyenne.

Remarque 11.3. *Le même phénomène est observé pour le stroke volume de LCS.*

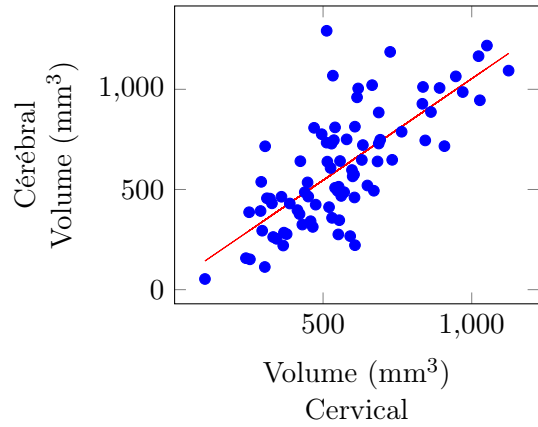
On observe, Fig. 11.16, la corrélation entre la pression intracrânienne moyenne et l'amplitude de la pression intracrânienne ainsi que la corrélation (régression de type puissance) entre la pression intracrânienne moyenne et la compliance calculée au niveau cervical.

Cette corrélation montre qu'il est possible, avec une précision toutefois assez faible, de déterminer la pression intracrânienne absolue en connaissant uniquement le gradient de pression.

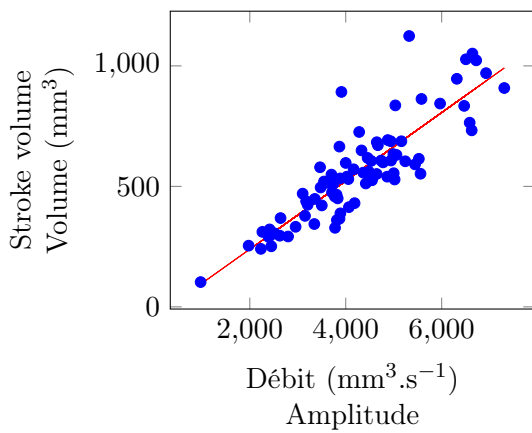
De la même façon, la compliance du système cérébrospinal est estimable, avec une assez bonne précision, à partir de l'amplitude de la pression intracrânienne uniquement.



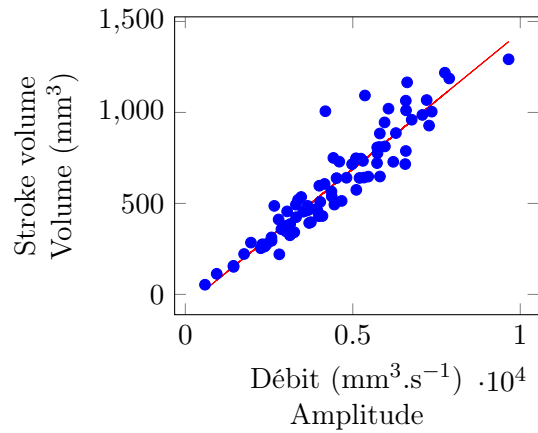
(a) Amplitude du débit de LCS ($R = 0.64$, $p < 0.001$).



(b) Stroke volume de LCS ($R = 0.74$, $p < 0.001$).

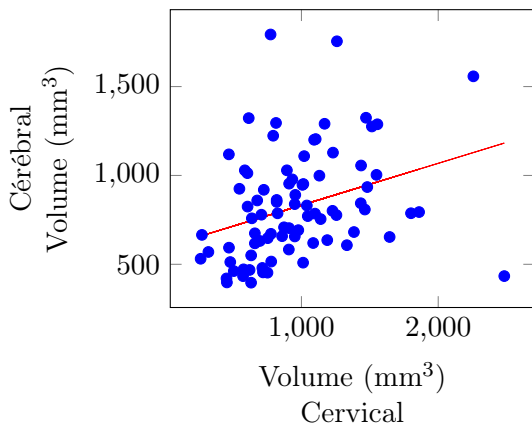


(c) Niveau cervical ($R = 0.87$, $p < 0.001$).

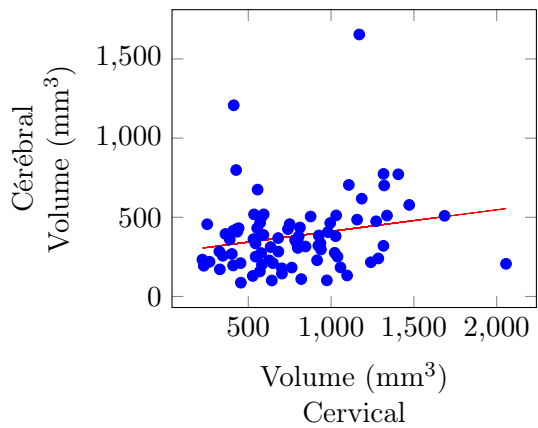


(d) Niveau cérébral ($R = 0.93$, $p < 0.001$).

FIGURE 11.13 – **Flux de LCS.** Relations entre les amplitudes et les stroke volumes aux niveaux cervical et cérébral.

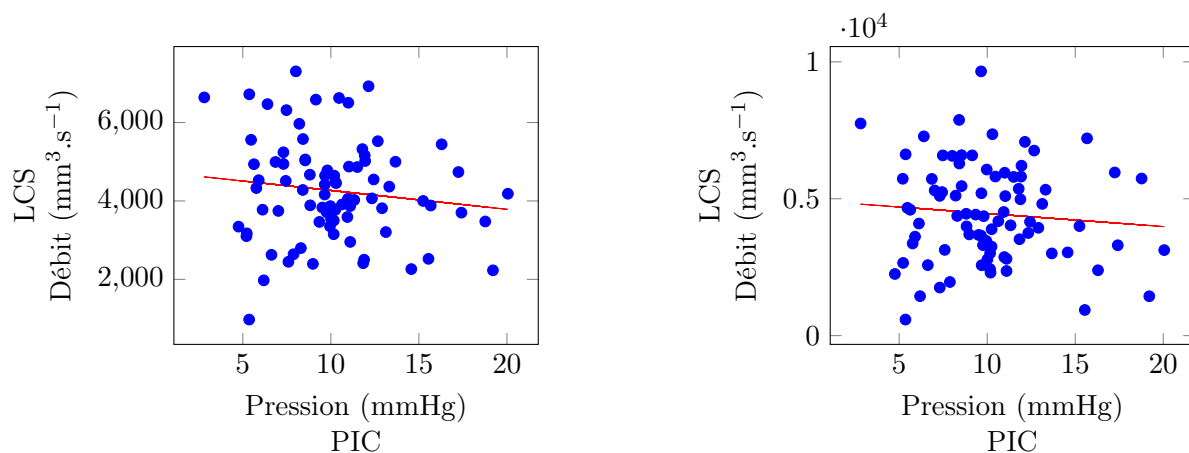


(a) Amplitude du volume artério-veineux ($R = 0.32$, $p < 0.005$).



(b) Amplitude du changement de volume intracrânien ($R = 0.21$, $p > 0.05$).

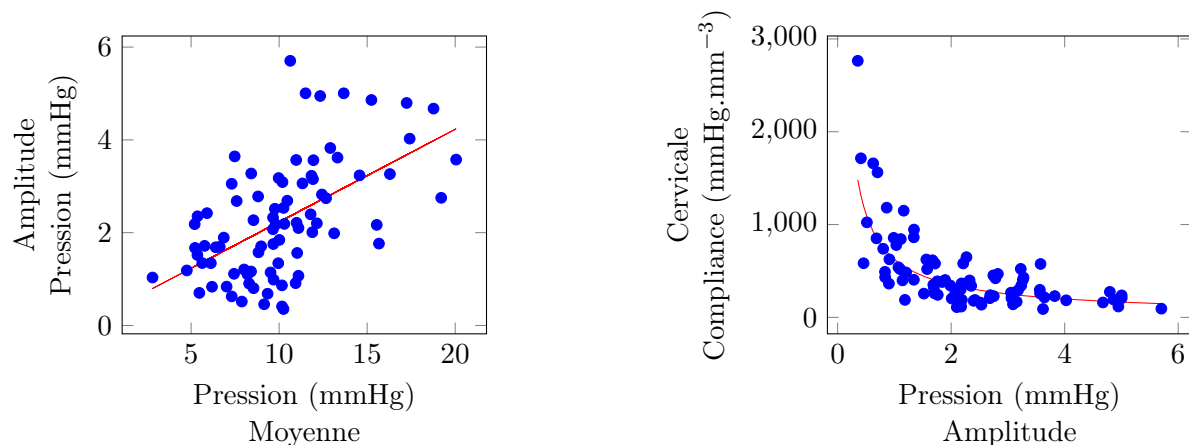
FIGURE 11.14 – **Volumes.** Relations entre les amplitudes du volume artério-veineux et du changement de volume intracrânien.



(a) Pression intracrânienne et LCS cervical ($R = 0.13$, $p > 0.1$).

(b) Pression intracrânienne et LCS cérébral ($R = 0.1$, $p > 0.1$).

FIGURE 11.15 – **Pression intracrânienne.** Relations entre la pression intracrânienne moyenne et l'amplitude du LCS.



(a) Pression intracrânienne moyenne et gradient de pression intracrânienne ($R = 0.56$, $p < 0.001$).

(b) Pression intracrânienne moyenne et compliance cervicale ($R = 0.75$, $p < 0.001$).

FIGURE 11.16 – **Pression intracrânienne.** Relations avec la pression intracrânienne moyenne.

Remarque 11.4. *Les résultats exposés ici sont extraits à partir d'une population pathologique. Ils représentent donc le comportement du système cérébrospinal dans des conditions non physiologiques : suspicion d'hydrocéphalie et population relativement âgée.*

Chapitre 12

Utilisation dans les modèles numériques

12.1 Flux

Les courbes de flux mesurés par IRM de flux et/ou calculées par le logiciel d'analyse MRiCP [33] sont utilisées dans les modèles «bifurcation», Chap. 6 et «système cérébrospinal simplifié», Chap. 7.

Dans le modèle «bifurcation», la courbe de flux de LCS (mesure IRM de flux) est directement utilisée à l'entrée du modèle, et la courbe de volume artériovoineux (calcul du logiciel d'analyse MRiCP) est utilisée dans les modèles de Windkessel.

Dans le modèle «système cérébrospinal simplifié», les courbes de flux artériel, veineux et de LCS (mesure IRM de flux) sont directement utilisées en entrée du modèle.

En IRM de flux, la résolution temporelle du signal est de 32 points par cycle cardiaque (variable selon les protocoles d'acquisition IRM); avec le logiciel d'analyse MRiCP elle est définie par l'utilisateur (typiquement 100 points par cycle cardiaque). Cette résolution n'est pas en adéquation avec les besoins du calcul numérique qui peut, selon le pas de temps choisi, nécessiter une résolution supérieure.

Afin de remédier à ce problème de résolution temporelle, on procède à une approximation par décomposition de Fourier du signal mesuré ou calculé. Cela apporte, par ailleurs, une périodicité au signal qui n'est pas parfaite lors des mesures.

Le signal discret $s^n, 0 \leq n \leq N$ représentant un cycle cardiaque, est décomposé en une série de Fourier finie :

$$s^n = a_0 + \sum_{k=1}^F a_k \cos(k\omega n) + b_k \sin(k\omega n) \quad (12.1)$$

avec $a_0, a_i, b_i, 1 \leq i \leq F$ les coefficients de Fourier, $\omega = \frac{2\pi}{T}$, où T est le temps d'un cycle cardiaque et F l'ordre de la série de Fourier.

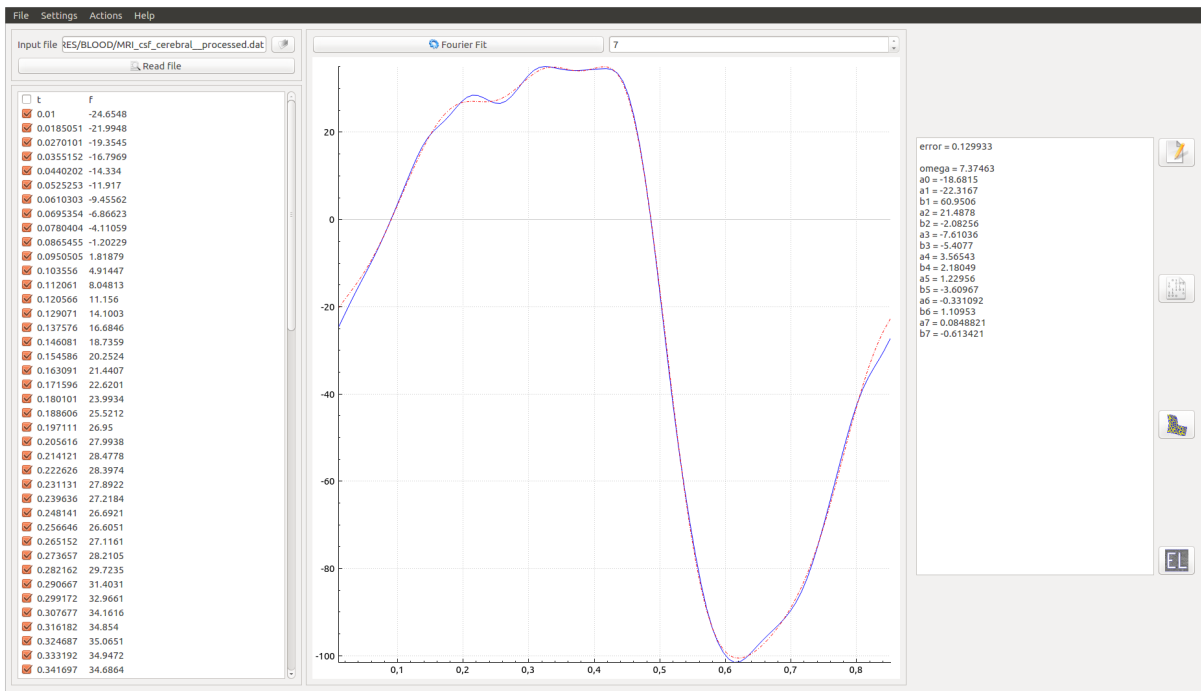
Les coefficients $a_0, a_i, b_i, 1 \leq i \leq F$ sont calculés à l'aide des formules suivantes :

$$a_0 = \frac{1}{T} \sum_{j=0}^N s^j \quad (12.2)$$

$$a_i = \frac{2}{T} \sum_{j=0}^N s^j \cos(i\omega j); \quad b_i = \frac{2}{T} \sum_{j=0}^N s^j \sin(i\omega j) \quad (12.3)$$

Un logiciel automatisé a été créé pour l'occasion [32].

Exemple 12.1. *Approximation d'une mesure de LCS au niveau cérébral à l'ordre 7.*



12.2 Pression

Les courbes de pression, obtenues par mesure invasive de la pression intracrânienne durant un test d'infusion, sont utilisées moyennées sur un cycle cardiaque à l'aide du logiciel d'analyse MRiCP dans l'état basal uniquement. Ces courbes moyennes nous servent de comparaison aux résultats de simulation obtenus.

12.3 Compliance

Le résultat de compliance obtenu dans le logiciel d'analyse MRiCP est utilisé dans le modèle «bifurcation» pour paramétrer les compliances des ventricules cérébraux et des ESA cérébraux à l'état basal.

Chapitre 13

Conclusion

Un logiciel automatisé d'analyse des données extraites des IRM de flux et des monitorings de pression intracrânienne a été créé pour former une base de données sur une population de patients.

Cette étude sur une population de patients suspectés d'hydrocéphalie nous apporte, en plus de données utiles pour la simulation numérique, un aperçu du comportement du système cérébrospinal en situation réelle pathologique.

La reproductibilité de l'analyse a été démontrée par une analyse par deux opérateurs différents sur une partie de la population.

La description de la population globale met en évidence le comportement général du système cérébrospinal, alors que la description de la population par groupes («normal» et «pathologique») permet de comprendre plus finement l'impact de la pathologie (l'hydrocéphalie) sur le comportement du système.

Cette description peut également servir de base pour la détermination des paramètres à observer pour définir plus simplement les groupes de la population, «normal» ou «pathologique».

Par ailleurs, les résultats des différents paramètres artériels, veineux, de LCS, de volumes artérioveineux, de changement de volume intracrânien et de pression intracrânienne permettent de valider les mesures, et d'envisager des pistes pour déterminer la pression intracrânienne absolue en ayant uniquement recours au gradient de pression.

Pour finir, la méthode mise en œuvre pour utiliser les données mesurées dans nos modèles numériques est détaillée.

Conclusion générale

13.1 Conclusion

Les différents algorithmes utilisés, après avoir été détaillés, ont été vérifiés à l'aide de cas tests (Taylor-Green, Ethier-Steinman) et validés à l'aide de comportements bien connus (écoulement de Poiseuille, effet annulaire de Richardson) des écoulements de fluide et de comparaisons avec des mesures IRM. Les ordres de convergence retrouvés numériquement correspondent aux attentes théoriques et les comportements simulés sont en adéquation avec les résultats théoriques. La comparaison entre les simulations et les mesures IRM confirme une bonne représentation du comportement fluide et fluide-structure des algorithmes utilisés.

L'utilité des modèles de Windkessel dans la modélisation des compartiments cérébraux pour obtenir une répartition physiologique des flux a été démontrée. La stabilité des algorithmes fluide est également validée théoriquement et numériquement pour un couplage avec des modèles de Windkessel.

Les paramètres de fréquence cardiaque et de compliance / module d'Young ont été mis en évidence comme ayant un impact sur la distribution du LCS entre les deux compartiments cérébraux : les ventricules cérébraux et les espaces subarachnoïdiens cérébraux.

Les résultats obtenus sur les deux modèles dans des cas réels montrent un comportement proche du comportement physiologique du système cérébrospinal, et des valeurs de gradients de pression correspondantes aux mesures effectuées sur des patients à l'aide d'un monitoring de pression intracrânienne. La pression intracrânienne calculée à l'aide de données d'un témoin est dans un ordre de grandeur physiologique.

Des différences significatives sont observées sur la compliance globale du système, et la répartition de celle-ci entre les ventricules cérébraux et les espaces subarachnoïdiens cérébraux, entre le cas pathologique et le cas sain pour le premier modèle. Une différence sur le module d'Young du cerveau (qui représente l'élasticité) est également observée entre les deux cas réels dans le second modèle. On peut donc suggérer un rôle prépondérant de la compliance cérébrale dans le développement de certaines pathologies.

L'analyse des données recueillies par IRM de flux et par monitoring de pression intracrânienne permet également d'avoir une description en situation réelle du comportement du système cérébrospinal que l'on peut comparer à nos résultats de simulation. Différentes corrélations et différences significatives ont été observées grâce à cette analyse. Cette analyse permet également d'ouvrir de nouveaux axes de recherche pour la détermination de la pression intracrânienne absolue de manière non invasive et du diagnostic des patients.

13.2 Perspectives

En première perspective de cette thèse, une étude de l'impact des paramètres en utilisant le second modèle, et en prenant en compte les améliorations, peut être réalisée pour observer les différences de comportement du système cérébrospinal.

Le second modèle avec entrées de LCS, artérielles et veineuses mérite une étude plus approfondie quant à sa stabilité. Il peut être également utile d'étudier ce modèle uniquement avec une entrée artérielle et observer la sortie veineuse et le va-et-vient de LCS en plaçant, par exemple, des modèles de Windkessel aux sorties veineuses et de LCS.

Il est également possible d'étudier l'impact des paramètres sur le comportement du système cérébrospinal, dans le second modèle avec entrée artérielle uniquement, en utilisant un signal d'entrée artériel proche d'un signal physiologique, Fig. 13.1.

Dans des conditions physiologiques, la partie systolique varie peu lorsque le cycle cardiaque varie ; c'est la partie diastolique qui s'étend ou se rétracte.

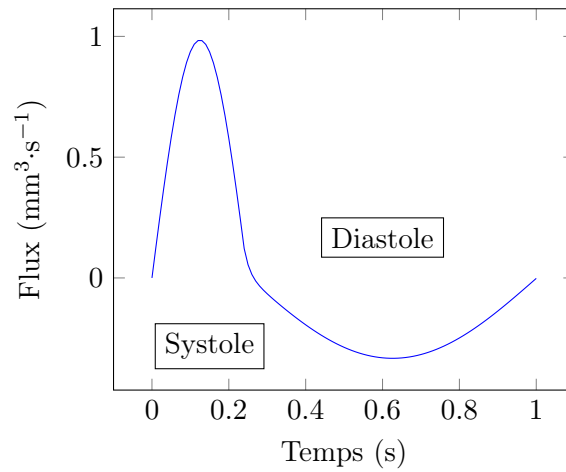


FIGURE 13.1 – **Signal artériel.** Signal physiologique comprenant une partie systolique et une partie diastolique.

Finalement, une étude similaire avec des modèles en 3 dimensions pourrait être réalisée afin de s’assurer que le comportement obtenu est le même. Une géométrie réaliste peut également être utilisée. Cette option n’a pas été explorée dans cette thèse à cause des temps de calculs très élevés des simulations 3D. Un tel problème peut être résolu avec l’algorithme de *splitting* parallélisé (fluide seul) ou avec l’algorithme d’interaction fluide-structure parallèle de FEEL++.

Concernant l’étude de la stabilité, une étude plus complète reprenant l’existence, l’unicité et la stabilité du couplage entre modèle 2D/3D et modèle 0D avec des conditions mêlées pourrait être faite.

Pour ce qui est de la validation croisée IRM/simulation numérique des algorithmes, une précision supérieure pourrait être amenée en modélisant les longueurs de tubes en entrée/sortie du fantôme par des modèles 1D. Une telle modélisation ferait intervenir des conditions de Neumann en entrée et en sortie du fantôme, pour lesquels une attention particulière doit être portée à la stabilité numérique de l’algorithme. L’utilisation de la méthode des caractéristique dans une gamme de pression physiologique permettrait de garantir la stabilité.

Concernant l’analyse des données de flux et de pression, une thèse dans le domaine biomédical commence sur ce sujet et l’étude va donc être approfondie. Une avancée significative pourrait consister en une simplification de la classification des sujets sains et pathologiques, uniquement en ayant recours au profil hydrodynamique des patients suspectés d’hydrocéphalie (sans étude biochimique, psychologique, ...), et donc de manière non invasive.

Annexes

Annexe A

Abréviations

ESA : Espaces SubArachnoïdiens.

IRM : Imagerie par Résonance Magnétique.

ARM : Angiographie par Résonance Magnétique.

LCS : Liquide CérébroSpinal.

PIC : Pression IntraCrânienne.

CVI : Changement de Volume Intracrânien.

ALE : Arbitrairement Lagrangien Eulérien.

bpm : battements par minute.

SAS: SubArachnoid Spaces.

MRI: Magnetic Resonance Imaging.

MRA: Magnetic Resonance Angiography.

CSF: CerebroSpinal Fluid.

ICP: IntraCranial Pressure.

IVC: Intracranial Volume Change.

BDF: Backward Differential Formula.

ALE: Arbitrary Lagrangian Eulerian.

bpm: beats per minute.

ROI: Region Of Interest.

SU: Speed-Up.

Annexe B

Publications et congrès

B.1 Publications

Numerical Cerebrospinal System Modeling in Fluid-Structure Interaction

S. Garnotel, S. Salmon, O. Balédent

(En préparation)

Numerical Modeling of the Intracranial Pressure using Windkessel Models

S. Garnotel, S. Salmon, O. Balédent

(En préparation)

Phantom Project: Development and Validation of the Pipeline From MRA Acquisition to MRA Simulation

A. Ancel, A. Fortin, S. Garnotel, O. Miraucourt, R. Tarabay

ESAIM: Proceedings and Surveys, 2016

Modélisation du Système Cérébrospinal en Interaction Fluide-Structure

S. Garnotel, S. Salmon, O. Balédent

CANUM Proceedings, 2016

Integration of CSF pressure and flow data to further investigate nature of cerebrospinal compliance in hydrocephalus

S. Garnotel, E. Schmidt, Z. Czosnyka, M. Baronchini, G. Pagé, O. Balédent

Fluids and Barriers of the CNS, 2015

A Windkessel Model for the Cerebrospinal Fluid System

S. Garnotel, S. Salmon, O. Balédent

4th International Conference on Computational & Mathematical Biomedical Engineering Proceedings, 2015

B.2 Congrès

Meeting of the International Society for Hydrocephalus and CSF Disorders, 8 au 10 octobre 2016, Carthagène (Colombie)

Présentation «*Numerical Modeling of the Cerebrospinal System: MRI and Simulation to Determine Hydrocephalus Mechanisms*»

16th International Symposium on Intracranial Pressure and Neuromonitoring, 28 juin au 2 juillet 2016, Boston (USA)

Poster «*Numerical Cerebrospinal System Modeling in Fluid-Structure Interaction*»

CANUM 2016, 9 au 13 mai 2016, Obernai

Présentation «*Modélisation du Système Cérébrospinal en Interaction Fluide-Structure*»

Meeting of the International Society for Hydrocephalus and CSF Disorders, 19 au 21 septembre 2015, Banff (Canada)

Présentation «*Integration of CSF Pressure and Flow Data to Further Investigate Nature of Cerebrospinal Compliance in Hydrocephalus*»

3rd Cerebrospinal Fluid Dynamics Symposium, 9 et 10 juillet 2015, Amiens

Poster «*Intracranial Compliance Changes During Cardiac Cycle in Hydrocephalus Patients*»

International Conference on Computational and Mathematical Biomedical Engineering, 29 juin au 1^{er} juillet 2015, Cachan

Présentation «*A Windkessel model for the cerebrospinal fluid system*»

Congrès SMAI 2015, 8 au 12 juin 2015, Les Karellis

Présentation «*Modélisation numérique de la pression intracrânienne à l'aide de modèles de Windkessel*»

Meeting of the International Society for Hydrocephalus and CSF Disorders, 5 au 8 septembre 2014, Bristol (Royaume-Uni)

Poster «*CSF oscillations in the cranium: toward the ventricles or the subarachnoid spaces*»

B.3 Écoles d'été

SEME 2016, 27 juin au 1^{er} juillet 2016, Centrale Marseille

Simulation spatio-temporelle d'images de perfusion par rehaussement de contraste (Oléa Médical)

CEMRACS 2015, 19 juillet au 7 août 2015, CIRM, Marseille

Coupling Multi-Physics Models Involving Fluids Phantom Project

Annexe C

Algorithmes FREEFEM++

Tous les algorithmes fluide sont présentés dans le cas d'un tube 2D ou 3D avec une entrée sinusoïdale de profil de Poiseuille.

L'algorithme fluide-structure est donné dans un tube 2D ou 3D avec une paroi déformable.

C.1 Stokes

C.1.1 2D

```
//Construction du Maillage
int nn = 10;

real L = 5.;
real D = 1.;

int wall = 1;
int inlet = 2;
int outlet = 3;

border b1(t=0., 1.) {x=L*t; y=0.; label=wall;};
border b2(t=0., 1.) {x=L; y=D*t; label=outlet;};
border b3(t=0., 1.) {x=L-L*t; y=D; label=wall;};
border b4(t=0., 1.) {x=0.; y=D-D*t; label=inlet;};

int nnL = max(2., L*nn);
int nnD = max(2., D*nn);

mesh Th = buildmesh(b1(nnL) + b2(nnD) + b3(nnL) + b4(nnD));
plot(Th, cmm="Maillage");

//Espaces éléments finis
fespace Uh(Th, P2);
Uh u1, u2;
Uh ulold, uold;
Uh v1, v2;

fespace Ph(Th, P1);
Ph p;
Ph q;

//Définition du problème
real rho = 1.;
real mu = 1.;

real T = 1.;
real dt = 1.e-2;

func f1 = 0.;
```

```

func f2 = 0.;

real t = 0.;

problem S ([u1, u2, p],[v1, v2, q])
  = int2d(Th) (
    (1./dt) * ( u1*v1 + u2*v2 )
    + mu * (
      dx(u1)*dx(v1) + dy(u1)*dy(v1)
      + dx(u2)*dx(v2) + dy(u2)*dy(v2)
    )
    - p*q*1.e-9
    - p*dx(v1) - p*dy(v2)
    - dx(u1)*q - dy(u2)*q
  )
  - int2d(Th) (
    (1./dt) * (u1old*v1 + u2old*v2)
  )
  - int2d(Th) ( f1*v1 + f2*v2 )
  + on(inlet, u1=cos(2.*pi*t)*(1.-(y-D/2.)^2/(D/2.)^2), u2=0.)
  + on(wall, u1=0., u2=0.)
  ;

//Résolution
for (int i = 0; i < T/dt; i++){
  //Mise à jour
  t += dt;
  u1old = u1;
  u2old = u2;

  //Résolution
  S;
  plot(p, cmm="Pression - t = "+t, WindowIndex=1);
  plot([u1, u2], cmm="Vitesse - t = "+t, WindowIndex=2);
}

```

Détails de l'algorithme :

- construction du maillage en tube en définissant ses bords,
- définition des espaces éléments finis, ici $\mathbb{P}^2\mathbb{P}^1$ afin de respecter la condition Inf-Sup, et des variables du problème,
- définition du problème de Stokes,
- résolution à chaque pas de temps et affichage de la solution calculée.

La résolution du problème s'exécute en 12.5 s. La vitesse et la pression au temps final sont données Fig. C.1.

C.1.2 3D

```

load "msh3"

//Construction du Maillage
int nn = 5;

real L = 5.;
real D = 1.;
real R = D/2.;

int wall = 1;
int inlet = 2;
int outlet = 3;

border b0(t=0., 2.*pi){x=R*cos(t); y=R*sin(t); label=0;};

int nnL = max(2., L*nn);
int nnR = max(2., 2.*pi*R*nn);

mesh Th0 = buildmesh(b0(nnR));

```

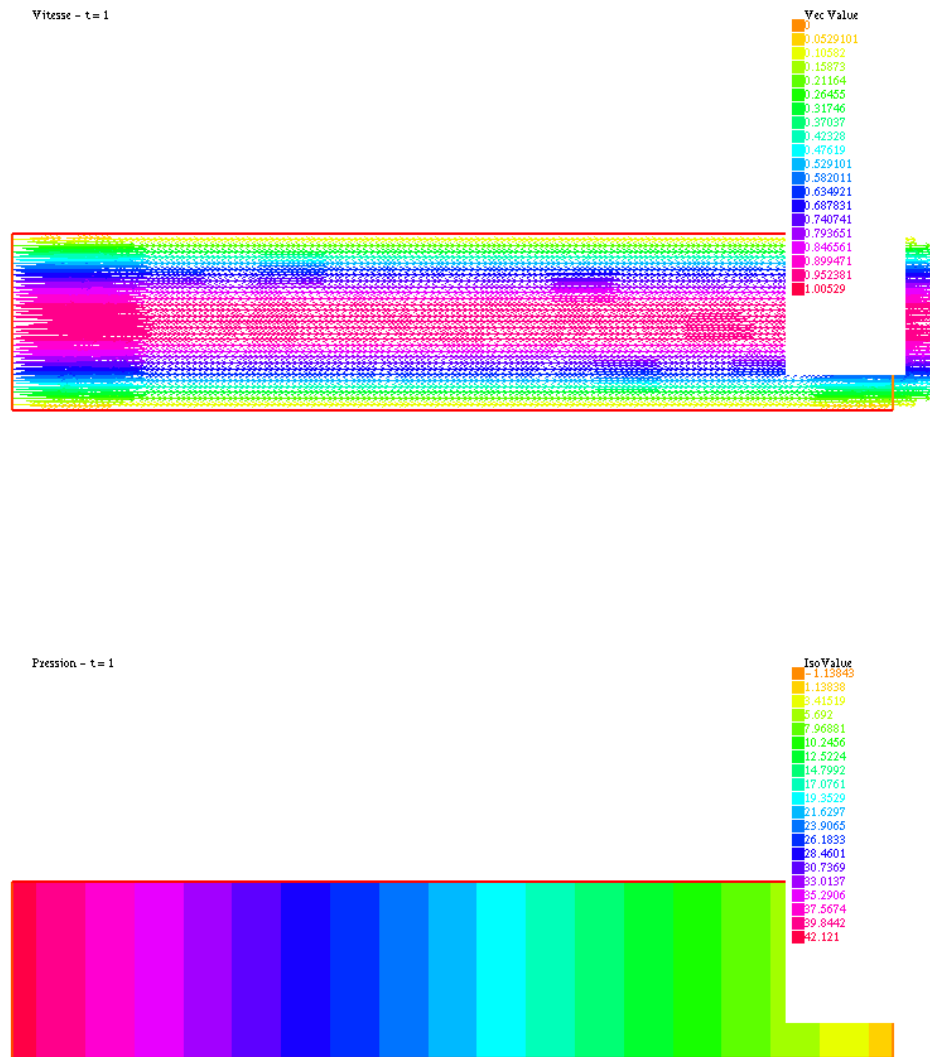


FIGURE C.1 – Vitesse et pression du fluide au temps final. Stokes 2D.

```

int[int] ldown = [0, inlet];
int[int] lmid = [0, wall];
int[int] lup = [0, outlet];
mesh3 Th = buildlayers(Th0, nnL, zbound=[0., L], labeldown=ldown, labelmid=lmid, labelup=lup);
savemesh(Th, "medit/maillage.mesh");
plot(Th, cmm="Maillage");

//Espaces éléments finis
fespace Uh(Th, P2);
Uh u1, u2, u3;
Uh ulold, u2old, u3old;
Uh v1, v2, v3;

fespace Ph(Th, P1);
Ph p;
Ph q;

```

```

//Définition du problème
real rho = 1.;
real mu = 1.;

real T = 1.;
real dt = 1.e-2;

func f1 = 0.;
func f2 = 0.;
func f3 = 0.;

real t = 0.;

problem S ([u1, u2, u3, p],[v1, v2, v3, q])
= int3d(Th) (
    (1./dt) * ( u1*v1 + u2*v2 + u3*v3 )
    + mu * (
        dx(u1)*dx(v1) + dy(u1)*dy(v1)
        + dx(u2)*dx(v2) + dy(u2)*dy(v2)
        + dx(u3)*dx(v3) + dy(u3)*dy(v3)
    )
    - p*q*1.e-9
    - p*dx(v1) - p*dy(v2) - p*dz(v3)
    - dx(u1)*q - dy(u2)*q - dz(u3)*q
)
- int3d(Th) (
    (1./dt) * (u1old*v1 + u2old*v2 + u3old*v3)
)
- int3d(Th) ( f1*v1 + f2*v2 + f3*v3 )
+ on(inlet, u1=0., u2=0., u3=cos(2.*pi*t)*(1.-(x^2+y^2)/R^2))
+ on(wall, u1=0., u2=0., u3=0.)
;

//Résolution
for (int i = 0; i < T/dt; i++){
    //Mise à jour
    t += dt;
    u1old = u1;
    u2old = u2;
    u3old = u3;

    //Résolution
    S;
    plot(p, cmm="Pression - t = "+t, WindowIndex=1);
    plot([u1, u2, u3], cmm="Vitesse - t = "+t, WindowIndex=2);

    //Sauvegarde
    {
        Ph uu1, uu2, uu3;

        uu1 = u1;
        uu2 = u2;
        uu3 = u3;

        ofstream file("medit/vitesse_"+i+".bb");
        file << "3 3 " << uu1[.n] << " 2 " << endl;
        for (int j = 0; j < uu1[.n]; j++){
            file << uu1[][j] << "\t" << uu2[][j] << "\t" << uu3[][j] << endl;
        }
    }
    {
        ofstream file("medit/pression_"+i+".bb");
        file << "3 1 " << p[.n] << " 2 " << endl;
        for (int j = 0; j < p[.n]; j++){
            file << p[][j] << endl;
        }
    }
}

```

La programmation de l'algorithme en 3 dimensions est similaire au cas 2 dimensions. Une sauvegarde des résultats à chaque pas de temps, pour une visualisation ultérieure sur Medit [30], est ajoutée au code.

La résolution du problème s'exécute en 203.7 s. La vitesse et la pression au temps final sont données Fig. C.2.

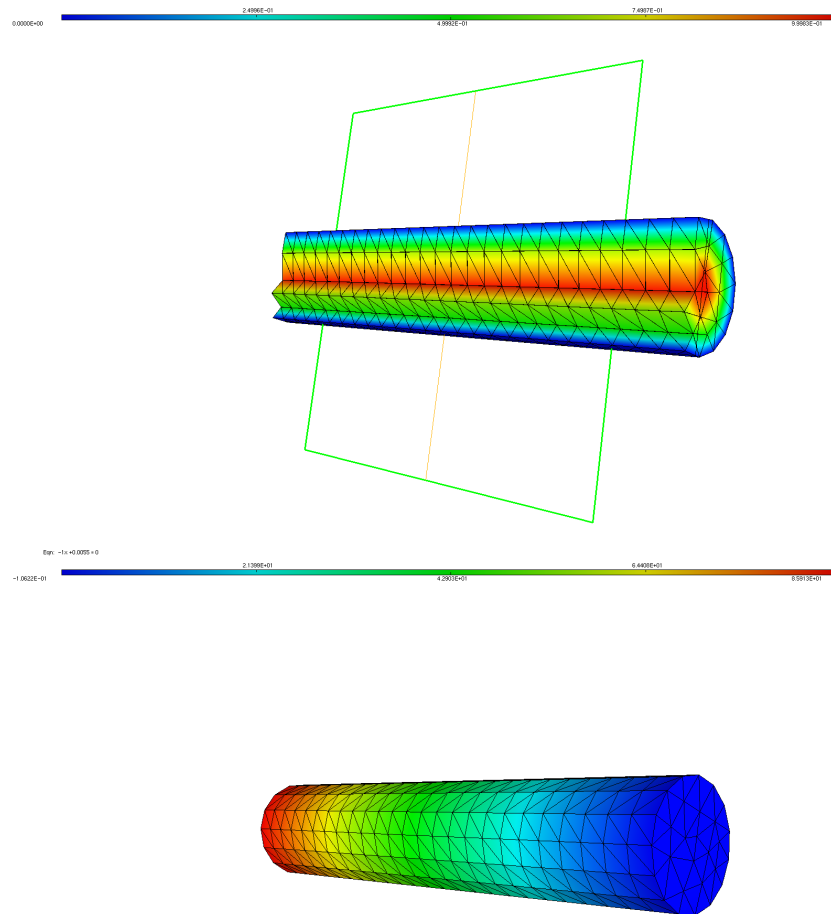


FIGURE C.2 – Vitesse et pression du fluide au temps final. Stokes 3D.

Le temps de calcul est très largement supérieur dans le cas 3D. On a un rapport de :

$$\frac{T_{3D}}{T_{2D}} = 16.3 \quad (\text{C.1})$$

Cela signifie que pour une dimension supplémentaire, on a un temps de calcul 16 fois supérieur.

C.2 Navier-Stokes, méthode des caractéristiques

C.2.1 2D

```
//Construction du Maillage
int nn = 10;
```

```

real L = 5.;
real D = 1.;

int wall = 1;
int inlet = 2;
int outlet = 3;

border b1(t=0., 1.){x=L*t; y=0.; label=wall;};
border b2(t=0., 1.){x=L; y=D*t; label=outlet;};
border b3(t=0., 1.){x=L-L*t; y=D; label=wall;};
border b4(t=0., 1.){x=0.; y=D-D*t; label=inlet;};

int nnL = max(2., L*nn);
int nnD = max(2., D*nn);

mesh Th = buildmesh(b1(nnL) + b2(nnD) + b3(nnL) + b4(nnD));
plot(Th, cmm="Maillage");

//Espaces éléments finis
fespace Uh(Th, P2);
Uh u1, u2;
Uh ulold, u2old;
Uh v1, v2;

fespace Ph(Th, P1);
Ph p;
Ph q;

//Définition du problème
real rho = 1.;
real mu = 1.;

real T = 1.;
real dt = 1.e-2;

func f1 = 0.;
func f2 = 0.;

real t = 0.;

problem NS ([u1, u2, p],[v1, v2, q])
= int2d(Th) (
    (1./dt) * ( u1*v1 + u2*v2 )
    + mu * (
        dx(u1)*dx(v1) + dy(u1)*dy(v1)
        + dx(u2)*dx(v2) + dy(u2)*dy(v2)
    )
    - p*q*1.e-9
    - p*dx(v1) - p*dy(v2)
    - dx(u1)*q - dy(u2)*q
)
- int2d(Th) (
    (1./dt) * convect([u1old,u2old], -dt, u1old) * v1
    + (1./dt) * convect([u1old,u2old], -dt, u2old) * v2
)
- int2d(Th) ( f1*v1 + f2*v2 )
+ on(inlet, u1=cos(2.*pi*t)*(1.-(y-D/2.)^2/(D/2.)^2), u2=0.)
+ on(wall, u1=0., u2=0.)
;

//Résolution
for (int i = 0; i < T/dt; i++){
    //Mise à jour
    t += dt;
    ulold = u1;
    u2old = u2;

    //Résolution
    NS;
}

```



```

    plot(p, cmm="Pression - t = "+t, WindowIndex=1);
    plot([u1, u2], cmm="Vitesse - t = "+t, WindowIndex=2);
}

```

Détails de l'algorithme :

- construction du maillage en tube en définissant ses bords,
- définition des espaces éléments finis, ici $\mathbb{P}^2\mathbb{P}^1$ afin de respecter la condition Inf-Sup, et des variables du problème,
- définition du problème de Navier-Stokes (opérateur `convect` pour la méthode des caractéristiques),
- résolution à chaque pas de temps et affichage de la solution calculée.

La résolution du problème s'exécute en 14 s. La vitesse et la pression au temps final sont données Fig. C.3.

C.2.2 3D

```

load "msh3"

//Construction du Maillage
int nn = 5;

real L = 5.;
real D = 1.;
real R = D/2.;

int wall = 1;
int inlet = 2;
int outlet = 3;

border b0(t=0., 2.*pi){x=R*cos(t); y=R*sin(t); label=0;};

int nnL = max(2., L*nn);
int nnR = max(2., 2.*pi*R*nn);

mesh Th0 = buildmesh(b0(nnR));

int[int] ldown = [0, inlet];
int[int] lmid = [0, wall];
int[int] lup = [0, outlet];
mesh3 Th = buildlayers(Th0, nnL, zbound=[0., L], labeldown=ldown, labelmid=lmid, labelup=lup);
savemesh(Th, "medit/maillage.mesh");
plot(Th, cmm="Maillage");

//Espaces éléments finis
fespace Uh(Th, P2);
Uh u1, u2, u3;
Uh ulold, u2old, u3old;
Uh v1, v2, v3;

fespace Ph(Th, P1);
Ph p;
Ph q;

//Définition du problème
real rho = 1.;
real mu = 1.;

real T = 1.;
real dt = 1.e-2;

func f1 = 0.;
func f2 = 0.;
func f3 = 0.;

real t = 0.;

```

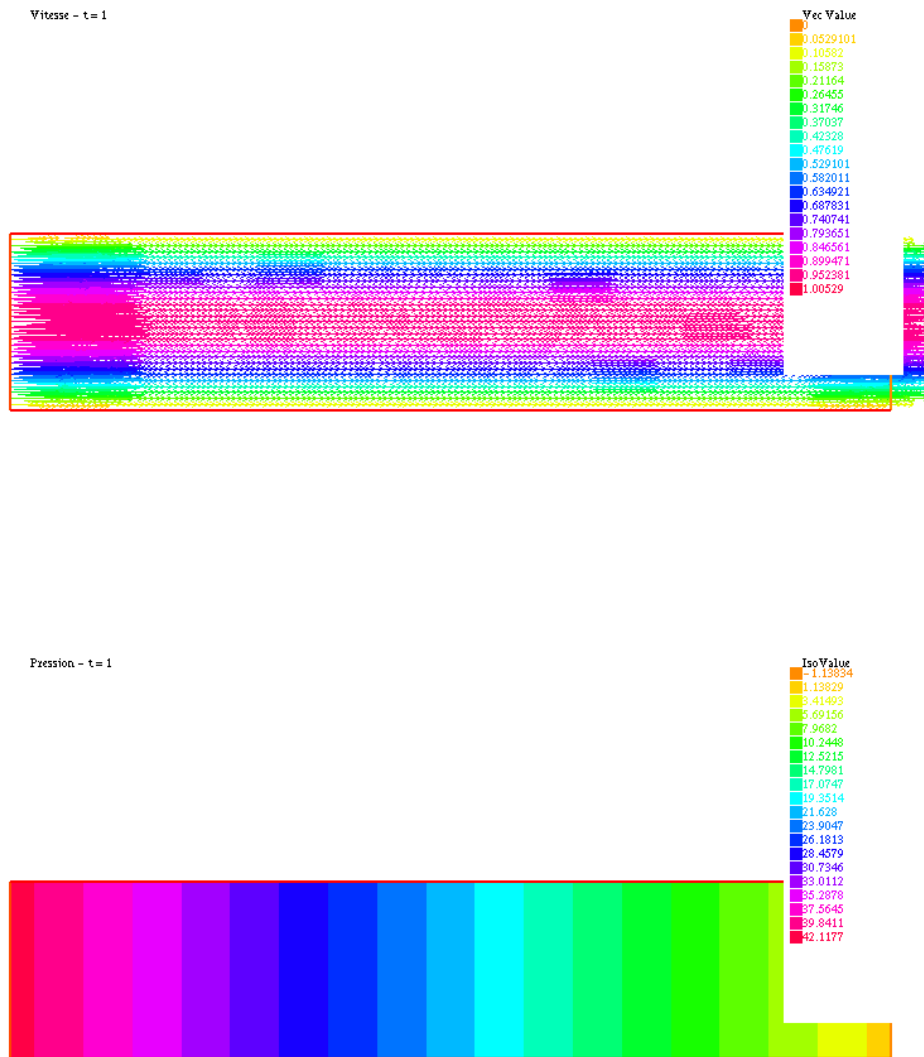


FIGURE C.3 – Vitesse et pression du fluide au temps final. Méthode des caractéristiques 2D.

```

problem NS ([u1, u2, u3, p],[v1, v2, v3, q])
= int3d(Th) (
  (1./dt) * ( u1*v1 + u2*v2 + u3*v3 )
  + mu * (
    dx(u1)*dx(v1) + dy(u1)*dy(v1) + dz(u1)*dz(v1)
    + dx(u2)*dx(v2) + dy(u2)*dy(v2) + dz(u2)*dz(v2)
    + dx(u3)*dx(v3) + dy(u3)*dy(v3) + dz(u3)*dz(v3)
  )
  - p*q*1.e-9
  - p*dx(v1) - p*dy(v2) - p*dz(v3)
  - dx(u1)*q - dy(u2)*q - dz(u3)*q
)
- int3d(Th) (
  (1./dt) * convect([u1old,u2old,u3old], -dt, u1old) * v1
  + (1./dt) * convect([u1old,u2old,u3old], -dt, u2old) * v2
)

```

```

        + (1./dt) * convect([u1old,u2old,u3old], -dt, u3old) * v3
    )
- int3d(Th) ( f1*v1 + f2*v2 + f3*v3 )
+ on(inlet, u1=0., u2=0., u3=cos(2.*pi*t)*(1.-(x^2+y^2)/R^2))
+ on(wall, u1=0., u2=0., u3=0.)
;

//Résolution
for (int i = 0; i < T/dt; i++){
    //Mise à jour
    t += dt;
    u1old = u1;
    u2old = u2;
    u3old = u3;

    //Résolution
    NS;
    plot(p, cmm="Pression - t = "+t, WindowIndex=1);
    plot([u1, u2, u3], cmm="Vitesse - t = "+t, WindowIndex=2);

    //Sauvegarde
    {
        Ph uu1, uu2, uu3;

        uu1 = u1;
        uu2 = u2;
        uu3 = u3;

        ofstream file("medit/vitesse_"+i+".bb");
        file << "3 3 " << uu1[.n] << " 2 " << endl;
        for (int j = 0; j < uu1[.n]; j++){
            file << uu1[][j] << "\t" << uu2[][j] << "\t" << uu3[][j] << endl;
        }
    }
    {
        ofstream file("medit/pression_"+i+".bb");
        file << "3 1 " << p[.n] << " 2 " << endl;
        for (int j = 0; j < p[.n]; j++){
            file << p[][j] << endl;
        }
    }
}

```

On procède de la même façon que pour l'algorithme en 2 dimensions. On ajoute une sauvegarde des résultats à chaque pas de temps pour une visualisation ultérieure sur Medit [30].

La résolution du problème s'exécute en 226.4 s. La vitesse et la pression au temps final sont données Fig. C.4.

Le temps de calcul est très largement supérieur dans le cas 3D. On a un rapport de :

$$\frac{T_{3D}}{T_{2D}} = 16.2 \quad (\text{C.2})$$

Cela signifie que pour une dimension supplémentaire, on a un temps de calcul 16 fois supérieur (du même ordre de grandeur que pour le problème de Stokes).

C.3 Navier-Stokes, algorithme d'Uzawa

C.3.1 2D

```

//Construction du Maillage
int nn = 10;

real L = 5.;
real D = 1.;

```

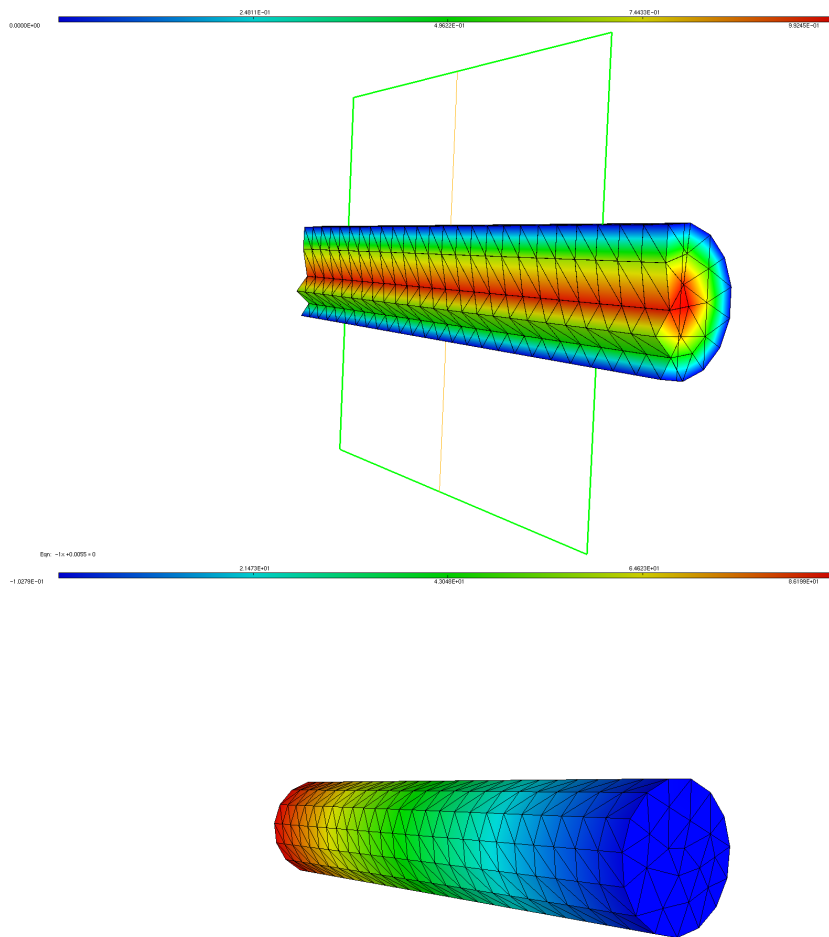


FIGURE C.4 – Vitesse et pression du fluide au temps final. Méthode des caractéristiques 3D.

```

int wall = 1;
int inlet = 2;
int outlet = 3;

border b1(t=0., 1.){x=L*t; y=0.; label=wall;};
border b2(t=0., 1.){x=L; y=D*t; label=outlet;};
border b3(t=0., 1.){x=L-L*t; y=D; label=wall;};
border b4(t=0., 1.){x=0.; y=D-D*t; label=inlet;};

int nnL = max(2., L*nn);
int nnD = max(2., D*nn);

mesh Th = buildmesh(b1(nnL) + b2(nnD) + b3(nnL) + b4(nnD));
plot(Th, cmm="Maillage");

//Espaces éléments finis
fespace Uh(Th, P2);
Uh u1, u2;

fespace Ph(Th, P1);
Ph p;

```

```

//Définition du problème
real rho = 1.;
real mu = 1.;

real T = 1.;
real dt = 1.e-2;

func f1 = 0.;
func f2 = 0.;

real t = 0.;

macro grad(a) [dx(a), dy(a)] //

varf vP (p, q)
  = int2d(Th) (
      grad(p)' * grad(q)
    )
  + int1d(Th, outlet) (
      1000. * p * q
    )
  ;

varf vM (p, q)
  = int2d(Th, qft=qf2pT) (
      p * q
    )
  ;

varf vA (u, v)
  = int2d(Th) (
      (1./dt) * u * v
      + mu * (grad(u)' * grad(v))
    )
  + on(wall, inlet, u=0.)
  ;

varf vB1 (p, v)
  = int2d(Th) (
      p * dx(v)
    )
  ;

varf vB2 (p, v)
  = int2d(Th) (
      p * dy(v)
    )
  ;

varf vOn1 (u, v)
  = on(inlet, u = cos(2.*pi*t)*(1.-(y-D/2.)^2/(D/2.)^2))
  + on(wall, u = 0.)
  ;

varf vOn2 (u, v)
  = on(inlet, u = 0.)
  + on(wall, u = 0.)
  ;

varf vNL1(u, v)
  = int2d(Th) (
      (1./dt) * convect([u1, u2], -dt, u1) * v
    )
  + int2d(Th) (
      f1 * v
    )
  + vOn1
  ;

varf vNL2(u, v)
  = int2d(Th) (
      (1./dt) * convect([u1, u2], -dt, u2) * v
  )

```

```

    )
    + int2d(Th) (
        f2 * v
    )
    + vOn2
;

matrix P = vP(Ph, Ph, solver=sparsesolver);
matrix M = vM(Ph, Ph, solver=sparsesolver);

matrix A = vA(Uh, Uh, solver=sparsesolver);
matrix B1 = vB1(Ph, Uh);
matrix B2 = vB2(Ph, Uh);

real[int] bd1 = vOn1(0, Uh);
real[int] bd2 = vOn2(0, Uh);

func real[int] Uzawa(real[int] pp) {
    real[int] b1 = bd1;      b1 += B1*pp;
    real[int] b2 = bd2;      b2 += B2*pp;

    u1[] = A^-1 * b1;
    u2[] = A^-1 * b2;

    pp = B1' * u1[];
    pp += B2' * u2[];
    pp = -pp;

    return pp;
}

func real[int] Precon(real[int] p) {
    real[int] pP = P^-1 * p;
    real[int] pM = M^-1 * p;
    real[int] pp = (1./dt) * pP + mu * pM;
    return pp;
}

//Résolution
real eps = 1.e-6;
for (int i = 0; i < T/dt; i++){
    //Mise à jour
    t += dt;
    bd1 = vNL1(0, Uh);
    bd2 = vNL2(0, Uh);

    //Résolution
    LinearCG(Uzawa, p[], precon=Precon, nbiter=1e3, veps=eps);
    eps = -abs(eps);
    plot(p, cmm="Pression - t = "+t, WindowIndex=1);
    plot([u1, u2], cmm="Vitesse - t = "+t, WindowIndex=2);
}

```

Détails de l'algorithme (définition du problème uniquement) :

- définition des formulations variationnelles (type `varf`) pour le préconditionneur,
- définition des formulations variationnelles du problème, de la condition d'incompressibilité et des conditions aux limites,
- construction des matrices représentatives des formes linéaires et bilinéaires (type `matrix`),
- définition des fonctions de résolution par l'algorithme d'Uzawa et de préconditionnement.

La résolution du problème s'exécute en 2.8 s. La vitesse et la pression au temps final sont données Fig. C.5.

Par rapport à la méthode des caractéristiques, qui donne le même résultat, le facteur d'accélération (SU , *speed-up*) est donné par :

$$SU = \frac{\text{Temps caractéristiques 2D}}{\text{Temps Uzawa 2D}} = \frac{14 \text{ s}}{2.8 \text{ s}} = 5 \quad (\text{C.3})$$

L'algorithme d'Uzawa 2D est donc 5 fois plus rapide que l'algorithme des caractéristiques 2D.

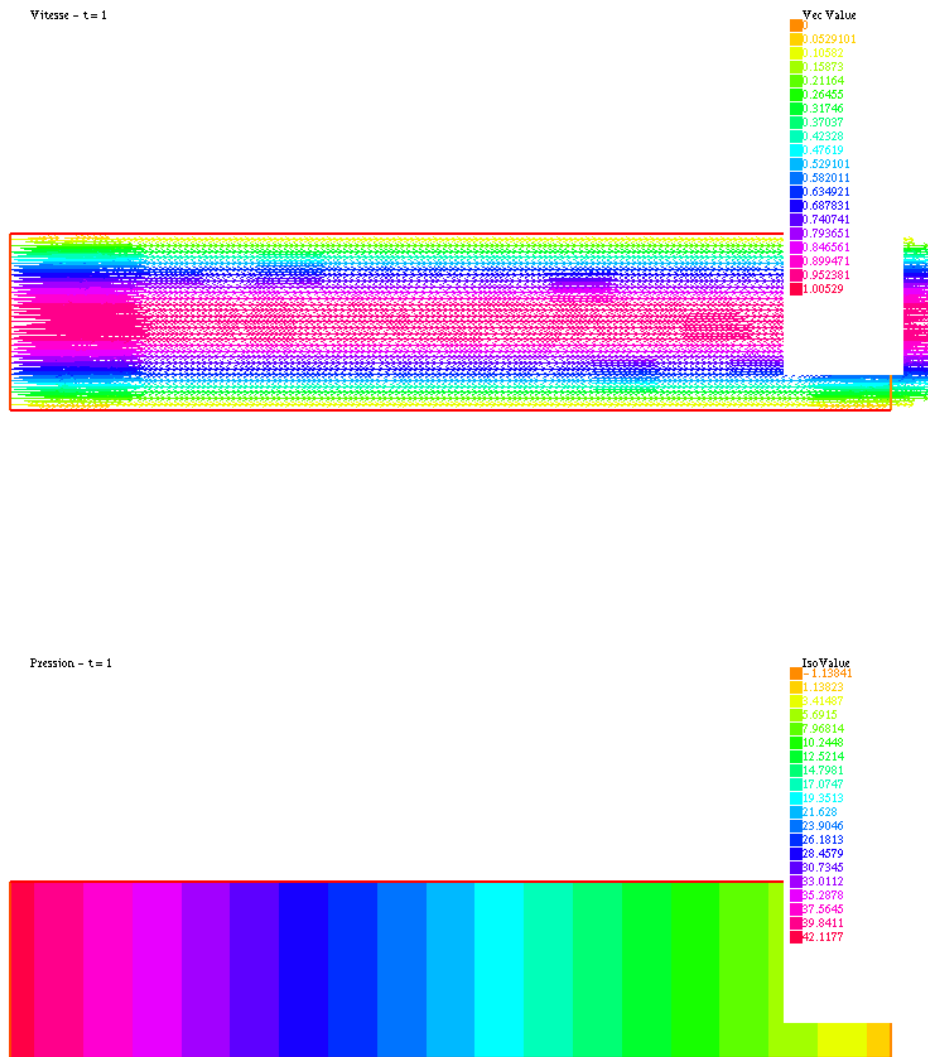


FIGURE C.5 – Vitesse et pression du fluide au temps final. Algorithme d’Uzawa 2D.

C.3.2 3D

```
load "msh3"

//Construction du Maillage
int nn = 5;

real L = 5.;
real D = 1.;
real R = D/2.;

int wall = 1;
int inlet = 2;
int outlet = 3;

border b0(t=0., 2.*pi){x=R*cos(t); y=R*sin(t); label=0;};
```

```

int nnL = max(2., L*nn);
int nnR = max(2., 2.*pi*R*nn);

mesh Th0 = buildmesh(b0(nnR));

int[int] ldown = [0, inlet];
int[int] lmid = [0, wall];
int[int] lup = [0, outlet];
mesh3 Th = buildlayers(Th0, nnL, zbound=[0., L], labeldown=ldown, labelmid=lmid, labelup=lup);
savemesh(Th, "medit/maillage.mesh");
plot(Th, cmm="Maillage");

//Espaces éléments finis
fespace Uh(Th, P2);
Uh u1, u2, u3;

fespace Ph(Th, P1);
Ph p;

//Définition du problème
real rho = 1.;
real mu = 1.;

real T = 1.;
real dt = 1.e-2;

func f1 = 0.;
func f2 = 0.;
func f3 = 0.;

real t = 0.;

macro grad(a) [dx(a), dy(a), dz(a)] //

varf vP (p, q)
  = int3d(Th) (
    grad(p)' * grad(q)
  )
  + int2d(Th, outlet) (
    1000. * p * q
  )
  ;

varf vM (p, q)
  = int3d(Th, qft=qf2pT) (
    p * q
  )
  ;

varf vA (u, v)
  = int3d(Th) (
    (1./dt) * u * v
    + mu * (grad(u)' * grad(v))
  )
  + on(wall, inlet, u=0)
  ;

varf vB1 (p, v)
  = int3d(Th) (
    p * dx(v)
  )
  ;

varf vB2 (p, v)
  = int3d(Th) (
    p * dy(v)
  )
  ;

varf vB3 (p, v)
  = int3d(Th) (

```



```

                p * dz(v)
            )
        ;

varf vOn1 (u, v)
    = on(inlet, u = 0.)
    + on(wall, u = 0.)
    ;

varf vOn2 (u, v)
    = on(inlet, u = 0.)
    + on(wall, u = 0.)
    ;

varf vOn3 (u, v)
    = on(inlet, u = cos(2.*pi*t)*(1.-(x^2+y^2)/R^2))
    + on(wall, u = 0.)
    ;

varf vNL1(u, v)
    = int3d(Th) (
        (1./dt) * convect([u1, u2, u3], -dt, u1) * v
    )
    + int3d(Th) (
        f1 * v
    )
    + vOn1
    ;

varf vNL2(u, v)
    = int3d(Th) (
        (1./dt) * convect([u1, u2, u3], -dt, u2) * v
    )
    + int3d(Th) (
        f2 * v
    )
    + vOn2
    ;

varf vNL3(u, v)
    = int3d(Th) (
        (1./dt) * convect([u1, u2, u3], -dt, u3) * v
    )
    + int3d(Th) (
        f3 * v
    )
    + vOn3
    ;

matrix P = vP(Ph, Ph, solver=sparsesolver);
matrix M = vM(Ph, Ph, solver=sparsesolver);

matrix A = vA(Uh, Uh, solver=sparsesolver);
matrix B1 = vB1(Ph, Uh);
matrix B2 = vB2(Ph, Uh);
matrix B3 = vB3(Ph, Uh);

real[int] bd1 = vOn1(0, Uh);
real[int] bd2 = vOn2(0, Uh);
real[int] bd3 = vOn3(0, Uh);

func real[int] Uzawa(real[int] pp) {
    real[int] b1 = bd1;      b1 += B1*pp;
    real[int] b2 = bd2;      b2 += B2*pp;
    real[int] b3 = bd3;      b3 += B3*pp;

    u1[] = A^-1 * b1;
    u2[] = A^-1 * b2;
    u3[] = A^-1 * b3;

    pp = B1' * u1[];
    pp += B2' * u2[];
    pp += B3' * u3[];
}

```

```

    pp = -pp;

    return pp;
}

func real[int] Precon(real[int] p){
    real[int] pP = P^-1 * p;
    real[int] pM = M^-1 * p;
    real[int] pp = (1./dt) * pP + mu * pM;
    return pp;
}

//Résolution
real eps = 1.e-6;
for (int i = 0; i < T/dt; i++){
    //Mise à jour
    t += dt;
    bd1 = vNL1(0, Uh);
    bd2 = vNL2(0, Uh);
    bd3 = vNL3(0, Uh);

    //Résolution
    LinearCG(Uzawa, p[], precon=Precon, nbiter=1e3, veps=eps);
    eps = -abs(eps);
    plot(p, cmm="Pression - t = "+t, WindowIndex=1);
    plot([u1, u2, u3], cmm="Vitesse - t = "+t, WindowIndex=2);

    //Sauvegarde
    {
        Ph uu1, uu2, uu3;

        uu1 = u1;
        uu2 = u2;
        uu3 = u3;

        ofstream file("medit/vitesse_"+i+".bb");
        file << "3 3 " << uu1[.n] << " 2 " << endl;
        for (int j = 0; j < uu1[.n]; j++){
            file << uu1[][j] << "\t" << uu2[][j] << "\t" << uu3[][j] << endl;
        }
    }
    {
        ofstream file("medit/pression_"+i+".bb");
        file << "3 1 " << p[.n] << " 2 " << endl;
        for (int j = 0; j < p[.n]; j++){
            file << p[][j] << endl;
        }
    }
}

```

On procède de la même façon que pour l'algorithme en 2 dimensions. On ajoute une sauvegarde des résultats à chaque pas de temps pour une visualisation ultérieure sur Medit [30].

La résolution du problème s'exécute en 30.6 s. La vitesse et la pression au temps final sont données Fig. C.6.

Le temps de calcul est largement supérieur dans le cas 3D. On a un rapport de :

$$\frac{T_{3D}}{T_{2D}} = 10.9 \quad (\text{C.4})$$

Cela signifie que pour une dimension supplémentaire, on a un temps de calcul 11 fois supérieur (mieux que pour la méthode des caractéristiques).

Par rapport à la méthode des caractéristiques, qui donne le même résultat, le facteur d'accélération est donné par :

$$SU = \frac{\text{Temps caractéristiques 3D}}{\text{Temps Uzawa 3D}} = \frac{226.4 \text{ s}}{30.6 \text{ s}} = 7.4 \quad (\text{C.5})$$

L'algorithme d'Uzawa 3D est donc plus de 7 fois plus rapide que l'algorithme des caractéristiques 3D.

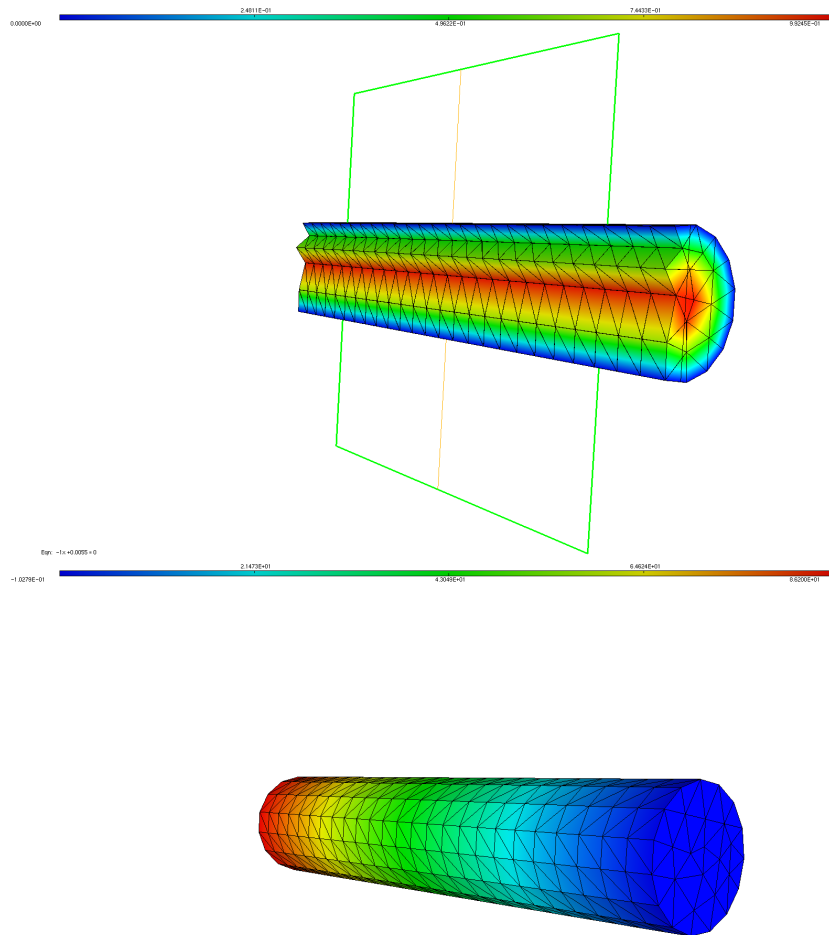


FIGURE C.6 – Vitesse et pression du fluide au temps final. Algorithme d’Uzawa 3D.

C.4 Navier-Stokes, algorithme de splitting

C.4.1 2D

```
//Construction du Maillage
int nn = 10;

real L = 5.;
real D = 1.;

int wall = 1;
int inlet = 2;
int outlet = 3;

border b1(t=0., 1.) {x=L*t; y=0.; label=wall;};
border b2(t=0., 1.) {x=L; y=D*t; label=outlet;};
border b3(t=0., 1.) {x=L-L*t; y=D; label=wall;};
border b4(t=0., 1.) {x=0.; y=D-D*t; label=inlet;};

int nnL = max(2., L*nn);
int nnD = max(2., D*nn);
```

```

mesh Th = buildmesh(b1(nnL) + b2(nnD) + b3(nnL) + b4(nnD));
plot(Th, cmm="Maillage");

//Espaces éléments finis
fespace Uh(Th, P2);
Uh u1, u2;
Uh ulk, u2k;
Uh ulk1, u2k1;
Uh ulstar, u2star;
Uh g1, g2;
Uh gg1, gg2;
Uh gg1k, gg2k;
Uh gg1k1, gg2k1;

fespace Ph(Th, P1);
Ph p;

//Définition du problème
real rho = 1.;
real mu = 1.;

real T = 1.;
real dt = 1.e-2;

func f1 = 0.;
func f2 = 0.;

real t = 0.;

real omega=0.5;
func uin = cos(2.*pi*t)*(1.-(y-D/2.)^2/(D/2.)^2);

macro grad(a) [dx(a), dy(a)] //
macro ugrad(a1, a2, b) (a1*dx(b) + a2*dy(b)) //
macro div(a1, a2) (dx(a1) + dy(a2)) //

varf vA(u, v)
= int2d(Th) (
    (rho/dt) * 1.5 * u * v
    + mu * grad(u)' * grad(v)
)
+ on(inlet, u=0.)
+ on(wall, u=0.)
;

varf vB1(u1, v1)
= int2d(Th) (
    f1 * v1
    + (rho/dt) * (2.*ulk - 0.5*ulk1) * v1
    + p * dx(v1)
    - (2.*ugrad(ulk, u2k, ulk) - ugrad(ulk1, u2k1, ulk1)) * v1
)
+ int1d(Th, outlet) (
    g1 * v1
)
+ on(inlet, u1=-uin*N.x)
+ on(wall, u1=0.)
;

varf vB2(u2, v2)
= int2d(Th) (
    f2 * v2
    + (rho/dt) * (2.*u2k - 0.5*u2k1) * v2
    + p * dy(v2)
    - (2.*ugrad(ulk, u2k, u2k) - ugrad(ulk1, u2k1, u2k1)) * v2
)
+ int1d(Th, outlet) (
    g2 * v2
)
+ on(inlet, u2=0.)
+ on(wall, u2=0.)

```

```

;

varf vPoisson(p, q)
  =-int2d(Th) (
    grad(p)'*grad(q)
  )
  + on(outlet, p=0)
;

varf vOnPoisson(p, q)
  = int2d(Th) (
    (rho/dt) * (2.*div(u1k, u2k) - 0.5*div(u1k1, u2k1)) * q
  )
  - int2d(Th) (
    (f1 - (2.*ugrad(u1k, u2k, u1k) - ugrad(u1k1, u2k1, u1k1))) * dx(q)
    + (f2 - (2.*ugrad(u1k, u2k, u2k) - ugrad(u1k1, u2k1, u2k1))) * dy(q)
  )
  - int1d(Th, inlet, wall) (
    mu * (dx(u2star) - dy(u1star))*(dy(q)*N.x - dx(q)*N.y)
  )
  + int1d(Th, inlet) (
    (rho/dt)* (
      (1.5*gg1 - 2.*gg1k + 0.5*gg1k1) * N.x
      + (1.5*gg2 - 2.*gg2k + 0.5*gg2k1) * N.y
    ) * q
  )
  + on(outlet, p
    = mu * (
      dx(u1star) * N.x * N.x
      + dy(u2star) * N.y * N.y
    )
    - omega * mu * div(u1star, u2star)
    - (g1*N.x + g2*N.y)
  )
;

matrix A = vA(Uh, Uh, solver=sparsesolver);
matrix Poisson = vPoisson(Ph, Ph, solver=sparsesolver);

//Résolution
u1 = 0; u1k = u1;
u2 = 0; u2k = u2;
p = 0;
g1 = 0;
g2 = 0;
gg1 = uin*N.x; gg1k = gg1;
gg2 = uin*N.y; gg2k = gg2;
for (int i = 0; i < T/dt; i++){
  //Mise à jour
  t += dt;

  u1k1 = u1k;
  u2k1 = u2k;

  u1k = u1;
  u2k = u2;

  gg1k1 = gg1k;
  gg2k1 = gg2k;

  gg1k = gg1;
  gg2k = gg2;

  u1star = 2.*u1k-u1k1;
  u2star = 2.*u2k-u2k1;

  //Résolution
  //pression
  real[int] OnPoisson = vOnPoisson(0,Ph);
  p[] = Poisson^-1 * OnPoisson;

  //vitesse

```

```

real[int] B1 = vB1(0,Uh);
u1[] = A^-1 * B1;

real[int] B2 = vB2(0,Uh);
u2[] = A^-1 * B2;

plot(p, cmm="Pression - t = "+t, WindowIndex=1);
plot([u1, u2], cmm="Vitesse - t = "+t, WindowIndex=2);
}

```

Détail de l'algorithme :

- construction du maillage en définissant ses bords,
- définition des espaces éléments finis (pas de condition Inf-Sup à respecter) et des variables du problème au pas de temps $k + 1$, k et $k - 1$,
- définition du problème en vitesse puis en pression,
- initialisation au pas de temps k et $k - 1$,
- résolution du problème en pression puis en vitesse à chaque pas de temps et affichage de la solution calculée.

La résolution du problème s'exécute en 4.2 s. La vitesse et la pression au temps final sont données Fig. C.7.

Par rapport à la méthode des caractéristiques, le facteur d'accélération est donné par :

$$SU = \frac{\text{Temps caractéristiques 2D}}{\text{Temps splitting 2D}} = \frac{14 \text{ s}}{4.2 \text{ s}} = 3.33 \quad (\text{C.6})$$

Par rapport à l'algorithme d'Uzawa, le facteur d'accélération est donné par :

$$SU = \frac{\text{Temps Uzawa 2D}}{\text{Temps splitting 2D}} = \frac{2.8 \text{ s}}{4.2 \text{ s}} = 0.67 \quad (\text{C.7})$$

L'algorithme de splitting est donc plus rapide que la méthode des caractéristiques mais plus lent que l'algorithme d'Uzawa (en exécution non parallèle).

C.4.2 3D

```

load "msh3"

//Construction du Maillage
int nn = 5;

real L = 5.;
real D = 1.;
real R = D/2.;

int wall = 1;
int inlet = 2;
int outlet = 3;

border b0(t=0., 2.*pi){x=R*cos(t); y=R*sin(t); label=0;};

int nnL = max(2., L*nn);
int nnR = max(2., 2.*pi*R*nn);

mesh Th0 = buildmesh(b0(nnR));

int[int] ldown = [0, inlet];
int[int] lmid = [0, wall];
int[int] lup = [0, outlet];
mesh3 Th = buildlayers(Th0, nnL, zbound=[0., L], labeldown=ldown, labelmid=lmid, labelup=lup);
savemesh(Th, "medit/maillage.mesh");
plot(Th, cmm="Maillage");

```

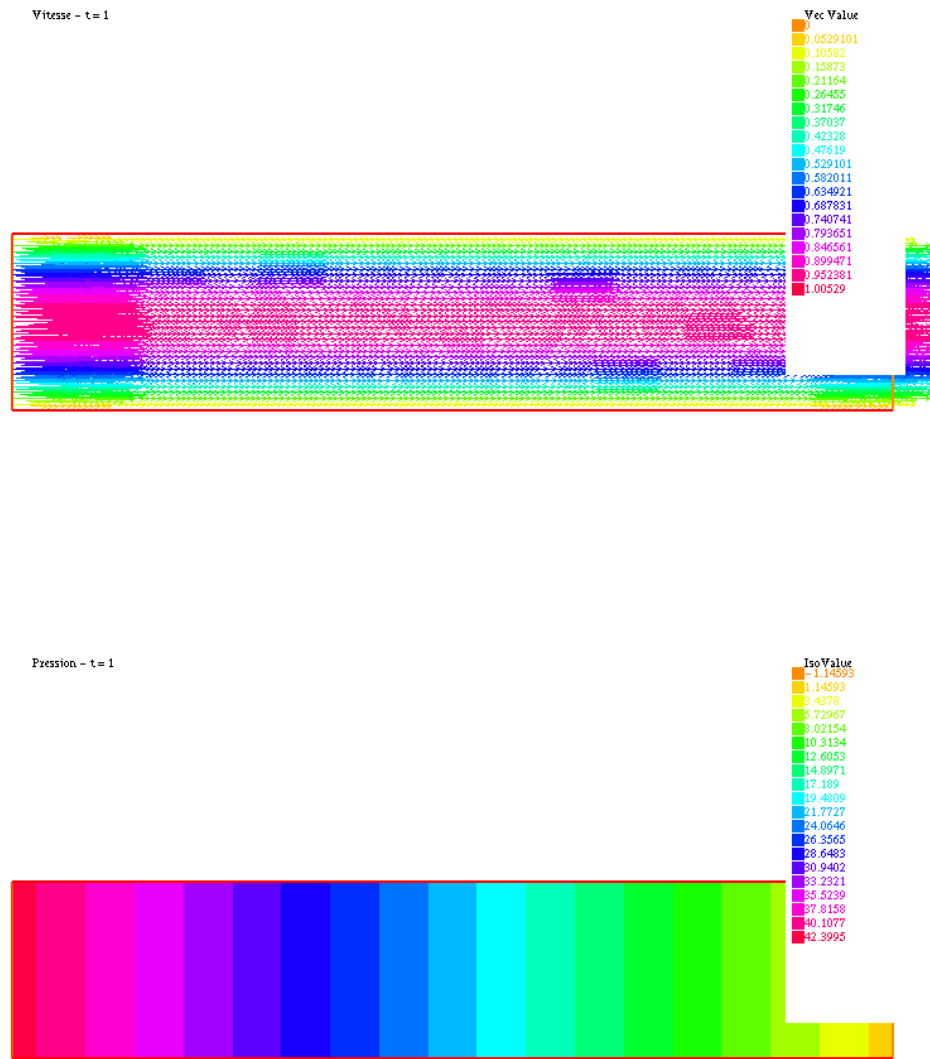


FIGURE C.7 – Vitesse et pression du fluide au temps final. Algorithme de splitting 2D.

```
//Espaces éléments finis
fespace Uh(Th, P2);
Uh u1, u2, u3;
Uh u1k, u2k, u3k;
Uh u1k1, u2k1, u3k1;
Uh ulstar, u2star, u3star;
Uh g1, g2, g3;
Uh gg1, gg2, gg3;
Uh gg1k, gg2k, gg3k;
Uh gg1k1, gg2k1, gg3k1;

fespace Ph(Th, P1);
Ph p;

//Définition du problème
```

```

real rho = 1.;
real mu = 1.;

real T = 1.;
real dt = 1.e-2;

func f1 = 0.;
func f2 = 0.;
func f3 = 0.;

real t = 0.;

real omega=0.5;
func uin = cos(2.*pi*t)*(1.-(x^2+y^2)/R^2);

macro grad(a) [dx(a), dy(a), dz(a)] //
macro ugrad(a1, a2, a3, b) (a1*dx(b) + a2*dy(b) + a3*dz(b)) //
macro div(a1, a2, a3) (dx(a1) + dy(a2) + dz(a3)) //

varf vA(u, v)
= int3d(Th) (
    (rho/dt) * 1.5 * u * v
    + mu * grad(u)' * grad(v)
)
+ on(inlet, u=0.)
+ on(wall, u=0.)
;

varf vB1(u1, v1)
= int3d(Th) (
    f1 * v1
    + (rho/dt) * (2.*u1k - 0.5*u1k1) * v1
    + p * dx(v1)
    - (2.*ugrad(u1k, u2k, u3k, u1k) - ugrad(u1k1, u2k1, u3k1, u1k1)) * v1
)
+ int2d(Th, outlet) (
    g1 * v1
)
+ on(inlet, u1=0.)
+ on(wall, u1=0.)
;

varf vB2(u2, v2)
= int3d(Th) (
    f2 * v2
    + (rho/dt) * (2.*u2k - 0.5*u2k1) * v2
    + p * dy(v2)
    - (2.*ugrad(u1k, u2k, u3k, u2k) - ugrad(u1k1, u2k1, u3k1, u2k1)) * v2
)
+ int2d(Th, outlet) (
    g2 * v2
)
+ on(inlet, u2=0.)
+ on(wall, u2=0.)
;

varf vB3(u3, v3)
= int3d(Th) (
    f3 * v3
    + (rho/dt) * (2.*u3k - 0.5*u3k1) * v3
    + p * dz(v3)
    - (2.*ugrad(u1k, u2k, u3k, u3k) - ugrad(u1k1, u2k1, u3k1, u3k1)) * v3
)
+ int2d(Th, outlet) (
    g3 * v3
)
+ on(inlet, u3=-uin*N.z)
+ on(wall, u3=0.)
;

varf vPoisson(p, q)
=-int3d(Th) (
    grad(p)' * grad(q)
)

```



```

    )
    + on(outlet, p=0)
    ;

varf vOnPoisson(p, q)
= int3d(Th) (
    (rho/dt) * (2.*div(u1k, u2k, u3k) - 0.5*div(u1k1, u2k1, u3k1)) * q
    )
- int3d(Th) (
    (f1 - (2.*ugrad(u1k, u2k, u3k, u1k) - ugrad(u1k1, u2k1, u3k1, u1k1))) * dx(q)
    + (f2 - (2.*ugrad(u1k, u2k, u3k, u2k) - ugrad(u1k1, u2k1, u3k1, u2k1))) * dy(q)
    + (f3 - (2.*ugrad(u1k, u2k, u3k, u3k) - ugrad(u1k1, u2k1, u3k1, u3k1))) * dz(q)
    )
- int2d(Th, inlet, wall) (
    mu * (dy(u3star) - dz(u2star)) * (dz(q) * N.y - dy(q) * N.z)
    + mu * (dz(u1star) - dx(u3star)) * (dx(q) * N.z - dz(q) * N.x)
    + mu * (dx(u2star) - dy(u1star)) * (dy(q) * N.x - dx(q) * N.y)
    )
+ int2d(Th, inlet) (
    (rho/dt) * (
        (1.5*gg1 - 2.*gg1k + 0.5*gg1k1) * N.x
        + (1.5*gg2 - 2.*gg2k + 0.5*gg2k1) * N.y
        + (1.5*gg3 - 2.*gg3k + 0.5*gg3k1) * N.z
    ) * q
    )
+ on(outlet, p)
= mu * (
    dx(u1star) * N.x * N.x
    + dy(u2star) * N.y * N.y
    + dz(u3star) * N.z * N.z
    )
- omega * mu * div(u1star, u2star, u3star)
- (g1*N.x + g2*N.y + g3*N.z)
)
;

matrix A = vA(Uh, Uh, solver=sparsesolver);
matrix Poisson = vPoisson(Ph, Ph, solver=sparsesolver);

//Résolution
u1 = 0; u1k = u1;
u2 = 0; u2k = u2;
u3 = 0; u3k = u3;
p = 0;
g1 = 0;
g2 = 0;
g3 = 0;
gg1 = uin*N.x; gg1k = gg1;
gg2 = uin*N.y; gg2k = gg2;
gg3 = uin*N.z; gg3k = gg3;
for (int i = 0; i < T/dt; i++){
    //Mise à jour
    t += dt;

    u1k1 = u1k;
    u2k1 = u2k;
    u3k1 = u3k;

    u1k = u1;
    u2k = u2;
    u3k = u3;

    gg1k1 = gg1k;
    gg2k1 = gg2k;
    gg3k1 = gg3k;

    gg1k = gg1;
    gg2k = gg2;
    gg3k = gg3;

    u1star = 2.*u1k-u1k1;
    u2star = 2.*u2k-u2k1;

```

```

u3star = 2.*u3k-u3k1;

//Résolution
//pression
real[int] OnPoisson = vOnPoisson(0,Ph);
p[] = Poisson^-1 * OnPoisson;

//vitesse
real[int] B1 = vB1(0,Uh);
u1[] = A^-1 * B1;

real[int] B2 = vB2(0,Uh);
u2[] = A^-1 * B2;

real[int] B3 = vB3(0,Uh);
u3[] = A^-1 * B3;

plot(p, cmm="Pression - t = "+t, WindowIndex=1);
plot([u1, u2, u3], cmm="Vitesse - t = "+t, WindowIndex=2);

//Sauvegarde
{
    fespace Ph1(Th, P1);
    Ph1 uu1, uu2, uu3;

    uu1 = u1;
    uu2 = u2;
    uu3 = u3;

    ofstream file("medit/vitesse_"+i+".bb");
    file << "3 3 " << uu1[].n << " 2 " << endl;
    for (int j = 0; j < uu1[.n]; j++){
        file << uu1[][j] << "\t" << uu2[][j] << "\t" << uu3[][j] << endl;
    }
}
{
    ofstream file("medit/pression_"+i+".bb");
    file << "3 1 " << p[.n] << " 2 " << endl;
    for (int j = 0; j < p[.n]; j++){
        file << p[][j] << endl;
    }
}
}

```

On procède de la même façon que pour l'algorithme en 2 dimensions. On ajoute une sauvegarde des résultats à chaque pas de temps pour une visualisation ultérieure sur Medit [30].

La résolution du problème s'exécute en 53.5 s. La vitesse et la pression au temps final sont données Fig. C.8.

Le temps de calcul est largement supérieur dans le cas 3D. On a un rapport de :

$$\frac{T_{3D}}{T_{2D}} = 12.7 \quad (\text{C.8})$$

Cela signifie que pour une dimension supplémentaire, on a un temps de calcul 13 fois supérieur (mieux que la méthode des caractéristiques, moins bien que l'algorithme d'Uzawa).

Par rapport à la méthode des caractéristiques, le facteur d'accélération est donné par :

$$SU = \frac{\text{Temps caractéristiques 3D}}{\text{Temps splitting 3D}} = \frac{226.4 \text{ s}}{53.5 \text{ s}} = 4.23 \quad (\text{C.9})$$

Par rapport à l'algorithme d'Uzawa, le facteur d'accélération est donné par :

$$SU = \frac{\text{Temps Uzawa 3D}}{\text{Temps splitting 3D}} = \frac{30.6 \text{ s}}{53.5 \text{ s}} = 0.57 \quad (\text{C.10})$$

L'algorithme de splitting est donc plus rapide que la méthode des caractéristiques mais plus lent que l'algorithme d'Uzawa (en exécution non parallèle).

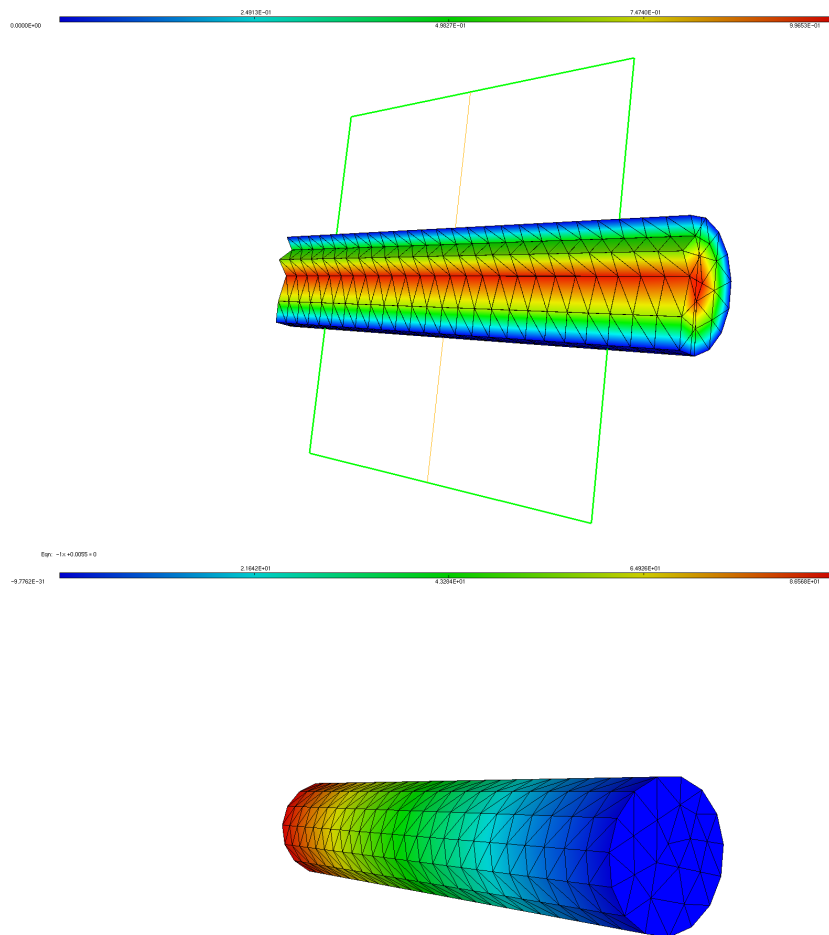


FIGURE C.8 – Vitesse et pression du fluide au temps final. Algorithme de splitting 3D.

C.5 Interaction fluide-structure

C.5.1 2D

```
//Construction du maillage
int nn = 20;
real H = 1.;
real L = 6.;
real e = 0.1;

int sigma1 = 101;
int sigma2 = 102;
int sigma3 = 103;
int gammaD = 104;
int gammaN = 105;
int gamma0 = 106;

mesh Th0;
{
  border m0(t=0., 1.) {x=L-L*t; y=H; label=gamma0;};
  border m1(t=0., 1.) {x=0.; y=H-H*t; label=sigma1;};
  border m2(t=0., 1.) {x=L*t; y=0.; label=sigma2;};
  border m3(t=0., 1.) {x=L; y=H*t; label=sigma3;};
  border m4(t=0., 1.) {x=0.; y=H+e-e*t; label=gammaD;};
}
```

```

border m5(t=0., 1.){x=L; y=H+e*t; label=gammaD;};
border m6(t=0., 1.){x=L-L*t; y=H+e; label=gammaN;};

real nnL = max(2., L*nn);
real nnH = max(2., H*nn);
real nne = max(2., e*nn);

Th0 = buildmesh(m0(-nnL) + m4(nne) + m5(nne) + m6(nnL) + m1(nnH) + m2(nnL) + m3(nnH));
}

//Espaces éléments finis
fespace FSI2h0(Th0, [P1b, P1b]);
FSI2h0 [u10, u20];
FSI2h0 [v10, v20];
FSI2h0 [ulpointpoint0, u2pointpoint0];

//Définition du problème
real t = 0.;

int Num = 100;
real dt = 0.001;
real T = Num*dt;

real rhoF = 1.;
real muF = 0.035;

real rhoS = 1.1;
real E = 3.*1.e6;
real nu = 0.3;

real lambdaS = (nu*E)/((1.-2.*nu)*(1.+nu));
real muS = E/(2.*(1.+nu));

real delta = 0.5;
real theta = 0.3;

func f1F = 0.;
func f2F = 0.;

func f1S = 0.;
func f2S = 0.;

func h1in = 1.e3 * (1. - cos(2.*pi*t/0.025)) * (t <= 0.025);
func h2in = 0.;

func h1out = 0.;
func h2out = 0.;

real sqrt2 = sqrt(2.);
macro div(a1, a2) (dx(a1) + dy(a2)) //
macro grad(a1) [dx(a1), dy(a1)] //
macro epsilon(a1, a2) [dx(a1), dy(a2), (dy(a1)+dx(a2))/sqrt2] //
macro as(a1, a2, b1, b2) (lambdaS * div(a1, a2) * div(b1, b2)
+ (2.*muS) * (epsilon(a1, a2)' * epsilon(b1, b2))) //

fespace Rh0(Th0, P0);
Rh0 reg = region;

int nstruct = reg(L/2., H+e/2.);
int nfluid = reg(L/2., H/2.);

cout << "Fluid region = " << nfluid << endl;
cout << "Structure region = " << nstruct << endl;

t = 0.;

[u10, u20] = [0., 0.];
[v10, v20] = [0., 0.];
[ulpointpoint0, u2pointpoint0] = [0., 0.];

```

```

//Résolution
mesh Th = Th0;
mesh Thhat = Th;
ofstream flowfile("./flow.dat");
for (int i = 0; i < Num; i++){
  cout << "iter = " << (i+1) << endl;

  //Mise à jour
  t += dt;

  Th = Thhat;

  fespace Rh(Th, P0);

  Rh caracF = 1.*(region==nfluid);
  Rh caracS = 1.*(region==nstruct);

  //Espaces éléments finis
  fespace FSI1h(Th, [P1b, P1b]);
  FSI1h [v1rondtilde, v2rondtilde];
  FSI1h [philhat, phi2hat];

  fespace FSI2h(Th, [P1b, P1b]);
  FSI2h [u1, u2];
  FSI2h [v1, v2];
  FSI2h [ulpointpoint, u2pointpoint];

  FSI2h [u1hat, u2hat];
  FSI2h [v1hat, v2hat];
  FSI2h [ulpointpointthat, u2pointpointthat];

  FSI2h [w1hat, w2hat];

  fespace FSI2hP(Th, P1);
  FSI2hP phat;
  FSI2hP qhat;

  fespace FSIh(Th, P1);
  FSIh T1, T2;

  u1[] = u10[];
  u2[] = u20[];

  v1[] = v10[];
  v2[] = v20[];

  ulpointpoint[] = ulpointpoint0[];
  u2pointpoint[] = u2pointpoint0[];

  //Résolution
  varf vAL ([u, uu], [v, vv])
    = int2d(Th) (
      grad(u)' * grad(v)
      + grad(uu)' * grad(vv)
    )
    + on(sigma1, sigma2, sigma3, gammaD, gammaN, gamma0, u=0., uu=0.)
    ;

  varf vbL ([u, uu], [v, vv])
    = on(sigma1, sigma2, sigma3, gammaD, gammaN, u=0., uu=0.)
    + on(gamma0, u=v1, uu=v2)
    ;

  {
    matrix AL = vAL(FSI1h, FSI1h, solver=sparsesolver);
    real[int] BL = vbL(0, FSI1h);

    v1rondtilde[] = AL^-1 * BL;
    v2rondtilde[] = AL^-1 * BL;
  }

  varf vA ([phi1, phi2], [psi1, psi2])
    = int2d(Th, nfluid) (

```

```

        (rhoF/dt) * [phi1, phi2]' * [psi1, psi2]
    )
+ int2d(Th, nfluid) (
    rhoF * (
        (
            (v1-v1rondtilde) * dx(phi1)
            + (v2-v2rondtilde) * dy(phi1)
        ) * psi1
        + (
            (v1-v1rondtilde) * dx(phi2)
            + (v2-v2rondtilde) * dy(phi2)
        ) * psi2
    )
    )
+ int2d(Th, nfluid) (
    2.*muF * (
        epsilon(phi1, phi2)' * epsilon(psi1, psi2)
    )
    )
+ int2d(Th, nstruct) (
    (2.*rhoS/dt) * [phi1, phi2]' * [psi1, psi2]
    )
+ int2d(Th, nstruct) (
    (2.*theta*dt) * as(phi1, phi2, psi1, psi2)
    )
+ on(sigma2, gammaD, phi1=0., phi2=0.)
;

varf vB ([phi1, phi2], [psi])
= -int2d(Th, nfluid) (
    div(phi1, phi2) * psi
    )
;

varf vM (phi, psi)
= int2d(Th) (
    phi * psi * 1.e-6
    )
;

varf vLF ([phi1, phi2], [psi1, psi2])
= int2d(Th, nfluid) (
    (rhoF/dt) * [v1, v2]' * [psi1, psi2]
    )
+ int2d(Th, nfluid) (
    [f1F, f2F]' * [psi1, psi2]
    )
+ int1d(Th, sigma1) (
    [h1in, h2in]' * [psi1, psi2]
    )
+ int1d(Th, sigma3) (
    [h1out, h2out]' * [psi1, psi2]
    )
+ on(sigma2, phi1=0., phi2=0.)
;

varf vLS ([phi1, phi2], [psi1, psi2])
= int2d(Th, nstruct) (
    [f1S, f2S]' * [psi1, psi2]
    )
+ int2d(Th, nstruct) (
    (2.*rhoS/dt) * [v1, v2]' * [psi1, psi2]
    )
+ int2d(Th, nstruct) (
    rhoS * [u1pointpoint, u2pointpoint]' * [psi1, psi2]
    )
- int2d(Th, nstruct) (
    as(u1, u2, psi1, psi2)
    )
- int2d(Th, nstruct) (
    (dt*(1.-2.*theta)) * as(v1, v2, psi1, psi2)
    )
- int2d(Th, nstruct) (

```

```

        (dt*dt*(0.5-2.*theta)) * as(ulpointpoint, u2pointpoint, psi1, psi2)
    )
+ on(gammaD, phi1=0., phi2=0.)
;

{
    matrix A = vA(FSI2h, FSI2h);
    matrix B = vB(FSI2h, FSI2hP);
    matrix M = vM(FSI2hP, FSI2hP);

    real[int] LF = vLF(0, FSI2h);
    real[int] LS = vLS(0, FSI2h);
    real[int] L = LF + LS;

    matrix G = [[A, B'], [B, M]];
    set(G, solver=sparsesolver);

    real[int] zero(phat[.n]);
    for (int i = 0; i < phat[.n] ; i++){
        zero[i] = 0.;
    }
    real[int] LG = [L, zero];

    {
        real[int] sol = G^-1 * LG;

        for (int k = 0; k < vlhat[.n]; k++){
            vlhat[k] = sol[k];
        }
        for (int k = 0; k < v2hat[.n]; k++){
            v2hat[k] = sol[k];
        }
        for (int k = 0; k < phat[.n]; k++){
            phat[k] = sol[vlhat[.n]+k];
        }
    }
}

[ulpointpointthat, u2pointpointthat] = [
    (vlhat - v1)/(dt*delta) - ulpointpoint*(1.*delta)/delta,
    (v2hat - v2)/(dt*delta) - u2pointpoint*(1.*delta)/delta];

[ulhat, u2hat] = [
    u1 + dt*v1 + (dt*dt)*( (0.5-theta)*ulpointpoint + theta*ulpointpointthat ),
    u2 + dt*v2 + (dt*dt)*( (0.5-theta)*u2pointpoint + theta*u2pointpointthat )];

//Movemesh
T1 = x + dt*v1rondtilde*caracF + (ulhat-u1)*caracS;
T2 = y + dt*v2rondtilde*caracF + (u2hat-u2)*caracS;

Thhat = movemesh(Th, [T1, T2]);

//Mise à jour
[u1, u2] = [ulhat, u2hat];
[v1, v2] = [vlhat, v2hat];
[ulpointpoint, u2pointpoint] = [ulpointpointthat, u2pointpointthat];

//Sauvegarde
u10[] = u1[];
u20[] = u2[];

v10[] = v1[];
v20[] = v2[];

ulpointpoint0[] = ulpointpoint[];
u2pointpoint0[] = u2pointpoint[];
}

```

Détails de l'algorithme :

- comme pour l'algorithme fluide, on procède d'abord à la construction du maillage, qui représente le maillage initial dans ce cas,

- on définit un espace élément finis et des variables associées qui serviront de sauvegardes entre les itérations,
- on définit les paramètres de la simulation, du fluide et de la structure,
- on définit ensuite le problème global à l'intérieur des itérations en temps,
- le problème est résolu et on peut modifier le maillage en conséquence,
- les données sont sauvegardées et optionnellement enregistrées.

La résolution du problème s'exécute en 167 s. La vitesse et la pression au temps $t = 35$ ms sont données Fig. C.9.

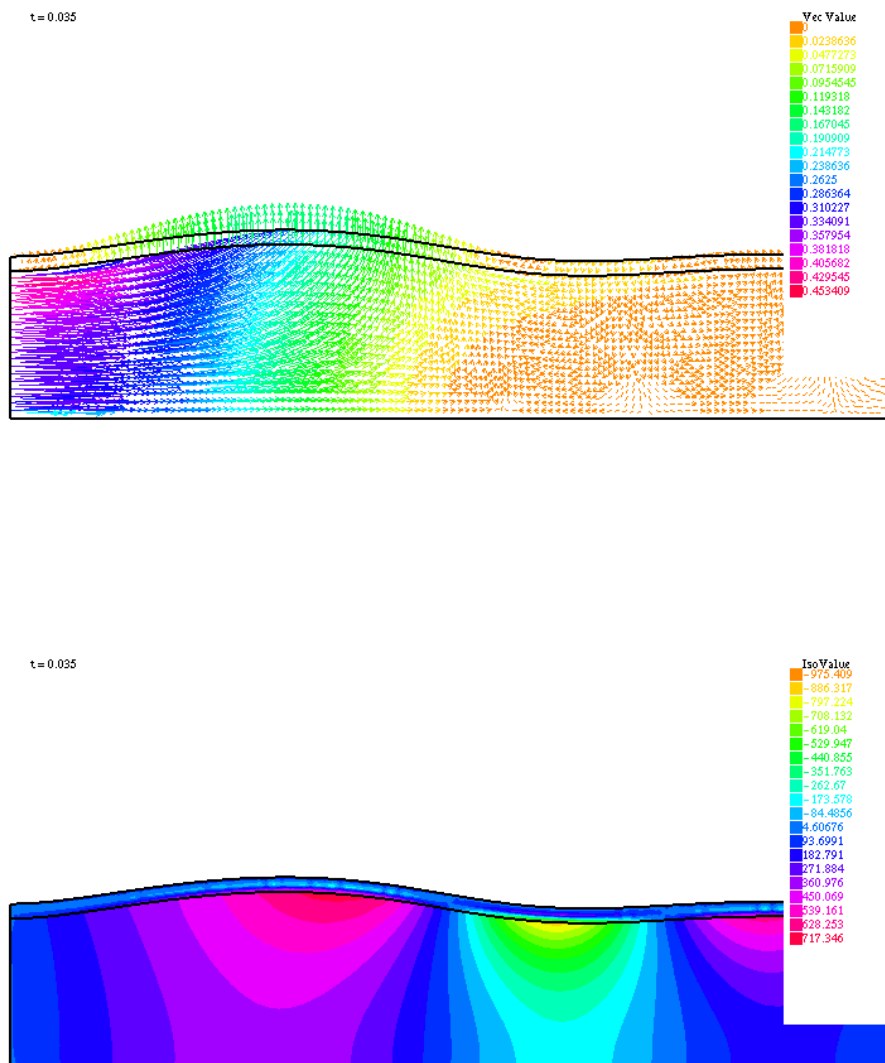


FIGURE C.9 – Vitesse et pression du fluide au temps $t = 35$ ms. Interaction fluide-structure 2D.

C.5.2 3D

```

load "msh3"

//Construction du maillage
int nn = 4;
real mH = 1.;
real mL = 6.;
real me = 0.1;

int sigma1 = 101;
int sigma3 = 103;
int gammaD = 104;
int gammaN = 105;
int gamma0 = 106;

mesh3 Th0;
{
    real nnL = max(2., mL*nn);
    real nnH = max(2., 2.*pi*mH*nn);
    real nne = max(2., 2.*pi*(mH+me)*nn);

    border m0(t=0., 2.*pi){x=mH*cos(t); y=mH*sin(t); label=gamma0;};
    border m1(t=0., 2.*pi){x=(mH+me)*cos(t); y=(mH+me)*sin(t); label=gammaN;};

    mesh Th00 = buildmesh(m0(nnH) + m1(nnH));

    int[int] ldown = [1, sigma1, 0, gammaD];
    int[int] lmid = [gamma0, gamma0, gammaN, gammaN];
    int[int] lup = [1, sigma3, 0, gammaD];
    Th0 = buildlayers(Th00, nnL, zbound=[0., mL], labeldown=ldown, labelmid=lmid, labelup=lup);
}

//Espaces éléments finis
fespace FSI2h0(Th0, [P1b, P1b, P1b]);
FSI2h0 [u10, u20, u30];
FSI2h0 [v10, v20, v30];
FSI2h0 [u1pointpoint0, u2pointpoint0, u3pointpoint0];

//Définition du problème
real t = 0.;

int Num = 100;
real dt = 0.001;
real T = Num*dt;

real rhoF = 1.;
real muF = 0.035;

real rhoS = 1.1;
real E = 3.*1.e6;
real nu = 0.3;

real lambdaS = (nu*E)/((1.-2.*nu)*(1.+nu));
real muS = E/(2.*(1.+nu));

real delta = 0.5;
real theta = 0.3;

func f1F = 0.;
func f2F = 0.;
func f3F = 0.;

func f1S = 0.;
func f2S = 0.;
func f3S = 0.;

func h1in = 0.;
func h2in = 0.;
func h3in = 1.e3 * (1. - cos(2.*pi*t/0.025)) * (t <= 0.025);

```

```

func hlout = 0.;
func h2out = 0.;
func h3out = 0.;

real sqrt2 = sqrt(2.);
macro div(a1, a2, a3) (dx(a1) + dy(a2) + dz(a3)) //
macro grad(a1) [dx(a1), dy(a1), dz(a1)] //
macro epsilon(a1, a2, a3) [dx(a1), dy(a2), dz(a3),
    (dz(a2)+dy(a3))/sqrt2, (dz(a1)+dx(a3))/sqrt2, (dy(a1)+dx(a2))/sqrt2] //
macro as(a1, a2, a3, b1, b2, b3) (lambdaS * div(a1, a2, a3) * div(b1, b2, b3)
    + (2.*muS) * (epsilon(a1, a2, a3)' * epsilon(b1, b2, b3))) //

fespace Rh0(Th0, P0);
Rh0 reg = region;

int nstruct = reg(mH+me/2., 0., 0.);
int nfluid = reg(0., 0., 0.);

cout << "Fluid region = " << nfluid << endl;
cout << "Structure region = " << nstruct << endl;

t = 0.;

[u10, u20, u30] = [0., 0., 0.];
[v10, v20, v30] = [0., 0., 0.];
[u1pointpoint0, u2pointpoint0, u3pointpoint0] = [0., 0., 0.];

//Résolution
mesh3 Th = Th0;
mesh3 Thhat = Th;
ofstream flowfile("./flow.dat");
for (int i = 0; i < Num; i++){
    cout << "iter = " << (i+1) << endl;

    //Mise à jour
    t += dt;

    Th = Thhat;

    fespace Rh(Th, P0);

    Rh caracF = 1.*(region==nfluid);
    Rh caracS = 1.*(region==nstruct);

    //Espaces éléments finis
    fespace FSI1h(Th, [P1b, P1b, P1b]);
    FSI1h [v1rondtilde, v2rondtilde, v3rondtilde];
    FSI1h [philhat, phi2hat, phi3hat];

    fespace FSI2h(Th, [P1b, P1b, P1b]);
    FSI2h [u1, u2, u3];
    FSI2h [v1, v2, v3];
    FSI2h [ulpointpoint, u2pointpoint, u3pointpoint];

    FSI2h [u1hat, u2hat, u3hat];
    FSI2h [v1hat, v2hat, v3hat];
    FSI2h [ulpointpointhat, u2pointpointhat, u3pointpointhat];

    FSI2h [w1hat, w2hat, w3hat];

    fespace FSI2hP(Th, P1);
    FSI2hP phat;
    FSI2hP qhat;

    fespace FSIh(Th, P1);
    FSIh T1, T2, T3;

    u1[] = u10[];
    u2[] = u20[];
    u3[] = u30[];

```

```

v1[] = v10[];
v2[] = v20[];
v3[] = v30[];

ulpointpoint[] = ulpointpoint0[];
u2pointpoint[] = u2pointpoint0[];
u3pointpoint[] = u3pointpoint0[];

//Résolution
varf vAL ([u, uu, uuu], [v, vv, vvv])
= int3d(Th) (
    grad(u)' * grad(v)
    + grad(uu)' * grad(vv)
    + grad(uuu)' * grad(vvv)
)
+ on(sigma1, sigma3, gammaD, gammaN, gamma0, u=0., uu=0., uuu=0.)
;

varf vbL ([u, uu, uuu], [v, vv, vvv])
= on(sigma1, sigma3, gammaD, gammaN, u=0., uu=0., uuu=0.)
+ on(gamma0, u=v1, uu=v2, uuu=v3)
;

{
    matrix AL = vAL(FSI1h, FSI1h, solver=sparsesolver);
    real[int] BL = vbL(0, FSI1h);

    v1rondtilde[] = AL^-1 * BL;
    v2rondtilde[] = AL^-1 * BL;
    v3rondtilde[] = AL^-1 * BL;
}

varf vA ([phi1, phi2, phi3], [psi1, psi2, psi3])
= int3d(Th, nfluid) (
    (rhoF/dt) * [phi1, phi2, phi3]' * [psi1, psi2, psi3]
)
+ int3d(Th, nfluid) (
    rhoF * (
        (
            (v1-v1rondtilde) * dx(phi1)
            + (v2-v2rondtilde) * dy(phi1)
            + (v3-v3rondtilde) * dz(phi1)
        ) * psi1
        + (
            (v1-v1rondtilde) * dx(phi2)
            + (v2-v2rondtilde) * dy(phi2)
            + (v3-v3rondtilde) * dz(phi2)
        ) * psi2
        + (
            (v1-v1rondtilde) * dx(phi3)
            + (v2-v2rondtilde) * dy(phi3)
            + (v3-v3rondtilde) * dz(phi3)
        ) * psi3
    )
)
+ int3d(Th, nfluid) (
    2.*muF * (
        epsilon(phi1, phi2, phi3)' * epsilon(psi1, psi2, psi3)
    )
)
+ int3d(Th, nstruct) (
    (2.*rhoS/dt) * [phi1, phi2, phi3]' * [psi1, psi2, psi3]
)
+ int3d(Th, nstruct) (
    (2.*theta*dt) * as(phi1, phi2, phi3, psi1, psi2, psi3)
)
+ on(gammaD, phi1=0., phi2=0., phi3=0.)
;

varf vB ([phi1, phi2, phi3], [psi])
= -int3d(Th, nfluid) (
    div(phi1, phi2, phi3) * psi
)

```

```

;

varf vM (phi, psi)
= int3d(Th) (
    phi * psi * 1.e-6
)
;

varf vLF ([phi1, phi2, phi3], [psi1, psi2, psi3])
= int3d(Th, nfluid) (
    (rhoF/dt) * [v1, v2, v3]' * [psi1, psi2, psi3]
)
+ int3d(Th, nfluid) (
    [f1F, f2F, f3F]' * [psi1, psi2, psi3]
)
+ int2d(Th, sigma1) (
    [h1in, h2in, h3in]' * [psi1, psi2, psi3]
)
+ int2d(Th, sigma3) (
    [h1out, h2out, h3out]' * [psi1, psi2, psi3]
)
;

varf vLS ([phi1, phi2, phi3], [psi1, psi2, psi3])
= int3d(Th, nstruct) (
    [f1S, f2S, f3S]' * [psi1, psi2, psi3]
)
+ int3d(Th, nstruct) (
    (2.*rhoS/dt) * [v1, v2, v3]' * [psi1, psi2, psi3]
)
+ int3d(Th, nstruct) (
    rhoS * [ulpointpoint, u2pointpoint, u3pointpoint]' * [psi1, psi2, psi3]
)
- int3d(Th, nstruct) (
    as(u1, u2, u3, psi1, psi2, psi3)
)
- int3d(Th, nstruct) (
    (dt*(1.-2.*theta)) * as(v1, v2, v3, psi1, psi2, psi3)
)
- int3d(Th, nstruct) (
    (dt*dt*(0.5-2.*theta))
    * as(ulpointpoint, u2pointpoint, u3pointpoint, psi1, psi2, psi3)
)
+ on(gammaD, phi1=0., phi2=0., phi3=0.)
;

{

matrix A = vA(FSI2h, FSI2h);
matrix B = vB(FSI2h, FSI2hP);
matrix M = vM(FSI2hP, FSI2hP);

real[int] LF = vLF(0, FSI2h);
real[int] LS = vLS(0, FSI2h);
real[int] L = LF + LS;

matrix G = [[A, B'], [B, M]];
set(G, solver=sparsesolver);

real[int] zero(phat[].n);
for (int i = 0; i < phat[].n; i++){
    zero[i] = 0.;
}
real[int] LG = [L, zero];

{

real[int] sol = G^-1 * LG;

for (int k = 0; k < vlhat[].n; k++){
    vlhat[][k] = sol[k];
}
for (int k = 0; k < v2hat[].n; k++){
    v2hat[][k] = sol[k];
}

}

}

```

```

        for (int k = 0; k < v3hat[.n]; k++){
            v3hat[][k] = sol[k];
        }
        for (int k = 0; k < phat[.n]; k++){
            phat[][k] = sol[v1hat[.n+k];
        }
    }

    [ulpointpointhat, u2pointpointhat, u3pointpointhat] = [
        (v1hat - v1)/(dt*delta) - ulpointpoint*(1.*delta)/delta,
        (v2hat - v2)/(dt*delta) - u2pointpoint*(1.*delta)/delta,
        (v3hat - v3)/(dt*delta) - u3pointpoint*(1.*delta)/delta];

    [ulhat, u2hat, u3hat] = [
        u1 + dt*v1 + (dt*dt)*( (0.5-theta)*ulpointpoint + theta*ulpointpointhat ),
        u2 + dt*v2 + (dt*dt)*( (0.5-theta)*u2pointpoint + theta*u2pointpointhat ),
        u3 + dt*v3 + (dt*dt)*( (0.5-theta)*u3pointpoint + theta*u3pointpointhat )];

    //Movemesh
    T1 = x + dt*v1rondtilde*caracF + (ulhat-u1)*caracS;
    T2 = y + dt*v2rondtilde*caracF + (u2hat-u2)*caracS;
    T3 = z + dt*v3rondtilde*caracF + (u3hat-u3)*caracS;

    Thhat = movemesh3(Th, transfo=[T1, T2, T3]);

    //Mise à jour
    [u1, u2, u3] = [ulhat, u2hat, u3hat];
    [v1, v2, v3] = [v1hat, v2hat, v3hat];
    [ulpointpoint, u2pointpoint, u3pointpoint] = [
        ulpointpointhat, u2pointpointhat, u3pointpointhat];

    //Sauvegarde
    u10[] = u1[];
    u20[] = u2[];
    u30[] = u3[];

    v10[] = v1[];
    v20[] = v2[];
    v30[] = v3[];

    ulpointpoint0[] = ulpointpoint[];
    u2pointpoint0[] = u2pointpoint[];
    u3pointpoint0[] = u3pointpoint[];
}

```

L'algorithme est similaire au cas 2 dimensions.

La résolution du problème s'exécute en 800.7 s. La vitesse et la pression au temps $t = 35$ ms sont données Fig. C.10.

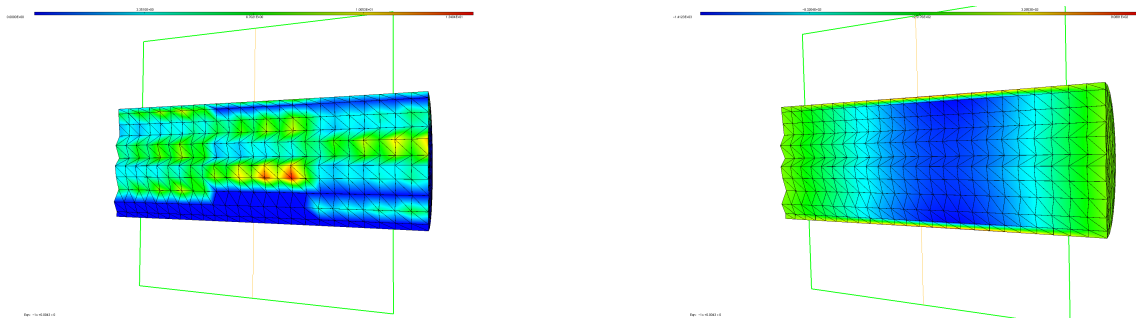


FIGURE C.10 – Vitesse et pression du fluide au temps $t = 35$ ms. Interaction fluide-structure 3D.

Le temps de calcul est largement supérieur dans le cas 3D. On a un rapport de :

$$\frac{T_{3D}}{T_{2D}} = 4.79 \quad (\text{C.11})$$

Cela signifie que pour une dimension supplémentaire, on a un temps de calcul 5 fois supérieur.

Annexe D

Projet ANR VIVABRAIN



D.1 Présentation

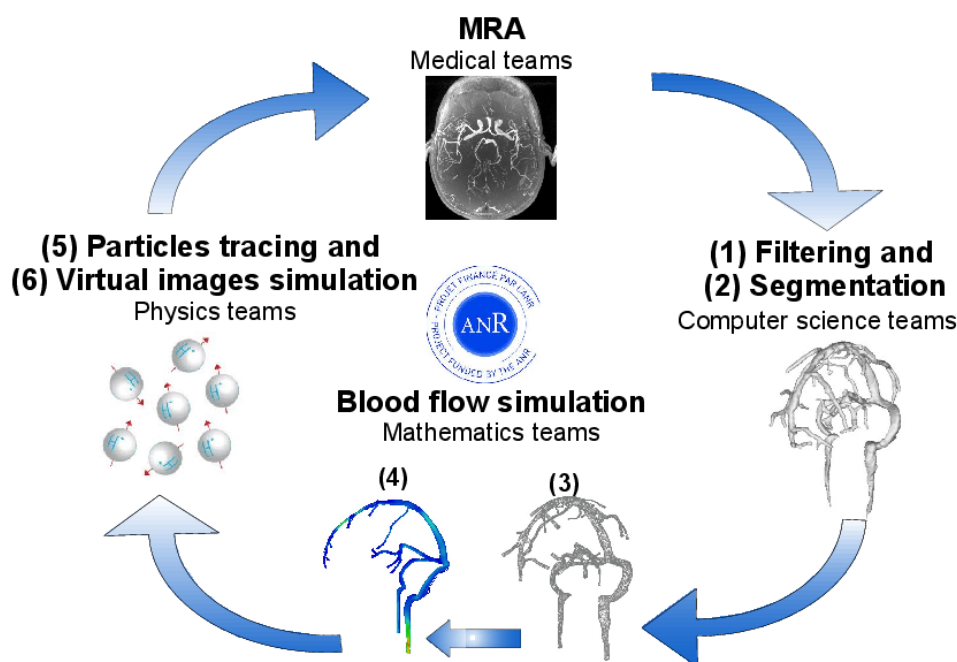


FIGURE D.1 – **Projet VIVABRAIN**. Représentation du pipeline.

Le but du projet ANR VIVABRAIN [6], Fig. D.1, est de mettre à disposition un pipeline de validation des différents outils utilisés dans la simulation d'écoulements sanguins.

Il se compose :

- de l'acquisition d'images ARM (angiographie par résonance magnétique) par les équipes médicales,
- de la segmentation des images ARM par les équipes informatiques,
- du maillage volumique et la simulation des écoulements sanguins par les équipes mathématiques,
- de la simulation d'ARM par les équipes physiques.

En comparant les ARM acquises par les équipes médicales et les ARM simulées par les équipes physiques, l'erreur commise tout au long de la boucle (segmentation + simulations) peut être quantifiée.

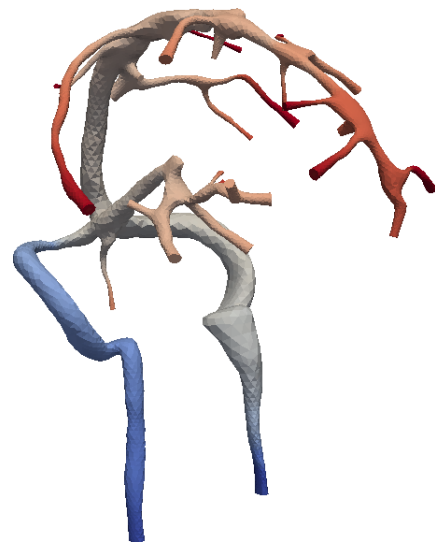
Un autre atout de ce projet est de mettre à disposition des images ARM en accès libre pour permettre d'améliorer les différents algorithmes de segmentation d'images, de reconstruction de la géométrie et de simulation.

D.2 Participation

D.2.1 Équipes mathématiques



(a) Imagerie IRM.



(b) Champ de pression sur le maillage reconstruit.

FIGURE D.2 – Réseau veineux.

Des algorithmes de simulation de la dynamique des fluides ont été développés pour répondre au besoin de la simulation des flux sanguins, après l'étape de reconstruction de maillage depuis les données IRM, Fig. D.2. L'algorithme de Navier-Stokes utilisant la méthode des caractéristiques couplée à l'algorithme d'Uzawa est choisi pour sa rapidité, et l'algorithme de splitting pour sa scalabilité (possible de l'utiliser sur des machines parallèles). Des outils de visualisation ont également été mis en place.

D.2.2 Équipes physiques

Afin de simuler des images angiographiques, les équipes physiques ont besoin des positions de spins dans la géométrie étudiée pendant le temps de l'acquisition à partir des simulations numériques d'écoulements sanguins.

Un tracé de particules a été effectué avec le logiciel PARAVIEW [1], Fig. D.3, suivie d'une conversion dans un format accepté par JEMRIS [72] à l'aide d'un script Python¹. Un tri des particules est finalement réalisé afin de retirer les particules entrant en collision avec les parois, celles-ci pouvant provoquer des artefacts lors de la simulation d'images ARM.

Ces algorithmes vont être intégrés dans l'outil de simulation AngioTK [57].

1. <https://www.python.org/>

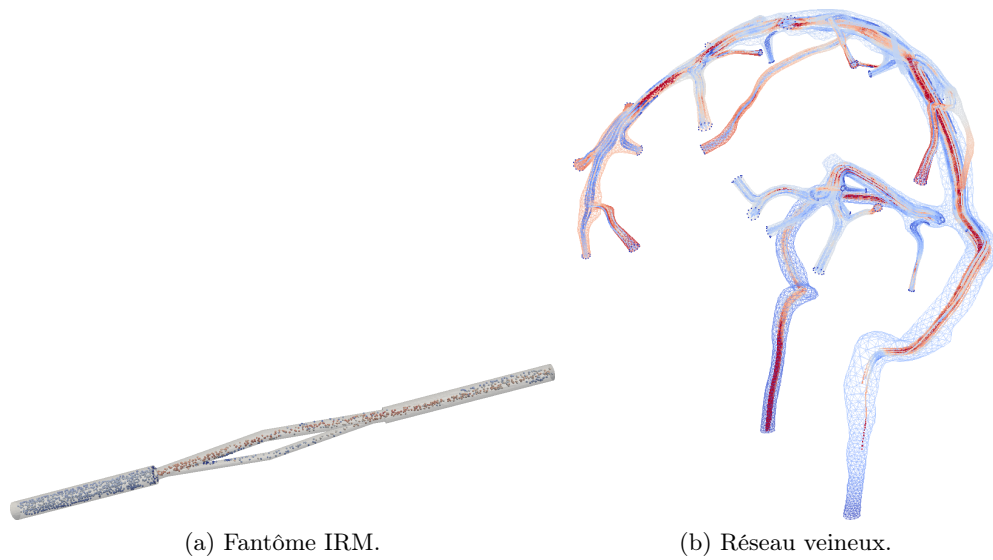


FIGURE D.3 – **Tracé de particules.** Fantôme IRM et réseau veineux.

Annexe E

CEMRACS 2015

PHANTOM PROJECT: DEVELOPMENT AND VALIDATION OF THE PIPELINE FROM MRA ACQUISITION TO MRA SIMULATIONS

ALEXANDRE ANCEL¹, ALEXANDRE FORTIN², SIMON GARNOTEL³, OLIVIA MIRAUCOURT²
AND RANINE TARABAY¹

Abstract. The aim of this project is to validate the *Vivabrain* pipeline with a physical phantom from real MRI acquisition to MRI simulations through image segmentation and computational fluid dynamics (CFD) simulations. For that purpose, we set up three comparison benchmarks. The first benchmark compares dimensions of the reconstructed geometry from real MRI acquisition to the physical phantom dimensions. The second aims to validate the CFD simulations by comparing the outputs of two simulations, one carried out using FEEL++ and the other using FREEFEM++. The CFD outputs are also compared to MRI flow measurement data. The goal of the last comparison benchmark is to compare the MRI simulations outputs to the numerical fluid simulations.

INTRODUCTION

In the last 20 years, progress in medical imaging has led to the development of modalities devoted to visualizing vascular structures. These angiography images progressively proved their usefulness in various clinical applications, in particular for cerebrovascular issues. This project is within the context of the ANR project *Vivabrain* [2]. The goal of this project is to develop a pipeline for the generation of virtual Magnetic Resonance Angiography (MRA) of the human brain, associated to their anatomical (3D) and hemodynamic (3D+t) models (providing a ground-truth). The interdisciplinary program follows four steps, see Fig. 1. We first start from real MRA data from which we extract the vascular volumes. We then generate the 3D vascular mesh (see [12] for more details). We perform 3D+t simulations of blood flow in the complex (arterial and venous) mesh. Finally, we run the simulations of MR acquisition of angiography data from these 3D+t models.

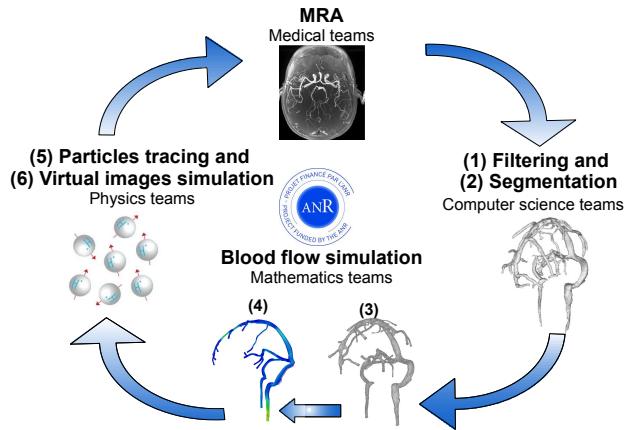
AngioTK [1] is a software framework developed in this context, using open source software. This framework takes MRI data as input and produces virtual angiographies as output. As show in Fig. 1, AngioTK proceeds in several steps:

- (1) Filtering: Filter the initial MRI data, to ease the extraction of arterial/venous data;
- (2) Segmentation: Separate the veins/arteries from the background;
- (3) Mesh processing: Process the segmented data to prepare it for numerical simulations. This task involves several subtasks using centerlines notably;
- (4) Numerical simulations: simulations of blood flow in the processed mesh (Feel++/FreeFem++);
- (5) Particles tracing: Generates particles for virtual angiographies;
- (6) Virtual angiography: Uses JEMRIS to simulate virtual angiographies.

¹ University of Strasbourg, IRMA / UMR 7501, Strasbourg, France

² University of Reims Champagne-Ardenne, LMR, Reims, France

³ University of Picardie Jules Verne, BioFlowImage Laboratory, Amiens, France

FIGURE 1. The *Vivabrain* project task loop.

In this paper, our purpose is to evaluate the accuracy of the image segmentation, the (3D+t) blood flow simulations and the MRI simulations. To this end, we started from a physical phantom, see Fig. 2, so that we have an exact description of its geometry that can be used to evaluate the error made when extracting the 3D volume from the MRI segmentation. For the computational fluid dynamic validation, we have chosen to run simulations using two finite elements libraries, FEEL++ [13] and FREEFEM++ [10]. A first level of validation of the computational fluid dynamics (CFD) simulations is to compare the numerical outputs obtained by using FEEL++ and FREEFEM++ libraries, running equivalent simulations on the same mesh. The second level of comparison is to compare the FEEL++ and FREEFEM++ numerical output data to the experimental MRI measurements. Lastly, the validation of the MRI simulations will be performed by comparing their outputs to the MRI acquisition and to the exact geometry dimensions.

The first part of this paper is dedicated to the description of the image segmentation, the numerical model and methods for the simulations of blood flow and MRI simulations. In the second part, we describe the benchmark setup used for the previously mentioned comparisons. The third part of this paper is dedicated to the image segmentation evaluation. Finally, we present the blood flow and MRI simulations results on the phantom. In particular, we run a cross-validation between the FEEL++ and FREEFEM++ libraries to estimate the precision of the step of blood flow simulations in the phantom and the complex cerebral venous network, which is studied in the *Vivabrain* project.

1. NUMERICAL METHODS

1.1. Image segmentation

In the first part of the pipeline, we need to extract the vascular volumes from the MRA images. Here, to segment the phantom, we use an active contour model called “snake” whose principle is to evolve a curve to detect the objects boundaries in a given image. Let Ω be a bounded open set of \mathbb{R}^2 , with $\partial\Omega$ its boundary. Let $I : \bar{\Omega} \rightarrow \mathbb{R}^2$ be a given image of bounded variation and $\mathcal{C}(s) : [0, 1] \rightarrow \mathbb{R}^2$ be a parameterized curve, at least of class C^2 . The classical snake model consists of minimizing the energy functional $J_1(\mathcal{C})$ given by:

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

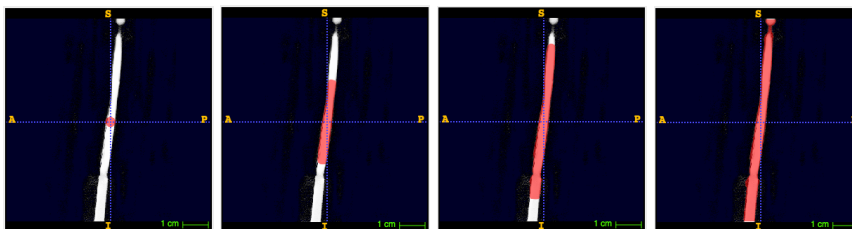


FIGURE 2. The physical phantom.

where α , β and λ are positive parameters. The first two terms control the smoothness of the contour (the internal energy), while the third term attracts the contour towards the object (the external energy). The snake minimizes the energy $J_1(\mathcal{C})$ by trying to locate the curve at the points of maxima $|\nabla I(\mathcal{C}(s))|$, acting as an edge-detector. In general, $-|\nabla I|$ can be replaced by $g(|\nabla I|)$ where g is a positive and decreasing function such as $\lim_{x \rightarrow \infty} g(x) = 0$ (e.g. $g(x) = \frac{1}{1+x}$ or $g(x) = \exp(-x)$). ITK-SNAP¹ implements the 3D geodesic active contour method developed by Caselles et al. [7], which allows topological changes of the curve, contrary to the classical snake. The new energy $J_2(\mathcal{C})$ to be minimized is given by:

$$J_2(\mathcal{C}) = 2 \int_0^1 |\mathcal{C}'(s)| \cdot g(|\nabla I(\mathcal{C}(s))|) ds \quad (2)$$

This is a problem of geodesic computation in a Riemannian space, according to a metric induced by the image I . This minimization problem is solved by a gradient descent which follows an evolution equation and the curve changes over time as it can be seen in Fig. 3.

FIGURE 3. Evolution of the snake on the phantom at iterations $n = 0$, $n = 500$, $n = 1000$, $n = 1500$.

1.2. Fluid simulations

In this section, we introduce the equations of the fluid dynamics and the numerical methods used to compute the blood flow in the extracted volumes. Let $\Omega \subset \mathbb{R}^3$ denote the bounded connected domain under investigation and let $\partial\Omega = \Gamma_D \cup \Gamma_N$ be its boundary, where Γ_D and Γ_N are the sections of the boundary where we will respectively impose a Dirichlet condition and a Neumann condition. In the context of blood flow, it is assumed

¹<http://www.itksnap.org>

that the unsteady incompressible Navier-Stokes equations Eq. (3,4) hold. They read as:

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p - \mu \Delta \mathbf{u} = \mathbf{f} \quad \text{in } \Omega \quad (3)$$

$$\operatorname{div}(\mathbf{u}) = 0 \quad \text{in } \Omega \quad (4)$$

where \mathbf{u} and p are respectively the velocity and pressure of the fluid, ρ and μ are the density and the dynamic viscosity of the fluid, and \mathbf{f} is an external force taken here equal to zero (gravity is neglected).

Let us consider a Dirichlet boundary condition at the inflow and a Neumann boundary condition at the outflow:

$$\mathbf{u} = \mathbf{u}_{in} \quad \text{on } \Gamma_D \quad (5)$$

$$\boldsymbol{\sigma}(\mathbf{u}, p) \cdot \mathbf{n} = \mathbf{0} \quad \text{on } \Gamma_N \quad (6)$$

with the stress tensor that writes for a Newtonian fluid:

$$\boldsymbol{\sigma}(\mathbf{u}, p) = \mu \nabla \mathbf{u} - p \mathbf{I} \quad (7)$$

1.3. Numerical algorithms

Space discretization Let δ be a discretization parameter. We define $\hat{K} \subset \mathbb{R}^d$ ($d = 1, 2, 3$) a reference elementary convex, *e.g.* a simplex or a hypercube. We denote by \mathcal{T}_δ a finite collection of nonempty, disjoint open simplices or hypercubes $\mathcal{T}_\delta = \{K = \varphi_{K,k}^{\text{geo}}(\hat{K})\}$ forming a partition of Ω , where $\varphi_{K,k}^{\text{geo}}$ is the polynomial of degree k that maps \hat{K} to K , which is also called the geometric transformation. We denote by $\mathbb{P}^N(\hat{K})$ and $\mathbb{P}^N(K)$ the spaces of polynomials of total degree less or equal than N defined on \hat{K} and K respectively. We denote Ω_δ the discrete domain, and $\Gamma_{D,\delta}$ and $\Gamma_{N,\delta}$ the discrete sections of the boundary where we set a Dirichlet and a Neumann boundary condition, respectively. Following these notations we can define $P_c^N(\Omega_\delta)$ and $[P_c^N(\Omega_\delta)]^d$:

$$P_c^N(\Omega_\delta) = \{v \in C^0(\Omega_\delta) \mid v \circ \varphi_{K,k}^{\text{geo}} \in \mathbb{P}^N(\hat{K}) \ \forall K \in \mathcal{T}_\delta\}, \quad [P_c^N(\Omega_\delta)]^d = \prod_1^d P_c^N(\Omega_\delta). \quad (8)$$

In the following, we introduce the discrete spaces associated to the velocity and the pressure respectively:

$$\begin{aligned} H_{(\mathbf{u}_{in}, \Gamma_{D,\delta})}^1(\Omega_\delta) &= \{f \in H^1(\Omega_\delta) / f|_{\Gamma_{D,\delta}} = \mathbf{u}_{in}\}, \\ V_\delta &= \{v \in H_{(\mathbf{u}_{in}, \Gamma_{D,\delta})}^1(\Omega_\delta) \cap [P_c^M(\Omega_\delta)]^d\}, \\ Q_\delta &= \{v \in P_c^N(\Omega_\delta)\}. \end{aligned}$$

In order to ensure the existence, the uniqueness and the stability of the solution of the discrete problem, the spaces V_δ and Q_δ must satisfy the so called Inf-Sup condition:

$$\exists \beta_\delta > 0 \mid \inf_{q_\delta \in Q_\delta} \sup_{\mathbf{v}_\delta \in V_\delta} \frac{\int_{\Omega_\delta} q_\delta \operatorname{div}(\mathbf{v}_\delta)}{\|q_\delta\|_{0,\Omega_\delta} \|\mathbf{v}_\delta\|_{1,\Omega_\delta}} \geq \beta_\delta. \quad (9)$$

We will consider the generalized Taylor-Hood finite element for the velocity-pressure discretization, that is to say we look for the velocity in $[P_c^{N+1}(\Omega_h)]^d$, $N \geq 1$ and the pressure in $P_c^N(\Omega_h)$ which satisfy the inf-sup condition. The resulting approximate velocity and pressure fields are respectively denoted by \mathbf{u}_δ and p_δ . Typically the Taylor-Hood finite element ($N = 1$) is considered to solve this problem.

Time discretization Let T , be the time run of the simulations and $I = [0, T]$, be the time interval. We subdivide I into \mathcal{N} sub-intervals $I^n = (t^n, t^{n+1})$ with $n = 1, \dots, \mathcal{N}$ and where $t^{n+1} - t^n = \Delta t$, the time step

assumed constant over time.

1.3.1. FEEL++

We denote $(\mathbf{u}_\delta^n, p_\delta^n)$ the approximate discrete solution at time t^n . We first start by discretizing the time derivative of the velocity by choosing an implicit scheme, the so-called *backward differentiation formulation* (BDF) of order two. We then choose a semi-implicit scheme with a second order extrapolation in order to manage the non-linear term $\mathbf{u} \cdot \nabla \mathbf{u}$. The weak formulation then reads: find $(\mathbf{u}_\delta^{n+1}, p_\delta^{n+1}) \in V_\delta \times Q_\delta$ such that $\forall \mathbf{v}_\delta \in \{\mathbf{v} \in [H^1(\Omega_\delta)]^3 \cap [P_c^M(\Omega_\delta)]^3 \mid \mathbf{v} = \mathbf{0} \text{ on } \Gamma_{D,\delta}\}, \forall q_\delta \in Q_\delta$

$$\begin{aligned} \rho \int_{\Omega_\delta} \frac{\frac{3}{2}\mathbf{u}_\delta^{n+1} - 2\mathbf{u}_\delta^n + \frac{1}{2}\mathbf{u}_\delta^{n-1}}{\Delta t} \cdot \mathbf{v}_\delta + \rho \int_{\Omega_\delta} ((2\mathbf{u}_\delta^n - \mathbf{u}_\delta^{n-1}) \cdot \nabla \mathbf{u}_\delta^{n+1}) \cdot \mathbf{v}_\delta + \mu \int_{\Omega_\delta} \nabla \mathbf{u}_\delta^{n+1} : \nabla \mathbf{v}_\delta \\ - \int_{\Omega_\delta} p_\delta^{n+1} \operatorname{div}(\mathbf{v}_\delta) = 0 \\ \int_{\Omega_\delta} q_\delta \operatorname{div}(\mathbf{u}_\delta^{n+1}) = 0 \end{aligned}$$

The choice of this discretization is motivated by the fact that BDF schemes with extrapolation yield to stable time discretizations of the Navier-Stokes equations at the continuous level.

The notation $\mathbf{A} : \mathbf{B}$ corresponds to the inner product of two tensor fields, it can be explicitly written as:

$$\mathbf{A} : \mathbf{B} = \operatorname{Tr}(\mathbf{A}^T \mathbf{B})$$

1.3.2. FREEFEM++

In this section, we describe the numerical method used in FREEFEM++ to solve the Navier-Stokes equations: the method of characteristics [6].

In order to manage the non-linearity, we consider the term $\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u}$ as a particular derivative, so for every particle we can write:

$$\frac{dX}{dt}(x, s; t) = \mathbf{u}(t, X(x, s; t)) \quad (10)$$

$$X(x, s; s) = x \quad (11)$$

where $X(x, s; t)$ is the particle position at time t , which was in x at time s .

That gives:

$$\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) (t^n, x) \sim \frac{\mathbf{u}(t^{n+1}, x) - \mathbf{u}(t^n, X^n(x))}{\Delta t} \quad (12)$$

with $X^n(x) = x - \mathbf{u}(t^n, x)\Delta t + \mathcal{O}(\Delta t^2)$.

We finally have:

$$\frac{\rho}{\Delta t} (\mathbf{u}^{n+1} - \mathbf{u}^n \circ X^n) - \mu \Delta \mathbf{u}^{n+1} + \nabla p^{n+1} = 0 \quad (13a)$$

$$\operatorname{div}(\mathbf{u}^{n+1}) = 0 \quad (13b)$$

The weak formulation can be written as: find $(\mathbf{u}_\delta^{n+1}, p_\delta^{n+1}) \in V_\delta \times Q_\delta$ such that $\forall \mathbf{v}_\delta \in \{\mathbf{v} \in [H^1(\Omega_\delta)]^3 \cap [P_c^{N+1}(\Omega_\delta)]^3 \mid \mathbf{v} = \mathbf{0} \text{ on } \Gamma_{D,\delta}\}, \forall q_\delta \in Q_\delta$

$$\int_{\Omega_\delta} \frac{\rho}{\Delta t} \mathbf{u}_\delta^{n+1} \cdot \mathbf{v}_\delta + \int_{\Omega_\delta} \frac{\rho}{\Delta t} (\mathbf{u}_\delta^n \circ \mathbf{X}_\delta^n) \cdot \mathbf{v}_\delta + \mu \int_{\Omega_\delta} \nabla \mathbf{u}_\delta^{n+1} : \nabla \mathbf{v}_\delta - \int_{\Omega_\delta} p_\delta^{n+1} \operatorname{div}(\mathbf{v}_\delta) = 0 \quad (14)$$

$$\int_{\Omega_\delta} q_\delta \operatorname{div}(\mathbf{u}_\delta^{n+1}) = 0 \quad (15)$$

This method is implemented using the `convect` operator of `FREEFEM++`.

The method of characteristics is unconditionally stable [14].

1.4. MRI simulations

MRI simulations were initially developed to optimize and test the MRI sequences. This method allows to predict easily, quickly and at a low cost the result of any complex experiment, with any physical parameters. Another motivation is educational, for basic understanding of MRI physics. Finally, this can be used for physical model validation, as it is the case in the *Vivabrain* project, where we test our Computational Fluid Dynamics models in the last part of the pipeline.

1.4.1. JEMRIS MRI simulator

JEMRIS is an advanced MRI simulator software written in C++, open-source and freely modifiable [17]. Several optional Matlab GUIs offer great freedom to design original sequences, including arbitrary pulse shapes and gradients, as well as management of various physical parameters. Many off-resonance factors are taken into account, such as chemical shift, concomitant gradient fields, magnetic susceptibility, etc., and possibly other phenomena accompanying the imaging process (molecular diffusion, Gaussian noise, patient move, ...).

1.4.2. Principle: Isochromat summation

In MRI, the signal collected over time is generated by the temporal variations of the macroscopic magnetization of tissues. This signal contains all the information needed to reconstruct the final image.

The most popular technique for MRI simulations is isochromat summation. The sample to be imaged is divided into equal subvolumes called isochromats (see Fig. 4). Those subvolumes are supposed to possess uniform physical properties: spin relaxation times $T1$, $T2$, $T2^{*2}$, equilibrium magnetization M_0 and magnetic susceptibility χ .

The measurable signal emitted by one isochromat is obtained by numerically solving Bloch equations. The whole MR signal is then obtained by summing the contribution of each isochromat over the entire sample.

1.4.3. Bloch equations: Temporal evolution of magnetization

Bloch equation gives the temporal evolution of tissue magnetization for one isochromat:

$$\frac{d\mathbf{M}}{dt} = \gamma \mathbf{M} \times \mathbf{B} - \hat{\mathbf{R}}(\mathbf{M} - \mathbf{M}_0) \quad (16)$$

where \mathbf{M} is the magnetization vector of the tissue, γ is the gyromagnetic ratio of hydrogen, \mathbf{B} is the external magnetic field and $\hat{\mathbf{R}}$ the relaxation matrix containing $T1$ and $T2$.

The magnetic field term $\mathbf{B}(\mathbf{r}, t)$ contains all the MR sequence elements (gradients and radio frequency pulses). Its expression is given by:

$$\mathbf{B}(\mathbf{r}, t) = [\mathbf{G}(t) \cdot \mathbf{r} + \Delta B(\mathbf{r}, t)] \mathbf{e}_z + \mathbf{B}_1(\mathbf{r}, t) \quad (17)$$

² $T2^*$ time is due to local magnetic field inhomogeneities. It results in a shorter transversal relaxation time.

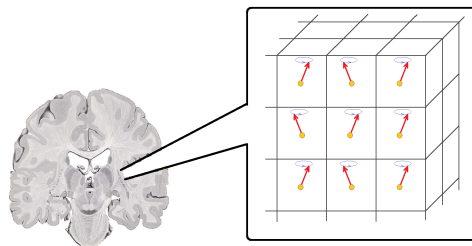


FIGURE 4. Cutting the sample into isochromats. A magnetization vector is associated to each isochromat and its evolution is monitored during the acquisition sequence. The collected MR signal corresponds to the transverse component of the magnetization.

where $\mathbf{G}(t)$ is the gradients sequence, \mathbf{r} is the isochromat position, $\Delta B(\mathbf{r}, t)$ corresponds to the field inhomogeneities due to off-resonance and non-uniform gradients, and $\mathbf{B}_1(\mathbf{r}, t)$ the radio frequency pulses sequence.

In the Lagrangian version presented here, Bloch equations are a simple ordinary differential equation (ODE) system. The software we use for our simulations, JEMRIS, solves those equations with the ODE solver CVODE, which is part of the SUNDIALS³ libraries. Numerical solving is based on Adams-Moulton linear multistep method.

1.4.4. Flow motion simulations

Natively, Bloch equation solving in JEMRIS is only dedicated to simulate static tissues. So, we provide an extension to the software in order to simulate fluid travelling [8]. Movements are taken into account with a Lagrangian approach, which requires to determine each individual spin trajectory. While solving Bloch equations, this approach avoids the need to use a different treatment for static tissues and flowing particles. We simply vary the position of the spin over time, which changes the field value seen by the particle:

$$\mathbf{r} = \mathbf{r}(t) \quad \Rightarrow \quad \mathbf{B}(\mathbf{r}, t) = [\mathbf{G}(t) \cdot \mathbf{r}(t) + \Delta B(\mathbf{r}, t)] \cdot \mathbf{e}_z + \mathbf{B}_1(\mathbf{r}, t)$$

By default, JEMRIS only allows to specify one trajectory for all spins, in order to simulate movement of a rigid sample. So we added a specific class to the C++ code to allow users to specify a different trajectory for each spin. A simple example of MRI simulations performed with our modified version of JEMRIS is shown on Fig. 5. As it appears on that image, we used four spins with specific individual behaviors: two static spins (at the top of the image), and two moving spins with diagonal trajectories (at the bottom).

With this new version, it thus becomes possible to describe flow phenomena, after converting flow data to the proper format (Fig. 6).

2. BENCHMARK SETUP

The aim of this benchmark is to validate the CFD simulations by comparing the numerical outputs with experimental data. For that purpose, a physical phantom was designed as a double bifurcation fluid circuit so that it can reproduce acceleration, deceleration and recirculation, all of which can be encountered in real vascular networks. The characteristics dimensions of the model were carefully chosen so that it mimics human cerebral little vessels, like lingual emissary veins.

The inlet branch, the left channel, has a diameter of 5 mm and a length of 40 mm, the lower left and lower right branches have a diameter of 2 mm and a length of 30 mm each, the upper left and upper right branches have a diameter of 3 mm and a length of 30 mm, and finally the outlet branch, the right channel, has a diameter

³<https://computation.llnl.gov/casc/sundials/main.html>

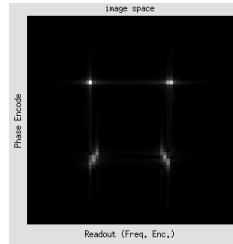


FIGURE 5. Our modifications allow JEMRIS to simulate individual trajectories for each spin. Here, an example of simple gradient echo MRI sequence with two static spins at the top and two moving spins at the bottom.

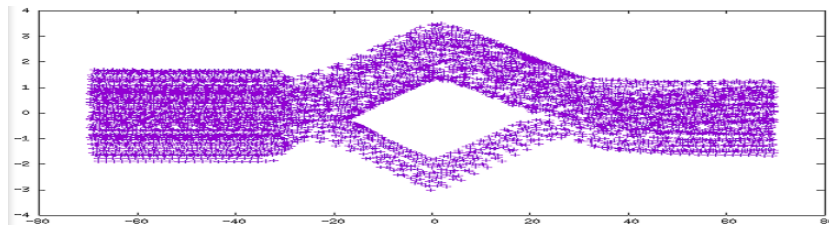


FIGURE 6. The Lagrangian approach we used to simulate flow motion requires to convert the velocity field determined with Navier-Stokes equations into individual particles trajectories, with a path tracer code.

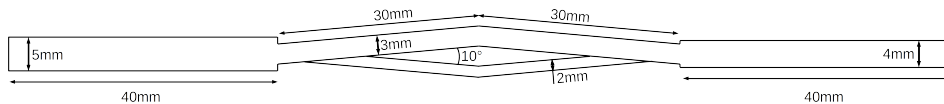


FIGURE 7. The physical phantom.

of 4 mm and a length of 40 mm. The angle between the two branches of the bifurcation is 10° (see Fig. 7). The physical phantom is machining in a rectangular block of Plexiglas that we consider as a non deformable material.

The fluid should have been chosen so that it satisfies isothermal and incompressible Newtonian blood properties, but instead, for the sake of simplicity, we used water to perform our experiments. The constant density and the dynamic viscosity are then $0.001g/mm^3$ and $0.001Pa \cdot s$, respectively.

For the MRI measurements, we considered the following setup. We used a Masterflex roller pump at a flow rate of 100 mL/min to inject fluid into the phantom, which is in the MRI. The fluid then goes to a reservoir crossing a heart rate getting which is used for the PC-MRI acquisition, see Fig. 8.

We used a 3T Philips Achieva dStream MRI machine for all acquisitions at the University Hospital of Amiens, Picardie. The microscopic coils used for all acquisitions have an inner diameter of 47 mm.

Morphological MRI

In order to obtain the geometry of the phantom, we performed a morphological MRI.

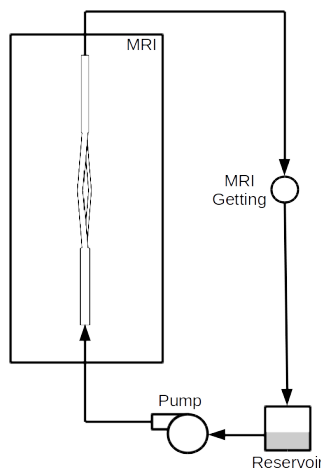


FIGURE 8. MRI benchmark.

The sequence used is a 3D phase contrast angiography with a flip angle of 12° for a pixel size of $0.5 \text{ mm} \times 0.5 \text{ mm}$ and a slice thickness of 0.6 mm .

Flow MRI

To obtain the flow measurements in the different branches of the phantom, we performed a phase contrast MRI (PC-MRI). We used 2D QFlow sequences with an echo time of 11 ms , a repetition time of 18 ms , a flip angle of 30° , a slice thickness of 1 mm and finally a pixel size of $0.35 \text{ mm} \times 0.35 \text{ mm}$. The field of view is $30 \text{ mm} \times 30 \text{ mm}$, the number of slices by cycle is 32 and the acquisition time is around 2 minutes . Encoding velocities are summarized in Tab. 1.

SLICE	ENCODING VELOCITY
Inlet	400 mm/s
Outlet	600 mm/s
Other	550 mm/s

TABLE 1. The characteristics of the flow MRI.

MR images were processed at the BioFlowImage laboratory with the home-made software Flow Analysis [3].

3. SEGMENTATION RESULTS

3.1. Comparison MRI geometry – realistic geometry

First, we compare the exact geometry with the MRI geometry to assess the accuracy of the MRI measurements. For this purpose, we need to segment the MRI images and we use two methods: the first is a simple thresholding (Fig. 9(a)) while the second is a snake included in the software ITK-SNAP and detailed in subsection 1.1 (Fig. 9(b)). To estimate the differences between the realistic geometry and the MRI geometry, we measure diameters at the radial slices given on Fig. 13 and we list results in Tab. 2.

We can notice that the thresholded MRI is thinner than the segmented MRI, while the segmented MRI is larger. This last issue is due to the snake which evolves outside the boundary of the desired object. In fact,

the MRI geometry can not be larger than the realistic geometry, so the thresholded MRI is closer to the reality. Nevertheless a lot of information is lost, certainly because of the smoothness. Another issue is the difficulty to match the two geometries. Indeed, the plane of the MRI acquisition is slightly inclined to the horizontal plane and as a result, we have to perform a transformation to align the two geometries.

In both cases, the input of the MRI does not correspond to the exact input, as well as the output. Indeed, the MRI tends to smooth the straight forms. These differences can skew the results of the MRI measurements in the next parts and the MRI acquisition should be improved in the future.

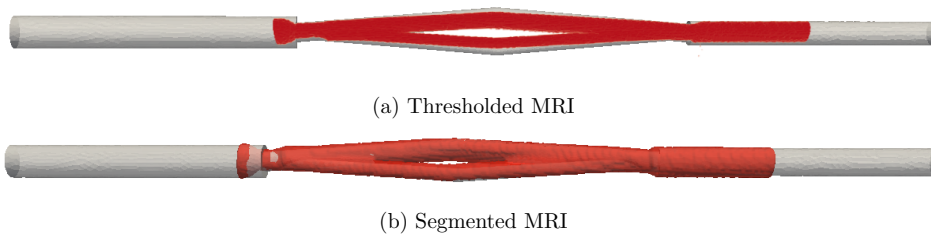


FIGURE 9. Comparison of the realistic geometry with MRI segmentations, the first is obtained by a simple threshold while the second by the classical snake approach.

RADIAL SLICES	IN	UL	LL	UC	LC	UR	LR	OUT
Realistic mesh	5.00	3.00	2.00	3.00	2.00	3.00	2.00	4.00
Thresholded MRI	3.36	2.70	1.97	2.45	1.73	2.70	1.97	3.12
Segmented MRI	5.10	5.78		3.60	2.30	5.27		4.45

TABLE 2. Comparison of the diameters (mm) at the radial slices : IN=inlet, UP=upper left, LL=lower left, UC=upper center, UL=upper left, UR=upper right, LR=lower right and OUT=outlet. As the UL and LL parts and the UR and LR parts respectively are stucked together in the segmented IRM, we compute the diameters on the entire left slice and on the entire right slice respectively which are theoretically equal to 5.11mm.

3.2. Comparison AngioTK geometry – realistic geometry

We used AngioTK up until the numerical simulation step as a tool to compare ground truth geometries built from real measurements and geometries extracted from MRI. In Fig. 10, we illustrate the different steps of mesh processing that lead to a mesh suitable for numerical simulations.

From left to right and from top to bottom, we have:

- The original MRI data visualized as a block (a) and with volume rendering (b);
- The initial mesh extracted from MRI data (c). This extraction is performed based on the level-set method;
- We then build centerlines (d) on the extracted mesh. Those centerlines carry additional information, like the radius of the maximal inscribed sphere at each point;
- Based on the centerlines, we generate a new volume of data (e) and extract a mesh suited for numerical simulations (f); this mesh is then opened in (g) to define adequate sections for applying boundary conditions;
- We eventually generate a volumic mesh using tetrahedra and partition the mesh for parallel simulations in (h) (here on 16 processors).

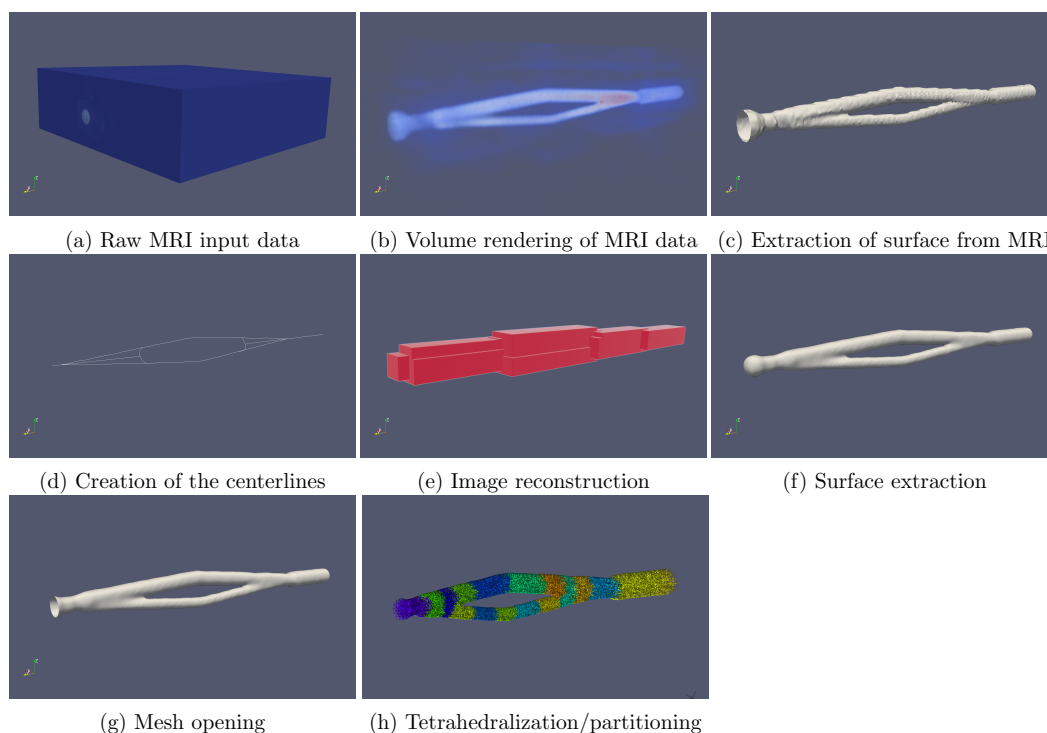


FIGURE 10. A sample execution of the AngioTK pipeline on the phantom dataset.

Let us first have a look at the differences between the original MRI data and the final mesh generated with AngioTK in Fig. 11.

Fig. 11 (a) and (b) show that the MRI did not capture the entire Phantom during the acquisition. Indeed, we are losing a part of inlet and outlet tubes through which the water flowed. Fig. 11 (c) show the mesh extracted and processed with AngioTK. Fig. 11 (d) superimposes the original MRI data and the extracted mesh. It shows that we correctly reconstructed the shape of the Phantom through MRI processing and mesh generation. However, AngioTK did not accurately capture the beginning of the inlet and the end of the outlet parts. This is due to the fact that the centerlines are computed using the radius of the maximal sphere inscribed inside the mesh. Thus capturing the required parts would require a finer tuning of the whole process and extending the inlet and outlet parts. A second issue appears at the two bifurcations of the tube. In the real phantom, the tubes are cut through plexiglas, so as they diverge/converge, the separation is sharp. We can however notice with the reconstruction that the tubes are joined together for a certain distance when diverging/converging. This is again due to the reconstruction of the phantom using spheres, as AngioTK is designed to be used with veins and arteries.

To measure the accuracy of the reconstruction, we illustrate in Tab. 3 the differences between real data (cf. Fig. 7) and measured data in term of tube width, based on the data associated to the computed centerlines. We compare the tube width of internal sections to avoid the parts influenced by the use of AngioTK as previously mentioned. The sections of the mesh from which we extract the data are described in Fig. 12.

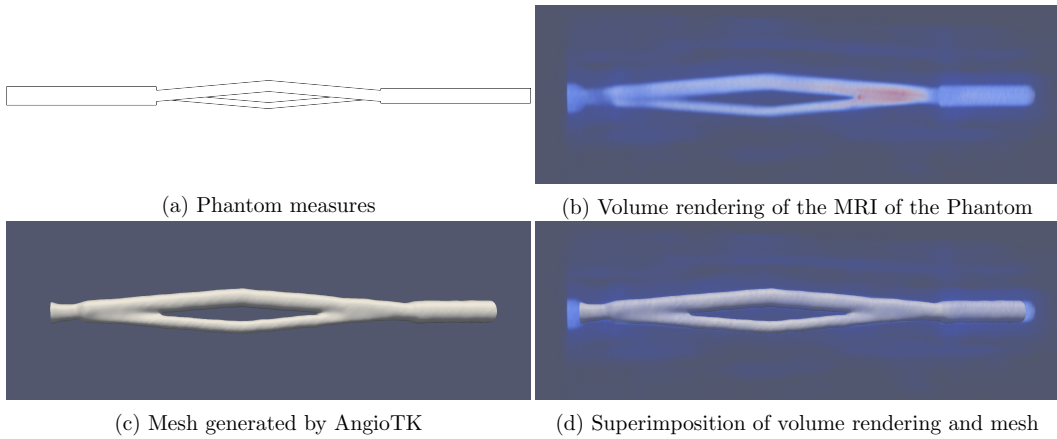


FIGURE 11. Comparison between the original MRI data and the final mesh generated by AngioTK.

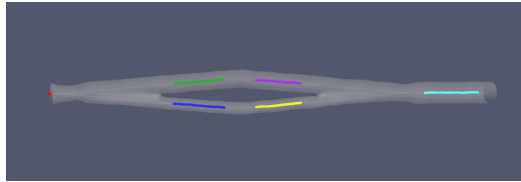


FIGURE 12. Sections on which we compute the minimal and maximal inscribed sphere radius. The different sections are the following: the inlet in red, the upper branch section 1 in green, the upper branch section 2 in purple, the lower branch section 1 in blue, the lower branch section 2 in yellow and the outlet in cyan.

For the size to be comparable, we decided to compute a ratio between the real phantom tube size and the size of the smallest and biggest inscribed sphere for the considered sample centerline points. Note that the maximal inscribed sphere information is computed at each centerline point. If the reconstruction is performed correctly, we should get a similar ratio between the different section sizes.

As it can be seen in Tab. 3, the ratios are close to 2 for the different sections, so we get a good match of the original size. The only problematic case is at the inlet, where it strongly differs from the other ratios. This is again due to the fact that we were not able to grasp the real shape of this section due to the incomplete acquisition.

4. SIMULATIONS RESULTS

The fluid flow simulations were performed using FEEL++ and FREEFEM++. The finite element spaces used to discretize the velocity and the pressure are the Inf-Sup stable Taylor-Hood finite elements $P2P1$.

For the FREEFEM++ simulations, we used the method of characteristics [4] described in section (1.2.2). In order to speed-up the resolution, this method is coupled with an iterative Uzawa Conjugate Gradient algorithm [15] with a Cahouet-Chabart preconditionner [5]. As for the FEEL++ simulations, we used a second

Section	Original size (mm)	Number of sample points	Size of inscribed sphere (no unit, min-max values)	Ratio (original size / inscribed sphere size)
Inlet branch (Fig. 12 in red)	5	2	[1.634, 1.770]	3.060, 2.824
Upper branch (Fig. 12 in green)	3	90	[1.489, 1.553]	2.014, 1.931
Upper branch (Fig. 12 in purple)	3	83	[1.552, 1.593]	1.932, 1.883
Lower branch (Fig. 12 in blue)	2	93	[0.985, 1.056]	2.03, 1.893
Lower branch (Fig. 12 in yellow)	2	83	[1.023, 1.079]	1.955, 1.853
Outlet branch (Fig. 12 in cyan)	4	97	[1.981, 2.024]	2.019, 1.976

TABLE 3. Size comparisons between the size of the real phantom and the data extracted from MRI data.

order finite difference scheme to approximate the time derivative and a second order extrapolation of the non-linear convective term [19], described in section (1.2.1). The resolution in the latter is made using the LU solver. The discretized Navier-Stokes equations were supplemented for both FEEL++ and FREEFEM++ simulations by a no-slip boundary conditions on the lateral surface of the computational domain, since we are dealing with viscous fluid. At the inflow and the outflow we imposed a Dirichlet and a Neumann boundary conditions, respectively.

Although the choice of a Neumann-type boundary conditions at the outflow in physiological flows may lead to possible instabilities with recirculations because of the loss of energy estimates, we did not use stability techniques and did not observe any instability at the channel exit because the exit is sufficiently far, and the Reynolds number is in the lower laminar range.

For the Dirichlet boundary conditions two flow profiles were implemented at the inlet of the channel: *i*) a Poiseuille profile with constant flow and *ii*) a pulsatile flow retrieved by using the experimental measurements of the flow at the inlet during a 5s MRI acquisition.

We start our simulations with a fluid at rest, that is, $p = 0$ and $\mathbf{u} = \mathbf{0}$ everywhere in Ω . In this part, in order to avoid additional geometric errors, the geometry was constructed using real dimensions described in Section (2). The meshing of the geometry was carried out using Gmsh 2.8.3 [9] and the partitioning using Metis [11] for the FEEL++ algorithm. An adapted mesh size was chosen depending on the diameter of the channel. A “ h_{max} ” was prescribed in the inlet and outlet channels, a “ $h_{average}$ ” was prescribed on the upper channel of the bifurcation and a “ h_{min} ” was set in the lower channel of the bifurcation. Three levels of refinements were performed in order to make a mesh convergence study. The corresponding mesh characteristics are reported in Tab. 4.

GEOMETRY	h_{min}	h_{max}	$h_{average}$	TETRAHEDRA	DOF
M1	0.2	0.5	0.3	157245	769662
M2	0.25	0.625	0.375	93655	469008
M3	0.3	0.75	0.45	60349	307510

TABLE 4. The characteristics of the three types of geometries.

The comparison is made in terms of axial component of the velocity and pressure along the centerline and at various radial sections (see Fig. 13) and normalized wall pressure difference along the length of the domain.

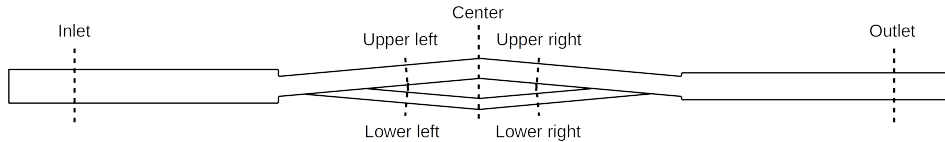


FIGURE 13. Radial slices where the velocity profiles are plotted.

4.1. Comparison Feel++– FreeFem++ in the phantom

The numerical fluid simulations performed in the phantom geometry using FEEL++ and FREEFEM++ are quantitatively of the same order of Fig. 14. But in order to carry out a cross validation of these two finite elements method libraries, we compared pressure and velocity profiles at six different radial sections previously defined, and along the centerline.

A constant velocity of 82mm/s (corresponding to the lowest velocity of the MRI measurements that we describe later) is imposed at the inlet for one second. The results are shown for the final time step. The simulation time run was fixed at 2.02s, the time for the fluid to cross all the geometry domain. A steady state was reached at this time.

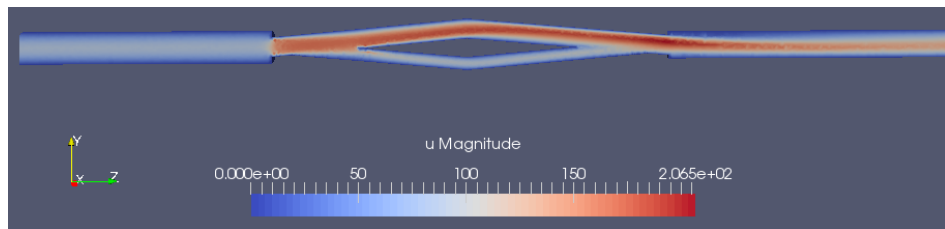


FIGURE 14. Velocity magnitude, M3 mesh, constant flow.

Pressure and velocity along the centerline are shown in Fig. 15 for FEEL++ and FREEFEM++ with a mesh convergence. We can see that FREEFEM++ gives a slightly higher pressure gradient than FEEL++, for the three levels of refinement. However, velocity profiles are very similar.

At the inlet section, the profiles overlap, retrieving the Poiseuille profile boundary condition imposed at the inlet. As for the outlet, the velocity profiles show also a good agreement.

Velocity profiles at the lower and upper channels, left and right sections are shown in Fig. 17. Some differences can be observed with the coarse mesh M3, but the mesh convergence shows that the results tend towards the same solution.

Most of the results are equivalent in this FEEL++–FREEFEM++ comparison. We believe that the observed differences can be due to the choice of algorithms for the two FEM libraries; one uses a second order time discretization while the other uses a first order one. FEEL++ algorithm equally uses a domain decomposition method that can impact the computed solutions.

Even if these differences exist, they are quite small and we can see a large correspondence of the results for the two libraries.

The Reynolds number in the constant and pulsatile flow cases varies between 200 and 700, thus the inertial terms could not be neglected. However, no recirculation or instabilities have been observed. Recirculations may

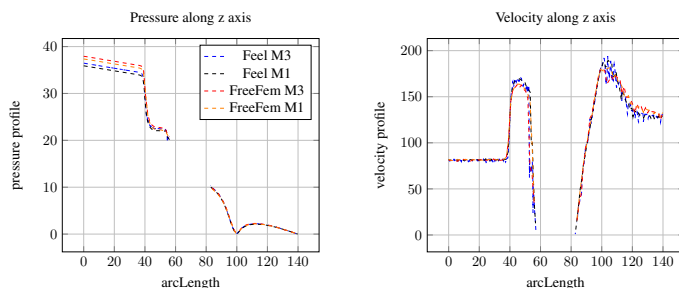


FIGURE 15. Comparison FEEL++ vs FREEFEM++ for a constant flow (V_{min}), $P2P1$: Velocity and pressure profiles along z axis.

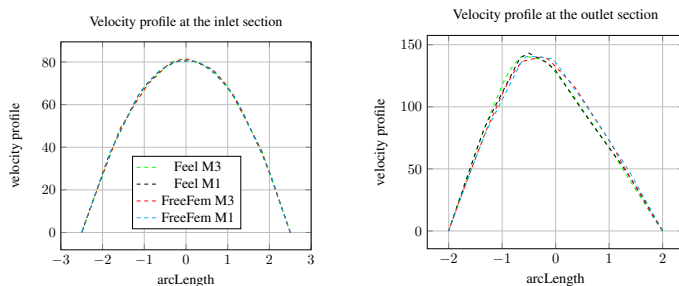


FIGURE 16. Comparison FEEL++ vs FREEFEM++ for a constant flow (V_{min}), $P2P1$: Velocity at the inlet and outlet sections.

be observed near the branch with a very fine mesh, but such simulation results are quite difficult to obtain in 3D due to necessary computational resources.

4.2. Comparison Numerical simulations – MRI measurements in the phantom

Now a pulsatile flow, obtained by PC-MRI measurement, is used to impose the velocity at the inlet of the phantom.

Comparison between FEEL++ and FREEFEM++ simulations and PC-MRI measurement is shown in Fig. 18 for the input and the output, and in Fig. 19 for the two branches.

The input and output flows are very similar, both between the two simulations and between simulations and experiment. Branches flow are equally similar for the two simulations but MRI measurements are slightly higher. These differences can be explained by the fact that PC-MRI images have to be manually segmented to obtain velocity and flow informations, a process that usually overestimates the real conditions of the experiment: operator dependent, manually segmented on a screen and dependent of the slice angle during the MRI sequence. At the inlet and the outlet, the pipe is quite large and the segmentation is very accurate, but in the two small branches greater errors may occur.

4.3. Comparison FEEL++– FreeFem++ in the cerebral venous network

The final goal of the *Vivabrain* project is to compute virtual cerebral angiography images. It is still too early to perform MR acquisition of complete angiographic images, but we have already reached the stage of blood

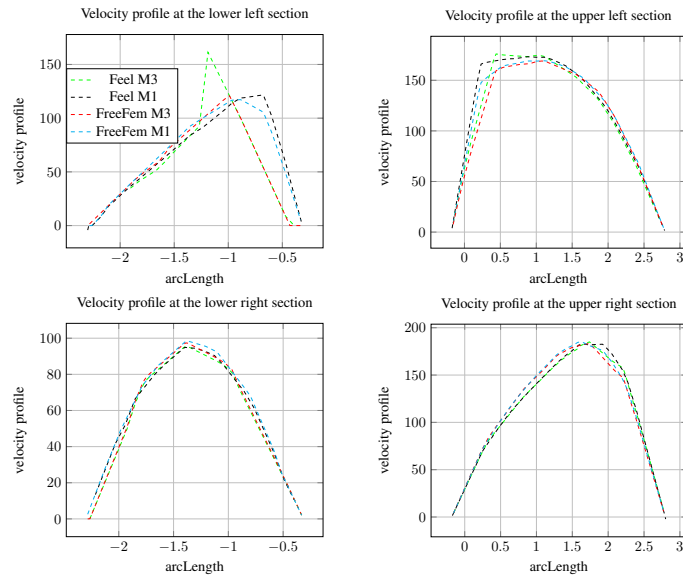


FIGURE 17. Comparison FEEL++ vs FREEFEM++ for a constant flow (V_{min}), *P2P1*: Velocity profile at the right and left sections in the upper and lower channels.

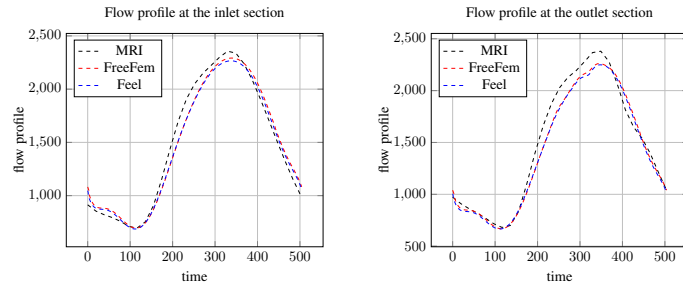


FIGURE 18. Comparison FEEL++ vs FREEFEM++ on the M3 mesh with a pulsatile flow, flow profile at the inlet and outlet sections during time.

flow simulations in the complex cerebral venous network. In this type of realistic geometry, a main issue remains the experimental validation of the results. If the physical phantom can reach that goal in the case of simple bifurcations, such approaches are hardly tractable for complex network. Therefore, finding a way to validate this key step is essential. In this part, we run a cross-validation between the FEEL++ and FREEFEM++ results in this realistic geometry.

First, we describe briefly an appropriate model for flow in the cerebral venous network (see [12] for more details). At the macroscopic scale, the brain venous network (see Fig. 20) is composed by – input – veins (7)–(11) draining the blood into the superior sagittal sinus (2) and the straight sinus (3), until their confluence

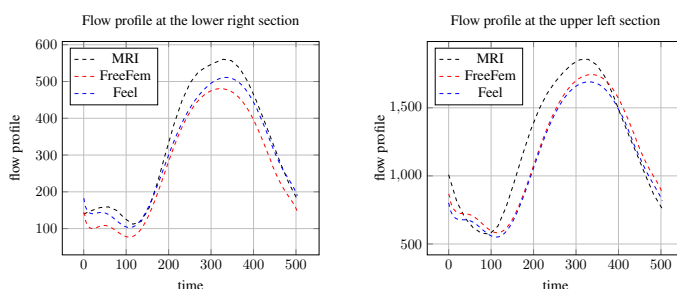


FIGURE 19. Comparison FEEL++ vs FREEFEM++ on the M3 mesh with a pulsatile flow, flow profile at the left sections in the upper and lower channels during time.

(4). The blood then passes into the transverse (5) and sigmoid parts (6) of the lateral sinuses, and reaches an extracranial area, composed of the – output – internal jugular veins (1).

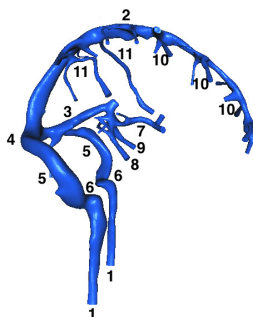


FIGURE 20. Cerebral venous network.

Concerning the flow in these vessels, we adopt standard assumptions: (i) the blood density is constant; (ii) the flow is assumed to be incompressible and isothermal; (iii) blood is supposed to be Newtonian. Another important issue is the relevance of using either a complex fluid-structure interaction model or a fluid model (hence neglecting the interaction with the vessel wall). Intracranial veins are quite constrained in the brain, thus they are considered as rigid.

The inflow is situated at the microcirculation exit, far from the thorax, hence far from the effects of breathing and cardiac pumping [18]. Since flow is not fully developed, both Poiseuille (steady case) or Womersley (unsteady case) flows are not appropriate. A constant velocity of small magnitude, due to microcirculation exit, is then employed. In summary, the boundary conditions are: (i) inflow: a steady profile; (ii) outflow: homogeneous natural conditions; (iii) lateral boundary: no-slip condition, since walls are assumed to be rigid.

Taking into account all the previous hypotheses leads to consider the Navier-Stokes equations for modeling the blood flow dynamics. As a first approximation, we solve instead the corresponding steady Stokes problem with the same boundary conditions as those considered in the Navier-Stokes equations, and with a dynamic viscosity of $3.5 \text{ mm}^2/s$.

Concerning the Dirichlet boundary conditions on the inlet and the wall, there are two ways to impose them (see Fig. 21). The first way is to first impose a constant velocity at the inlet and then a no-slip condition

(zero velocity) on the wall. In this case, the velocity is null on the wall and on the elements which are at the intersection between the inlet and the wall sections of the boundary. The second way is to first impose the no-slip condition on the wall and then to impose the constant velocity at the inlet, therefore the velocity is zero on the wall, except for the elements which are at the intersection between the inlet and the wall sections of the boundary where it is equal to the constant velocity imposed on the inlet section. None of these methods is completely satisfactory and we are discussing to impose a smooth function at the inlet, which preserves the H^1 framework, for a futur work.



FIGURE 21. Different ways to impose the boundary conditions.

We run the Stokes problem on the realistic geometry and we take an initial inflow of $6000 \text{ mm}^3/\text{s}$ which corresponds to an initial velocity of 28 mm/s . The areas of the input and output veins calculated by the two libraries are identical, contrary to the flows. These data are described in Tab. 5 and are given in percentage of the total area and total flow. On the one hand, we can notice for the two libraries that the total inflow is preserved in the Inlet/Wall case but is significantly lower than the initial flow (respectively $5164 \text{ mm}^3/\text{s}$ in `FREEFEM++` and $5406 \text{ mm}^3/\text{s}$ in `FEEL++`). On the other hand, the total inflow is no longer preserved in the Wall/Inlet but the inflow ($-5985 \text{ mm}^3/\text{s}$ and $-5987 \text{ mm}^3/\text{s}$) and the outflow ($6227 \text{ mm}^3/\text{s}$ and $6175 \text{ mm}^3/\text{s}$) are closer than the initial flow. Finally, the inflows and outflows are very close in `FREEFEM++` and `FEEL++` and regarding in percentage of the total flow, they are substantially equal. This would ensure a correct simulations.

4.4. Comparison MRI simulation – Numerical simulations

MRI simulations were performed using JEMRIS [17] with constant flow computed from numerical simulations in the phantom. The input required in order to perform the simulations are the trajectories of a fixed number of particles, that are the positions along the flow at each time step, for the whole MRI sequence duration. For constant and pulsatile flow, a VTK [16] script was implemented to save the *particle tracer* of a sphere of point sources with its corresponding computed velocity field.

We show below a simulated phase contrast image of constant flow obtained with JEMRIS (Fig. 22). About 19 000 spins trajectories were calculated on a time interval of 5s (700 MB of data) from numerical data calculated in amount for constant flow with `FEEL++` and `FREEFEM++`. We used a phase contrast sequence with resolution 128, matrix 180×30 , TE of 8 ms, TR of 100 ms, Venc of 400 mm/s and $N_{\text{ex}}=1$. Calculations took only 15 minutes with 20 CPU on Romeo supercomputer from Reims⁴.

This velocity image was then filtered with its corresponding signal magnitude image (Fig. 23). To do this, we applied a mask to set to zero the pixels of the velocity map when the corresponding signal intensity in that pixel was inferior to a fixed threshold value. Thus, only the significant values of velocity present in the phantom geometry appear on the final image. We can notice an important lack of spins in the lower branch because of the low number of flow particles traveling along that path.

The velocities measured in each branch give right order of magnitude compared to initial data calculated with `FEEL++` and `FREEFEM++` (Tab. 6 and Fig. 24). However, the low resolution used for MRI simulation lead to an averaging of the velocities in the voxels⁵ (partial volume effect). This phenomenon lead to an underestimation of the peak value in each branch of the phantom, as the values measured are averaged on a wide range of velocities. This effect is particularly severe in the thinnest branch, where the gradients of

⁴<https://romeo.univ-reims.fr/>

⁵Voxel: 3D pixel.

LABEL	AREA (%)	FLOW INLET/WALL		FLOW WALL/INLET	
		FREEFEM++ (%)	FEEL++ (%)	FREEFEM++ (%)	FEEL++ (%)
INPUT					
1	5.9	5.12	5.11	5.09	5.09
2	1.98	1.96	1.97	1.98	1.98
3	1.85	1.83	1.84	1.86	1.86
4	1.26	1.23	1.24	1.26	1.26
5	1.45	1.43	1.44	1.42	1.45
6	4.94	4.98	4.97	4.94	4.94
8	1.25	1.23	1.24	1.25	1.25
9	2.52	2.52	2.52	2.52	2.52
10	6.05	6.09	6.07	6.05	6.05
11	5.23	5.24	5.24	5.21	5.22
12	6.22	6.28	6.26	6.22	6.22
13	1.56	1.54	1.55	1.56	1.56
14	0.85	0.83	0.84	0.85	0.847
15	6.42	6.41	6.41	6.42	6.42
16	7.28	7.29	7.29	7.28	7.28
17	1.70	1.68	1.68	1.70	1.70
18	1.89	1.87	1.88	1.90	1.90
19	0.96	0.93	0.94	0.96	0.96
20	2.43	2.40	2.41	2.43	2.43
21	0.52	0.51	0.51	0.52	0.52
22	2.64	2.61	2.62	2.64	2.64
23	6.30	6.34	6.33	6.30	6.31
25	3.54	3.54	3.54	3.54	3.54
26	5.43	5.46	5.45	5.43	5.43
27	3.92	3.92	3.92	3.93	3.92
28	5.78	5.83	5.81	5.78	5.78
29	6.34	6.36	6.35	6.34	6.34
30	1.18	1.16	1.17	1.18	1.18
31	3.40	3.39	3.39	3.40	3.40
TOTAL	214.85	-5163.93	-5405.517	-5984.6	-5987.07
	mm ²	mm ³ /s	mm ³ /s	mm ³ /s	mm ³ /s
OUTPUT					
7	45.08	69.18	69.1	69.19	69.2
24	54.92	30.82	30.9	30.81	30.8
TOTAL	37.64	5163.78	5405.51	6227.29	6175.05
	mm ²	mm ³ /s	mm ³ /s	mm ³ /s	mm ³ /s

TABLE 5. Comparison between flows in cerebral veins, calculated by FreeFem++ and Feel++. These flows are given in percentage of total inflow and outflow.

velocities are very high inside a voxel, leading to an important underestimation, even more increased by the lack of particles.

Those problems should be solved, in the future, by increasing the total number of particles (to increase the number of spins in the lower branch) and by enhancing image resolution (in order to limit partial volume effects). This will lead, on the other hand, to an increase of data files volume and of total computation time. Simulations of pulsatile flow will also be performed in short term works, but it will probably need to reuse the

same trajectories of a few number of cardiac cycles translated in time, in order to cover the whole MRI sequence duration. Otherwise, the trajectories data volume could rapidly become prohibitive. Another step to improve our simulations in the future will be to check the effect of spatial and temporal spins density variations on the quality of the velocity measured.

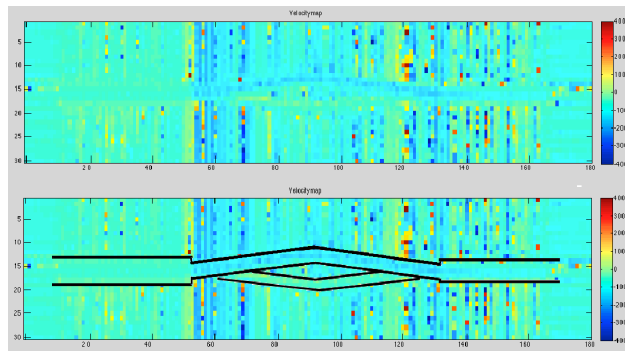


FIGURE 22. Above: Velocity map simulated with phase contrast sequence in JEMRIS. Below: Superimposition of the simulated image with the phantom geometry. The remainder of the image is noise due to air.

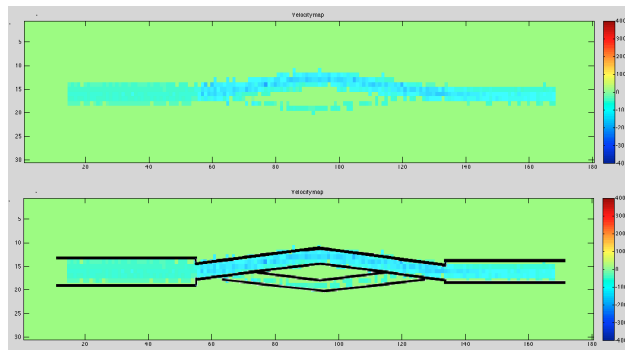


FIGURE 23. Above: Same image as Fig. 22, filtered with its corresponding magnitude image. Below: The filtered velocity map matches well with the initial phantom geometry, but we can notice a lack of particles in the lower branch.

5. CONCLUSION

The MRI-realistic geometry comparison shows qualitative differences between real and MRI-measured diameters of each part of the phantom. This study emphasizes the necessity of segmentation or threshold procedure improvement for a physiological use.

The FEEL++-FREEFEM++ comparison in the phantom shows a huge correlation between the computed solutions. These kinds of comparison allow us to confirm the good agreement of the solution regarding to a

BRANCH	FEEL++/ FREEFEM++	JEMRIS
Inlet	80mm/s	70mm/s
Outlet	140mm/s	110mm/s
Upper branch	170mm/s	150mm/s
Lower branch	110mm/s	60mm/s

TABLE 6. Comparison of velocity peaks measured in the center of each branch with JEMRIS vs Numerical simulations. In each branch, the low resolution used for MRI simulations lead to an averaging and an underestimation of the peak velocity (type of partial volume effect), particularly severe in the thinnest branch. This could be corrected with more particles trajectories and higher resolution.

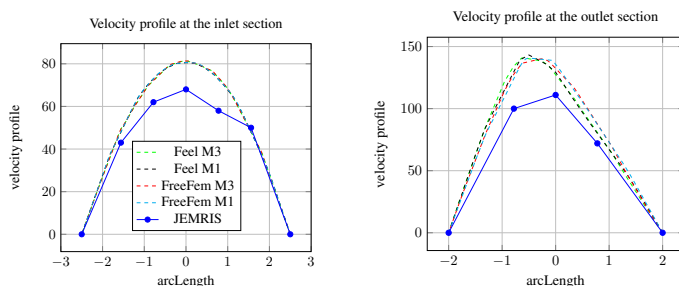


FIGURE 24. Velocity profiles at the inlet and outlet sections measured with JEMRIS. Velocities measured give a good order of magnitude compared to initial numerical simulations data but we can observe an underestimation of velocity peak, due to low resolution and partial volume effects.

physiological use. The FEEL++-FREEFEM++-PC-MRI comparison shows qualitative agreement of the flow in all the phantom. In the realistic geometry of the cerebral venous network, a good agreement of the solutions is equally found between FEEL++ and FREEFEM++ results. However, geometrical errors shown in the first part of the pipeline should not be neglected. For example, if a lower section is used to perform simulations, a higher pressure will be calculated than the physiological one.

The JEMRIS - FEEL++ FREEFEM++ comparison with constant flow showed a good qualitative agreement for the velocity profiles, but an underestimation of peak velocities. This should easily be corrected with more particles trajectories and higher image resolution in the virtual MRI simulations.

To conclude, all the steps of the pipeline are developed and now, we have to adapt them for a realistic geometry.

6. ACKNOWLEDGMENTS

Thanks to Stéphanie Salmon for the supervision of this project. Thanks to Stéphanie Salmon, Marcela Szopos, Christophe Prud'homme, Nicolas Passat, Hugues Talbot, Olivier Balédent, Emmanuel Durand for the supervision of the authors' PhD theses. Thanks to Vincent Chabannes and Odysée Merveille for their help during the CEMRACS. Thanks to Gwenaél Pagé for the MRI acquisitions. Thanks to PFT Innovaltech for machining the phantom. This project was funded by Agence Nationale de la Recherche (Grant Agreement ANR-12-MONU-0010 (VIVABRAIN)). This work was granted access to the HPC resources of Aix-Marseille Université financed by the project Equip@Meso (ANR-10-EQPX-29-01) of the program "Investissements d'Avenir" supervised by

the Agence Nationale de la Recherche. We would like to thanks the IRMIA Labex at IRMA, Strasbourg for funding and Cemosis for the support and development environment provided for this project.

REFERENCES

- [1] AngioTK. <http://www.cemosis.fr/interdisciplinary-bio-project/2015/09/01/angiotk/>.
- [2] VIVABRAIN project. <http://vivabrain.fr>. Accessed: 2014-05-14.
- [3] O. Balédent, I. Idy-Peretti, et al. Cerebrospinal fluid dynamics and relation with blood flow: A magnetic resonance study with semiautomated cerebrospinal fluid segmentation. *Investigative Radiology*, 36(7):368–377, 2001.
- [4] J.P. Benque, B. Ibler, and G. Labadie. A finite element method for Navier-Stokes equations. In *Numerical Methods for Non-Linear Problems*, volume 1, pages 709–720, 1980.
- [5] J. Cahouet and J.P. Chabard. Some fast 3D finite element solvers for the generalized Stokes problem. *International Journal for Numerical Methods in Fluids*, 8:869–895, 1988.
- [6] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods - Fundamentals in Single Domains*. Springer-Verlag, Berlin Heidelberg, 2006.
- [7] V. Caselles, R. Kimmel, and G. Sapiro. On geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.
- [8] A. Fortin, S. Salmon, and E. Durand. Extension of an mri simulator software for phase contrast angiography experiments. In *Biomedical Simulation*, volume 8789 of *Lecture Notes in Computer Science*, pages 150–154, 2014.
- [9] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.
- [10] F. Hecht. New development in Freefem++. *Journal of Numerical Mathematics*, 20:251–265, 2012.
- [11] G. Karypis and V. Kumar. Metis-unstructured graph partitioning and sparse matrix ordering system, version 2.0. 1995.
- [12] O. Miracourt, O. Génévaux, M. Szopos, M. Thiriet, H. Talbot, S. Salmon, and N. Passat. 3D CFD in complex vascular systems: A case study. In *Biomedical Simulation*, volume 8789 of *Lecture Notes in Computer Science*, pages 86–94, 2014.
- [13] C. Prudhomme, V. Chabannes, G. Pena, and S. Veys. Feel++: Finite element embedded language in c++. *Free Software available at <http://www.feelpp.org>. Contributions from A. Samake, V. Doyeux, M. Ismail and S. Veys*.
- [14] A. Quarteroni. *Numerical Models for Differential Problems*, volume 2. Springer, 2009.
- [15] J.E. Roberts and J.M. Thomas. Mixed and hybrid methods. *Handbook of Numerical Analysis.*, 2:523–639, 1991.
- [16] W.J. Schroeder, B. Lorensen, and K. Martin. *The visualization toolkit*. Kitware, 2004.
- [17] T. Stocker, K. Vahedipour, D.I. Pflugfelder, and N.J. Shah. High-performance computing MRI simulations. *Magnetic Resonance in Medicine*, 64:186–193, 2010.
- [18] M. Thiriet. *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, 2008.
- [19] S. Turek. A comparative study of some time-stepping techniques for the incompressible Navier-Stokes equations: From fully implicit nonlinear schemes to semi-implicit projection methods. *International Journal for Numerical Methods in Fluids*, 22(2):987–1011, 1996.

Annexe F

SEME 2016

2D spatio-temporal simulation of perfusion data following the tracer kinetic field theory

Simon GARNOTEL, Ihab HAIDAR, Thi Khuyen LE

SEME 2016

Contents

1	Introduction	1
2	Weak form	2
2.1	Time discretization	3
2.2	Reformulation of the problem	3
3	Numerical simulation	4
3.1	Numerical scheme	4
3.2	Numerical stability	4
3.2.1	Without compartment	4
3.2.2	With compartments	5
4	Results	6
4.1	No compartments	6
4.2	Two compartments, same tissues	7
4.3	Two compartments, different transfer constants	8
4.4	Two compartments, different tissues	9
5	Conclusion and perspectives	10
A	FREEFEM++ algorithms	12
A.1	No compartment	12
A.2	Two compartments	13

1 Introduction

The perfusion MRI technique aim to characterize the mcirculation by analyzing the concentration-time profiles of an indicator in blood.

Sourbron [3] introduce a theory called the «tracer-kinetic field theory» , represented by the equation Eq. (1), that quantify the concentration along the time function of different parameters of the tissues. To better represents the system, a multi-compartmental formulation is designed, including Q compartments.

$$\frac{\partial C^q}{\partial t} = -\nabla \cdot (\mathbf{u}^q C^q) + \nabla \cdot (D^q \nabla C^q) - K^q C^q + \sum_{p \neq q} K^{qp} C^p - \sum_{p \neq q} K^{pq} C^q \quad (1)$$

Where C^q is the concentration of the compartment q , \mathbf{u}^q is the velocity, D^q is the diffusivity, K^q is the decay rate and K^{qp} and K^{pq} are the transfer constant respectively from the compartment p to the compartment q and from q to p .

In order to determine tissues parameters from concentration-time profiles, it is necessary to solve an inverse problem. So, to validate this technique, a resolution of the direct problem have to be done first to check the calculated parameters from a concentration-time curve, Fig. 1.

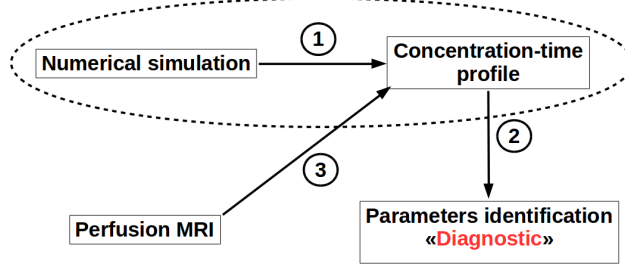


Figure 1: **Aim of this study.** 1: Direct problem, 2: Inverse problem, 3: real data.

2 Weak form

Lemma 1.

$$\int_{\Omega} \nabla \cdot \mathbf{u}q = - \int_{\Omega} \mathbf{u} \cdot \nabla q + \int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n}q \quad (2)$$

Proof. In 2 dimensions, the first integral reads:

$$\int_{\Omega} \frac{\partial u_1}{\partial x} q + \frac{\partial u_2}{\partial y} q \quad (3)$$

Using an integration by parts, it holds:

$$- \int_{\Omega} u_1 \frac{\partial q}{\partial x} + u_2 \frac{\partial q}{\partial y} + \int_{\partial\Omega} u_1 n_x q + u_2 n_y q = - \int_{\Omega} \mathbf{u} \cdot \nabla q + \int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n}q \quad (4)$$

□

Let Ω be a domain of \mathbb{R}^2 bounded by $\partial\Omega$. The weak form is obtained by integrating Eq. (1) on Ω using a the test function $\varphi \in H_0^1(\Omega)$.

$$\int_{\Omega} \frac{\partial C^q}{\partial t} \varphi = - \int_{\Omega} \nabla \cdot (\mathbf{u}^q C^q) \varphi + \int_{\Omega} \nabla \cdot (D^q \nabla C^q) \varphi - \int_{\Omega} K^q C^q \varphi + \sum_{p \neq q} \int_{\Omega} (K^{qp} C^p - K^{pq} C^q) \varphi \quad (5)$$

Using the integration by parts formula define in Lemma (1), it holds:

$$\int_{\Omega} \frac{\partial C^q}{\partial t} \varphi - \int_{\Omega} C^q \mathbf{u}^q \cdot \nabla \varphi + \int_{\Omega} D^q \nabla C^q \cdot \nabla \varphi + \int_{\Omega} K^q C^q \varphi - \sum_{p \neq q} \int_{\Omega} (K^{qp} C^p - K^{pq} C^q) \varphi + \int_{\partial\Omega} (C^q \mathbf{u} - D^q \nabla C^q) \cdot \mathbf{n} \varphi = 0 \quad (6)$$

As we the weak form is defined in $H_0^1(\Omega)$, we get:

$$\int_{\Omega} \frac{\partial C^q}{\partial t} \varphi - \int_{\Omega} C^q \mathbf{u}^q \cdot \nabla \varphi + \int_{\Omega} D^q \nabla C^q \cdot \nabla \varphi + \int_{\Omega} K^q C^q \varphi - \sum_{p \neq q} \int_{\Omega} (K^{qp} C^p - K^{pq} C^q) \varphi = 0 \quad (7)$$

2.1 Time discretization

A finite difference is used to handle with the time derivative:

$$\frac{\partial A}{\partial t} \approx \frac{A^{n+1} - A^n}{\Delta T} \quad (8)$$

With ΔT the time step, $A^n = A(n\Delta t)$ and $T = N\Delta t$.

The weak form is given (in $H^1(\Omega)$) by:

$$\int_{\Omega} \frac{C_{n+1}^q - C_n^q}{\Delta t} \varphi - \int_{\Omega} C_{n+1}^q \mathbf{u}_{n+1}^q \cdot \nabla \varphi + \int_{\Omega} D_{n+1}^q \nabla C_{n+1}^q \cdot \nabla \varphi + \int_{\Omega} K^q C_{n+1}^q \varphi - \sum_{p \neq q} \int_{\Omega} (K^{qp} C_{n+1}^p - K^{pq} C_{n+1}^q) \varphi + \int_{\partial\Omega} (C_{n+1}^q \mathbf{u} - D_{n+1}^q \nabla C_{n+1}^q) \cdot \mathbf{n} \varphi = 0 \quad (9)$$

In $H_0^1(\Omega)$, the discretized weak form reads:

$$\int_{\Omega} \frac{C_{n+1}^q - C_n^q}{\Delta t} \varphi - \int_{\Omega} C_{n+1}^q \mathbf{u}_{n+1}^q \cdot \nabla \varphi + \int_{\Omega} D_{n+1}^q \nabla C_{n+1}^q \cdot \nabla \varphi + \int_{\Omega} K^q C_{n+1}^q \varphi - \sum_{p \neq q} \int_{\Omega} (K^{qp} C_{n+1}^p - K^{pq} C_{n+1}^q) \varphi = 0 \quad (10)$$

2.2 Reformulation of the problem

In order to numerical solve the equation Eq.(1), we have to correctly define the initial condition and the boundary conditions.

Let Ω be a domain of \mathbb{R}^2 bounded by $\partial\Omega = \Gamma_D \cup \Gamma_R$. The equation Eq.(1) explicitly written with its initial and boundary conditions reads:

$$\begin{aligned} \frac{\partial C^q}{\partial t} + \nabla \cdot \mathbf{u}^q C^q - \nabla \cdot D^q \nabla C^q + K^q C^q &= \sum_{p \neq q} K^{qp} C^p - \sum_{p \neq q} K^{pq} C^q & [0, T] \times \Omega & \quad (11a) \end{aligned}$$

$$C^q = h^q \quad [0, T] \times \Gamma_D \quad (11b)$$

$$(C_{n+1}^q \mathbf{u} - D_{n+1}^q \nabla C_{n+1}^q) \cdot \mathbf{n} = g^q \quad [0, T] \times \Gamma_R \quad (11c)$$

$$C^q = C_0^q \quad \{0\} \times \Omega \quad (11d)$$

Where Γ_D is the boundary associated to the Dirichlet condition Eq. (11b) and Γ_R is the boundary associated to the Robin condition Eq. (11c). Eq. (11d) is the initial condition.

3 Numerical simulation

3.1 Numerical scheme

In order to solve numerically the equation Eq. (11), we design an explicit coupling between the different compartments $q \in \{1, \dots, Q\}$. The scheme writes:

Find C_{n+1}^q such as $\forall \varphi$:

$$\int_{\Omega} \frac{C_{n+1}^q - C_n^q}{\Delta t} \varphi - \int_{\Omega} C_{n+1}^q \mathbf{u}_{n+1}^q \cdot \nabla \varphi + \int_{\Omega} D_{n+1}^q \nabla C_{n+1}^q \cdot \nabla \varphi + \int_{\Omega} K_{n+1}^q C_{n+1}^q \varphi - \int_{\Omega} \sum_{p \neq q} K_n^{qp} C_n^p \varphi + \int_{\Omega} \sum_{p \neq q} K_n^{pq} C_n^q \varphi = 0 \quad (12)$$

We impose a Dirichlet condition equal to zero for the wall and an input function C_{in} on the lower left corner.

The partial differential equations of all compartments are solve in the same time in a monolithic problem.

In practice [2], only the neighbors pixels have taken into account in the transfer of concentration. For each pixel, only four neighbors pixels are considered in 2 dimensions, Fig.2.

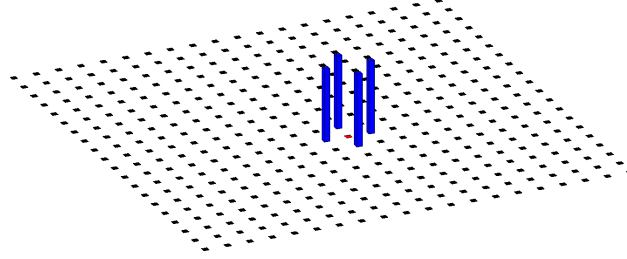


Figure 2: **Numerical scheme.** Illustration of the neighbors pixels. The current pixel is in red and its neighbors in blue, other pixels are in black.

The problem is solved using FREEFEM++ [1]. Algorithms are available in Appendix A.

3.2 Numerical stability

3.2.1 Without compartment

Theorem 1. *The scheme (12), without compartment, is unconditionally stable.*

Proof. Firstly, we consider the weak form in $H_0^1(\Omega)$, Eq. (12), without compartment. In order to prove the stability (in the sense of energy), we use C_{n+1}^q as the test function ϕ .

$$\int_{\Omega} \frac{C_{n+1} - C_n}{\Delta t} C_{n+1} - \int_{\Omega} C_{n+1} \mathbf{u} \cdot \nabla C_{n+1} + \int_{\Omega} D \nabla C_{n+1} \cdot \nabla C_{n+1} + \int_{\Omega} K C_{n+1} C_{n+1} = 0 \quad (13)$$

Using the relation $(C_{n+1} - C_n)C_{n+1} = \frac{1}{2}(|C_{n+1}|^2 - |C_n|^2 + |C_{n+1} - C_n|^2)$, we get:

$$\begin{aligned} \int_{\Omega} \frac{1}{2} |C_{n+1}|^2 - \int_{\Omega} \frac{1}{2} |C_n|^2 + \int_{\Omega} \frac{1}{2} |C_{n+1} - C_n|^2 - \int_{\Omega} \Delta t C_{n+1} \mathbf{u} \cdot \nabla C_{n+1} \\ + \int_{\Omega} \Delta t D |\nabla C_{n+1}|^2 + \int_{\Omega} \Delta t K |C_{n+1}|^2 = 0 \end{aligned} \quad (14)$$

Making simplifications and use the fact that $\frac{1}{2} |C_{n+1} - C_n|^2 \geq 0$, it holds:

$$\begin{aligned} \int_{\Omega} \left(\frac{1}{2} + \Delta t K \right) |C_{n+1}|^2 - \int_{\Omega} \Delta t C_{n+1} \mathbf{u} \cdot \nabla C_{n+1} + \int_{\Omega} \Delta t D |\nabla C_{n+1}|^2 \\ \leq \int_{\Omega} \frac{1}{2} |C_n|^2 \end{aligned} \quad (15)$$

Using the Lemma (1), we get in $H_0^1(\Omega)$:

$$\begin{aligned} \int_{\Omega} \left(\frac{1}{2} + \Delta t K \right) |C_{n+1}|^2 + \int_{\Omega} \Delta t \nabla \cdot \mathbf{u} |C_{n+1}|^2 + \int_{\Omega} \Delta t D |\nabla C_{n+1}|^2 \\ \leq \int_{\Omega} \frac{1}{2} |C_n|^2 \end{aligned} \quad (16)$$

As we consider an incompressible flow, $\nabla \cdot \mathbf{u} = 0, \forall q$, we get:

$$\int_{\Omega} \left(\frac{1}{2} + \Delta t K \right) |C_{n+1}|^2 + \int_{\Omega} \Delta t D |\nabla C_{n+1}|^2 \leq \int_{\Omega} \frac{1}{2} |C_n|^2 \quad (17)$$

As $D \geq 0, \forall q$, it holds:

$$\int_{\Omega} \left(\frac{1}{2} + \Delta t K \right) |C_{n+1}|^2 \leq \int_{\Omega} \frac{1}{2} |C_n|^2 \quad (18)$$

The result holds with a discrete Gronwall lemma. \square

3.2.2 With compartments

Theorem 2. *The scheme (12), with compartments, is conditionnaly stable under the Courant-Friedrichs-Lewy (CFL) condition:*

$$\Delta t \leq \frac{1}{2 \min_q \sum_{p=1}^Q K^{pq}} \quad (19)$$

Proof. We now consider the weak form in $H_0^1(\Omega)$, Eq. (12), with compartments.

$$\begin{aligned} \int_{\Omega} \frac{C_{n+1}^q - C_n^q}{\Delta t} C_{n+1}^q - \int_{\Omega} C_{n+1}^q \mathbf{u}^q \cdot \nabla C_{n+1}^q + \int_{\Omega} D^q \nabla C_{n+1}^q \cdot \nabla C_{n+1}^q \\ + \int_{\Omega} K^q C_{n+1}^q C_{n+1}^q - \sum_{p \neq q} \int_{\Omega} (K^{qp} C_n^p - K^{pq} C_n^q) C_n^q = 0 \end{aligned} \quad (20)$$

Using the same techniques as the numerical stability without compartments, it easily holds:

$$\int_{\Omega} \left(\frac{1}{2} + \Delta t K^q \right) |C_{n+1}^q|^2 - \int_{\Omega} \Delta t \sum_{p \neq q} (K^{qp} C_n^p - K^{pq} C_n^q) C_n^q \leq \int_{\Omega} \frac{1}{2} |C_n^q|^2 \quad (21)$$

Making simplifications, we get:

$$\int_{\Omega} \left(\frac{1}{2} + \Delta t K^q \right) |C_{n+1}^q|^2 \leq \int_{\Omega} \left(\frac{1}{2} - \Delta t \sum_{p \neq q} K^{pq} \right) |C_n^q|^2 + \int_{\Omega} \Delta t \sum_{p \neq q} K^{qp} C_n^p C_n^q \quad (22)$$

Using the relation $\sum_{p \neq q} K^{qp} C_n^p C_n^q = \sum_p K^{qp} C_n^p C_n^q - K^{qq} |C_n^q|^2$, we get:

$$\int_{\Omega} \left(\frac{1}{2} + \Delta t K^q \right) |C_{n+1}^q|^2 \leq \int_{\Omega} \left(\frac{1}{2} - \Delta t \sum_{p=1}^Q K^{pq} \right) |C_n^q|^2 + \int_{\Omega} \Delta t \sum_{p=1}^Q K^{qp} C_n^p C_n^q \quad (23)$$

The result holds with a discrete Gronwall lemma if:

$$\frac{1}{2} - \Delta t \sum_{p=1}^Q K^{pq} \geq 0 \Leftrightarrow \Delta t \leq \frac{1}{2 \sum_{p=1}^Q K^{pq}}, \forall q \in \{1, \dots, Q\} \quad (24)$$

That leads to:

$$\Delta t \leq \frac{1}{2 \min_q \sum_{p=1}^Q K^{pq}} \quad (25)$$

□

The stability is then proved under some conditions:

- $\nabla \cdot \mathbf{u}^q = 0, \forall q \in \{1, \dots, Q\}$
- Respect of the CFL condition

Numerical simulations have to respect these conditions.

4 Results

4.1 No compartments

Tissues parameters from [4] are taken into account for this numerical simulation.

The field of view is a square with a side of $L = 40$ cm. The pixel size is fixed at $2 \text{ cm} \times 2 \text{ cm}$.

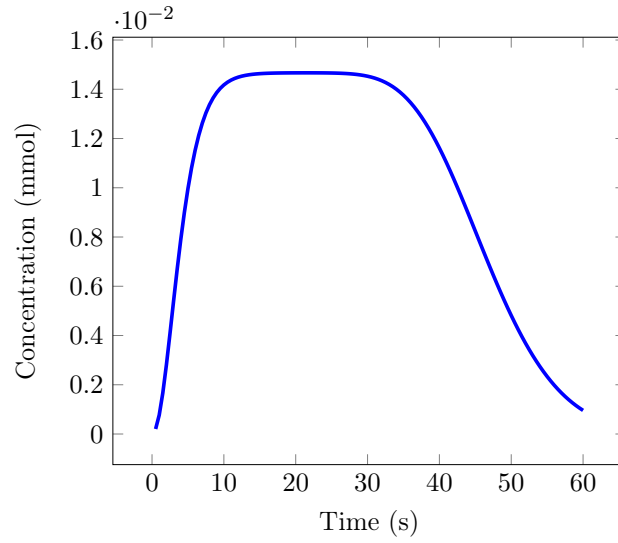
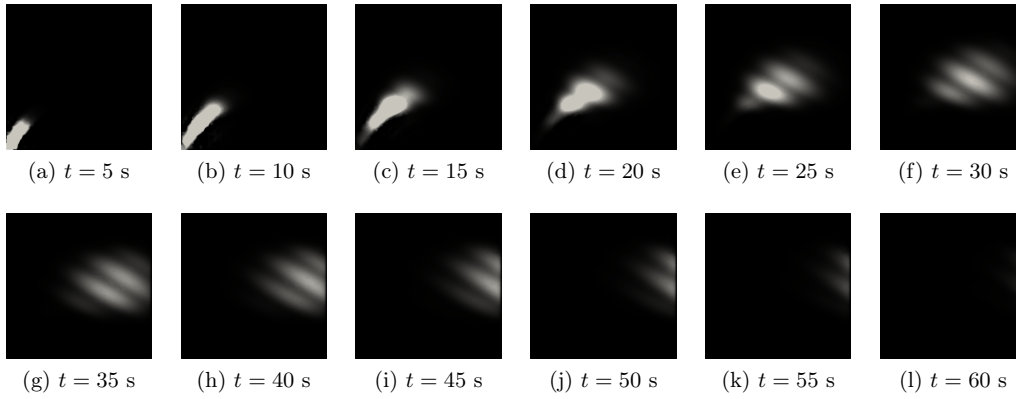
We have:

- $K = 0$
- $D = \exp \left(- \left(\frac{0.4x}{L-0.5} \right)^2 - \left(\frac{0.2y}{L-0.5} \right)^2 \right)$.
- $v = 0.7 + 0.5 \sin \left(\frac{2\pi(3x+5y)}{L} \right)$
- $F = \left(\frac{x}{L} - \frac{x^2}{L^2} + 0.5, \frac{y}{L-1} \frac{x}{L-1} \right)^t$
- $u = \frac{F}{v} + \frac{D}{v^2} \nabla v$
- $C_{in} = t^2 \exp \left(\frac{-t}{1.5} \right)$

Simulation is performed during a total time $T = 60$ s with a time step $\Delta t = 0.5$ s. Calculation time is 57 s with one CPU (Intel Core i5 3.10 GHz).

The mean concentration during the simulation time is shown fig. 3.

The mean concentration increases during the injection before to stabilize it. The concentration decrease is due to the concentration exit of the field of view. To illustrate this phenomenon, concentration at different time steps is shown Fig. 4.

Figure 3: **No compartment.** Mean concentration during time.Figure 4: **No compartment.** Concentration at different time steps.

4.2 Two compartments, same tissues

The tissues parameters are identically defined in the two compartments. The transfer constants are define as $K_1^{1,2} = K_1^{2,1} = K_2^{1,2} = K_2^{2,1} = 0.01$.

Identical transfer constants are taken to observe an equal redistribution of the concentration in the two compartments. Input function C_{in} is divided in 1/3, 2/3 for the first and the second compartment respectively.

Simulation is performed during a total time $T = 60$ s with a time step $\Delta t = 0.5$ s. Calculation time is 133 s with one CPU (one core of an Intel Core i5 3.10 GHz).

The mean concentration during the simulation time is shown fig. 5 for the global concentration and for each compartment.

As the transfer constants are equal, the concentration in the compartment 2 tends to the concentration in the compartment 1.

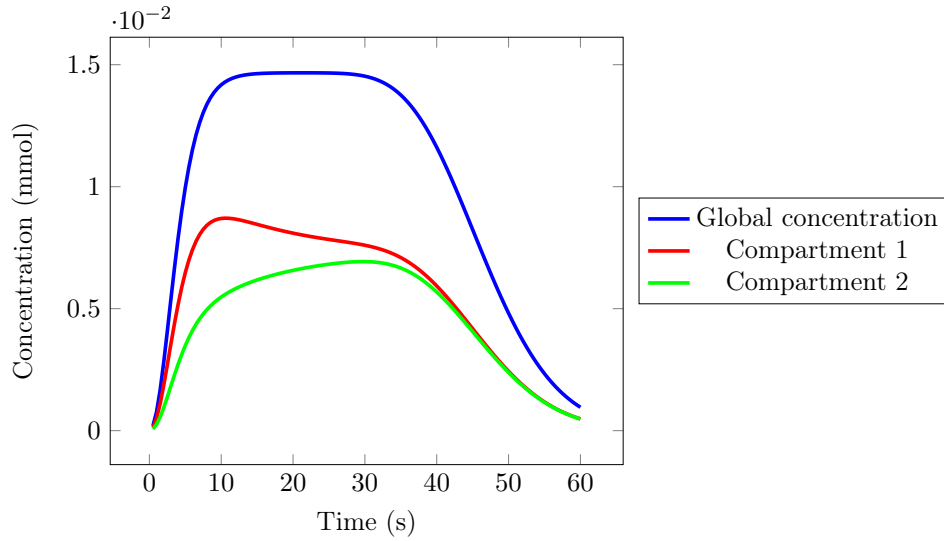


Figure 5: **Two compartments, same tissues.** Mean concentration during time

The global concentration function of time is the same as the first experience as we do not have modified any tissues parameters.

4.3 Two compartments, different transfer constants

We now define the transfer constants as $K_1^{1,2} = K_2^{1,2} = 0.01$ and $K_1^{2,1} = K_2^{2,1} = 0.1$. Other parameters are identical to the previous simulation.

The mean concentration during the simulation time is shown fig. 6 for the global concentration and for each compartment.

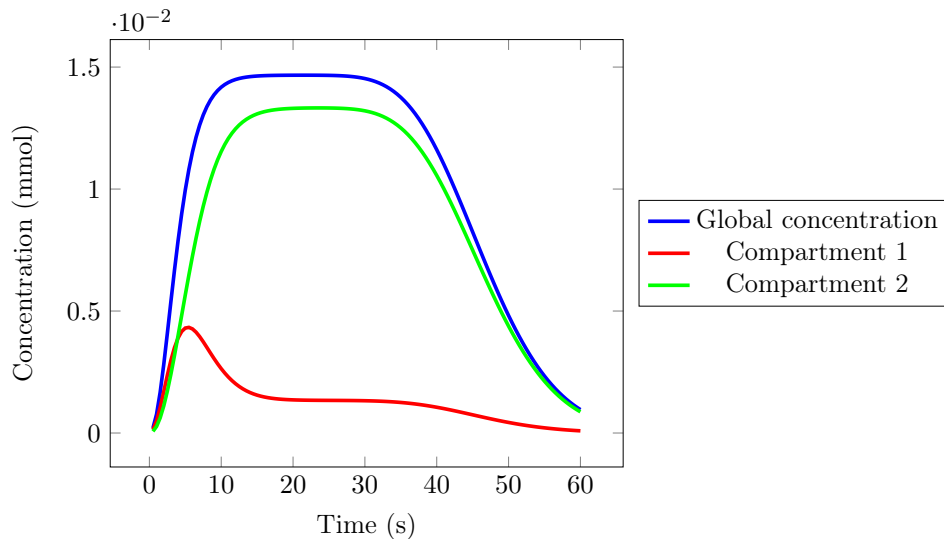


Figure 6: **Two compartments, different transfer constants.** Mean concentration during time

As the transfer constant that takes concentration to the other compartment in the first compartment is smaller than in the second ($K_1^{1,2} < K_1^{2,1}$) and the transfer constant that gives concentration to the other compartment in the first compartment is higher than in the second ($K_2^{1,2} < K_1^{2,1}$), concentration in the first compartment is absorbed by the second.

The global concentration function of time is the same as the first experience as we do not have modified any tissues parameters.

4.4 Two compartments, different tissues

In this simulation, parameters of the first two compartments simulation are taken and we change the velocity parameters of the two compartments as defined here:

- $u_1 = (0.3, 0)'$
- $u_2 = (-0.6, -0.6)'$

All other parameters are the same as previous simulations. The input function is taken here equal to zero and the initial concentration is defined by the following equation:

$$C_0^1 = 2(19.5 \leq x \leq 22.5)(19.5 \leq y \leq 22.5) \quad (26a)$$

$$C_0^2 = 0 \quad (26b)$$

The mean concentration during the simulation time is shown fig. 7 for the global concentration and for each compartment.

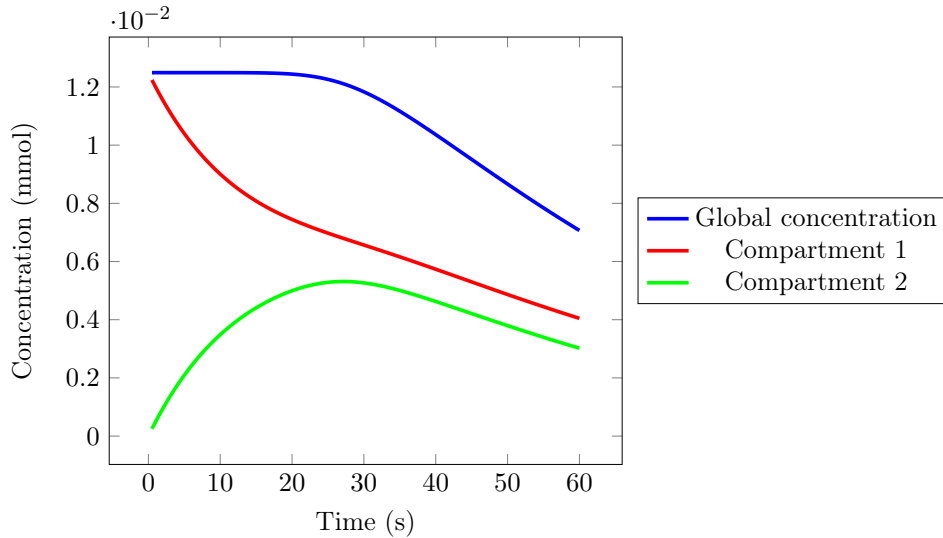


Figure 7: **Two compartments, different tissues.** Mean concentration during time

The concentration begin at his maximum here because there is not source term. Compartment 2 concentration increase due to transfer constants and the global concentration decrease with the concentration output of the field of view.

Concentration at different time steps is illustrate Fig. 8 for the global concentration and for each compartment.

Global concentration show a division of the concentration from the center to the left bottom and the right upper corners. We can see with the concentration in the first and the

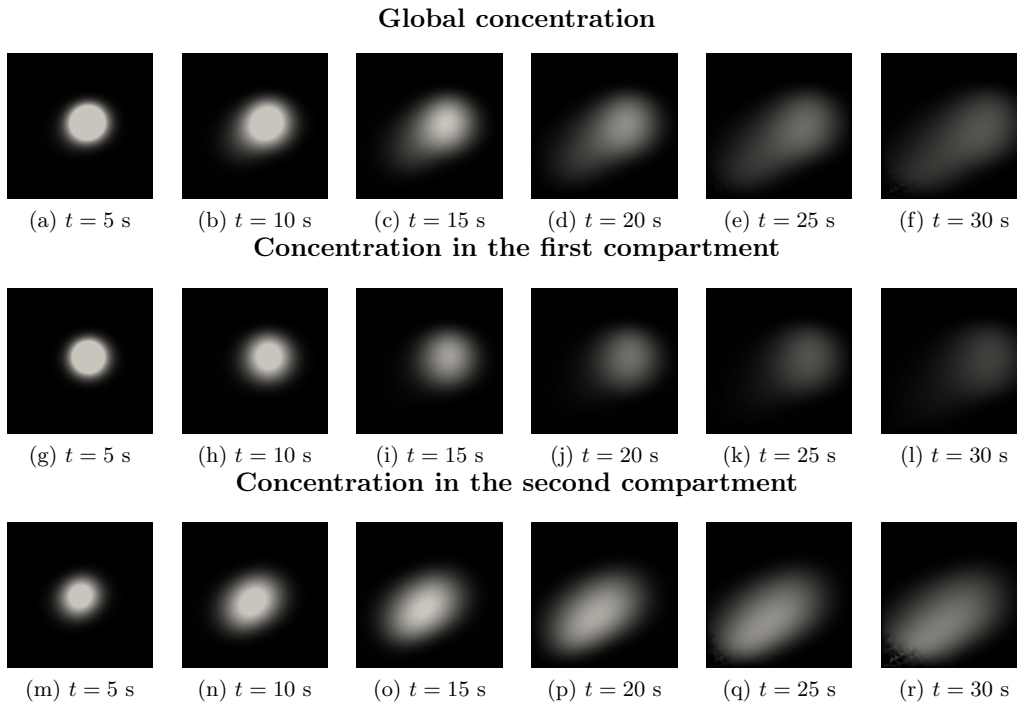


Figure 8: **Two compartments, different tissues.** Concentration at different time steps. *Gray-scale coloring is not in the same range for the different compartments.*

second compartment the role of each compartment in the separation, following the velocity parameter u .

5 Conclusion and perspectives

A phantom was designed using fixed parameters. Sourbron's model equation was discretized in time and an explicit numerical scheme was realized to solve the problem with `FREEFEM++`.

Simulations have been performed in some test cases and similar results are observed in [4].

An overview of the problem have been realized. A complete analysis should be done existence, uniqueness of solution and a correct choose of the functional space.

References

- [1] Frédéric Hecht. New development in FreeFem++. *Journal of Numerical Mathematics*, 20(3-4):251–266, 2012.
- [2] Martin Pellerin, Thomas E Yankeelov, and Martin Lepage. Incorporating contrast agent diffusion into the analysis of DCE-MRI data. *Magnetic resonance in medicine*, 58(6):1124–1134, 2007.
- [3] Steven Sourbron. A tracer-kinetic field theory for medical imaging. *IEEE transactions on medical imaging*, 33(4):935–946, 2014.

- [4] Steven Sourbron. Reconstructing the one-compartment tracer-kinetic field with diffusion and convection. In *ISMRM proceedings*, 2015.

A FREEFEM++algorithms

A.1 No compartment

```

load "iovtk"

//Parameters
real TotalT = 60.;
real DeltaT = 0.5;
real t = 0.;

//Mesh
real nn = 1;
real L = 40.;
int NPixels = 20;
mesh Th;
{
    real LPixel = L/NPixels;
    border b1(t=0., 1.) {x=LPixel*t; y=0.; label=1;};
    border b2(t=0., 1.) {x=LPixel + (L-LPixel)*t; y=0.; label=0;};
    border b3(t=0., 1.) {x=L; y=L*t; label=0;};
    border b4(t=0., 1.) {x=L-L*t; y=L; label=0;};
    border b5(t=0., 1.) {x=0.; y=L - (L-LPixel)*t; label=0;};
    border b6(t=0., 1.) {x=0.; y=LPixel - LPixel*t; label=1;};

    real nnL = max(2., L*nn);
    real nnLPixel = max(2., LPixel*nn);
    real nnLLPixel = max(2., nnL - nnLPixel);

    Th = buildmesh(b1(nnLPixel) + b2(nnLLPixel) + b3(nnL) + b4(nnL)
        + b5(nnLLPixel) + b6(nnLPixel));
}

//FESpace
fespace Ch(Th, P2);
Ch Cn1, Cn;
Ch d;
Ch v;

//Problem variables
macro div(a1, a2)          (dx(a1) + dy(a2))          //
macro grad(a)              [dx(a), dy(a)]              //

//K
func K = 0.;

//D
func D = exp(-((x/L-0.5)/0.4)^2 - ((y/L-0.5)/0.2)^2);

//Volume
func V = (0.7+0.2*sin(2*pi*(3*x+5*y)/L));
v = V;

//Flow
func Fx = (x/L - (x*x)/(L*L) + 0.5);
func Fy = ((y/L-1)*(x/L-1));

//Velocity
func U1 = ((Fx / v) + (D / (v*v)) * dx(v));
func U2 = ((Fy / v) + (D / (v*v)) * dy(v));

//Source
func Source = (t)^2 * exp(-t/1.5);

//Problem

```

```

problem Sourbron(Cn1, d, solver=sparsesolver)
= int2d(Th) (
    (1./DeltaT) * (1.) * Cn1 * d
)
+ int2d(Th) (
    (1./DeltaT) * (-1.) * Cn * d
)
- int2d(Th) (
    Cn1 * [U1, U2]' * grad(d)
)
+ int2d(Th) (
    K * Cn1 * d
)
+ int2d(Th) (
    D * grad(Cn1)' * grad(d)
)
+ on(1, Cn1=Source)
+ on(0, Cn1=0.)
;

//Time iterations
Cn = 0.;
Cn1 = 0;
for (int i = 0; i < TotalT/DeltaT; i++){
    //Update
    t += DeltaT;

    v = V;
    Cn = Cn1;

    //Solve
    Sourbron;

    //Plot
    plot(Cn1, cmm="Concentration, t = "+t, fill=true, value=true);

    //Calculate
    real TotalC = int2d(Th) (Cn1) / int2d(Th) (1.);
    real MinC = Cn1[].min;
    real MaxC = Cn1[].max;

    //Save
    int[int] fforder = [1];
    savevtk("vtk/result_0compartment_"+i+".vtu", Th, Cn1,
        dataname="Concentration", order=fforder);

    if (i == 0){
        ofstream file("concentration_0compartment.dat");
        file << t << "\t" << TotalC << "\t" << MinC << "\t" << MaxC << endl;
    }
    else{
        ofstream file("concentration_0compartment.dat", append);
        file << t << "\t" << TotalC << "\t" << MinC << "\t" << MaxC << endl;
    }
}

```

A.2 Two compartments

```

load "iovtk"

//Parameters
real TotalT = 60.;
real DeltaT = 0.5;
real t = 0.;

//Mesh
real nn = 1;

```



```

real L = 40.;
int NPixels = 20;
real LPixel = L/NPixels;
mesh Th;
{
    real LPixel = L/NPixels;
    border b1(t=0., 1.) {x=L*Pixel*t; y=0.; label=1;};
    border b2(t=0., 1.) {x=L*Pixel + (L-L*Pixel)*t; y=0.; label=0;};
    border b3(t=0., 1.) {x=L; y=L*t; label=0;};
    border b4(t=0., 1.) {x=L-L*t; y=L; label=0;};
    border b5(t=0., 1.) {x=0.; y=L - (L-L*Pixel)*t; label=0;};
    border b6(t=0., 1.) {x=0.; y=L*Pixel - L*Pixel*t; label=1;};

    real nnL = max(2., L*nn);
    real nnLPixel = max(2., L*Pixel*nn);
    real nnLLPixel = max(2., nnL - nnLPixel);

    Th = buildmesh(b1(nnLPixel) + b2(nnLLPixel) + b3(nnL) + b4(nnL)
        + b5(nnLLPixel) + b6(nnLPixel));
}

//FESpace
fespace Ch(Th, P2);
Ch Cn1q, Cnq;
Ch Cn1p, Cnp;
Ch dq, dp;
Ch vq, vp;

//Problem variables
macro div(a1, a2)          (dx(a1) + dy(a2))          //
macro grad(a)              [dx(a), dy(a)]              //

//K
func Kq = 0.;
func Kp = 0.;

//D
func Dq = exp(-((x/L-0.5)/0.4)^2 - ((y/L-0.5)/0.2)^2);
func Dp = exp(-((x/L-0.5)/0.4)^2 - ((y/L-0.5)/0.2)^2);

//Volume
func Vq = (0.7+0.2*sin(2*pi*(3*x+5*y)/L));
func Vp = (0.7+0.2*sin(2*pi*(3*x+5*y)/L));
vq = Vq;
vp = Vp;

//Flow
func Fxq = (x/L - (x*x)/(L*L) + 0.5);
func Fyq = ((y/L-1)*(x/L-1));
func Fxp = (x/L - (x*x)/(L*L) + 0.5);
func Fyp = ((y/L-1)*(x/L-1));

//Velocity
func U1q = ((Fxq / vq) + (Dq / (vq*vq)) * dx(vq));
func U2q = ((Fyq / vq) + (Dq / (vq*vq)) * dy(vq));
func U1p = ((Fxp / vp) + (Dp / (vp*vp)) * dx(vp));
func U2p = ((Fyp / vp) + (Dp / (vp*vp)) * dy(vp));

//Kqp
real[int] KqpValue = [0.01, 0.01];
func Kqpq = KqpValue[0] * (
    (x < (NPixels-1)*LPixel) * Cnp(x+LPixel, y)
    + (y < (NPixels-1)*LPixel) * Cnp(x, y+LPixel)
    + (x > L*Pixel) * Cnp(x-L*Pixel, y)
    + (y > L*Pixel) * Cnp(x, y-L*Pixel)
);

func Kqpp = KqpValue[1] * (

```

```

    (x < (NPixels-1)*LPixel) * Cnq(x+LPixel, y)
  + (y < (NPixels-1)*LPixel) * Cnq(x, y+LPixel)
  + (x > LPixel) * Cnq(x-LPixel, y)
  + (y > LPixel) * Cnq(x, y-LPixel)
);

//Kpq
real[int] KpqValue = [0.01, 0.01];
func Kpqq = KpqValue[0] * (
  (x < (NPixels-1)*LPixel) * Cnq(x+LPixel, y)
  + (y < (NPixels-1)*LPixel) * Cnq(x, y+LPixel)
  + (x > LPixel) * Cnq(x-LPixel, y)
  + (y > LPixel) * Cnq(x, y-LPixel)
);

func Kppp = KpqValue[1] * (
  (x < (NPixels-1)*LPixel) * Cnp(x+LPixel, y)
  + (y < (NPixels-1)*LPixel) * Cnp(x, y+LPixel)
  + (x > LPixel) * Cnp(x-LPixel, y)
  + (y > LPixel) * Cnp(x, y-LPixel)
);

//Source
func Source = (t)^2 * exp(-t/1.5);
func Sourceq = (2./3.)*Source;
func Sourcep = (1./3.)*Source;

//Problem
problem Sourbron([Cnlq, Cnlp], [dq, dp], solver=sparsesolver)
= int2d(Th) (
  (1./DeltaT) * (1.) * [Cnlq, Cnlp]' * [dq, dp]
)
+ int2d(Th) (
  (1./DeltaT) * (-1.) * [Cnq, Cnp]' * [dq, dp]
)
- int2d(Th) (
  Cnlq * [U1q, U2q]' * grad(dq)
  + Cnlp * [U1p, U2p]' * grad(dp)
)
+ int2d(Th) (
  Kq * Cnlq * dq
  + Kp * Cnlp * dp
)
+ int2d(Th) (
  Dq * grad(Cnlq)' * grad(dq)
  + Dp * grad(Cnlp)' * grad(dp)
)
- int2d(Th) (
  Kpqq * dq
  + Kppp * dp
)
+ int2d(Th) (
  Kpqq * dq
  + Kppp * dp
)
+ on(1, Cnlq=Sourceq)
+ on(1, Cnlp=Sourcep)
+ on(0, Cnlq=0.)
+ on(0, Cnlp=0.)
;

//Iterations
Cnq = 0.;
Cnlq = 0;
Cnp = 0.;
Cnlp = 0;
for (int i = 0; i < TotalT/DeltaT; i++){
  //Update

```

```

t += DeltaT;

vq = Vq;
vp = Vp;

Cnq = Cn1q;
Cnp = Cn1p;

//Solve
Sourbron;
Ch Cn1Total = Cn1q + Cn1p;

//Plot
plot(Cn1Total, cmm="Concentration, t = "+t, fill=true, value=true);

//Calculate
real TotalC = int2d(Th)(Cn1Total) / int2d(Th)(1.);
real MinC = Cn1Total[].min;
real MaxC = Cn1Total[].max;

real TotalCq = int2d(Th)(Cn1q) / int2d(Th)(1.);
real MinCq = Cn1q[].min;
real MaxCq = Cn1q[].max;

real TotalCp = int2d(Th)(Cn1p) / int2d(Th)(1.);
real MinCp = Cn1p[].min;
real MaxCp = Cn1p[].max;

//Save
int[int] fforder = [1, 1, 1];
savevtk("vtk/result_2compartments_"+i+".vtu", Th, Cn1Total, Cn1q, Cn1p,
        dataname="Concentration Compartment0 Compartment1", order=fforder);

if (i == 0){
    ofstream file("concentration_2compartments.dat");
    file << t << "\t" << TotalC << "\t" << MinC << "\t" << MaxC
        << "\t" << TotalCq << "\t" << MinCq << "\t" << MaxCq
        << "\t" << TotalCp << "\t" << MinCp << "\t" << MaxCp
        << endl;
}
else{
    ofstream file("concentration_2compartments.dat", append);
    file << t << "\t" << TotalC << "\t" << MinC << "\t" << MaxC
        << "\t" << TotalCq << "\t" << MinCq << "\t" << MaxCq
        << "\t" << TotalCp << "\t" << MinCp << "\t" << MaxCp
        << endl;
}
}
}

```


Bibliographie

- [1] J. Ahrens, B. Geveci, and C. Law. Paraview: An end-user tool for large-data visualization. *The Visualization Handbook*, page 717, 2005.
- [2] M.J. Albeck, S.E. Børgesen, F. Gjerris, J.F. Schmidt, and P.S. Sørensen. Intracranial pressure and cerebrospinal fluid outflow conductance in healthy subjects. *Journal of Neurosurgery*, 74(4):597–600, 1991.
- [3] N. Alperin, M. Mazda, T. Lichtor, and S.H. Lee. From cerebrospinal fluid pulsation to noninvasive intracranial compliance and pressure measured by MRI flow studies. *Current Medical Imaging Reviews*, 2:117–129, 2006.
- [4] N.J. Alperin, S.H. Lee, F. Loth, P.B. Raksin, and T. Lichtor. MR-intracranial pressure (ICP): A method to measure intracranial elastance and pressure noninvasively by means of MR imaging: Baboon and human study 1. *Radiology*, 217(3):877–885, 2000.
- [5] A. Amery, H. Wasir, C. Bulpitt, J. Conway, R. Fagard, P. Lijnen, and T. Reybrouck. Aging and the cardiovascular system. *Acta Cardiologica*, 33(6):443–467, 1977.
- [6] ANR. VIVABRAIN Project. Accessed: 2014-05-14.
- [7] I. Babuška. The finite element method with lagrangian multipliers. *Numerische Mathematik*, 20(3):179–192, 1973.
- [8] O. Balédent. *Imaging of the cerebrospinal*, volume 256. Cambridge University Press, 2014.
- [9] O. Balédent, C. Gondry-Jouet, M.E. Meyer, G. De Marco, D. Le Gars, M.C. Henry-Feugeas, and I. Idy-Peretti. Relationship between cerebrospinal fluid and blood dynamics in healthy volunteers and patients with communicating hydrocephalus. *Investigative Radiology*, 39(1):45–55, 2004.
- [10] O. Balédent, M.C. Henry-Feugeas, and I. Idy-Peretti. Cerebrospinal fluid dynamics and relation with blood flow: A magnetic resonance study with semiautomated cerebrospinal fluid segmentation. *Investigative Radiology*, 36(7):368–377, 2001.
- [11] G.A. Bateman. Vascular compliance in normal pressure hydrocephalus. *American Journal of Neuroradiology*, 21(9):1574–1585, 2000.
- [12] J.P. Benque, B. Ibler, and G. Labadie. A finite element method for Navier-Stokes equations. In *Numerical Methods for Non-Linear Problems*, volume 1, pages 709–720, 1980.

- [13] R. Bouzerar, M. Czosnyka, Z. Czosnyka, and O. Balédent. *Physical phantom of craniospinal hydrodynamics*. Springer, 2012.
- [14] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 8(2):129–151, 1974.
- [15] P.C. Buijs, M.J. Krabbe-Hartkamp, C.J. Bakker, E.E. De Lange, L.M. Ramos, M.M. Breteler, and W.P. Mali. Effect of age on cerebral blood flow: Measurement with ungated two-dimensional phase-contrast MR angiography in 250 adults. *Radiology*, 209(3):667–674, 1998.
- [16] J. Cahouet and J.P. Chabard. Some fast 3D finite element solvers for the generalized Stokes problem. *Internatioanl Journal for Numerical Methods in Fluids*, 8:869–895, 1988.
- [17] E.R. Cardoso, J.O. Rowan, and S. Galbraith. Analysis of the cerebrospinal fluid pulse wave in intracranial pressure. *Journal of Neurosurgery*, 59(5):817–821, 1983.
- [18] B. Chaarani, J. Zmudka, C. Capel, C. Gondry-Jouet, R. Bouzerar, and O. Balédent. Csf and brain tissue repartition in aging brain disorders. In *Hydrocephalus*, 2013.
- [19] V. Chabannes. *Vers la simulation numérique des écoulements sanguins*. PhD thesis, Université de Grenoble, 2013.
- [20] The C++ Standards Committee. C++. <http://www.open-std.org/JTC1/SC22/WG21/>.
- [21] The Qt Company. Qt. <http://www.qt.io/>.
- [22] M. Czosnyka and J.D. Pickard. Monitoring and interpretation of intracranial pressure. *Journal of Neurology, Neurosurgery & Psychiatry*, 75(6):813–821, 2004.
- [23] M.S. Daners, S. Bottan, L. Guzzella, D. Poulidakos, and V. Kurtcuoglu. Craniospinal pressure–volume dynamics in phantom models. *IEEE Transactions on Biomedical Engineering*, 59(12):3482–3490, 2012.
- [24] T. Dutta-Roy, A. Wittek, and K. Miller. Biomechanical modelling of normal pressure hydrocephalus. *Journal of Biomechanics*, 41(10):2263–2271, 2008.
- [25] E. Eichhammer. Qcutomplot, Qt plotting widget. <http://www.qcustomplot.com/>.
- [26] C.R. Ethier and D.A. Steinman. Exact fully 3D Navier-Stokes solution for benchmarking. *International Journal for Numerical Methods in Fluids*, 19(5):369–376, 1994.
- [27] J. Fouchet-Incaux. Artificial boundaries and formulations for the incompressible Navier–Stokes equations: Applications to air and blood flows. *SeMA Journal*, 64(1):1–40, 2014.
- [28] J. Fouchet-Incaux. *Modélisation, analyse numérique et simulations autour de la respiration*. PhD thesis, Paris 11, 2015.
- [29] O. Frank. Die grundform des arteriellen pulses. *Zeitschrift für Biologie*, 37(483-526):459, 1899.

- [30] P. Frey. Medit: An interactive mesh visualization software, 2001.
- [31] M. Frigo and S.G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005.
- [32] S. Garnotel. FourierFit. <https://github.com/sgarnotel/FourierFit>.
- [33] S. Garnotel. MRiCP. <https://github.com/sgarnotel/MRiCP>.
- [34] C. Geuzaine and J.F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.
- [35] A.E. Green and G.I. Taylor. Mechanism of the production of small eddies from larger ones. In *Proceedings of the Royal Society A*, volume 158, pages 499–521, 1937.
- [36] G. Green. *An essay on the application of mathematical analysis to the theories of electricity and magnetism*. Mayer&Müller, 1828.
- [37] J.L. Guermond and J. Shen. A new class of truly consistent splitting schemes for incompressible flows. *Journal of Computational Physics*, 192(1):262–276, 2003.
- [38] R.B. Hamilton, K. Baldwin, P. Vespa, M. Bergsneider, and X. Hu. Subpeak regional analysis of intracranial pressure waveform morphology based on cerebrospinal fluid hydrodynamics in the cerebral aqueduct and prepontine cistern. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 3935–3938. IEEE, 2012.
- [39] F. Hecht. New development in FreeFem++. *Journal of Numerical Mathematics*, 20(3-4):251–266, 2012.
- [40] D. Hibbitt, B. Karlsson, and P. Sorensen. *ABAQUS/standard: User's Manual (Version 6.2)*, volume 1. Pawtucket, RI: Hibbitt, Karlsson & Sorensen Inc, 2001.
- [41] F. Jędrzejewski. *Introduction aux méthodes numériques*. Springer Science & Business Media, 2006.
- [42] R.N. Kalaria. Cerebral vessels in ageing and Alzheimer's disease. *Pharmacology & Therapeutics*, 72(3):193–214, 1996.
- [43] G.E. Karniadakis, M. Israeli, and S.A. Orszag. High-order splitting methods for the incompressible navier-stokes equations. *Journal of computational physics*, 97(2):414–443, 1991.
- [44] W. Koch. Local instability characteristics and frequency determination of self-excited wake flows. *Journal of Sound and Vibration*, 99(1):53–83, 1985.
- [45] M. Kopelman, A. Glik, S. Greenberg, and I. Shelef. Intracranial nonjugular venous pathways: A possible compensatory drainage mechanism. *American Journal of Neuroradiology*, 34(7):1348–1352, 2013.
- [46] N. Lenfeldt, L.O.D. Koskinen, A.T. Bergenheim, J. Malm, and A. Eklund. CSF pressure

- assessed by lumbar puncture agrees with intracranial pressure. *Neurology*, 68(2):155–158, 2007.
- [47] A.A. Linninger, M. Xenos, D.C. Zhu, M.R. Somayaji, S. Kondapalli, and R.D. Penn. Cerebrospinal fluid flow in the normal and hydrocephalic human brain. *IEEE Transactions on Biomedical Engineering*, 54(2):291–302, 2007.
- [48] A. Lokossou, S. Garnotel, G. Pagé, L. Balardy, Z. Czosnyka, P. Payoux, E.A. Schmidt, and O. Balédent. ICP monitoring and phase contrast MRI to investigate intracranial compliance. In *16th International Symposium on Intracranial Pressure & Neuromonitoring Proceedings*, 2016.
- [49] A. Marmarou, K. Shulman, and J. Lamorgese. Compartmental analysis of compliance and outflow resistance of the cerebrospinal fluid system. *Journal of Neurosurgery*, 43(5):523–534, 1975.
- [50] A. Marmarou, K. Shulman, and R.M. Rosende. A nonlinear analysis of the cerebrospinal fluid system and intracranial pressure dynamics. *Journal of Neurosurgery*, 48(3):332–344, 1978.
- [51] M. Mase, T. Miyati, K. Yamada, H. Kasai, M. Hara, and Y. Shibamoto. Non-invasive measurement of intracranial compliance using cine MRI in normal pressure hydrocephalus. In *Intracranial Pressure and Brain Monitoring XII*, pages 303–306. Springer, 2005.
- [52] T. Matsumoto, H. Nagai, Y. Kasuga, and K. Kamiya. Changes in intracranial pressure (ICP) pulse wave following hydrocephalus. *Acta Neurochirurgica*, 82(1-2):50–56, 1986.
- [53] M.J. McAuliffe, F.M. Lalonde, D. McGarry, W. Gandler, K. Csaky, and B.L. Trus. Medical image processing, analysis and visualization in clinical research. In *Computer-Based Medical Systems, 2001. CBMS 2001. Proceedings. 14th IEEE Symposium on*, pages 381–386. IEEE, 2001.
- [54] O. Miraucourt and S. Garnotel. Numtourcfd. numtourcfd.univ-reims.fr.
- [55] G. Pagé. *Quantification et caractérisation des écoulements sanguins dans l’arborescence vasculaire de la région cervico-faciale par IRM de flux : évaluation et application*. PhD thesis, Université de Picardie Jules Verne, 2016.
- [56] J.L. Poiseuille. *Recherches expérimentales sur le mouvement des liquides dans les tubes de très-petits diamètres*. Imprimerie Royale, 1844.
- [57] ANR Vivabrain Project. AngioTK. <http://www.cemosis.fr/interdisciplinary-bio-project/2015/09/01/angiotk/>.
- [58] A. Quarteroni. *Numerical Models for Differential Problems*, volume 2. Springer, 2009.
- [59] S. Qvarlander, B. Lundkvist, L.O.D. Koskinen, J. Malm, and A. Eklund. Pulsatility in CSF dynamics: Pathophysiology of idiopathic normal pressure hydrocephalus. *Journal of Neurology, Neurosurgery & Psychiatry*, 84(7):735–741, 2013.
- [60] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation

- for Statistical Computing, Vienna, Austria, 2013.
- [61] O. Reynolds. An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels. *Proceedings of the Royal Society of London*, 35(224-226):84–99, 1883.
- [62] E.G. Richardson and E. Tyler. The transverse velocity gradient near the mouths of pipes in which an alternating or continuous flow of air is established. *Proceedings of the Physical Society*, 42(1):1, 1929.
- [63] J.E. Roberts and J.M. Thomas. Mixed and hybrid methods. *Handbook of Numerical Analysis*, 2:523–639, 1991.
- [64] N. Rott. Note on the history of the Reynolds number. *Annual Review of Fluid Mechanics*, 22(1):1–12, 1990.
- [65] H. Rouvière and A. Delmas. *Anatomie humaine: descriptive, topographique et fonctionnelle. Tête et cou*, volume 1. Elsevier Masson, 2002.
- [66] A. Samaké, V. Chabannes, C. Daversin, V. Doyeux, M. Ismail, G. Pena, C. Prud’homme, C. Trophime, and S. Veys. Advances in Feel++: A domain specific embedded language in C++ for partial differential equations. In *Eccomas’12 - European Congress on Computational Methods in Applied Sciences and Engineering*, 2012.
- [67] S.E. Sankari. *Cerebral hydrodynamics imaging: from physiology to pathology*. PhD thesis, Université de Picardie Jules Verne, 2009.
- [68] E. Schmidt. PROLIPHYC – cerebrospinal fluid proteome in hydrocephalus – trial details.
- [69] C.A. Schneider, W.S. Rasband, and K.W. Eliceiri. NIH image to ImageJ: 25 years of image analysis. *Nature Methods*, 9(7):671–675, 2012.
- [70] Z. Sheng, M. Thiriet, and F. Hecht. An efficient numerical method for the equations of steady and unsteady flows of homogeneous incompressible Newtonian fluid. *Journal of Computational Physics*, 230(3):551–571, 2011.
- [71] Z. Sheng, M. Thiriet, and F. Hecht. 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, 2013.
- [72] T. Stöcker, K. Vahedipour, D. Pflugfelder, and N.J. Shah. High-performance computing MRI simulations. *Magnetic Resonance in Medicine*, 64(1):186–193, 2010.
- [73] S. Sy and C.M. Murea. Algorithm for solving fluid-structure interaction problem on a global moving mesh. In *Proceedings of IV European Conference on Computational Mechanics, Paris, France*, 2010.
- [74] MeCS. Plateforme modélisation et calcul scientifique. <https://www.mecs.u-picardie.fr/>.
- [75] ROMEO. Centre de calcul ROMEO. <https://romeo.univ-reims.fr/>.

- [76] S. Uchida. The pulsating viscous flow superposed on the steady laminar motion of incompressible fluid in a circular pipe. *Zeitschrift für Angewandte Mathematik und Physik ZAMP*, 7(5):403–422, 1956.
- [77] Neurosurgery Unit University of Cambridge. ICM+. <http://www.neurosurg.cam.ac.uk/pages/ICM/about.php>.
- [78] M. Ursino and C.A. Lodi. A simple mathematical model of the interaction between intracranial pressure and cerebral hemodynamics. *Journal of Applied Physiology*, 82(4):1256–1269, 1997.
- [79] J.M. Valdueza, T. von Münster, O. Hoffman, S. Schreiber, and K.M. Einhäupl. Postural dependency of the cerebral venous outflow. *The Lancet*, 355(9199):200–201, 2000.
- [80] I.E. Vignon-Clementel, C.A. Figueroa, K.E. Jansen, and C.A. Taylor. 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, 2006.
- [81] M.H. Wilson. Monro-kellie 2.0: The dynamic vascular and venous pathophysiological components of intracranial pressure. *Journal of Cerebral Blood Flow & Metabolism*, 2016.
- [82] J.R. Womersley. Method for the calculation of velocity, rate of flow and viscous drag in arteries when the pressure gradient is known. *The Journal of Physiology*, 127(3):553–563, 1955.