



HAL
open science

Schémas semi-implicites et de diffusion-redistanciation pour la dynamique des globules rouges

Arnaud Sengers

► **To cite this version:**

Arnaud Sengers. Schémas semi-implicites et de diffusion-redistanciation pour la dynamique des globules rouges. Analyse fonctionnelle [math.FA]. Université Grenoble Alpes, 2019. Français. NNT : 2019GREAM032 . tel-02341602

HAL Id: tel-02341602

<https://theses.hal.science/tel-02341602>

Submitted on 31 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTE UNIVERSITE GRENOBLE ALPES

Spécialité : **Mathématiques Appliquées**

Arrêté ministériel : 25 mai 2016

Présentée par

«Arnaud SENGERS »

Thèse dirigée par **Emmanuel MAITRE**
et co-dirigée par **Mourad ISMAIL**

préparée au sein du **Laboratoire Jean Kuntzmann**
dans l'**École Doctorale Mathématiques, Sciences et
technologies de l'information, Informatique**

**Schémas semi-implicites et de
diffusion-redistanciation pour la
dynamique des globules rouges.**

**Semi-implicit and diffusion-
redistanciation schemes for the
dynamic of red blood cells.**

Thèse soutenue publiquement le **19 Juillet 2019**,
devant le jury composé de :

Monsieur Georges-Henri COTTET

Professeur des Universités, Université Grenoble Alpes, Président

Monsieur Cédric GALUSINSKI

Professeur des Universités, Université de Toulon, Rapporteur

Madame Marcela SZOPOS

Professeur des Universités, Université Paris Descartes, Rapporteur

Monsieur Élie BRETIN

Maître de Conférences, INSA de Lyon, Examineur

Monsieur Emmanuel MAITRE

Professeur des Universités, Grenoble INP, Directeur de thèse

Monsieur Mourad ISMAIL

Maître de Conférences, Université Grenoble Alpes, Co-directeur de thèse

Monsieur Thibaut MÉTIVET

Post-Doctorant, Université de Strasbourg, Invité



SOMMAIRE

1	Interaction Fluide-Structure et Méthode <i>levelset</i>	23
1.1	Introduction	25
1.2	Physique des vésicules et des globules rouges	27
1.3	Interaction Fluide-Structure	31
1.3.1	Méthodes purement lagrangiennes	33
1.3.2	Méthodes hybrides Eulériennes/Lagrangiennes	34
1.3.3	Méthodes purement Eulériennes	36
1.3.4	Méthodes de suivi d'interface et de capture d'interface pour les méthodes eulériennes.	40
1.4	Méthode de frontière immergée eulérienne	43
1.4.1	Modèle fluide	44
1.4.2	Physique d'une membrane élastique immergée	45
1.4.3	Méthode de frontière immergée	46
1.4.4	Fonction <i>levelset</i>	47
1.4.5	Équation de transport	51
1.4.6	Réinitialisation de la fonction <i>levelset</i>	51
1.4.7	Couplage fluide/structure pour des modèles de vésicules	56
1.4.8	Multiplés vésicules immergées et force hydrodynamique	58
1.4.9	Méthode champ de phase et liens avec la méthode <i>levelset</i> des modèles de vésicules immergées	60
1.5	Discrétisation du couplage et analyse de stabilité	64
1.5.1	Reformulation du système sous une formulation vertex	67
1.5.2	Analyse de stabilité du schéma explicite	68
1.6	Premier schéma de prédiction	72
1.6.1	Dérivation du schéma semi-implicite	72
1.6.2	Analyse de stabilité du schéma semi-implicite	73
1.6.3	Résultats numériques du schéma de prédiction	75

1.6.4	Lien avec les méthodes de mouvements générés par diffusion	77
2	Mouvement d'interface généré par diffusion	79
2.1	Introduction : le schéma de Merriman,Bence,Osher	81
2.2	Intuition	83
2.3	Méthode de Convolution-Seuillage	85
2.3.1	Mouvement par courbure moyenne multiphasique avec jonctions	85
2.3.2	Mouvement par courbure moyenne avec conservation de volume	87
2.3.3	Mouvement par flot de Willmore	89
2.3.4	Un défaut majeur : la précision proche de l'interface.	93
2.4	Méthode de Convolution-Redistanciation	94
2.4.1	Développement de la convolution de la fonction distance au voisinage de l'interface	95
2.4.2	Développement de la convolution d'une fonction $f(d)$ au voisinage de l'interface	100
2.4.3	Mouvement par courbure moyenne par Convolution-Redistanciation	101
2.4.4	Vérification numérique de convergence sur le cercle	102
2.4.5	Mouvement par courbure moyenne en dimension 2	103
2.4.6	Extraction de la courbure par diffusion	105
2.4.7	Mouvement par courbure moyenne avec conservation de volume par convolution-redistanciation	109
2.4.8	Flot de Willmore par convolution-redistanciation en dimension 2	110
2.4.9	Vérification numérique du mouvement par flot de Willmore en dimension 2	111
2.4.10	Flot de Willmore par convolution-redistanciation en dimension 3	112
2.4.11	Vérification numérique du mouvement par flot de Willmore en dimension 3	114
2.5	Convergence théorique du mouvement par courbure moyenne	118
2.5.1	Notations	119
2.5.2	L'espace des fonctions à variations bornées $BV(\Omega)$	119
2.5.3	L'approche de Almgren-Taylor-Wang et Luckhaus-Sturzenhecker	120
2.5.4	L'approche de Laux-Otto	124
2.5.5	L'approche de Chambolle	126
2.5.6	Convergence du schéma convolution-redistanciation	129
3	Flot de Willmore sous contrainte de volume et d'aire, application aux schémas semi-implicites	131
3.1	Méthodes de conservation de volume et d'aire	132

3.1.1	Méthodes de conservation de volume et aire en formulation <i>levelset</i> et pour les méthodes de convolution-seuillage	133
3.1.2	Conservation simultanée du volume et de l'aire	135
3.1.3	Comparaison avec une méthode par multiplicateurs de Lagrange	138
3.1.4	Mouvement par flot de Willmore à volume et aire constants	141
3.2	Obtention numérique de la forme d'équilibre d'un globule rouge	145
3.2.1	Cas de la dimension 2	145
3.2.2	Cas de la dimension 3	150
3.3	Méthode de prédiction dans le cas des globules rouges	155
3.3.1	Minimisation pour une équation de transport discrétisée par le schéma d'Euler explicite	156
3.3.2	Minimisation pour une équation de transport discrétisée par un schéma d'Euler implicite	158
3.3.3	Perspectives de la méthode	159
4	Conclusion et perspectives	161

LISTE DES FIGURES

1.1	Forme d'un globule rouge. Source : Shutterstock.	28
1.2	Représentation d'une vésicule et de la structure moléculaire de sa membrane. L'échelle entre la membrane phospholipidique et la taille de la vésicule n'est pas respectée. Source : Nasa Astrobiology Institute : http://astrobiology.nasa.gov/nai/	28
1.3	Forme d'équilibre pour un globule en dimension 2. Le taux de remplissage est d'environ 0.5.	31
1.4	Représentation de la carte ALE \mathcal{A}^T avec le domaine de référence Ω^* et le domaine déformé Ω^t	35
1.5	Illustration de la méthode des multiplicateurs de Lagrange.	38
1.6	Illustration de la répartition du domaine pour la méthode de la frontière élargie.	39
1.7	Représentation des méthodes de suivi d'interface. Les points bleus correspondent aux marqueurs lagrangiens. À gauche, pour la méthode de suivi d'interface, ils sont situés sur l'interface. À droite, pour la méthode de suivi de volume, ils sont situés sur l'ensemble du domaine solide.	41
1.8	Illustration de la méthode de volume de fluide. Chaque cellule contient la concentration de fluide qu'elle renferme. La courbe rouge représente l'interface.	42
1.9	Représentation de la fonction distance signée. L'interface est la ligne de niveau 0 de ϕ représentée en noire.	50
1.10	Représentation des fonctions d'Heaviside et de Dirac régularisée.	50
1.11	Méthode d'obtention d'un maillage \mathbb{P}^1 ayant les mêmes degrés de liberté que le maillage \mathbb{P}^2	54
1.12	Représentation des différentes fonctions <i>label</i> : L_0 à gauche, L_1 au centre et L_2 à droite.	59
1.13	Représentation des deux fonctions <i>levelset</i> ϕ_1 (à gauche) et ϕ_2 (à droite).	60

1.14	Représentation de la fonction de Ginzburg-Landau de type double potentiel. On observe deux minima qui permettent de séparer les deux phases.	62
1.15	Simulation du schéma explicite (1.103) pour un pas de temps $dt = 0.0035$ supérieur au seuil limite. La bulle explose rapidement.	76
1.16	Simulation du schéma de prédiction (1.141) pour un pas de temps $dt = 0.0055$ supérieur au seuil limite du schéma explicite. À noter une légère perte de volume.	76
2.1	Configuration dans un voisinage du point P.	83
2.2	Profil d'une étape de l'algorithme de Diffusion-Seuillage. La courbe noire décrit la fonction caractéristique initiale. La courbe en pointillés bleus représente la fonction soumise à une diffusion radiale pure qui ne déplace pas l'interface. En pointillés rouges, on obtient le résultat de l'équation (2.6) et la courbe rouge représente le seuillage de cette fonction.	84
2.3	Configuration du cas multiphasique. L'espace est recouvert de phases Ω_i se déplaçant par courbure moyenne et en tension les unes avec les autres.	86
2.4	Configuration à un point de jonction entre 3 phases. Les angles θ_i sont déterminés par la relation (2.10).	86
2.5	Configuration de l'étude.	90
2.6	Exemple de situation où la résolution du maillage est trop faible et l'interface reste coincée dans un cycle infini.	93
2.7	La fonction distance contient des informations à un niveau inférieur à la taille des mailles. La configuration à l'étape $k + 1$ a évolué même dans le cas de faibles déplacements.	94
2.8	Projection sur la courbe utilisant la distance signée.	97
2.9	Convergence de l'erreur sur le rayon les deux algorithmes de mouvement par courbure moyenne 2.4.3 : 7 et 2.4.3 : 8. On retrouve l'observation faite dans [ERT ⁺ 10] que l'on n'obtient pas l'ordre 2 avec la méthode 2.4.3 : 8 mais plutôt une division par deux de l'erreur.	105
2.10	Évolution de l'erreur des méthodes (2.82) et (2.83) en fonction du pas de temps. La courbure est calculée en dimension 2 sur un cercle de rayon $r = 0.25$ pour une discrétisation éléments finis avec des éléments de taille $h = 0.01$ et une interface d'épaisseur $1.5h$. L'erreur calculée est : $e_{\mathbb{L}^2} = \frac{\int_{\Omega} [H(x) - H_{th}(x)]^2 dx}{\int_{\Omega} H_{th}(x)^2 dx}$	107
2.11	Convergence de l'erreur \mathbb{L}^2 pour les différentes méthodes de calcul de la courbure.	108
2.12	Convergence de l'erreur sur le rayon de l'algorithme de mouvement par flot de Willmore en dimension 2 2.4.8 : 11.	113

2.13	Convergence numérique du tore T selon le flot de Willmore vers un tore de Clifford.	116
2.14	Évolution de l'énergie d'Helfrich au cours du temps. On observe une décroissance jusqu'au minimum puis la courbe croît à nouveau.	117
2.15	Évolution du rapport $\frac{a}{b}$ au cours du temps. On observe qu'il augmente vers $\sqrt{2}$. Cependant, on ne se stabilise pas à $\sqrt{2}$	118
2.16	Évolution de la quantité $a^2 + b^2$ au cours du temps. On observe une légère décroissance de celle-ci.	118
3.1	Choix de la ligne de niveau $\{\phi - \lambda = 0\}$ qui a le volume choisi, avec ici $\lambda > 0$	134
3.3	Ellipse initiale pour notre simulation avec un taux de remplissage autour de 0.5.	146
3.4	Convergence de l'algorithme 3.1.4 : 15 vers la forme d'équilibre .	147
3.5	Perte de la symétrie et léger déplacement du globule sur un maillage non structuré.	147
3.6	Évolution de l'énergie d'Helfrich au cours du temps en dimension 2.	148
3.7	Comparaison des formes d'équilibres obtenues par l'algorithme 3.1.4 : 15 avec celles de [KHM11] par la méthode de <i>Lattice-Boltzmann</i> pour des taux de gonflements de 0.9;0.8;0.7;0.6. Nous représentons notre résultat pour 0.5.	150
3.8	Formes d'équilibres pour différents taux de remplissage ν	151
3.9	Courbure calculée par diffusion pendant l'étape de correction de volume et d'aire présentant plusieurs irrégularités et artefacts qui s'accumulent au cours de la simulation	152
3.10	Évolution du profil vers une forme d'équilibre biconcave du globule pour la simulation en dimension 3.	153
3.11	Évolution de l'énergie d'Helfrich au cours du temps pour l'algorithme de flot de Willmore 3D avec conservation de volume et d'aire.	154
3.12	Formes d'équilibres d'un globule rouge pour différents taux de remplissage ν en dimension 3.	154

ABSTRACT

In this work, we propose new semi-implicit schemes to improve the numerical simulations of the motion of an immersed red blood cell. We consider the levelset method where the interface is described as the 0 isoline of an auxiliary function and the fluid-structure coupling is done by adding a source term in the fluid equation.

The idea of these semi-implicit scheme is to predict the position and the shape of the structure through a heat equation and to use this prediction to improve the accuracy of the source term in the fluid equation. This type of semi-implicit scheme has firstly been implemented in the case of a multiphase flow and an immersed elastic membrane and has shown better temporal stability than an explicit scheme, resulting in an improved global efficiency.

In order to extend this method to the case of a red blood cell, we propose an algorithm to compute the Willmore flow in dimensions 2 and 3. In the spirit of the diffusion generated motion methods, our method simulates a non linear four order flow by only solving heat equations. To ensure the conservation of the volume and area of the vesicle, we add to the method a correction step that slightly moves the interface so that we recover the constraints.

The combination of these two steps allows to compute the behavior of a red blood cell left at rest. We validate this method by obtaining the convergence to an equilibrium shape in both 2D and 3D. Finally we introduce a semi-implicit scheme in the case of a red blood cell that shows how we can use this method as a prediction in the complete coupling model.

Keywords: fluid-structure interaction, levelset method, semi-implicit scheme, diffusion generated motion of interface, convolution-redistanciation method, prediction scheme.

RÉSUMÉ

Dans ce travail, nous nous sommes intéressés à la mise en place de schémas semi-implicites pour l'amélioration des simulations numériques du déplacement d'un globule rouge dans le sang. Nous considérons la méthode *levelset* où l'interface fluide-structure est représentée comme la ligne de niveau 0 d'une fonction auxiliaire et le couplage est effectué en ajoutant un terme source dans l'équation fluide.

Le principe de ces schémas semi-implicites est de prédire la position et la forme de la structure par une équation de la chaleur et d'utiliser cette prédiction pour obtenir un terme de force plus précis dans l'équation fluide. Ce type de schémas semi-implicites a d'abord été mis en place dans le cadre d'un système diphasique ou d'une membrane élastique immergée pour avoir des conditions moins restrictives sur le pas de temps que pour un couplage explicite. Cela a permis d'en choisir un plus grand et ainsi augmenter l'efficacité globale de l'algorithme complet par rapport à un schéma explicite classique.

Afin d'étendre ce raisonnement au cas d'un globule rouge, nous proposons un algorithme pour simuler le flot de Willmore en dimensions 2 et 3. Notre méthode s'inspire des méthodes de mouvements d'interface générés par diffusion et nous arrivons à obtenir un flot non linéaire d'ordre 4 uniquement avec des résolutions d'équations de la chaleur. Pour assurer la conservation du volume et de l'aire d'un globule rouge, nous proposons ensuite une méthode de correction qui déplace légèrement l'interface afin de recoller aux contraintes.

La combinaison des deux étapes précédentes décrit le comportement d'un globule rouge laissé au repos. Nous validons cette méthode en obtenant une forme d'équilibre d'un globule rouge. Nous proposons enfin un schéma semi-implicite dans le cas d'un globule rouge qui ouvre la voie vers l'utilisation de cette méthode comme prédicteur de l'algorithme de couplage complet.

Mots clés : interaction fluide-structure, méthode *levelset*, schémas semi-implicites, mouvements d'interface générés par diffusion, méthode de convolution-seuillage, méthode de convolution-redistanciation, schéma de prédiction.

NOTATIONS

Quantités physiques et géométriques

- N : dimension de l'espace
- Ω : ensemble du domaine
- Ω_f : région occupée par le fluide
- Ω_s : région occupée par la structure
- Γ : interface fluide-structure
- u : champ de vitesse du fluide
- p : champ de pression
- μ_f : viscosité du fluide
- ρ_f : densité du fluide
- Re : nombre de Reynolds
- $D(u) = \frac{\nabla u + \nabla u^T}{2}$: tenseur des déformations de la vitesse du fluide
- χ : fonction indicatrice du domaine structure Ω_s
- ϕ : fonction *levelset* du domaine structure Ω_s
- $V(\phi)$: volume englobé par la fonction *levelset*, mesure de $\{\phi < 0\}$
- $A(\phi)$: aire de l'interface décrite par la fonction *levelset*, mesure de l'ensemble $\{\phi = 0\}$
- d : fonction distance signée à l'interface

- n : normale sortante à l'interface Γ
- H : courbure moyenne de l'interface Γ
- $\kappa = (N - 1)H$: courbure totale de l'interface Γ
- K : courbure de Gauss de l'interface Γ
- δ_ε : fonction delta régularisée d'épaisseur ε
- \mathcal{H}_ε : fonction Heaviside régularisée d'épaisseur ε
- G_t : noyau de la chaleur pour le temps t
- \times : produit vectoriel en dimension 3
- $*$: produit de convolution
- P_Γ : projection orthogonale sur l'interface Γ
- Δ_Γ : opérateur de Laplace-Beltrami
- W : terme de Willmore associé au gradient de l'énergie d'Helfrich
- \mathbb{I} : matrice identité
- \otimes : produit tensoriel

Paramètres numériques

- $\Delta x, h$: taille d'un élément, pas de discrétisation en espace
- δt : pas de temps
- ε : épaisseur de l'interface régularisée
- \mathbb{P}^n : éléments finis de Lagrange d'ordre n

INTRODUCTION

La rhéologie est l'étude de la déformation et de l'écoulement de la matière sous l'effet des certaines contraintes telles que des forces ou une géométrie du domaine particulières. Une branche importante de ce domaine est l'hémodynamique qui s'intéresse aux écoulements sanguins. Comme la compréhension du système cardio-vasculaire offre beaucoup de perspectives en médecine pour la prévention et le traitement des maladies associées, beaucoup d'efforts de recherche sont mis en œuvre pour la perfectionner. En complément des expérimentations physiques, des modèles mathématiques et des simulations numériques sont développés.

La composition du sang pour un individu sain est faite d'environ 55% de fluide, le plasma, et de 45% de globules rouges tandis que le reste des particules (globules blancs, plaquettes et diverses protéines) représente moins de 1% du volume occupé. Le sang est un fluide complexe, dit aussi non newtonien, dans le sens où ses lois d'écoulements ne relèvent pas de celles des fluides ordinaires, décrites par l'équation de Navier-Stokes. Pour décrire les écoulements sanguins avec précision, il convient en effet de prendre en compte l'impact des globules rouges sur ceux-ci. Leur déplacement et changement de forme incessants engendrent une rétroaction sur le fluide qu'il faut ajouter au système. On se retrouve donc devant un problème de couplage fluide-structure où l'on doit modéliser le comportement du plasma, des globules rouges immergés et de leur interactions. La contribution du reste des particules est négligée en raison de leur faible proportion.

L'étude des problèmes d'interactions fluide-structure constitue un des principaux challenges dans le domaine de mécanique des fluides numériques. La principale difficulté posée par ce type de problème vient de la différence de représentation entre deux domaines. En effet, il est naturel de représenter le fluide par une approche eulérienne et un solide par une formulation lagrangienne. Il existe deux approches pour aborder cette difficulté : l'approche hybride et l'approche uniforme. Pour la première, on conserve les deux formulations distinctes et on doit gérer les interactions, généralement par des conditions sur l'interface. L'approche uniforme consiste à représenter le fluide ou la structure différemment afin d'avoir une formulation uniforme pour l'ensemble du problème. Cela permet d'avoir une gestion plus simple du couplage. Avec une formulation purement lagrangienne, les différents objets sont représentés par un ensemble de marqueurs lagrangiens qui se déplacent avec le fluide. Au contraire, avec une méthode purement eulérienne, on utilise un maillage fixe qui décrit simultanément les deux domaines.

Le choix de la formulation doit être complété par une méthode pour représenter l'interface fluide-structure. On distingue deux catégories majeures : les méthodes de suivi d'interface et les méthodes de capture d'interface. Une méthode de suivi d'interface représente l'interface à l'aide d'un ensemble de marqueurs lagrangiens qui suivent son déplacement. Une méthode de capture d'interface décrit la position de l'interface à l'aide d'une fonction scalaire

auxiliaire qui est transportée par le fluide.

Dans cette thèse, nous choisissons une formulation purement eulérienne couplée à une méthode de capture d'interface : la méthode *levelset* introduite par Osher [OF06]. L'interface est représentée comme la ligne de niveau 0 de la fonction *levelset* qui est transportée par le fluide. L'action des globules rouges est prise en compte en ajoutant un terme source dans le membre de droite de l'équation de Navier-Stokes en s'inspirant de la méthode de frontière immergée de Peskin [Pes02]. Le terme source peut se diviser en deux composantes. D'une part, un terme de force est dû à la forme des globules qui cherchent à minimiser leur énergie de courbure. D'autre part, un terme élastique de pénalisation est ajouté afin de contraindre l'inextensibilité du globule, c'est-à-dire forcer l'aire de sa membrane à rester constante. Ce modèle est introduit par Cottet et Maitre dans [CM04, CM06]. Le système doit également être complété par des conditions aux bords qui sont déterminées selon l'application considérée.

Une fois notre modèle mathématique complètement déterminé, nous pouvons centrer notre étude sur les aspects numériques de ce dernier. Pour un problème de couplage fluide-structure, la discrétisation requiert une attention particulière. En plus des contraintes liées à la discrétisation de chaque équation, il y a une condition de stabilité portant sur la discrétisation temporelle du couplage. Un schéma implicite résolvant le système d'un bloc est généralement stable. Sa résolution est cependant souvent trop coûteuse d'un point de vue algorithmique pour être utilisée dans la pratique. On lui préfère alors le schéma explicite suivant : on résout d'abord l'équation de Navier-Stokes en explicitant les termes de force puis on utilise le champ de vitesse obtenu pour l'équation d'advection de la fonction *levelset*. Cette solution est plus simple en termes de solveur et moins coûteuse en temps de calcul que le schéma implicite. Le choix du pas de temps est néanmoins typiquement limité par une condition de type CFL restrictive d'ordre $\frac{3}{2}$ en le pas de maillage.

Dans [CM16], une nouvelle stratégie est proposée afin d'améliorer cette restriction sur le pas de temps. L'idée est de prédire la forme de la membrane pour avoir une meilleure expression des forces dans l'équation de Navier-Stokes. La prédiction est obtenue en résolvant une équation de la chaleur dont le coût en temps de calcul est faible comparé à celui de l'équation fluide. Ceci constitue le point de départ de la thèse. Dans [CM16], la méthode a été testée dans le cas d'un système diphasique et d'une membrane élastique immergée avec une résolution par différences finies.

L'objectif de cette thèse est de voir comment étendre ce principe de prédiction, notamment au cas d'un globule rouge. On procède en dressant un parallèle entre notre première méthode de prédiction et la classe des méthodes de mouvements générés par diffusion. Introduite par Merriman, Bence et Osher [MBO92], cette classe de méthodes consiste à déplacer une interface en alternant une étape de diffusion d'une fonction indicatrice et une étape de seuillage où on se ramène à une fonction indicatrice. Cet algorithme a été modifié dans [ERT⁺10] dans le

cas d'une fonction distance signée afin de pallier des problèmes de précision de maillage proche de l'interface. Nous avons adapté cette idée pour obtenir un déplacement par flot de Willmore en dimensions 2 et 3. L'intérêt est que l'on simule un flot non linéaire d'ordre 4 uniquement avec des résolutions d'équations de la chaleur. Ceci est particulièrement utile pour des simulations avec des éléments \mathbb{P}^1 où la précision est compliquée à maintenir lors des dérivations successives.

Pour simuler le comportement d'un globule rouge, il faut ajouter au déplacement par flot de Willmore une contrainte de conservation de volume et d'aire. Nous opérons une descente de gradient selon le flot de Willmore en relâchant les contraintes puis en y recollant *a posteriori*. Pour faire cette deuxième étape, nous adaptons une idée de Smolianski [Smo05] qui consiste à choisir comme nouvelle interface la ligne de niveau permettant d'obtenir un volume cible. Nous proposons de choisir comme nouvelle interface l'ensemble $\{\phi = \lambda + \mu H\}$, avec λ et μ des constantes choisies pour obtenir un volume et une aire cibles. Nous montrons l'efficacité de notre méthode en simulant l'obtention d'une forme d'équilibre d'un globule rouge laissé au repos. Nous concluons ce travail par la mise en place d'un schéma semi-implicite dans le cas d'un globule rouge qui ouvre la voie vers l'utilisation de cette méthode comme prédicteur de l'algorithme de couplage complet.

Une autre approche possible pour augmenter la stabilité numérique est la méthode de schémas augmentés. De tels schémas ont été proposés par Bresch et al. [BDZ15, BCNV16, BJ18] dans le cas d'un fluide compressible. L'idée est d'ajouter une nouvelle inconnue homogène à une vitesse dans le système. Cela permet de dériver des estimations *a priori* pour étudier la convergence théorique du schéma ou d'améliorer les conditions de stabilité numérique en obtenant des membres de droites antisymétriques l'un par rapport à l'autre. Ayant préféré consacrer cette thèse à l'étude des schémas de prédictions, nous proposons en annexe B les débuts de réflexion sur la possibilité d'adapter ces travaux à notre contexte. Le principe est de faire jouer à notre fonction *levelset* le rôle de la fonction de densité du cas compressible. La nouvelle inconnue s'exprimerait alors en fonction du gradient de la fonction *levelset*.

Pour nos simulations, nous utilisons dans cette thèse la librairie d'éléments finis FEEL++¹. Il s'agit d'une librairie expressive dans le sens où l'écriture du code se rapproche fortement de l'expression mathématique des problèmes variationnels. Elle fournit la possibilité de faire des calculs massivement parallèles dont la gestion est faite en interne. Il suffit à l'utilisateur de choisir le nombre de processeurs alloués pour la simulation. De même, les solveurs pour les problèmes différentiels sont également gérés en interne et il suffit de donner la formulation variationnelle du problème. C'est pour ces différentes raisons que nous avons choisi d'utiliser FEEL++ dans cette thèse. Cela nous permet de proposer un code disponible et simple d'utilisation pour les problèmes que

¹<http://www.feelpp.org>

nous avons traités. Nous proposons en annexe A, une introduction à la librairie ainsi qu'un exemple de code pour montrer la simplicité d'implémentation et sa similarité avec sa formulation mathématique. Les codes de calculs pour la forme d'équilibre d'un globule rouge en dimension 2 et 3 sont disponibles sous Github ².

La thèse est organisée comme suit :

Dans le Chapitre 1, nous proposons une introduction générale des différentes méthodes numériques. Nous nous focaliserons sur les méthodes eulériennes et en particulier la méthode *levelset* [OF06]. Ensuite, nous détaillons le modèle utilisé dans cette thèse [CM04, CM06], qui est une adaptation aux méthodes de capture d'interface de la méthode de frontière immergée de Peskin [Pes02]. On termine ce premier chapitre par la présentation du schéma semi-implicite par prédiction [CM16]. Nous menons une analyse numérique pour montrer son efficacité dans le cas d'un système diphasique ou d'une membrane élastique immergée.

Le Chapitre 2 est consacré à l'étude des mouvements d'interface générés par diffusion dont le principe est proche de celui de notre étape de prédiction. Introduite par Merriman, Bence et Osher [MBO92] dans le contexte de la méthode champ de phase pour simuler un mouvement par courbure moyenne, la méthode de convolution-seuillage a depuis été étendue à des mouvements géométriques plus généraux. Un défaut majeur de ces méthodes est la nécessité d'avoir une précision fine proche de l'interface et d'avoir recours à un remaillage du domaine. Dans [ERT⁺10], les auteurs proposent de diffuser la fonction distance signée à l'interface pour obtenir un mouvement géométrique piloté par diffusion de surface. Nous nous inspirons de cette idée pour obtenir un déplacement par flot de Willmore en dimension 2. Nous proposons ensuite une méthode originale pour étendre l'algorithme au cas de la dimension 3 en diffusant également le carré de la fonction distance signée. On conclut le chapitre en faisant une présentation rapide des différents résultats de convergence théorique existants pour les méthodes de convolution-seuillage.

Dans le Chapitre 3, on s'intéresse au déplacement par flot de Willmore avec conservation de volume et d'aire. Après un passage en revue des différentes méthodes de conservation existantes dans le contexte *levelset*, nous introduisons un algorithme efficace permettant d'obtenir une interface englobant un volume et une aire cibles. Nous le combinons alors avec notre algorithme de diffusion pour simuler un déplacement par flot de Willmore avec conservation de volume et d'aire. Dans notre approche, on relâche les contraintes dans un premier temps pour y recoller *a posteriori* afin d'obtenir un algorithme efficace en termes de temps de calcul. Nous testons notre méthode en obtenant une forme d'équilibre d'un globule rouge laissé au repos. Nous utilisons ensuite cet algorithme comme prédicteur pour un schéma semi-implicite de couplage dans le cas du globule rouge.

²<https://github.com/sengers/EquilibriumShape>

CHAPITRE 1

INTERACTION FLUIDE-STRUCTURE ET MÉTHODE *LEVELSET*

Sommaire

1.1	Introduction	25
1.2	Physique des vésicules et des globules rouges	27
1.3	Interaction Fluide-Structure	31
1.3.1	Méthodes purement lagrangiennes	33
1.3.2	Méthodes hybrides Eulériennes/Lagrangiennes	34
1.3.3	Méthodes purement Eulériennes	36
1.3.4	Méthodes de suivi d'interface et de capture d'interface pour les méthodes eulériennes.	40
1.4	Méthode de frontière immergée eulérienne	43
1.4.1	Modèle fluide	44
1.4.2	Physique d'une membrane élastique immergée	45
1.4.3	Méthode de frontière immergée	46
1.4.4	Fonction <i>levelset</i>	47
1.4.5	Équation de transport	51
1.4.6	Réinitialisation de la fonction <i>levelset</i>	51
1.4.7	Couplage fluide/structure pour des modèles de vésicules	56
1.4.8	Multiplés vésicules immergées et force hydrodynamique	58
1.4.9	Méthode champ de phase et liens avec la méthode <i>level-</i> <i>set</i> des modèles de vésicules immergées	60

1.5	Discrétisation du couplage et analyse de stabilité	64
1.5.1	Reformulation du système sous une formulation vertex	67
1.5.2	Analyse de stabilité du schéma explicite	68
1.6	Premier schéma de prédiction	72
1.6.1	Dérivation du schéma semi-implicite	72
1.6.2	Analyse de stabilité du schéma semi-implicite	73
1.6.3	Résultats numériques du schéma de prédiction	75
1.6.4	Lien avec les méthodes de mouvements générés par diffusion	77

1.1 Introduction

L'étude des problèmes couplant un fluide et une structure élastique s'est fortement développée durant ces dernières années. Il existe en effet de nombreuses applications physiques rentrant dans ce cadre ; on pense par exemple à l'écoulement d'un fluide autour d'une aile d'avion ou d'un bateau. Dans cette thèse, on se consacre au cas d'une vésicule en suspension dans le sang, en particulier les globules rouges car ils représentent 99% des cellules présentes dans le sang. Ce sujet a déjà été source de nombreuses études et expérimentations par les biologistes et physiciens [ALV02, CM99, MVA⁺06, AV08, VPM09]. Avec l'explosion des moyens informatiques et de la puissance de calcul, les simulations numériques permettent de compléter les expériences physiques qui peuvent se révéler très coûteuses à réaliser.

La principale difficulté liée à l'étude de ces phénomènes réside dans le fait que le fluide influe sur la structure et réciproquement. Il est alors nécessaire de coupler les équations du problème, cette tâche étant particulièrement délicate. En effet, le choix de la discrétisation du couplage impose des conditions sur le pas de temps. Selon les paramètres physiques, un nombre de Reynolds élevé ou une élasticité forte dans notre cas, ces conditions peuvent se révéler trop restrictives pour la mise en place de simulations efficaces. Dans ce chapitre, on propose un aperçu général des différentes approches du couplage fluide-structure existantes dans la littérature. On les distinguera en trois catégories : purement lagrangiennes, eulérienne-lagrangiennes et purement eulériennes. Un fluide est naturellement décrit en formulation eulérienne alors qu'un solide l'est en formulation lagrangienne. Il s'agit alors soit d'adapter la formulation d'un des milieux pour unifier les représentations soit d'assurer un bon compromis entre les deux formulations.

L'objet de ce chapitre est de proposer un passage en revue de la littérature sur le couplage fluide-structure d'une membrane élastique et d'un fluide. On présente d'abord les méthodes purement lagrangiennes et hybrides. Parmi les méthodes purement lagrangiennes, on compte principalement la méthode *SPH* et la méthode *Vortex*. La méthode *SPH*, pour *smoothed particle hydrodynamics* a été introduite par Lucy [Luc77] et Monaghan [Mon94] puis étendue dans [Mon94, CL03] au cas des fluides incompressibles et enfin adaptée au contexte fluide-structure dans [AGS07]. Une revue générale de la méthode est proposée dans [WCW⁺16]. La méthode *Vortex* consiste à modifier l'équation de fluide en la remplaçant par une équation sur la vorticité de la vitesse. Une référence complète se trouve dans le livre de Cottet [CKP⁺00] et on cite également [Kou05, CCB⁺08, RBCK10, MGCM15] pour des références plus récentes sur le sujet. En ce qui concerne les méthodes hybrides, la plus répandue pour les problèmes de couplage fluide-structure est la méthode *Arbitrary Lagrangian Eulerian*, introduite par Donea dans [DGH82] dont une synthèse est proposée dans [DHPRF04]. Le principe de la méthode est d'utiliser un maillage se déplaçant à une certaine

vitesse, un maillage fixe correspondant à l'approche eulérienne et un maillage suivant l'interface correspondant à l'approche lagrangienne. On mentionne les travaux de Fernandez, Gerbeau et Grandmont [GGM01, FGG06, FMV15, FLV15] qui développent la méthode et résolvent notamment le problème de l'effet de masse ajoutée [COP97].

Dans cette thèse, on adoptera le troisième formalisme, à savoir une méthode purement eulérienne. Initiées par [Hym52, Sau63], ces méthodes permettent notamment d'utiliser un maillage fixe pour l'ensemble du domaine. Plusieurs méthodes permettent de traiter la présence de la structure selon le principe de domaine fictif : la méthode de pénalisation [Ca194, ABF99, Mau09], voir [CC08] pour le cas de solides rigides immergés ; la méthode des multiplicateur de Lagrange [GPHJ99, GPH⁺01, SP05] ; la méthode de frontière élargie introduite par Maury [Mau01] et développée avec Ismail [Ism04, BIM11].

Dans ce contexte eulérien, il existe deux approches classiques pour représenter l'interface : les méthodes de suivi d'interface et les méthodes de capture d'interface. Les méthodes de suivi d'interface proposent d'introduire des marqueurs lagrangiens qui suivent explicitement celle-ci. Deux contributeurs importants sur le sujet sont le groupe de Glimm et al [GGL⁺88, GGG⁺98, GGL⁺98] et Tryggvason [TBE⁺01, UT92, TT09]. Les méthodes de capture d'interface proposent de représenter l'interface implicitement grâce à une fonction auxiliaire qui est transportée par le fluide. Les trois approches les plus populaires sont la méthode *levelset* [OF01, OF06], *Volume of fluide*, dite *VOF*, [SZ03, RR02] et la méthode champ de phase [CL85, Kob93, KM99], voir [DLW04, DLRW05, DLL07, BKM05, Bib05] pour le couplage fluide-structure avec une vésicule. On citera également la méthode *coupled VOF/levelset* qui cherche à combiner les avantages des deux approches en utilisant deux fonctions auxiliaires.

Une avancée significative pour les méthodes eulériennes a été la méthode de frontière immergée, proposée par Peskin [Pes02]. Le couplage y est exprimé à l'aide d'un terme source dans l'équation de Navier-Stokes. Cependant, la localisation des forces élastiques nécessite l'utilisation de marqueurs lagrangiens et les interpolations liées à leur utilisation sur le maillage fixe peuvent induire des pertes de volume.

Dans le cas particulier d'une membrane élastique immergée, Cottet et Maitre ont proposé une adaptation de la méthode de frontière immergée en un modèle totalement eulérien par capture d'interface [CM04, CM06]. Une fonction *levelset* est utilisée pour décrire l'interface fluide-structure et la norme de son gradient permet d'exprimer l'étirement de la membrane et les forces élastiques associées. L'absence de marqueurs lagrangiens permet une meilleure conservation du volume et un traitement naturel des grandes déformations et changements topologiques. Il s'agit du modèle que l'on considérera dans de cette thèse.

Une autre problématique d'importance lors de la simulation numérique de problème fluide-structure est la discrétisation du couplage. Un schéma implicite résoudra simultanément les équations fluide et d'advection. Si cette approche

est inconditionnellement stable numériquement, elle est trop coûteuse en temps de calcul pour être envisageable. L'alternative est de découpler la résolution des équations en résolvant d'abord l'équation fluide en exprimant les forces à partir de valeurs explicitées de la fonction *levelset*. On utilise alors la nouvelle vitesse dans l'équation d'advection afin d'obtenir la valeur de la fonction *levelset*. Une telle stratégie est plus simple et moins coûteuse à résoudre algorithmiquement. Le choix du pas de temps est néanmoins limité par une condition type CFL plus restrictive que celle de l'advection dans le cas d'une membrane avec un coefficient de rigidité très élevé, cf Brackbill [BKZ92], Galusinski et Vigneaux [GV07, GV08] :

$$\delta t \leq C (\Delta x)^{\frac{3}{2}} \quad (1.1)$$

Dans [CM16], Cottet et Maitre proposent une nouvelle stratégie. Elle consiste à prédire la position de la membrane au pas de temps futur pour avoir une meilleure expression des forces élastiques dans l'équation de Navier-Stokes. On montrera alors que l'on peut obtenir des simulations plus efficaces avec ce schéma prédictif. Ce point constitue le point de départ de la thèse.

1.2 Physique des vésicules et des globules rouges

Les vésicules sont des membranes fermées en suspension dans un milieu aqueux. La membrane est constituée d'une bi-couche phospholipidique. Les molécules formant la membrane possèdent une tête hydrophile dirigée vers le fluide et une queue hydrophobe pointée vers l'intérieur de la membrane, cf figure 1.2. Les vésicules fabriquées artificiellement par les physiciens sont souvent de l'ordre de $10\mu\text{m}$.

Les globules rouges humains sont des cellules d'environ $7\mu\text{m}$ de diamètre et dénuées de noyau ou organelle. De ce fait, il s'agit d'une des cellules les plus simples du corps humain à étudier. L'intérieur d'un globule rouge est essentiellement composé d'hémoglobine et d'un cytosquelette. Le rôle principal des globules rouges est le transport d'oxygène.

Le comportement et les propriétés des vésicules se rapprochent de ceux des globules rouges. L'étude des vésicules constitue ainsi un point de départ vers la compréhension des propriétés visco-élastiques, dynamiques et rhéologiques de fluides tels que le sang. Un nombre conséquent d'études ont été menées expérimentalement sur la dynamique des vésicules [ALV02, CM99, MVA⁺06, AV08, VPM09].

Le problème consistant à déterminer la forme d'un globule rouge est célèbre du point de vue mathématique : il s'agit du problème de Willmore, qui correspond à chercher une surface d'aire et de volume englobé fixés et de courbure moyenne minimale. Les travaux de Canham [Can70], Helfrich [Hel73] et [Eva74] constituent les premiers travaux qui étudient les formes d'équilibres d'un globule rouge laissé au repos.

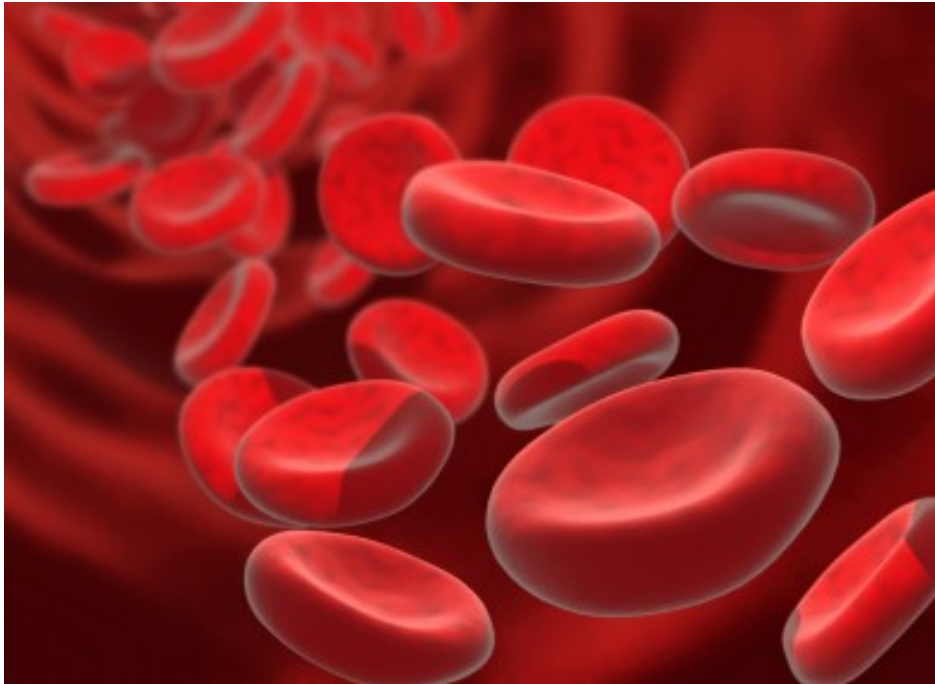


FIGURE 1.1 – Forme d'un globule rouge. Source : Shutterstock.

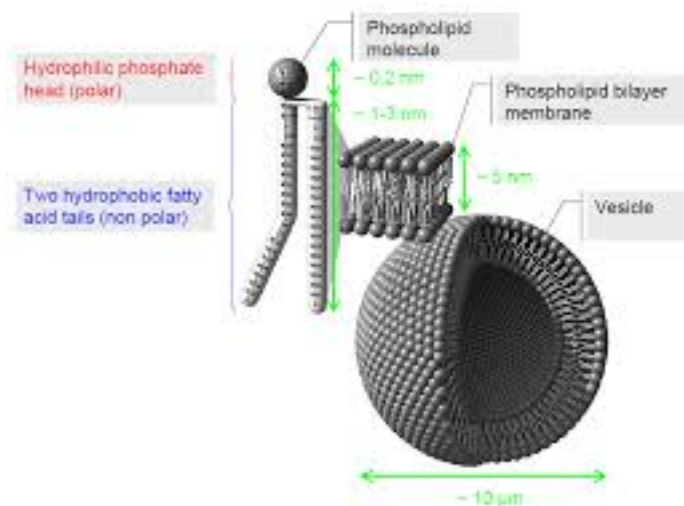


FIGURE 1.2 – Représentation d'une vésicule et de la structure moléculaire de sa membrane. L'échelle entre la membrane phospholipidique et la taille de la vésicule n'est pas respectée. Source : Nasa Astrobiology Institute : <http://astrobiology.nasa.gov/nai/>

Le comportement des vésicules est régi par l'énergie de courbure, aussi

appelée énergie de Helfrich, suivante :

$$E_c = \frac{\kappa_H}{2} \int_{\Gamma} H^2 d\sigma \quad (1.2)$$

où H désigne la courbure moyenne de la surface, κ_H le module de Helfrich et Γ la frontière de la membrane.

Dans cette thèse, on désignera par H la courbure moyenne et par κ la trace de l'endomorphisme de Weingarten, c'est-à-dire la somme des courbures principales. Dans un espace de dimension N , on a alors :

$$\kappa = (N - 1)H \quad (1.3)$$

De plus, K désignera la courbure de Gauss.

À partir d'une configuration donnée, on peut déterminer la direction de descente du gradient de l'énergie (1.2) en exprimant ses premières variations. Le détail des calculs suivants se trouve dans le livre de Willmore [Wil96]. En dimension 3, les variations de la courbure moyenne dans la direction ψ s'écrivent :

$$\delta H(\psi) = \frac{1}{2} (\Delta_{\Gamma} \psi + \psi(4H^2 - 2K)) \quad (1.4)$$

où Δ_{Γ} désigne l'opérateur de Laplace-Beltrami.

Tandis que celles de l'élément de surface infinitésimal sont données par :

$$\delta(d\sigma)(\psi) = -2H\psi d\sigma \quad (1.5)$$

La différentielle de E_c dans la direction ψ s'exprime alors par :

$$\begin{aligned} \delta E_c(\psi) &= \int_{\Gamma} 2H [\delta H(\psi)] d\sigma + \int_{\Gamma} H^2 \delta(d\sigma)(\psi) \\ &= \int_{\Gamma} H [\Delta_{\Gamma} \psi + \psi(4H^2 - 2K)] d\sigma - 2 \int_{\Gamma} H^2 \psi H d\sigma \end{aligned} \quad (1.6)$$

Par intégration par parties, on a $\int H \Delta_{\Gamma} \psi d\sigma = \int \psi \Delta_{\Gamma} H d\sigma$. Ainsi :

$$\delta E_c(\psi) = \int_{\Gamma} (\Delta_{\Gamma} H + 2H(H^2 - K)) \psi d\sigma \quad (1.7)$$

L'équation (1.7) donne la direction du gradient de l'énergie. Un calcul similaire donne le résultat suivant en dimension 2 :

$$\delta E_c(\psi) = \int_{\Gamma} \left(\Delta_{\Gamma} H + \frac{H^3}{2} \right) \psi ds \quad (1.8)$$

Ainsi, si l'on cherche à minimiser l'énergie, il convient de déplacer l'interface dans la direction :

$$W = \begin{cases} \Delta_{\Gamma} H + \frac{H^3}{2} & \text{en dimension 2} \\ \Delta_{\Gamma} H + 2H(H^2 - K) & \text{en dimension 3} \end{cases} \quad (1.9)$$

On conclut cette section en donnant les résultats pour les trois cas particuliers en dimension 3 qui nous intéressent :

- Sur les surfaces compactes sans bords de genre 0.
- Sur les surfaces compactes sans bords de genre 1.
- Sur les surfaces compactes sans bords de genre 0 avec conservation de volume et d'aire.

On rappelle que le genre d'une surface connexe est le nombre maximum de courbe fermées simples disjointes que l'on peut tracer à l'intérieur de la surface sans lui faire perdre son caractère connexe. Cette notion définit le concept intuitif de "trou" topologique.

Parmi les objets qui nous intéressent, les sphères et les ellipsoïdes sont des objets de genres 0, tandis que les tores sont des objets de genre 1.

Proposition 1.2.1. [Wil96] *Pour une surface compacte sans bord de genre 0, on a l'inégalité :*

$$\int_{\Gamma} H^2 d\sigma \geq 4\pi \quad (1.10)$$

Les seuls minimiseurs sont les sphères.

Théorème 1.2.2. [Wil96, MN14] *Pour une surface compacte sans bord de genre 1, on a l'inégalité :*

$$\int_{\Gamma} H^2 d\sigma \geq 2\pi^2 \quad (1.11)$$

Les seuls minimiseurs sont les tores de Clifford, c'est-à-dire les tores réguliers dont le ratio entre les deux axes est de $\sqrt{2}$.

La preuve que $2\pi^2$ est bien le minimum global sur les surfaces compactes de genre 1 n'a été établi qu'en 2012 dans [MN14]. Cependant, montrer que, parmi les tores réguliers, les tores de Clifford minimisent l'énergie d'Helfrich est plus aisé. En effet, l'énergie d'un tore régulier de rapport $\lambda > 1$ est donnée par :

$$E_c(T_\lambda) = \frac{\pi^2 \lambda^2}{\sqrt{\lambda^2 - 1}} \quad (1.12)$$

Le minimum est réalisé en $\lambda = \sqrt{2}$ pour le tore de Clifford.

Le cas des surfaces compactes sans bord de genre 0 avec conservation de volume et d'aire est le cas qui nous intéresse en particulier. En effet, il s'agit du cas des vésicules et des globules rouges. La forme bi-concave représentée dans

la littérature *Fig.1.1* est une forme minimale associée au volume réduit de la membrane. Pour un volume V et une aire A fixée, on définit le volume réduit ν , aussi appelé taux de remplissage ou taux de gonflement, par :

$$\nu = \begin{cases} \frac{4\pi V}{A^2} = \frac{V}{\pi \left[\frac{A}{2\pi}\right]^2} & \text{en dimension 2} \\ \frac{6\sqrt{\pi}V}{A^{\frac{3}{2}}} & \text{en dimension 3} \end{cases} \quad (1.13)$$

Cette quantité représente le ratio entre le volume de l'objet et le volume d'une sphère de même aire. Typiquement, le globule rouge humain a un taux de remplissage de 0.65. On donne à la figure 1.3 un exemple de forme d'équilibre pour un taux de remplissage d'environ 0.5.

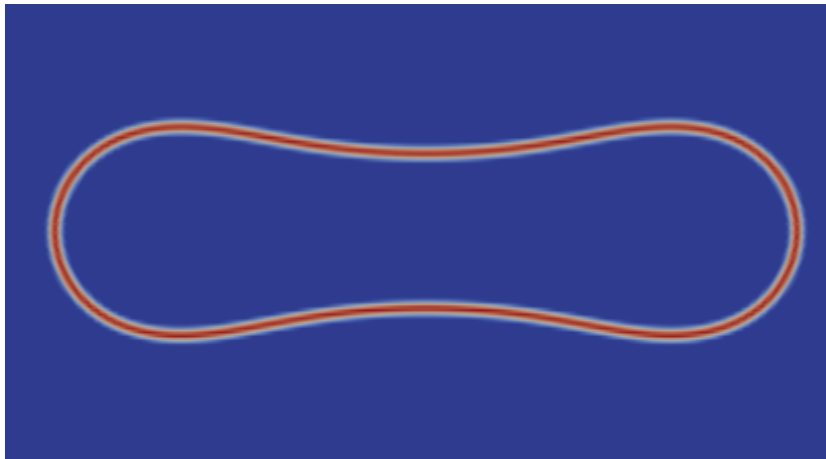


FIGURE 1.3 – Forme d'équilibre pour un globule en dimension 2. Le taux de remplissage est d'environ 0.5.

L'obtention des formes d'équilibre d'un globule rouge constitue un bon test pour vérifier le calcul du flot de Willmore avec conservation de volume et d'aire. Plusieurs méthodes existent pour résoudre ce problème, par exemple la méthode des intégrales de frontières et la méthode *Lattice-Boltzmann* [BRS⁺04, KHM11] ou encore la méthode des colliers de perles [ILL14]. Il s'agira du test de référence pour notre méthode que l'on proposera au Chapitre 3 où l'on comparera nos résultats avec ceux obtenus dans [KHM11].

1.3 Interaction Fluide-Structure

Les problèmes d'interactions entre un fluide et une structure relèvent d'une grande importance dans plusieurs domaines tels que la biomécanique (globules rouges dans le plasma, interaction du sang avec les valves cardiaques, les déplacements de micro-nageurs ou de cellules cancéreuses), l'aérodynamique, la

sédimentation de particules. La simulation numérique de ces différents phénomènes constitue une part essentielle pour leur compréhension.

La difficulté principale des problèmes d'interactions fluide-structure provient de la différence de représentation des deux milieux. En effet, il est naturel de représenter le fluide de manière eulérienne, c'est-à-dire qu'on caractérise son comportement en calculant sa vitesse en tout point du domaine. Plus précisément, on se fixe à chaque point du maillage et on observe au cours du temps l'évolution d'une quantité physique du fluide, le plus souvent sa vitesse. Comme on ne suit pas le déplacement de points physiques particuliers, il n'y a pas de précaution particulière, telle que le remaillage, à prendre entre les différentes itérations.

Pour la structure, il est naturel d'utiliser une représentation lagrangienne, c'est-à-dire que l'on définit des points physiques du solide dont on suit les déplacements au cours du temps. On suit la position de la particule positionnée initialement en x_0 à l'aide de la fonction $X(x_0, t)$ se déplaçant à une vitesse u et satisfaisant l'équation :

$$\frac{\partial X}{\partial t}(x_0, t) = u(X(x_0, t), t) \quad (1.14)$$

Numériquement, cela signifie que les sommets du maillage correspondent à des points physiques qui se déplacent. Ainsi, il est nécessaire que le maillage suive les déplacements du solide. Dès lors, cela implique *a minima* une déformation du maillage afin que les points du maillage suivent les points physiques de la structure. Lorsque ces déformations deviennent trop importantes, il devient nécessaire d'effectuer un remaillage. Cela permet de suivre avec précision les déplacements et éventuelles déformations du solide et, dans le cas d'un couplage, une localisation précise de l'interface. Cependant, de grandes déformations peuvent faire que les points du maillage s'accumulent ou s'isolent sur une partie du domaine. Il est alors nécessaire d'optimiser les méthodes numériques pour gérer ces cas de figure.

Pour un problème de couplage fluide-structure, la première question est le choix de la méthode de représentation du problème étudié. Il existe trois manières générales d'aborder cette problématique. On peut choisir d'adopter une représentation uniforme purement lagrangienne ou purement eulérienne. La difficulté dans ces cas est alors de gérer la représentation non naturelle d'un des deux milieux. L'autre approche est de conserver une représentation eulérienne/lagrangienne, mais le couplage des deux milieux n'est pas pris en compte par la représentation et un compromis doit être mis en place manuellement.

On propose dans ce qui suit un exposé des principales méthodes existantes pour chaque représentation avant de s'intéresser plus en détail aux méthodes purement eulériennes puisqu'il s'agit du choix que nous adopterons dans cette thèse.

1.3.1 Méthodes purement lagrangiennes

Les méthodes purement lagrangiennes consistent essentiellement à suivre le déplacement des points physiques des particules solides et fluides. Cette représentation n'est pas très naturelle pour un fluide en écoulement continu car il y a un renouvellement constant des particules et pas de point physique particulier à suivre.

Les deux méthodes lagrangiennes/lagrangiennes les plus utilisées sont la méthode dite "Smoother particle hydrodynamics" et la méthode dite de "Vortex".

La méthode "Smoothed particle hydrodynamics" ou plus succinctement SPH modélise le milieu à l'aide d'une suite de particules qui possèdent les caractéristiques physiques du fluide. Elle a été introduite par Lucy dans [Luc77] et par Gingold et Monaghan dans [GM77]. Le déplacement et les caractéristiques physiques de chaque particule sont obtenus à partir des équations physiques du mouvement et sont calculés par des moyennes pondérées des valeurs aux particules avoisinantes. Chaque particule contient ainsi les informations de la moyenne pondérée sur un volume de masse fixée, ce qui est particulièrement pertinent lors de la simulation du déplacement de gaz.

Cette méthode ne requiert pas l'utilisation d'un maillage pour discrétiser le domaine puisque les seules valeurs calculées sont les valeurs pour chaque particule.

Par exemple, la discrétisation de la valeur de la densité ρ_i de la particule i et l'équation de conservation de la masse s'expriment :

$$\begin{cases} \rho_i = \sum_{j=1}^N m_j W_{ij} \\ \frac{d\rho_i}{dt} = \sum_{j=1}^N m_j u_{ij} \nabla_i W_{ij} \end{cases} \quad (1.15)$$

où m_j désigne la masse de la particule j , u_{ij} la vitesse du fluide entre les particules i et j et W_{ij} est une fonction de poids pénalisant la distance entre les deux particules i et j .

Il convient de noter que l'on a remplacé par cette méthode une équation aux dérivées partielles par un système d'équations différentielles ordinaires.

La discrétisation précédente constitue un exemple concret de la méthode SPH. La discrétisation de l'équation de Navier-Stokes requiert plus d'attention et dépend des contraintes imposées aux paramètres par l'application physique et le phénomène considéré. On se réfère à [WCW⁺16] pour une synthèse sur les différentes extensions et applications existantes de la méthode SPH.

Cette méthode a été adaptée pour un fluide incompressible à surface libre par Monaghan dans [Mon94]. L'incompressibilité est assurée par un gradient de pression évalué à chaque particule ainsi qu'une force exprimant les variations locales de densité pour chaque particule. Des extensions adaptées pour le

couplage fluide/fluide [CL03] et fluide/structure [AGS07] étendent la méthode pour le couplage.

On peut citer plusieurs avantages de la méthode SPH. Du fait de l'absence de maillage, elle est bien adaptée aux problèmes à frontière libre et son implémentation est conceptuellement simple. Elle est particulièrement adaptée aux problèmes compressibles, car significativement moins coûteuse qu'une méthode faisant intervenir un maillage, et l'interface est bien représentée car celle-ci suit le déplacement des particules situées à la frontière des deux milieux. Elle gère efficacement les larges déformations. D'un point de vue computationnel, elle est cependant plus coûteuse que les autres méthodes que l'on va présenter dans la suite dans le cas incompressible. En effet, l'incompressibilité n'est pas pris en compte par le modèle et doit être imposée manuellement.

Dans le cas de fluides incompressibles, la méthode Vortex est plus adaptée que la méthode SPH car elle impose la contrainte d'incompressibilité directement dans sa formulation. Elle consiste à résoudre l'équation de Helmholtz plutôt que l'équation de Navier-Stokes, c'est-à-dire l'équation de transport de la vorticit   $\omega = \nabla \times u$.

Par exemple, pour l'équation de Navier-Stokes incompressible suivante :

$$\begin{cases} \rho (\partial_t u + u \cdot \nabla u) - \mu \Delta u + \nabla p = 0 \\ \operatorname{div} u = 0 \end{cases} \quad (1.16)$$

l'équation de Helmholtz associ  e est :

$$\begin{cases} \rho (\partial_t \omega + u \cdot \nabla \omega) - (\omega \cdot \nabla) u - \mu \Delta \omega = 0 \\ \operatorname{div} u = 0 \end{cases} \quad (1.17)$$

L'ensemble des marqueurs Lagrangiens utilis  s pour la r  solution num  rique de la m  thode Vortex sont localis  s aux r  gions o   ω est non nul, ce qui peut se r  v  ler tr  s avantageux dans certaines situations par rapport    la m  thode SPH. Cependant, le traitement des conditions aux bords est le d  faut majeur d'une m  thode de type Vortex, car il est n  cessaire de reformuler les conditions aux bords de la vitesse en conditions sur la vorticit  .

Comme on l'a d  j   mentionn  , il est n  cessaire de remailler si toutes les particules lagrangiennes se regroupent lors de grandes d  formations. On se r  f  re au livre de Cottet et Koumoutsakos [CKP⁺00] pour une r  f  rence compl  te sur la m  thode Vortex. On citera   galement les travaux [CCB⁺08, Kou05, RBCK10, MGCM15] pour des r  f  rences plus r  centes sur la m  thode.

1.3.2 M  thodes hybrides Eul  riennes/Lagrangiennes

La m  thode hybride la plus r  pandue pour les probl  mes de couplage fluide-structure est la m  thode "Arbitrary Lagrangian Eulerian" ou ALE, introduite par Donea dans [DGH82] et pass  e en revue dans [DHPRF04]. Il convient de citer

les travaux de Grandmont et al. dans le début des années 2000 [GGM01, FGG01, FGG07]. Il s'agit d'une méthode hybride où l'on combine les descriptions lagrangienne et eulérienne en utilisant un maillage mobile qui suit le déplacement normal de l'interface. Intuitivement, un maillage à vitesse nulle correspond à la représentation eulérienne et un maillage à vitesse égale à celle du solide correspond à la représentation lagrangienne. L'intérêt de cette méthode est de pouvoir suivre les déplacements de l'interface et d'avoir à l'intérieur une vitesse de déformation qui ne suit pas nécessairement la vitesse du fluide.

L'ingrédient principal de la méthode ALE est la construction d'une carte ALE qui décrit le déplacement du maillage initial Ω^* dans le temps, c'est-à-dire l'application qui donne à chaque point x^* du maillage Ω^* associée sa position à l'instant t dans le maillage déplacé Ω^t :

$$\begin{aligned} \mathcal{A}^T : \Omega^* &\rightarrow \Omega^t \\ x^* &\mapsto x(x^*, t) = \mathcal{A}^t(x^*) \end{aligned} \quad (1.18)$$

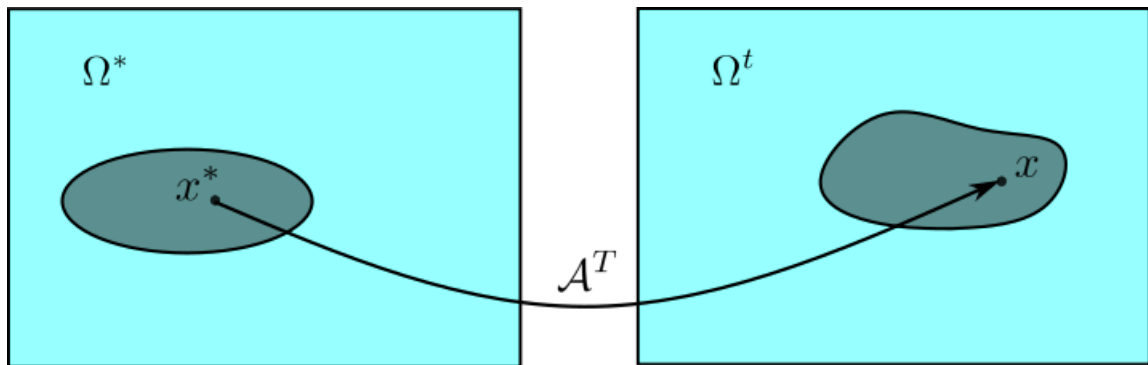


FIGURE 1.4 – Représentation de la carte ALE \mathcal{A}^T avec le domaine de référence Ω^* et le domaine déformé Ω^t .

L'intérêt de la carte ALE est qu'elle permet de ramener tous les calculs fluides à des calculs sur le domaine de référence Ω_f^* . Lorsque les déformations deviennent trop importantes, il est parfois nécessaire de remailler le domaine. Le nouveau domaine de référence devient alors le domaine obtenu à la fin du remaillage..

Les équations fluide et solide sont résolues séparément et les conditions de continuité sur la vitesse à l'interface ainsi que sur le tenseur des déformations sont résolues explicitement sur l'interface.

Des méthodes de constructions de carte ALE sont présentées dans [PPQ12, Cha13] utilisant l'opérateur d'extension harmonique ou les équations de Winslow.

La principale faiblesse de la méthode ALE est sa difficulté d'implémentation, notamment en dimension 3. Il est particulièrement délicat de gérer les grandes déformations et les changements topologiques. Ces événements impliquent

notamment un remaillage et la création d'une nouvelle carte ALE, procédures très coûteuses en dimension 3. La gestion de l'effet de masse ajoutée [COP97] a longtemps été un problème des méthodes ALE mais a été résolu dans les travaux de Fernandez, Gerbeau et Grandmont [FGG01, FGG07, FGG06, FLV15, FMV15].

1.3.3 Méthodes purement Eulériennes

La troisième alternative est d'utiliser une représentation complètement Eulérienne du problème fluide/structure. L'atout majeur de ce choix est que l'on ne force pas les points du maillage à suivre l'interface solide et que l'on peut alors utiliser un maillage fixe. On évite alors les défauts des méthodes lagrangiennes que l'on a identifiés avec les précédentes méthodes. La majorité des méthodes purement eulériennes de couplage fluide-structure reposent sur le principe des frontières fictives, à savoir que le fluide est défini sur l'ensemble du domaine. Une contrainte est appliquée aux régions occupées par la structure pour rendre compte de sa présence. On présente ici plusieurs méthodes qui reposent sur ce principe. L'idée originale de ces méthodes vient de Hyman dans [Hym52] puis est développée par Sau'lev dans [Sau63].

Méthode de pénalisation

La méthode de pénalisation a été introduite dans [Cal94] et développée dans [ABF99] pour simuler le déplacement d'un fluide incompressible autour d'obstacles solides immobiles. En suivant le précepte de la méthode de domaine fictif, on considère que le fluide remplit le domaine, notamment les zones contenant les obstacles. Cela permet d'éviter d'avoir à adapter le maillage à la géométrie des structures. L'idée est alors de pénaliser le fluide dans les zones où il est absent en ajoutant à l'équation de Navier-Stokes le terme f suivant :

$$f = \frac{u}{K} \quad (1.19)$$

où K est le paramètre de perméabilité du système.

L'équation de Navier-Stokes devient alors :

$$\rho_f (\partial_t u + (u \cdot \nabla)u) - \mu_f \Delta u + \nabla p = f \quad (1.20)$$

Idéalement, K vaut $+\infty$ dans la région fluide pour que le terme f n'influe pas sur l'équation et tend vers 0 dans le milieu solide. Le milieu solide est alors assimilé à un milieu poreux avec une perméabilité très faible auquel on applique la loi de Darcy avec une vitesse proportionnelle à K .

Numériquement, on choisit f égale à 0 sur le domaine fluide et $\frac{u}{K}$ sur le domaine rigide avec $K \ll 1$. Il est souvent nécessaire de régulariser cette expression sur l'interface, ce qui est fait en régularisant la fonction indicatrice. L'analyse numérique de cette méthode est présentée dans [Mau09]. Dans [Pat01],

une extension à un objet solide Ω_1 se déplaçant à une certaine vitesse u_1 à été proposée. Le problème à résoudre est le suivant :

$$\begin{cases} \rho_f (\partial_t u + (u \cdot \nabla)u) - \mu_f \Delta u + \nabla p = \rho g + \lambda \mathbf{1}_{(u_1-u)} \\ \operatorname{div} u = 0 \\ u_1 = u_1^t + w_1 \times (x - x_1^g) \\ m_1 \frac{du_1^t}{dt} = F_1 \\ J_1 \frac{dw_1}{dt} = T_1 \end{cases} \quad (1.21)$$

où J_1 est la matrice d'inertie, x_1^g le centre de gravité, m_1 la masse, u_1^t et w_1 les vitesses de translation et angulaires. La force F_1 et le moment T_1 sont donnés par :

$$\begin{cases} F_1 = F_{ext} + \int_{\Gamma_1} \sigma n_1 & \text{sur } \Gamma_1 \\ T_1 = T_{ext} + \int_{\Gamma_1} (x - x_1^g) \times \sigma n_1 & \text{sur } \Gamma_1 \end{cases} \quad (1.22)$$

où σ dénote le tenseur des contraintes, n_1 la normale sortante à l'interface Γ_1 .

La méthode de pénalisation a été adaptée au cas de solides rigides immergés dans un fluide en utilisant une formulation Vortex dans [CC08].

Méthode des multiplicateurs de Lagrange

La méthode des multiplicateurs de Lagrange, introduite dans [GPHJ99] et [GPH+01], propose d'introduire des multiplicateurs de Lagrange pour imposer la contrainte de rigidité à l'intérieur du solide et les conditions de Dirichlet sur l'interface fluide-structure.

Dans [GPHJ99], la contrainte imposée est :

$$u = u_1^t + w_1 \times (x - x_1^g) \quad \text{sur } \Omega_1 \quad (1.23)$$

Une amélioration de cette approche a été proposée dans [SP05] et utilise la contrainte de rigidité :

$$\begin{cases} \operatorname{div} D(u) = 0 & \text{sur } \Omega_1 \\ (D(u)) \cdot n_1 = 0 & \text{sur } \Gamma_1 \end{cases} \quad (1.24)$$

où $D(u) = \frac{\nabla u + \nabla u^T}{2}$ désigne le tenseur des déformations.

Cela équivaut à :

$$D(u) = 0 \quad \text{sur } \Omega_1 \quad (1.25)$$

On assigne alors l'équation de la contrainte de rigidité (1.23) au multiplicateur de Lagrange λ :

$$\begin{cases} \operatorname{div} D(\lambda) = 0 & \text{sur } \Omega_1 \\ (D(\lambda)) \cdot n_1 = 0 & \text{sur } \Gamma_1 \end{cases} \quad (1.26)$$

On ajoute alors dans l'équation de Navier-Stokes un terme de force $f = \operatorname{div} D(\lambda)$ en plus de la contrainte sur l'interface (1.24).

La résolution numérique de ce problème proposée dans [SP05] consiste en deux étapes en suivant le principe des schémas de Chorin [Cho67]. D'abord on résout l'équation de Navier-Stokes sans le terme de force lié au multiplicateur de Lagrange λ pour obtenir une vitesse intermédiaire \tilde{u} à divergence nulle. Ensuite, on calcule les multiplicateurs de Lagrange et on impose la contrainte de rigidité *a posteriori* pour obtenir la vitesse u du fluide.

Il existe deux approches pour le calcul des multiplicateurs de Lagrange. La première consiste à ne représenter les multiplicateurs de Lagrange que sur l'interface de la structure, figure de gauche de 1.5. Il suffit alors de mailler la surface. Pour la seconde approche, les multiplicateurs de Lagrange sont distribués sur l'ensemble du domaine solide. Il est alors nécessaire de mailler l'interface et l'intérieur du solide, voir la figure de droite de 1.5. Le choix de la méthode dépend de l'application physique modélisée.

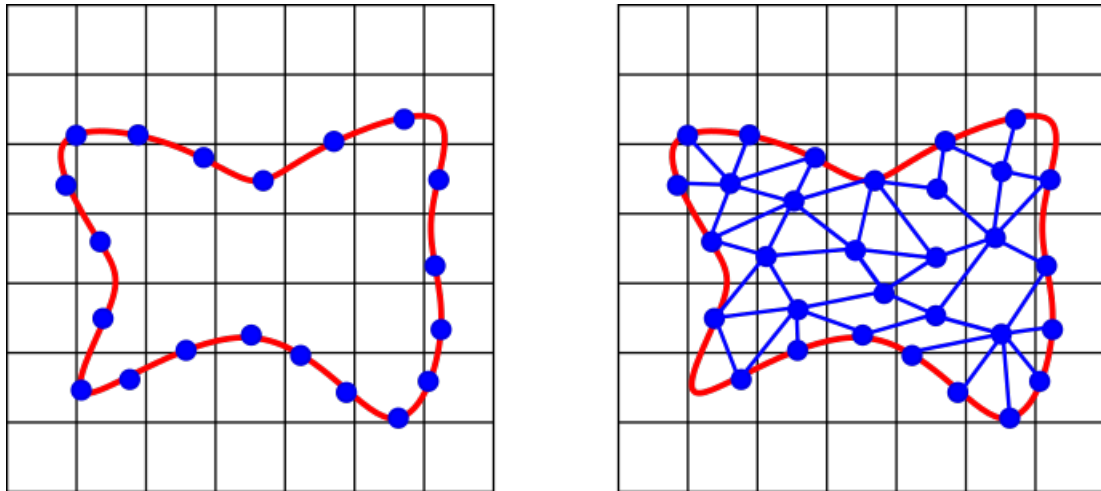


FIGURE 1.5 – Illustration de la méthode des multiplicateurs de Lagrange.

Méthode de frontière élargie

La méthode de frontière élargie a été introduite par Maury dans [Mau01] et adaptée par Ismail pour des problèmes elliptiques dans des domaines perforés [Ism04]. L'idée est de remplacer le problème initial en deux sous problèmes : un global et un local. Le problème global est résolu sur un maillage cartésien

fixe permettant l'utilisation des solveurs et pré-conditionneurs classiques. Le problème local est défini dans un voisinage des trous dans lequel on peut considérer un maillage fin pour mieux approcher la solution.

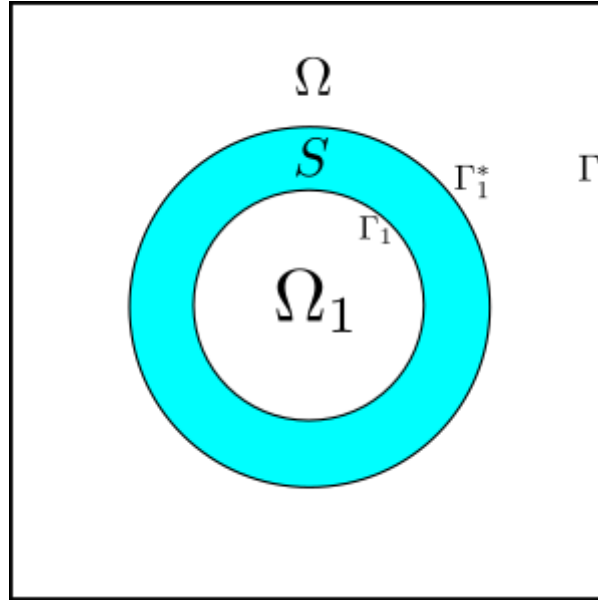


FIGURE 1.6 – Illustration de la répartition du domaine pour la méthode de la frontière élargie.

Le lien entre les résolutions locale et globale se fait par une interpolation d'un champ global sur l'interface artificielle Γ_1^* et une prescription d'un saut de la dérivée normale à travers Γ_1 .

Une analyse mathématique de la méthode a été proposée dans [BIM11] et de meilleurs résultats sont obtenus qu'avec la méthode de pénalisation H^1 car on impose la continuité des dérivées normales.

Pour fixer les idées, on présente l'exemple du problème de Poisson sur le domaine perforé donné par 1.6. La méthode de frontière élargie transforme le problème de Poisson suivant :

$$\begin{cases} -\Delta u = f \text{ dans } \Omega \setminus \Omega_1 \\ u = 0 \text{ sur } \Gamma \\ u = 0 \text{ sur } \Gamma_1 \end{cases} \quad (1.27)$$

en les deux sous-problèmes suivants :

$$\text{Problème Local : } \begin{cases} -\Delta v = f \text{ dans } S \\ v = u^* \text{ sur } \Gamma_1^* \end{cases} \quad (1.28)$$

$$\text{Problème Global : } \begin{cases} -\Delta u^* = f^* + \frac{\partial v}{\partial n} \delta_{\Gamma_1} \text{ dans } \Omega \\ u^* = 0 \text{ sur } \Gamma \end{cases} \quad (1.29)$$

où Γ_1 et Γ_1^* sont les frontières de la zone locale, f^* est l'extension de f par 0 sur Ω_1 . La condition aux limites sur Γ_1 est étendue à l'intérieur de la structure par le terme de saut $\partial_n v \delta_{\Gamma_1}$.

1.3.4 Méthodes de suivi d'interface et de capture d'interface pour les méthodes eulériennes.

Il existe deux approches classiques pour représenter l'interface : la méthode de suivi d'interface et la méthode de capture d'interface. On s'intéresse dans cette section à ces deux approches dans le cas d'une représentation eulérienne du couplage fluide-structure.

Méthodes de suivi d'interface et de suivi de volume

Il existe un grand nombre d'articles traitant de méthodes de suivi d'interface dans le contexte d'un maillage eulérien fixe. On citera les travaux de Glimm et ses collaborateurs [GGL⁺88, GGG⁺98, GGL⁺98], qui proposent notamment un package [DFG⁺06] permettant d'utiliser ces méthodes. On citera également les travaux de Tryggvason [TBE⁺01, UT92, TT09] centrés sur la simulation multi-fluidique incompressible et compressible. On se réfère à [Hym84] pour un aperçu des différentes méthodes existantes. La méthode de frontière immergée sera présentée plus tard, car elle sert de base au modèle *levelset* qui nous intéresse.

Avec la méthode de suivi d'interface, on représente l'interface explicitement à l'aide de marqueurs lagrangiens. Ces marqueurs sont initialisés sur l'interface et connectés entre eux pour former l'interface par une fonction linéaire par morceaux en 2D et une triangulation en 3D. Les marqueurs sont ensuite déplacés de manière lagrangienne. Plus précisément, si l'on note X_i la position du marqueur i , son évolution est décrite par :

$$\frac{dX_i}{dt} = u_i \quad (1.30)$$

où la vitesse u_i est la vitesse locale sur le maillage interpolée aux marqueurs lagrangiens.

Bien que le traitement de changements topologiques de l'interface soit une difficulté connue des méthodes de suivi d'interface, des algorithmes qui gèrent les intersections entre différentes parties de l'interface existent en dimension 2 depuis 1988 [GGL⁺88]. La gestion des bifurcations est gérée manuellement et l'extension à la dimension 3 est alors plus compliquée. En effet, en 2D, l'interface consiste uniquement en des points et des courbes. En dimension 3, l'interface contient également des surfaces et les bifurcations introduites par le mouvement des points (1.30) sont plus délicates à gérer qu'en dimension 2.

La courbure peut être calculée en approximant localement l'interface comme une surface quadratique :

$$z = Ax^2 + Bxy + Cy^2 + Dx + Ey \quad (1.31)$$

les coefficients étant obtenus à partir des 5 sommets les plus proches sur la surface. Le signe de la courbure est déterminé par l'orientation de la normale, la normale à un sommet étant interpolée à partir des normales aux faces voisines.

Certaines méthodes dites "Volume Tracking Methods" positionnent les marqueurs lagrangiens sur l'ensemble du domaine solide, cf *Fig.1.7* pour une représentation des deux approches. Parmi celles-ci, on trouve la méthode MAC "marker-and-cell" proposée par Harlow et Welch [HW65] pour simuler des flots incompressibles à surfaces libres. Comme les marqueurs localisent seulement une phase, il est nécessaire de reconstruire l'interface. La reconstruction peut être imprécise si certaines régions ont une faible densité de points.

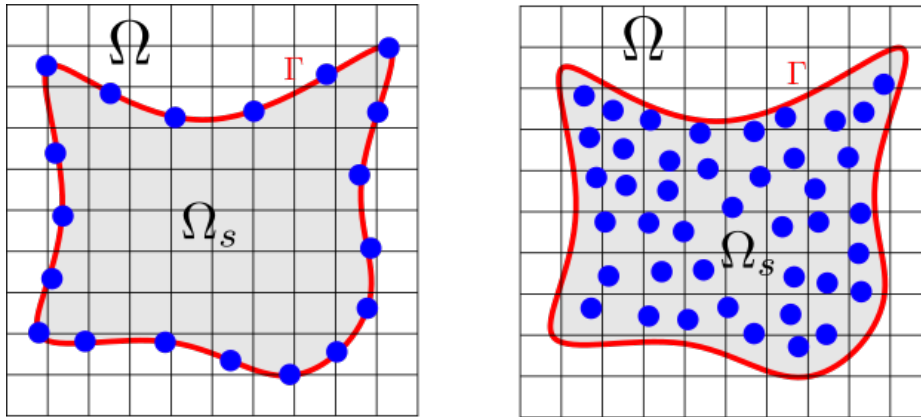


FIGURE 1.7 – Représentation des méthodes de suivi d'interface. Les points bleus correspondent aux marqueurs lagrangiens. À gauche, pour la méthode de suivi d'interface, ils sont situés sur l'interface. À droite, pour la méthode de suivi de volume, ils sont situés sur l'ensemble du domaine solide.

Dans la pratique, on ne positionne pas des marqueurs sur tout le domaine solide pour les méthodes de suivi de volume. On se contente de mettre des marqueurs sur une bande proche de l'interface. Cela permet de réduire le coût algorithmique mais peut engendrer des procédures supplémentaires en cas de grandes déformations.

En conclusion, les méthodes de suivi de volume rendent compte plus fidèlement des interactions entre les différentes phases que les méthodes de suivi d'interface. Cependant, elles impliquent un surcoût algorithmique car l'ensemble des marqueurs est de même dimension que l'espace. De plus, elles ne procurent pas d'informations sous-maillages car on doit reconstruire l'interface.

Méthodes de capture d'interface

L'approche alternative, appelée méthode de capture d'interface, consiste à localiser l'interface à partir des informations contenues dans une fonction scalaire. La champ de vitesse du fluide est calculé sur l'ensemble du maillage et la fonction scalaire est transportée par le fluide. Cette méthode gère naturellement les changements topologiques à travers la fonction scalaire. De plus, on peut obtenir des informations géométriques, typiquement la normale à l'interface et sa courbure, à partir de la fonction scalaire. Les trois approches les plus populaires sont la méthode *levelset*, la méthode champ de phase et la méthode *Volume of fluid*.

La méthode "Volume of fluid" décrit la fraction de fluide présente dans chacune des cellules du maillage, voir *Fig.1.8*. La fonction α vaut donc 0 dans le domaine solide, 1 dans le domaine fluide et une valeur intermédiaire sur l'interface. La fonction α peut être interprétée comme une fonction caractéristique discontinue du domaine fluide. Cette description est particulièrement adaptée à une méthode de différences finies, mais il existe des articles traitant de son implémentation sur des maillages non structurés [SZ03].

On transporte α par le champ de vitesse du fluide :

$$\partial_t \alpha + u \cdot \nabla \alpha = 0 \quad (1.32)$$

Pour une approche par différences finies, comme α est discontinue, les schémas standards risquent de diffuser l'interface. À l'instar des méthodes de suivi de volume, il est nécessaire de reconstruire l'interface pour obtenir des schémas satisfaisants. On citera [RR02, RK98, SZ03] pour différentes méthodes de reconstruction.

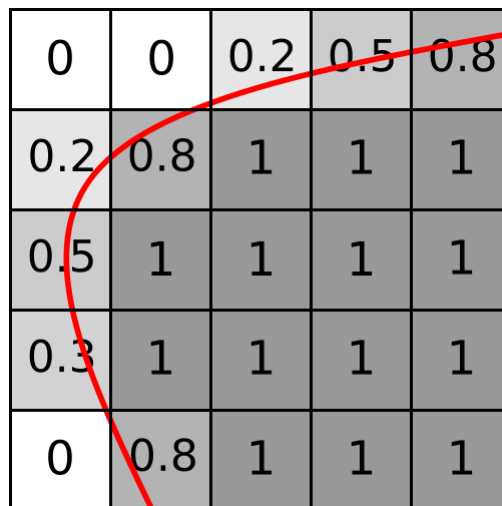


FIGURE 1.8 – Illustration de la méthode de volume de fluide. Chaque cellule contient la concentration de fluide qu'elle renferme. La courbe rouge représente l'interface.

L'avantage majeur de la méthode *Volume of fluid* est une très bonne conser-

vation de la masse. Son défaut principal est la faible précision des calculs de normales et de courbure à partir de α . Le caractère discontinu de α et la nécessité de reconstruire l'interface en sont les raisons.

La méthode couplée *levelset / Volume of fluid* propose de rectifier ce problème. Elle consiste à utiliser les deux approches simultanément afin de bénéficier des avantages de chacune. Elle a été introduite par Bourlioux [Bou95] et Sussman [Sus03]. L'idée est de calculer la fonction *levelset* ϕ , dont on verra les propriétés dans la suite, en plus de la fonction α . On peut alors calculer la normale et la courbure à l'interface avec plus de précision en utilisant ϕ et se servir de α pour la conservation de la masse. Au prix du surcoût algorithmique engendré par le calcul d'une fonction scalaire supplémentaire, on combine ainsi les avantages de chaque méthode. Cette méthode a été adaptée sur des maillages structurés [Sus03] et non structurés [YJL⁺06].

L'objet de la prochaine section est la présentation de la méthode *levelset* et du modèle que l'on utilisera dans cette thèse. On y présentera également la méthode champ de phase afin de les comparer.

L'intérêt de cette comparaison est double. D'une part, il s'agira de montrer que ces méthodes sont très proches conceptuellement. D'autre part, on étudie les méthodes de convolution-seuillage au Chapitre 2. Celles-ci ont initialement été introduites pour la méthode champ de phase et les fonctions indicatrices et nous en étudions une variante adaptée à la formulation *levelset* et les fonctions distances signées.

1.4 Méthode de frontière immergée eulérienne

Dans cette section, on présente le modèle de couplage fluide-structure que l'on va utiliser durant cette thèse. Il a été introduit par Cottet et Maitre en 2004 dans [CM04]. Il repose essentiellement sur une adaptation du modèle de frontière immergée de Peskin [Pes77] où l'on remplace la méthode de suivi d'interface qui nécessite l'utilisation de marqueurs lagrangiens par une méthode de capture d'interface. La méthode de capture d'interface utilisée est la méthode *levelset* introduite par Sethian et Osher dans [OS88], on se réfère à [OF06] pour un exposé complet. Plus précisément, l'interface est représentée implicitement comme la ligne de niveau 0 d'une fonction auxiliaire dont l'évolution suit une équation d'advection selon la vitesse du fluide. Le couplage est effectué comme celui de la méthode de frontière immergée, c'est-à-dire qu'un terme de force élastique est introduit au second membre de l'équation de Navier-Stokes.

On commence par introduire le modèle fluide sous sa formulation eulérienne naturelle. Ensuite, on présente la méthode de frontière immergée dont s'inspire le modèle. On présente alors les différentes propriétés analytiques et numériques de la méthode *levelset*. On conclut la section par une comparaison avec la méthode de champ de phase. L'intérêt de cette comparaison est double. D'une part, ces deux méthodes sont proches à la fois dans leur formulation et

dans leur concept. De plus, chacune présente des avantages et défauts distincts liés au choix de la fonction capturant l'interface. D'autre part, ce sont ces deux représentations dont on étudiera le mouvement par diffusion dans le deuxième chapitre.

1.4.1 Modèle fluide

On considère un domaine Ω_f dans lequel est plongé un fluide Newtonien, visqueux et incompressible. Il s'agit typiquement d'un liquide et notre cas d'application principal sera le plasma sanguin. On adopte les notations suivantes :

- $u(x, t)$ désigne le champ de vitesse du fluide.
- $p(x, t)$ désigne la pression.
- μ_f désigne la viscosité du fluide.
- ρ_f désigne la densité du fluide.

La contrainte d'incompressibilité est donnée par :

$$\operatorname{div} u = 0 \quad (1.33)$$

L'équation de conservation de la masse s'écrit, dans le cas où la viscosité n'est pas constante :

$$\partial_t \rho_f + u \cdot \nabla \rho_f = 0 \quad (1.34)$$

Le principe fondamental de la dynamique fournit l'équation régissant le comportement de u . Ce qui donne le système de Navier-Stokes incompressible :

$$\begin{cases} \rho_f (\partial_t u + (u \cdot \nabla)u) - \operatorname{div} 2\mu_f D(u) + \nabla p = F \\ \operatorname{div} u = 0 \end{cases} \quad (1.35)$$

où $D(u) = \frac{\nabla u + \nabla u^T}{2}$ est le gradient symétrique et F représente les forces extérieures appliquées au fluide. Il convient d'ajouter les conditions initiales et les conditions aux bords au système pour qu'il soit complet.

Pour un fluide de densité et viscosité constantes sans contraintes extérieures, le système peut être réécrit sous la forme adimensionnelle suivante :

$$\begin{cases} \partial_t u + (u \cdot \nabla)u - \frac{1}{Re} \Delta u + \nabla p = \frac{F}{\rho_f} \\ \operatorname{div} u = 0 \end{cases} \quad (1.36)$$

Le nombre de Reynolds Re est donné par :

$$Re = \frac{\rho_f L u_f}{\mu_f} \quad (1.37)$$

L désigne la longueur caractéristique de la géométrie du domaine, u_f désigne la vitesse caractéristique du fluide et μ_f désigne la viscosité du fluide. Le nombre de Reynolds permet d'identifier la nature de l'écoulement fluide, notamment son régime : turbulent, laminaire, transitoire. Essentiellement, lorsque $Re > 4000$, on parle d'un écoulement turbulent car le terme diffusif Δu a un effet négligeable par rapport au terme convectif $u \cdot \nabla u$. Au contraire, pour $Re < 2000$, on parle d'un écoulement laminaire et le terme diffusif domine la dynamique du fluide et les perturbations sont moindres.

Dans le cas de petits vaisseaux sanguins, le nombre de Reynolds est de l'ordre de 10^{-4} et on peut alors négliger les termes inertiels. On considère alors un système de Navier-Stokes quasi-statique, c'est-à-dire que l'on résout le système de Stokes suivant à chaque pas de temps :

$$\begin{cases} -\mu_f \Delta u + \nabla p = F \\ \operatorname{div} u = 0 \end{cases} \quad (1.38)$$

1.4.2 Physique d'une membrane élastique immergée

On considère que la surface est plongée dans un milieu fluide décrit par son champ de vitesse u . Chaque point de l'interface suit donc une trajectoire $x \mapsto \Psi(x, t)$ vérifiant l'équation :

$$\partial_t \Psi(x, t) = u(\Psi(x, t), t) \quad (1.39)$$

On considère que l'interface est décrite comme ligne de niveau 0 de la fonction ϕ

$$\Gamma(t) = \{x \in \mathbb{R}^3 / \phi(x, t) = 0\} \quad (1.40)$$

Soit $x \in \Gamma(0)$ un point de l'interface à l'instant initial. Son évolution temporelle est donc donnée par la trajectoire $\Psi(x, t)$ et on a alors la propriété suivante :

$$\phi(\Psi(x, t), t) = 0 \text{ et } \Gamma(t) = \{\Psi(x, t) / x \in \Gamma(0)\} \quad (1.41)$$

En dérivant la première équation par rapport à t , on obtient :

$$\partial_t \phi(\Psi(x, t), t) + u(\Psi(x, t), t) \cdot \nabla \phi(\Psi(x, t), t) = 0 \quad (1.42)$$

La fonction ϕ est décrite de manière eulérienne et est donc définie sur l'espace tout entier et on prolonge donc l'équation précédente sur l'espace tout entier :

$$\partial_t \phi + u \cdot \nabla \phi = 0 \quad (1.43)$$

Maintenant que l'on a donné les équations d'évolution des deux milieux, on s'intéresse à la méthode de couplage que l'on va utiliser dans notre modèle.

1.4.3 Méthode de frontière immergée

La méthode de frontière immergée, "Immersed Boundary Method" dans sa version originale, a été introduite par Peskin en 1977 dans [Pes77] pour la simulation des valves cardiaques. Le but de l'article est la modélisation d'un système fluide-structure où les structures sont des fibres uni-dimensionnelles agissant sur le fluide par des forces élastiques. Le couplage entre le fluide et la structure est réalisé à l'aide du principe des forces virtuelles.

On considère un maillage fixe pour représenter la région fluide Ω_f , remplie d'un fluide de vitesse u , de pression p et de densité ρ soumis au régime de Navier-Stokes :

$$\rho_f (\partial_t u + (u \cdot \nabla)u) - \mu_f \Delta u + \nabla p = F_e \quad (1.44)$$

où F_e désigne la force élastique des fibres agissant sur le fluide.

L'interface $\Gamma(t)$ est décrite par suivi d'interface et un ensemble de marqueurs Lagrangiens selon la paramétrisation $s \rightarrow X(s, t)$, pour $s \in [0, L]$. Le cas étudié impose que $\Gamma(t)$ soit un ensemble de courbes simples et fermées, chaque courbe correspondant à une fibre. Ces marqueurs se déplacent librement sur le maillage et sont interpolés avec les points du maillage pour le champ de vitesse.

Les forces élastiques F_e exercées sur le fluide apparaissent au second membre des équations de Navier-Stokes comme des masses de Dirac à support sur les fibres. L'expression des forces élastiques est dérivée du potentiel élastique suivant :

$$\mathcal{E}(X(s, t)) = \int_0^L E(|\partial_s X(s, t)|) ds \quad (1.45)$$

ce qui donne par dérivation l'expression de la force élastique :

$$\mathcal{F}_e(X(s, t), t) = \frac{1}{|\partial_s X(s, t)|} \frac{\partial}{\partial s} \left(E'(|\partial_s X(s, t)|) \frac{\partial_s X(s, t)}{|\partial_s X(s, t)|} \right) \quad (1.46)$$

Ces forces sont localisées sur les fibres à l'aide d'une masse de Dirac et le terme de force de l'équation de Navier-Stokes est finalement donné par :

$$F_e(x, t) = \mathcal{F}_e(x, t) \delta_{\Gamma(t)} \quad (1.47)$$

La méthode pour obtenir cette force élastique F_e est le principe des forces virtuelles.

Chaque fibre est transportée par le fluide et une interpolation est faite pour évaluer la vitesse des marqueurs sur le maillage. L'évolution de chaque point de l'interface vérifie :

$$\frac{\partial X(s, t)}{\partial t} = u(X(s, t), t) = \int_{\Omega} u(x, t) \delta(x - X(s, t)) ds \quad (1.48)$$

Le système décrivant l'intégralité du problème fluide-structure est alors donné par :

$$\begin{cases} \rho_f (\partial_t u + (u \cdot \nabla)u) - \mu_f \Delta u + \nabla p = F_e \\ F_e(x, t) = \int_0^L F(s, t) \delta(x - X(s, t)) ds \\ \frac{\partial X(s, t)}{\partial t} = u(X(s, t), t) = \int_{\Omega} u(x, t) \delta(x - X(s, t)) ds \end{cases} \quad (1.49)$$

L'atout majeur est le traitement extrêmement simple du couplage. En effet, il se fait naturellement en ajoutant un terme de force au second membre de l'équation de Navier-Stokes. Cependant, le déplacement des fibres est traité par une méthode de suivi de surface et cela nécessite l'introduction de marqueurs lagrangiens le long des courbes et d'interpolations aux points du maillage pour évaluer leurs vitesses. Cette interpolation est source d'erreurs qui peuvent mener à un passage de fluide à travers les fibres. De plus, les grandes déformations et les changements topologiques sont toujours sources de complications en ce qui concernent la gestion des marqueurs lagrangiens.

Différentes améliorations ont été proposées pour contrevenir à ces défauts, on citera par exemple [Bre12, GHS95, GHS93] ou à des cas plus généraux que celui de simples fibres [LC15, KBGD16]. On se réfère à [Pes02] et [MI05] pour des rapports complets sur la méthode de frontière immergée et les références vers les articles sur le sujet.

On présente dans ce qui suit le modèle de couplage fluide-structure que l'on utilisera dans cette thèse. Il a été introduit par Cottet et Maitre dans [CM04, CM06] et repose fortement sur une adaptation de la méthode de frontière immergée. L'idée est de remplacer l'approche de suivi d'interface utilisée dans le modèle de Peskin par une méthode de capture d'interface qui utilise la méthode *levelset* introduite par Osher et Sethian dans les années 1980. On se réfère à leur livre [OF06] et l'article [OF01] qui détaillent la plupart des résultats et des références de la méthode *levelset*. Essentiellement, on introduit une fonction auxiliaire ϕ dont la ligne de niveau 0 capture l'interface $\Gamma(t)$. Cette représentation implicite de l'interface permet de pallier aux inconvénients liés à l'utilisation des marqueurs lagrangiens.

L'atout majeur de la méthode est que l'on repose sur une formulation totalement eulérienne du problème. Ainsi, on n'a pas de problème de remaillage pour les grandes déformations qui sont traitées naturellement par la formulation. De plus, la gestion du couplage est faite simplement en suivant le principe de la méthode de frontière immergée.

1.4.4 Fonction *levelset*

On s'intéresse maintenant à la définition de notre fonction *levelset* qui va permettre de représenter le l'interface. La plupart des résultats concernant la fonc-

tion *levelset* que l'on va présenter se trouvent dans [OF06].

On considère un domaine Ω divisé en deux sous-domaines Ω_s et Ω_f séparés par une interface Γ . On appelle fonction distance signée à l'interface Γ , la fonction d vérifiant les propriétés suivantes :

$$\begin{cases} \forall x \in \Omega_s, d_\Gamma(x) = -\text{dist}(x, \Gamma) \\ \forall x \in \Gamma, d_\Gamma(x) = 0 \\ \forall x \in \Omega_f, d_\Gamma(x) = \text{dist}(x, \Gamma) \end{cases} \quad (1.50)$$

où $\text{dist}(x, \Gamma) = \min_{y \in \Gamma} \|x - y\|$ est la distance de x à l'interface.

La proposition suivante regroupe quelques propriétés classiques de la fonction distance.

Proposition 1.4.1. *On considère la fonction d_Γ distance signée à une interface Γ régulière (au moins de classe C^2). On a la relation suivante entre d_Γ et la projection orthogonale :*

$$d_\Gamma(x) = (x - P_\Gamma(x)) \cdot n(P_\Gamma(x)). \quad (1.51)$$

D'autre part, on a la propriété suivante :

$$|\nabla d_\Gamma| = 1 \quad (1.52)$$

De cette relation, on déduit : $\nabla_n d_\Gamma = \nabla \nabla d_\Gamma \cdot \nabla d_\Gamma = 0$.

De plus, on peut exprimer la normale et la courbure moyenne en fonction de d_Γ :

$$\begin{cases} n(x) = \nabla(d_\Gamma(x)) \\ \Delta d_\Gamma(x) = \kappa(x) = (N - 1)H(x). \end{cases} \quad (1.53)$$

Idéalement, on aimerait choisir comme fonction *levelset* la fonction distance. Cependant, la condition $|\nabla \phi| = 1$ n'est pas conservée par l'équation de transport. On choisit donc une fonction *levelset* de la forme :

$$\begin{cases} \forall x \in \Omega_s, \phi(x) < 0 \\ \forall x \in \Gamma, \phi(x) = 0 \\ \forall x \in \Omega_f, \phi(x) > 0 \end{cases} \quad (1.54)$$

Cette formulation permet de localiser l'interface Γ comme la ligne de niveau 0 de la fonction *levelset* ϕ mais également à séparer les deux phases Ω_s, Ω_f à l'aide de son signe.

D'autre part, contrairement à la fonction indicatrice, on va voir comment la fonction *levelset* contient des informations supplémentaires sur l'étirement de la structure. On notera également que la fonction *levelset* peut fournir des informations à une précision plus fine que celle du maillage.

Le choix du signe pour ϕ est arbitraire et on adopte la convention suivante : le domaine Ω_s sur lequel ϕ est négative désignera l'intérieur de la structure et le domaine Ω_f désignera le fluide.

On peut définir la fonction de Heaviside suivante pour localiser la région fluide :

$$\mathcal{H}(\phi) = \begin{cases} 0 & \text{pour } \phi < 0 \\ \frac{1}{2} & \text{pour } \phi = 0 \\ 1 & \text{pour } \phi > 0 \end{cases} \quad (1.55)$$

En pratique, on ne peut pas travailler avec une interface d'épaisseur nulle. On introduit alors un paramètre ε qui décrit l'épaisseur numérique de l'interface, cette idée ayant été introduite par Peskin dans [Pes77]. Le paramètre ε est généralement pris de l'ordre de $1.5h$ à $3h$, où h désigne la taille d'un élément du maillage. On se réfère à [UT92] pour la justification de la régularisation suivante :

$$\mathcal{H}_\varepsilon(\phi) = \begin{cases} 0 & \text{pour } \phi \leq -\varepsilon \\ \frac{1}{2} \left(1 + \frac{\phi}{\varepsilon} + \frac{\sin\left(\frac{\pi\phi}{\varepsilon}\right)}{\pi} \right) & \text{pour } |\phi| \leq \varepsilon \\ 1 & \text{pour } \phi > \varepsilon \end{cases} \quad (1.56)$$

La fonction Heaviside est utilisée pour définir des quantités ayant des valeurs différentes sur les différents domaines. Par exemple, dans le cas d'une simulation d'un flot diphasique, on définit la densité et la viscosité des deux fluides de la façon suivante :

$$\begin{cases} \rho = \rho_s + (\rho_f - \rho_s) \mathcal{H}_\varepsilon(\phi) \\ \mu = \mu_s + (\mu_f - \mu_s) \mathcal{H}_\varepsilon(\phi) \end{cases} \quad (1.57)$$

On peut également définir la fonction Dirac régularisée qui permet de localiser l'interface :

$$\delta_\varepsilon(\phi) = \begin{cases} 0 & \text{pour } \phi < -\varepsilon \\ \frac{1}{2\varepsilon} \left(1 + \cos\left(\frac{\pi\phi}{\varepsilon}\right) \right) & \text{pour } \phi \in [-\varepsilon; \varepsilon] \\ 0 & \text{pour } \phi > \varepsilon \end{cases} \quad (1.58)$$

Le choix de cette fonction Dirac est justifié par la propriété suivante :

Proposition 1.4.2. [Proposition 2.2] de [CM06] Pour une fonction ζ de classe C^∞ à support dans $[-1; 1]$ telle que $r \rightarrow \frac{1}{\varepsilon} \zeta\left(\frac{r}{\varepsilon}\right)$ converge vers le Dirac δ_0 en 0 lorsque $\varepsilon \rightarrow 0$ dans $\mathcal{D}'(\mathbb{R})$. Alors, pour ϕ suffisamment régulière telle que $|\nabla\phi| \geq \alpha_0 > 0$, on a lorsque $\varepsilon \rightarrow 0$:

$$\frac{1}{\varepsilon} \zeta\left(\frac{\phi}{\varepsilon}\right) |\nabla\phi| \rightarrow \delta_{\{\phi=0\}} \text{ dans } \mathcal{M}(\Omega) \quad (1.59)$$

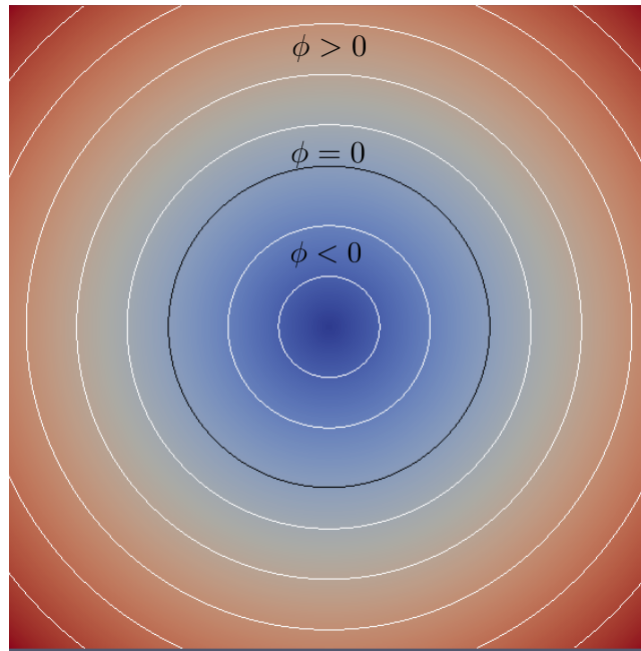


FIGURE 1.9 – Représentation de la fonction distance signée. L'interface est la ligne de niveau 0 de ϕ représentée en noire.

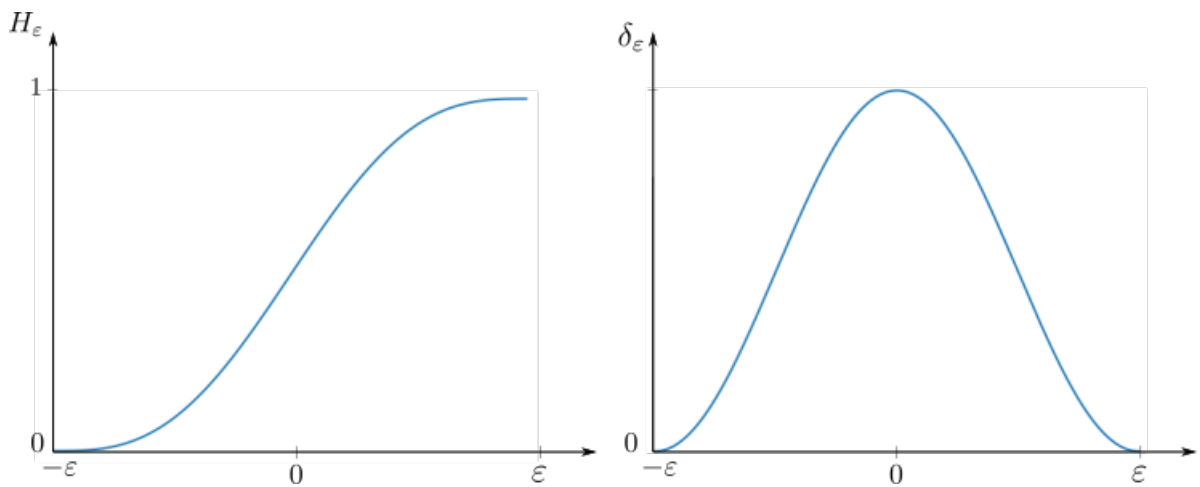


FIGURE 1.10 – Représentation des fonctions d'Heaviside et de Dirac régularisée.

Un atout majeur de la fonction *levelset* est que l'on peut facilement exprimer certaines quantités géométriques telles que la normale à la courbe ainsi que la courbure moyenne à l'aide de la *levelset* ϕ :

$$\begin{cases} n(\phi) = \frac{\nabla\phi}{|\nabla\phi|} \\ H(\phi) = \text{div} \frac{\nabla\phi}{|\nabla\phi|} \end{cases} \quad (1.60)$$

1.4.5 Équation de transport

Comme on l'a vu à la section 1.4.2, l'évolution de l'interface Γ est basée sur le transport de la fonction *levelset* ϕ par le champ de vitesse du fluide :

$$\begin{cases} \partial_t \phi + u \cdot \nabla \phi = 0 \\ \phi(\cdot, 0) = \phi_0 \end{cases} \quad (1.61)$$

Pour une vitesse u sans condition particulière, la solution peut avoir une régularité plus faible que la condition initiale ϕ_0 . De plus, si ϕ_0 est une fonction distance signée, la norme du gradient n'est pas conservée et il en est ainsi de même de la condition $|\nabla \phi| = 1$. En effet, $|\nabla \phi|$ vérifie l'équation suivante :

$$\partial_t |\nabla \phi| + u \cdot \nabla |\nabla \phi| = -|\nabla \phi| \frac{\nabla \phi \otimes \nabla \phi}{|\nabla \phi|^2} : \nabla u \quad (1.62)$$

Puisque l'on part d'une fonction distance, il faudrait que l'on ait la condition $|\nabla \phi| \frac{\nabla \phi \otimes \nabla \phi}{|\nabla \phi|^2} : \nabla u = 0$ à tout temps pour $|\nabla \phi|$ que soit conservé, ce qui n'est pas en général le cas. En fait, il s'agit de comprendre que la quantité $|\nabla \phi|$ mesure l'étirement de la membrane. Cette quantité est très importante car elle sert dans l'expression des forces élastiques et de courbure.

1.4.6 Réinitialisation de la fonction *levelset*

D'après (1.62), la norme du gradient de ϕ n'est pas conservée et décrit l'étirement de la membrane. Cependant, cela peut poser problème si sa valeur s'éloigne trop de 1. En effet, au cours des simulations, des erreurs numériques peuvent survenir si $|\nabla \phi|$ subit de fortes variations. Ces erreurs peuvent avoir des conséquences importantes notamment lors de la localisation d'interface par la masse de Dirac δ_ε . On cherche alors à maintenir $|\nabla \phi|$ près de 1 en procédant lorsque nécessaire à une étape dite de réinitialisation ou de renormalisation. L'étape de réinitialisation consiste à remplacer ϕ par la fonction distance signée à la ligne de niveau $\{\phi = 0\}$ actuelle. On présente dans ce qui suit succinctement différentes méthodes existantes pour réaliser cette étape de réinitialisation.

Renormalisation de la fonction *levelset*

Une méthode peu coûteuse proposée dans [CM06] consiste à remplacer ϕ par $\frac{\phi}{|\nabla \phi|}$. L'expression $\frac{\phi}{|\nabla \phi|}$ fournit une bonne approximation au premier ordre de la fonction distance au voisinage de l'interface, ce qui est particulièrement efficace pour calculer H_ε et δ_ε , voir [CM04] et [MMC⁺09] pour des applications numériques et des comparaisons avec la méthode d'Hamilton-Jacobi présentée dans la suite.

Cette méthode n'est pas utilisable si l'on a besoin des valeurs de ϕ loin de l'interface ou si la vitesse est trop irrégulière. En effet, dans ces cas de figure, la

valeur de $|\nabla\phi|$ peut énormément varier et prendre des valeurs éloignées de 1 et engendrer les problèmes suivants :

- Si $|\nabla\phi|$ est très faible, une division par ce terme peut engendrer des erreurs numériques significatives.
- Si $|\nabla\phi|$ est très élevé, le terme $\frac{\phi}{|\nabla\phi|}$ approche 0 et une certaine zone peut abusivement être traitée comme faisant partie de l'interface.

Il est alors nécessaire d'employer des méthodes plus coûteuses et plus précises.

Résolution d'une équation d'Hamilton-Jacobi

Dans [SSO94], les auteurs réinitialisent la fonction *levelset* en imposant au gradient de prendre la valeur 1. Cela est fait en résolvant l'équation suivante, dite de Hamilton-Jacobi ou équation Eikonale :

$$\partial_t\phi + \text{sign}(\phi_0)(|\nabla\phi| - 1) = 0 \quad (1.63)$$

En effet, $|\nabla\phi|$ va converger vers 1 lorsque l'on s'approche de l'état stationnaire où $\partial_t\phi = 0$. La fonction signe permet de propager le front dans deux directions opposées, de part et d'autre de l'interface.

L'équation (1.63) est une équation de transport à la vitesse $\text{sign}(\phi_0)\frac{\nabla\phi}{|\nabla\phi|}$ et se résout de la même manière que (1.61). Elle peut être formulée sous la forme suivante :

$$\partial_t\phi + \text{sign}(\phi_0)\frac{\nabla\phi}{|\nabla\phi|} \cdot \nabla\phi = \text{sign}(\phi_0) \quad (1.64)$$

On transporte alors ϕ dans la direction de la normale en imposant la condition $|\nabla\phi| = 1$ avec le terme source $\text{sign}(\phi_0)$. La condition $|\nabla\phi| = 1$ est propagée depuis l'interface à la vitesse 1 dans la direction normale et il est possible d'avoir une idée du nombre d'itérations en temps nécessaire pour imposer la condition sur un voisinage de l'interface d'épaisseur fixée.

Dans [SSO94], on montre que la fonction signe doit être régularisée pour des raisons numériques :

$$\text{sign}_{\varepsilon_s}(\phi) = \frac{\phi}{\sqrt{\phi^2 + \varepsilon_s^2}} \quad (1.65)$$

Le paramètre de régularisation ε_s doit lui aussi être choisi avec précaution.

Dans [EFFM02], les auteurs observent la perte complète de l'interface dans le cas d'un flot en cisaillement important et globalement une perte de masse significative survient lors de longues simulations. On se réfère à la thèse de Paul Vigneaux [Vig07] pour une étude plus précise de cette méthode ainsi qu'une description en détail des améliorations existantes.

Une idée proposée dans [GF00] est de remplacer l'équation d'advection (1.61) par l'équation suivante :

$$\partial_t \phi(x, t) + u \cdot \nabla \phi(x - \phi \nabla \phi, t) = 0 \quad (1.66)$$

Cependant l'équation devient alors non locale et cette méthode n'est pas utilisable dans le cas du couplage fluide-structure.

Dans [LXS17], les auteurs proposent de résoudre deux équations de type Hamilton-Jacobi supplémentaires afin de conserver localement près de l'interface la valeur de $|\nabla \phi|$ afin de conserver la valeur de l'étirement.

Méthode de fast marching

La méthode de *fast-marching* a été introduite en 1996 par Sethian dans [Set96] et une étude plus détaillée se trouve dans [Set99]. L'idée est de partir des valeurs proches de l'interface initialisée correctement et de propager le front sur tout le domaine de proche en proche en suivant le principe de l'algorithme de Dijkstra.

Plus précisément, on distingue chaque sommet du maillage en trois catégories :

- Traité (T) : les sommets dont la valeur de ϕ redistanciée est connue.
- Voisin (V) : les points dont la valeur redistanciée est inconnue et dont un voisin est "Alive".
- Lointain (L) : le reste des sommets du domaines qui sont des sommets à traiter.

À chaque itération, on choisit le sommet de l'ensemble "Voisin" dont la distance à l'interface est minimale. On lui assigne cette valeur et on le place dans l'ensemble "Traité". On ajoute alors ses éventuels voisins de "Lointain" dans la "Voisin" et on actualise la valeur de la fonction ϕ des sommets voisins de la "Voisin". On continue à itérer jusqu'à ce que tous les sommets soient dans la catégorie "Traité" et la fonction ϕ ainsi obtenue est une fonction distance signée (la gestion du signe est à prendre en compte lors du calcul des distances).

On initialise l'algorithme au niveau de l'interface en prenant comme valeur $\frac{\phi}{|\nabla \phi|}$ pour les sommets que l'on marque comme "Traité". On rappelle que cela représente une bonne approximation de la distance sur le voisinage de l'interface.

La complexité de la *fast-marching* est en $O(n \log n(n))$ pour n le nombre de sommets du maillage. La principale difficulté repose sur la gestion des voisins, notamment lorsqu'il faut trouver le minimum. On se réfère à [Vig07] pour une description détaillée d'une implémentation optimisée de l'algorithme.

Dans le cadre d'une résolution par éléments finis, l'algorithme que l'on a décrit ne fonctionne que pour des éléments de Lagrange d'ordre 1. Une solution pour les ordres supérieurs est de considérer le maillage iso- \mathbb{P}^1 associé, c'est-à-dire

Algorithme 1 : Algorithme de Fast-Marching. Étant donnée une fonction *levelset* ϕ , on forme la fonction distance signée à la ligne de niveau $\{\phi = 0\}$.

Initialisation :

- 1) Mettre tous les sommets de la bande d'interface dans "Traité" avec comme valeur $\frac{\phi}{|\nabla\phi|}$.
- 2) Ajouter les sommets voisins dans "Voisin".
- 3) Ajouter tous les autres sommets dans "Lointain".

Tant que "Voisin" n'est pas vide :

- 1) Trouver x tel que $\phi(x) = \min_{NB} \phi(y)$.
 - 2) Déplacer x vers "Traité" et actualiser les valeurs des voisins de x .
 - 3) Déplacer les voisins de x présents dans "Lointain" dans "Voisin".
-

un maillage \mathbb{P}^1 dont les sommets sont les degrés de liberté du maillage d'ordre supérieur.

La technique à adopter est alors de projeter le maillage d'ordre supérieur sur le maillage iso- \mathbb{P}^1 associé. On effectue la *fast-marching*, puis on projette à nouveau sur le maillage initial.

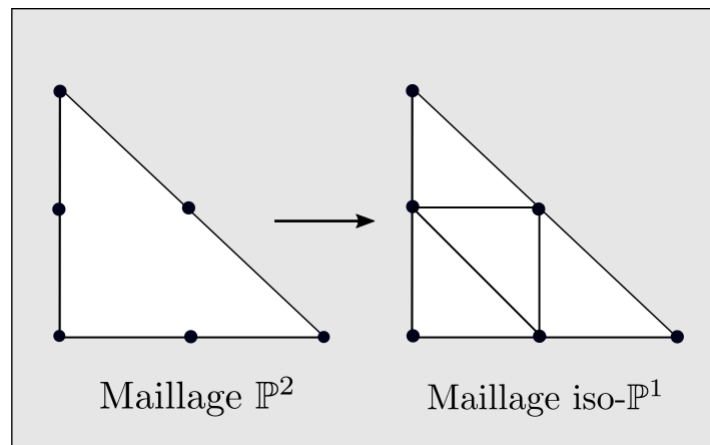


FIGURE 1.11 – Méthode d'obtention d'un maillage \mathbb{P}^1 ayant les mêmes degrés de liberté que le maillage \mathbb{P}^2 .

Dans nos simulations numériques sur FEEL++, on préférera la méthode de *fast-marching*. Le benchmark 3D [MCIP18] montre de meilleurs résultats de la méthode de *fast-marching* que celle d'Hamilton-Jacobi pour des simulations multi-fluidiques sur FEEL++.

Maintenant que l'on a présenté l'équation d'évolution de la fonction *levelset*, il nous reste à voir comment le couplage entre le fluide et la structure s'opère. On rappelle que cela est fait en adaptant le terme de force de la méthode de frontière immergée au contexte *levelset*.

Stabilisation de l'équation de transport

L'équation d'advection est une équation hyperbolique et sa stabilité est alors un problème important. En effet, si l'on ne prend pas de précautions des oscillations peuvent survenir. Dans le cadre de méthodes de différences ou volumes finis, des schémas numériques spécifiques *ENO*, *Essentially non-oscillatory* et *WENO*, *Weighted essentially non-oscillatory* ont été introduits dans [LOC94] pour gérer ces dernières. Elles consistent essentiellement à forcer la dérivation à être faite dans le sens du fluide.

Ce genre de techniques n'est pas utilisable pour des éléments finis car on ne discrétise pas le domaine selon une grille régulière. Plusieurs méthodes de stabilisation existent dans le cadre d'une méthode éléments finis, on citera les trois méthodes suivantes : *streamline upwind Petrov Galerkin SUPG* [BH82, Hug87], *Galerkin Least Squares (GALS)* [HFH89] et *subgrid scale SGS* [Hau02]. Il existe bien d'autres méthodes mais nous présentons celles-ci car ce sont celles qui ont été implémentées dans FEEL++.

Le principe est d'ajouter un terme de stabilisation sur chaque élément du domaine. Plus précisément, si l'on considère l'équation d'advection générale suivante :

$$\sigma\phi + u \cdot \nabla\phi = f \quad (1.67)$$

On définit l'opérateur \mathcal{L} et son adjoint de la manière suivante :

$$\begin{cases} \mathcal{L} = u \cdot \nabla + \sigma \\ \mathcal{L}^* = u \cdot \nabla - \sigma \end{cases} \quad (1.68)$$

L'équation (1.67) se réécrit alors :

$$\mathcal{L}(\phi) = f \quad (1.69)$$

et sa formulation faible éléments finis :

$$\forall \psi_h, \int_{\Omega} \mathcal{L}(\phi_h)\psi_h = \int_{\Omega} f\psi_h \quad (1.70)$$

Les méthodes de stabilisation modifient l'équation précédente en ajoutant sur chaque élément un terme $S_e(\phi_h, \psi_h)$:

$$\forall \psi_h, \int_{\Omega} \mathcal{L}(\phi_h)\psi_h + \sum_{e=1}^{Nb_{elt}} S_e(\phi_h, \psi_h) = \int_{\Omega} f\psi_h \quad (1.71)$$

Si l'on note Ω_e l'élément indexé par e , on définit S_e pour les différentes méthodes

par :

$$\begin{cases} S_e^{SUPG}(\phi_h, \psi_h) = \int_{\Omega_e} \frac{h}{2|u|} (u \cdot \nabla \phi_h) (\mathcal{L}(\phi_h) - f) dx \\ S_e^{GALS}(\phi_h, \psi_h) = \int_{\Omega_e} \frac{1}{\frac{2|u|}{h} + |\sigma|} \mathcal{L}(\psi_h) (\mathcal{L}(\phi_h) - f) dx \\ S_e^{SGS}(\phi_h, \psi_h) = \int_{\Omega_e} \frac{1}{\frac{2|u|}{h} + |\sigma|} (-\mathcal{L}^*(\psi_h)) (\mathcal{L}(\phi_h) - f) dx \end{cases} \quad (1.72)$$

Il convient de noter que les termes ajoutés conservent la consistance de la discrétisation grâce au facteur $(\mathcal{L}(\phi_h) - f)$, c'est-à-dire que $S_e \rightarrow 0$ lorsque $h, \delta t \rightarrow 0$. On se réfère aux articles cités pour une plus de détails sur le choix et les propriétés de ces termes de stabilisation.

1.4.7 Couplage fluide/structure pour des modèles de vésicules

On rappelle que la force élastique introduite dans [Pes02] est donnée par l'expression (1.45) et exprime un potentiel élastique dû à l'étirement de la membrane qui cherche à la ramener vers sa forme d'équilibre.

On reprend le même formalisme que l'on va ensuite adapter au contexte *levelset*, à savoir une paramétrisation $s \rightarrow X(t, s, r)$ de l'interface $\Gamma(t)$ de dimension 2, immergée dans un flot de dimension 3. En dimension 2, l'élément de surface est donné par $d\sigma = ds dr |X_r \times X_s|$ (\times désignant le produit vectoriel) et le potentiel élastique vaut :

$$\mathcal{E}(X) = \int_{\Omega} E (|\partial_r X \times \partial_s X|) d(s, r) \quad (1.73)$$

pour une certaine fonction E à définir.

Comme la formulation *levelset* permet d'exprimer la normale $|\partial_r X \times \partial_s X|$ sous la forme $\frac{\nabla \phi}{|\nabla \phi|}$, les auteurs montrent dans [CM04] que la force qui dérive de (1.73) s'exprime comme suit :

$$\begin{aligned} \mathcal{F}_e &= F_T + F_n \\ &= \left(\mathbb{I} - \frac{\nabla \phi \otimes \nabla \phi}{|\nabla \phi|^2} \right) \nabla (E'(|\nabla \phi|)) + E'(|\nabla \phi|) H(\phi) \frac{\nabla \phi}{|\nabla \phi|} \end{aligned} \quad (1.74)$$

Le terme F_T correspond à la partie tangentielle de la force tandis que le terme F_n correspond à la partie normale de la force.

Les calculs sont disponibles dans [CM04]. Ils reposent essentiellement sur l'expression des premières variations de l'énergie dans la direction ϕ suivie d'intégrations par partie. On exprime enfin l'égalité du type :

$$\partial_t \mathcal{E}(\phi) = - \int_{\Omega} F \cdot u dx. \quad (1.75)$$

Si $\frac{\nabla\phi}{|\nabla\phi|}$ désigne la normale à l'interface, le terme $\mathbb{I} - \frac{\nabla\phi \otimes \nabla\phi}{|\nabla\phi|^2}$ est l'opérateur de projection sur le plan tangent, que l'on notera dans la suite $\mathbb{P}_{\nabla\phi^\perp}$.

L'expression de la force est ensuite régularisée avec la masse de Dirac régularisée afin de concentrer les forces sur l'interface et notre force devient :

$$F_e[\phi] = \mathcal{F}_e(\phi) \frac{1}{\varepsilon} \zeta\left(\frac{\phi}{\varepsilon}\right) |\nabla\phi| \quad (1.76)$$

Il convient de noter que l'on a exprimé la force de couplage uniquement à l'aide de la fonction ϕ . Cet élément est crucial car on n'a alors besoin que de la connaissance des fonctions u, p, ϕ pour obtenir toutes les informations sur le système. Aucun marqueur lagrangien n'est nécessaire et on peut alors éviter les complications liées à leur utilisation.

Dans [CM06], une méthode alternative est proposée pour l'obtention de (1.76). L'idée est de dériver la force immédiatement depuis la régularisation du potentiel (1.81) :

$$\mathcal{E}(X) = \int_{\Omega} E(|\nabla\phi|) \frac{1}{\varepsilon} \zeta\left(\frac{\phi}{\varepsilon}\right) dx \quad (1.77)$$

Cette approche semble plus intrinsèque dans le sens où la régularisation est prise en compte lors de la dérivation de la force plutôt que simplement appliquée *a posteriori*.

La fonction E permet de distinguer les différentes situations physiques causant cette force élastique. Sa dérivée E' décrit essentiellement la réponse de la membrane à un changement d'aire. On note ν_e le coefficient de raideur et on distingue principalement les deux cas de figures suivants :

- Le cas $E'(r) = \nu_e (r - 1)$ correspond à une élasticité linéaire dont l'étirement au repos vaut 1. Plus la valeur de ν_e est élevée, plus la raideur de la membrane est importante et plus la membrane approche l'inextensibilité recherchée.
- Le cas $E'(r) = \nu_e$ correspond à une tension de surface. Elle permet de modéliser des flots multiphasiques de fluides.

Le système du modèle *levelset* pour une membrane élastique ou un système bi-fluide proposé par Cottet et Maitre s'écrit :

$$\begin{cases} \rho_f (\partial_t u + u \cdot \nabla u) + \nabla p - \mu_f \Delta u = F_e(\phi) \\ \operatorname{div} u = 0 \\ \partial_t \phi + u \cdot \nabla \phi = 0 \end{cases} \quad (1.78)$$

Dans le cas d'une membrane élastique, la force F_e a pour rôle de forcer la conservation de l'aire de la membrane. De plus, comme on transporte ϕ par un champ de vitesse à divergence nulle, son volume est conservé. Ainsi, l'aire et le périmètre de la membrane sont conservés par le système.

On rappelle que pour la simulation des vésicules et globules rouges, il est nécessaire de prendre en compte les effets de courbure. Ceux-ci s'expriment à partir de l'énergie de Helfrich (1.2). On dérive la force associée à partir de l'énergie de Willmore régularisée :

$$E_c[\phi] = \frac{1}{2} \int_{\Omega} H(\phi)^2 |\nabla \phi| \frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) dx \quad (1.79)$$

Le résultat de la dérivation de cette énergie donne la force de courbure F_c suivante :

$$F_c[\phi] = \operatorname{div} \left(-\frac{H(\phi)^2}{2} \frac{\nabla \phi}{|\nabla \phi|} + \frac{1}{|\nabla \phi|} \mathbb{P}_{\nabla \phi^\perp} (\nabla (|\nabla \phi| H(\phi))) \right) \frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) \nabla \phi \quad (1.80)$$

On se réfère à [MMC⁺09] pour la dérivation de l'expression de la force.

Le système de couplage fluide-structure complet du modèle *levelset* pour la simulation des globules dans le sang s'écrit :

$$\begin{cases} \rho_f (\partial_t u + u \cdot \nabla u) + \nabla p - \mu_f \Delta u = F_\varepsilon(\phi) + F_c(\phi) \\ \operatorname{div} u = 0 \\ \partial_t \phi + u \cdot \nabla \phi = 0 \end{cases} \quad (1.81)$$

1.4.8 Multiples vésicules immergées et force hydrodynamique

La concentration de globules rouges dans le sang est très importante. On peut utiliser la même fonction *levelset* pour représenter l'intégralité des globules pour modéliser leur déplacement. Cependant, ϕ ne dissocie pas les différents objets et il est alors impossible de rendre compte des forces de répulsions engendrées par la force hydrodynamique entre deux objets proches. Il est inenvisageable d'associer une *levelset* à chaque globule pour des raisons de coût algorithmique.

Une méthode proposée dans [JBM19] propose de gérer les effets de la force hydrodynamique pour une forte concentration de globules en ajoutant une seconde fonction *levelset*.

La méthode considérée consiste à définir une force répulsive similaire à une force électromagnétique. Si l'on considère deux objets Ω_i et Ω_j représentés par leur *levelset* respective ϕ_i et ϕ_j . On peut exprimer la distance de Ω_j à Ω_i par la valeur de Ω_j . La direction de la force de répulsion suivant la force hydrodynamique sera donnée par $\nabla \phi_j$.

Le modèle de force de contact de Ω_j sur Ω_i est le suivant :

$$F_{ij} = \rho(x) \frac{k}{\varepsilon} \zeta \left(\frac{\phi_i(x)}{\varepsilon} \right) \frac{\nabla \phi_j(x)}{\phi_j(x)} \exp \left(-\frac{\phi_j(x)}{\varepsilon_b} \right) \quad (1.82)$$

Elle est localisée sur l'interface Γ_{ij} grâce au Dirac régularisé $\frac{1}{\varepsilon} \zeta \left(\frac{\phi_i(x)}{\varepsilon} \right)$ et au terme exponentiel en ϕ_j . Il est nécessaire que la valeur des deux *levelset* soit petite pour que le Dirac ne soit pas nul et le terme exponentiel non négligeable.

La force totale correspond à la somme de toutes les forces F_{ij} :

$$F_{repulsion}(x) = \sum_{i \neq j} \rho(x) \frac{k}{\varepsilon} \zeta \left(\frac{\phi_i(x)}{\varepsilon} \right) \frac{\nabla \phi_j(x)}{\phi_j(x)} \exp \left(-\frac{\phi_j(x)}{\varepsilon_b} \right) \quad (1.83)$$

où ε_b et k sont des constantes de répulsion. k doit être choisi de l'ordre des vitesses relatives des objets.

L'idée de [JBM19] est de définir une seconde fonction *levelset* ϕ_2 et de labels L_0, L_1, L_2 permettant d'identifier les objets, cf Fig.1.12. ϕ_2 sert à mesurer la distance au deuxième plus proche voisin, cf Fig.1.13. Le label L_0 détermine le domaine où l'on se trouve, L_1 le domaine du plus proche voisin et L_2 le domaine du deuxième plus proche voisin.

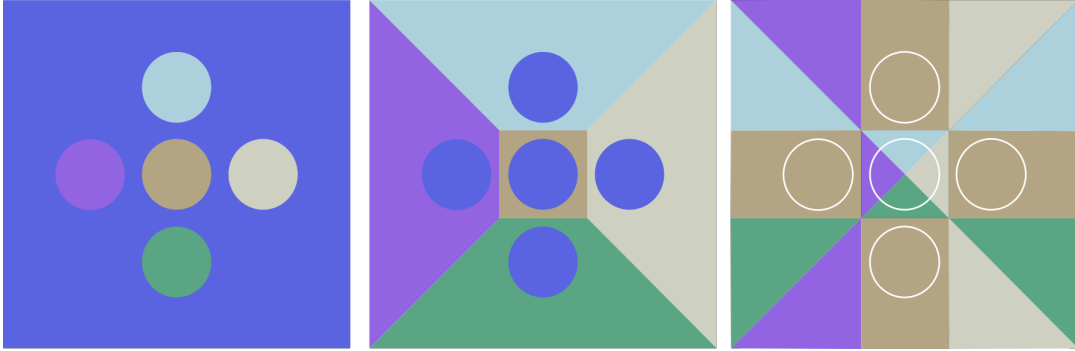


FIGURE 1.12 – Représentation des différentes fonctions *label* : L_0 à gauche, L_1 au centre et L_2 à droite.

On pourra alors remplacer ϕ_i par la première *levelset* ϕ et ϕ_j par la deuxième *levelset* ϕ_2 dans la force (1.83).

On décrit succinctement comment on gère la deuxième *levelset* ϕ_2 . Pour un point x de l'objet i , on aura :

- L_0 détermine le domaine où l'on se trouve : $L_0(x) = i$.
- L_1 détermine le domaine du plus proche voisin, c'est-à-dire le fluide : $L_1 = 0$.
- L_2 détermine le domaine du deuxième plus proche voisin, c'est-à-dire l'objet le plus proche, qui correspond à Ω_j : $L_2(x) = j$.

On a besoin de définir ces fonctions labels pour calculer la fonction ϕ_2 . Le calcul de ϕ_2 se fait à l'aide d'une double *fast-marching*. La différence avec l'algorithme classique 1.4.6 : 1 est que chaque sommet va passer deux fois dans la liste Voisin.

On initialise l'algorithme en calculant L_0 sur tous les sommets. On utilise $\frac{\phi_1}{|\nabla \phi_1|}$ comme valeur d'initialisation pour les points situés sur les interfaces et on

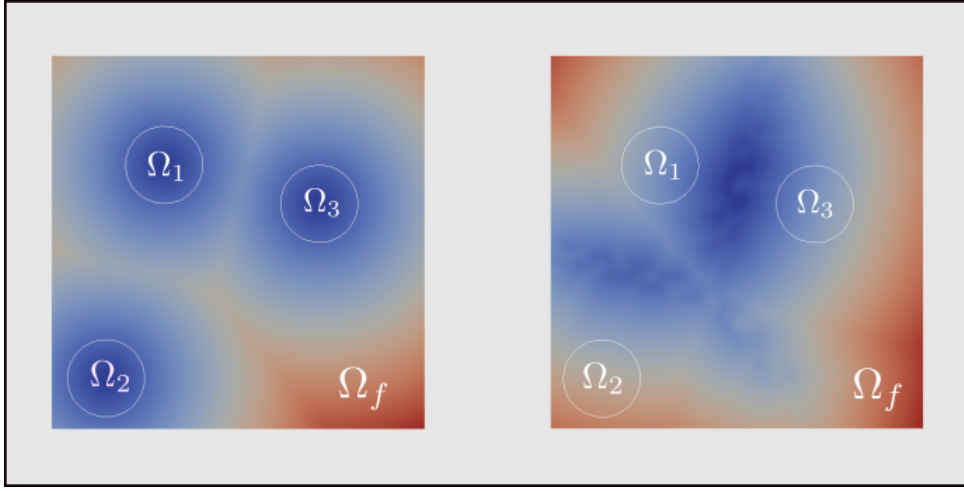


FIGURE 1.13 – Représentation des deux fonctions *levelset* ϕ_1 (à gauche) et ϕ_2 (à droite).

affecte leur label L_1 (il s'agit du fluide ou de l'objet dont on initialise l'interface selon le signe de ϕ_1).

L'algorithme se déroule alors comme l'algorithme de *fast-marching* classique. Le sommet dont la distance est minimale dans la liste Voisin est traité. Si son label L_1 et sa fonction ϕ_1 sont connus, c'est qu'il s'agit du deuxième passage et on calcule alors L_2 et ϕ_2 . Sinon, il s'agit du premier passage et on s'occupe de L_1 et ϕ_1 . Les détails de l'implémentation se trouvent dans [JBM19].

On peut alors adapter la force de répulsion hydrodynamique (1.83) au formalisme précédent :

$$F_{repulsion}(x) = \rho(x) \frac{k}{\varepsilon} \zeta \left(\frac{\phi_1(x)}{\varepsilon} \right) \frac{\nabla \phi_2(x)}{\phi_2(x)} \exp \left(-\frac{\phi_2(x)}{\varepsilon_b} \right) \quad (1.84)$$

Une étude est menée dans [JBM19] pour prouver que la gestion des répulsions suivant la force hydrodynamique avec (1.84) est satisfaisante.

1.4.9 Méthode champ de phase et liens avec la méthode *levelset* des modèles de vésicules immergées

Dans cette partie, on s'intéresse à la méthode de champ de phase et plus particulièrement aux modèles de vésicules par la méthode champ de phase. On compare ce modèle avec la méthode *levelset* (1.81).

Originellement introduite pour des problèmes de croissance cristalline [CL85], la méthode champ de phase a ensuite été étendue à différents domaines [Kob93, KR98, KM99].

Le principe de la méthode champ de phase est de séparer les phases en leur attribuant une valeur constante fixée. Un choix naturel, dans le cas d'un solide

immergé dans un fluide, est d'utiliser la fonction indicatrice χ de la structure :

$$\chi = \begin{cases} 1 & \text{si } x \in \Omega_s \\ 0 & \text{si } x \in \Omega_f \end{cases} \quad (1.85)$$

C'est cette description qui est adoptée pour les méthodes de convolution-seuillage que l'on étudiera au chapitre suivant.

La méthode champ de phase a été adaptée au contexte des vésicules conjointement par Du et al. [DLW04, DLRW05, DLL07] et Biben et al. [BKM05, Bib05]. Une étude des différentes méthodes champ de phase pour le flot de Willmore ainsi que des schémas numériques sont proposés dans [BMO15].

On s'intéresse maintenant plus en détail au modèle de champ de phase pour les vésicules introduit dans [BKM05, Bib05]. La description considérée est basée sur le modèle de Landau-Ginzburg. On choisit pour valeur des phases $\chi = \pm 1$ tout en assurant une transition régularisée à l'interface. On obtient χ en minimisant l'énergie libre suivante :

$$E_{\text{config}}(\chi) = \int_{\Omega} \frac{1}{4} (1 - \chi^2)^2 + \frac{\varepsilon^2}{2} |\nabla \chi|^2 dx \quad (1.86)$$

avec $\varepsilon > 0$.

Le premier terme de l'énergie permet de séparer les deux phases, cf Fig.1.14, tandis que le second terme assure une transition régulière sur l'interface. De plus, le deuxième terme permet d'imposer une épaisseur fixe pour l'interface de l'ordre du paramètre ε .

Par exemple, dans le cas particulier d'une interface plane (on prend ici le plan $Vect(x, y)$ en dimension 3), χ s'exprime explicitement par :

$$\chi(z) = \left(\frac{z}{\varepsilon\sqrt{2}} \right) \quad (1.87)$$

L'interface est ainsi d'épaisseur $\sqrt{2}\varepsilon$ et converge bien vers la fonction indicatrice discontinue. Il convient également de noter que $|\nabla \chi|$ est une approximation du Dirac.

L'énergie (1.86) est appelée de configuration car elle sert uniquement à définir χ la frontière entre les deux phases. Elle n'affecte pas la physique du système dans le sens où elle ne traduit pas l'effet des forces élastique et de courbure de la vésicule. Elle contient cependant des informations physiques telles que la tension de surface pouvant être indésirables selon la situation considérée.

De la même façon qu'avec la fonction *levelset*, on peut exprimer la normale et la courbure à l'aide de χ :

$$n(\chi) = \frac{\nabla \chi}{|\nabla \chi|} \text{ et } \kappa(\chi) = -\text{div } n \quad (1.88)$$

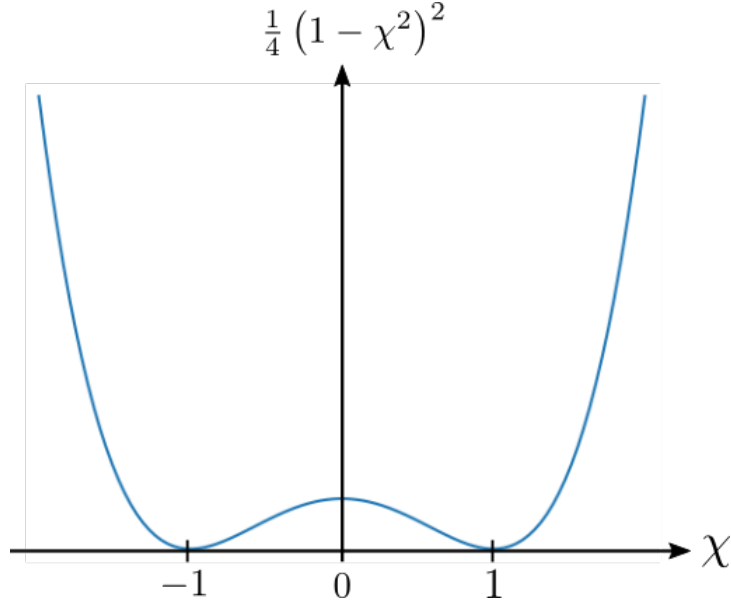


FIGURE 1.14 – Représentation de la fonction de Ginzburg-Landau de type double potentiel. On observe deux minima qui permettent de séparer les deux phases.

On peut alors exprimer les forces de courbure et élastique par l'énergie suivante :

$$E(\chi) = \frac{1}{2} \int \kappa(\chi)^2 \frac{|\nabla\chi|}{2} dx + \int \xi \frac{|\nabla\chi|}{2} dx \quad (1.89)$$

Le terme de gauche correspond à l'énergie d'Helfrich (1.2) et le second correspond à l'énergie élastique d'inextensibilité de la membrane.

Selon le même principe que pour la dérivation des forces de la méthode *levelset*, on obtient la force suivante pour la méthode champ de phase :

$$F(\chi) = \left[- \left\{ \Delta_\Gamma \kappa + \frac{\kappa_H}{2} (\kappa^2 - G) \right\} n + \xi \kappa n + \mathbb{P}_{\nabla\chi^\perp}(\nabla\xi) \right] \frac{|\nabla\chi|}{2} \quad (1.90)$$

Le terme ξ est un multiplicateur de Lagrange introduit pour assurer l'incompressibilité locale, c'est-à-dire pour imposer la condition :

$$(\mathbb{I} - n \otimes n) : \nabla u = 0 \quad (1.91)$$

Dans [BKM05, Bib05], les auteurs proposent de définir ξ sur tout le domaine mais de limiter son action à l'interface en considérant l'énergie suivante :

$$\int_{\Omega} \xi(x, t) \frac{|\nabla\chi|}{2} dx \quad (1.92)$$

Comme χ est choisie approchant la fonction de champ de phase (1.85) (en prenant -1 au lieu de 0 pour la valeur sur le fluide), $|\nabla\chi|$ s'approche d'une

fonction Dirac d'épaisseur ε et l'énergie précédente est bien concentrée sur l'interface. L'équation postulée dans [BKM05, Bib05] pour ξ est :

$$\partial_t \xi + u \cdot \nabla \xi = T (\mathbb{I} - n \otimes n) : \nabla u \quad (1.93)$$

où T est un paramètre à choisir suffisamment grand pour imposer $(\mathbb{I} - n \otimes n) : \nabla u \sim 0$.

L'équation fluide considérée dans les articles [BKM05] et [Bib05] est l'équation de Stokes.

$$\begin{cases} \varepsilon_u \partial_t u - \operatorname{div} \left(2\eta \frac{\nabla u + \nabla u^T}{2} \right) + \nabla p = F \\ \operatorname{div} u = 0 \end{cases} \quad (1.94)$$

où ε_u représente un paramètre de relaxation pris petit pour tendre vers un nombre de Reynolds nul et $\eta = \eta_f \frac{(1+\chi)}{2} + \eta_s \frac{(1-\chi)}{2}$ représente la viscosité régularisée qui prend en compte les viscosités des fluides internes et externes à la membrane.

La membrane étant transportée par le fluide, l'équation d'advection $\partial_t \chi + u \cdot \nabla \chi = 0$ décrit l'évolution de χ . Cependant, pour des raisons de stabilité et pour garantir que la fonction transportée minimise bien l'énergie de configuration, l'équation considérée est alors :

$$\partial_t \chi + u \cdot \nabla \chi = -\varepsilon_\chi \frac{\delta E_{config}}{\delta \chi} \quad (1.95)$$

où ε_χ désigne une constante cinétique pour fixer l'échelle temporelle et $\frac{\delta E_{config}}{\delta \chi}$ désigne les premières variations de l'énergie.

Un terme additionnel de courbure est ajouté pour supprimer la tension de surface contenue naturellement dans l'énergie de configuration. En effet, puisqu'il n'y a pas d'échange de matière entre l'intérieur et l'extérieur de la membrane, l'énergie de surface est nulle et il est nécessaire d'annuler cette tension de surface intrinsèquement comprise dans le modèle de Landau-Ginzburg. L'équation finale vaut :

$$\partial_t \chi + u \cdot \nabla \chi = \varepsilon_\chi \left[\chi(1 - \chi^2) + \varepsilon^2 (\Delta \chi + \kappa |\nabla \chi|) \right] \quad (1.96)$$

Une solution alternative à la présence de tension de surface proposée dans [JM08] est de corriger l'énergie de configuration afin de supprimer la tension de surface sans changer l'allure de la fonction.

Le système de champ de phase correspondant au système *levelset* (1.81) pour un régime de Stokes s'écrit alors :

$$\begin{cases} \varepsilon_u \partial_t u - \operatorname{div} \left(\eta \frac{\nabla u + \nabla u^T}{2} \right) + \nabla p = F \\ \operatorname{div} u = 0 \\ \partial_t \chi + u \cdot \nabla \chi = 0 \end{cases} \quad (1.97)$$

On observe que la dérivation du modèle (1.97) est très similaire au modèle *levelset*. Dans [MMPR12], les forces élastiques ont été mises sous forme divergente :

$$\begin{cases} F_e(\phi) = \operatorname{div} \left(E'(|\nabla\phi|) |\nabla\phi| \left[\mathbb{I} - \frac{\nabla\phi \otimes \nabla\phi}{|\nabla\phi|^2} \right] \right) & \text{levelset} \\ F_e(\chi) = \operatorname{div} \left(\xi |\nabla\chi| \left[\mathbb{I} - \frac{\nabla\chi \otimes \nabla\chi}{|\nabla\chi|^2} \right] \right) & \text{champ de phase} \end{cases} \quad (1.98)$$

On peut faire le lien entre $|\nabla\phi|$ et le multiplicateur de Lagrange ξ . On a vu que la norme du gradient de ϕ mesure l'étirement et qu'il vérifie l'équation de transport :

$$\partial_t |\nabla\phi| + u \cdot \nabla |\nabla\phi| = -|\nabla\phi| \frac{\nabla\phi \otimes \nabla\phi}{|\nabla\phi|^2} : \nabla u \quad (1.99)$$

On peut réécrire le terme de droite, pour un fluide à divergence nulle :

$$-|\nabla\phi| \frac{\nabla\phi \otimes \nabla\phi}{|\nabla\phi|^2} : \nabla u = |\nabla\phi| \left(\mathbb{I} - \frac{\nabla\phi \otimes \nabla\phi}{|\nabla\phi|^2} \right) : u \quad (1.100)$$

En divisant par $|\nabla\phi|$ et en multipliant par T , on obtient l'équation :

$$\partial_t (T \log |\nabla\phi|) + u \cdot \nabla (T \log |\nabla\phi|) = T (\mathbb{I} - \nabla\phi \otimes \nabla\phi) : \nabla u \quad (1.101)$$

On voit alors que $T \log |\nabla\phi|$ vérifie la même équation que celle postulée pour ξ (1.93). Comme les équations de transport ne sont pas les mêmes pour les deux méthodes, les champs de vitesse seront aussi différents et on ne peut pas conclure que ξ et $T \log |\nabla\phi|$ sont égaux. Cependant, il convient de rappeler qu'on choisit $E'(|\nabla\phi|) = \lambda(|\nabla\phi| - 1)$ pour une membrane élastique et que $|\nabla\phi| - 1$ et $\log |\nabla\phi|$ sont équivalents au voisinage de 1.

Cependant, on peut estimer qu'ils ont des comportements proches et il peut sembler surprenant que l'on n'ait pas une relation linéaire entre ξ et $|\nabla\phi|$ à partir de l'expression des forces (1.98).

1.5 Discrétisation du couplage et analyse de stabilité

Lors de la résolution numérique d'un système de couplage, en plus de la discrétisation de chaque terme, on doit choisir comment discrétiser le couplage entre le fluide et la structure. C'est ce qui nous intéresse ici.

Une approche monolithique consiste à résoudre le système entier avec le même solveur. Cette stratégie se révèle souvent inutilisable en pratique car chaque équation nécessite des procédures numériques spécifiques. On considère alors une approche séparée. Chaque équation aura son solveur et certaines procédures de stabilisation dédiées afin d'optimiser les performances des calculs. La

résolution des équations se fait successivement et une difficulté supplémentaire est introduite, à savoir le choix de discrétisation du couplage. En effet, lors de la résolution de l'équation de Navier-Stokes à l'instant $n + 1$, on doit choisir quelle valeur pour ϕ on prend dans l'expression du terme force. Typiquement, cela revient à choisir si le terme de force en ϕ est explicité en utilisant ϕ^n ou implicité en utilisant ϕ^{n+1} .

Pour fixer les notations on donne les deux schémas de couplage classiques. Le schéma implicite s'écrit :

$$\begin{cases} \rho \left(\frac{u^{n+1} - u^n}{\delta t} + u^{n+1} \cdot \nabla u^{n+1} \right) + \nabla p^{n+1} - \mu \Delta u^{n+1} = F_e(\phi^{n+1}) \\ \operatorname{div} u^{n+1} = 0 \\ \frac{\phi^{n+1} - \phi^n}{\delta t} + u^{n+1} \cdot \nabla \phi^{n+1} = 0 \end{cases} \quad (1.102)$$

Le schéma explicite s'écrit :

$$\begin{cases} \rho \left(\frac{u^{n+1} - u^n}{\delta t} + u^{n+1} \cdot \nabla u^{n+1} \right) + \nabla p^{n+1} - \mu \Delta u^{n+1} = F_e(\phi^n) \\ \operatorname{div} u^{n+1} = 0 \\ \frac{\phi^{n+1} - \phi^n}{\delta t} + u^{n+1} \cdot \nabla \phi^{n+1} = 0 \end{cases} \quad (1.103)$$

Remarque 1.5.1. *Le schéma explicite (1.103) est en réalité un schéma semi-implicite dans le sens où l'on résout d'abord l'équation de Navier-Stokes pour obtenir u^{n+1} . Puis on utilise la valeur de u^{n+1} pour la résolution de l'équation d'advection. Un schéma purement explicite utiliserait u^n . On choisit d'ignorer ce choix car il produit des résultats moins bons et pour une complexité algorithmique égale à celle de (1.103).*

Ainsi, on emploiera abusivement le terme "schéma explicite" pour désigner (1.103). On évite alors une ambiguïté future car on désignera parfois la méthode de prédiction par l'appellation "schéma-implicite".

On a choisi une discrétisation temporelle selon le schéma d'euler implicite pour les deux équations. On s'intéressera plus en détail sur les méthodes numériques pour chaque équation dans les parties numériques. On met l'accent ici uniquement sur la stabilité des équations en fonction du couplage.

En l'état, le schéma implicite (1.102) correspond à l'approche monolithique car les équations sont toujours couplées. En effet, chaque équation a besoin du terme à l'instant $n + 1$ fourni pour l'autre équation. On doit alors résoudre un problème couplant les équations de Navier-Stokes contenant un terme source non linéaire avec une équation de transport. Ceci se révèle souvent trop coûteux en pratique.

Dans cette section, on étudie la stabilité du système (1.103) en dimension 2 dans le cas d'une méthode de différences finies. On ne discute pas sur les conditions aux bords. En effet, l'objectif ici est de montrer l'inefficacité du schéma de couplage explicite dans certains cas et de justifier la nécessité de proposer une nouvelle méthode. On commence par linéariser le modèle (1.78) discrétisé par un couplage implicite ou explicite et on effectue une étude de stabilité de type Von Neumann. Il s'agit d'une méthode classique pour étudier la stabilité d'un schéma à l'aide d'une décomposition en série de Fourier.

On cherche à obtenir l'équation linéarisée pour le modèle (1.78) en dimension 2. On note $u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$. Le modèle se réécrit alors :

$$\begin{cases} \partial_t u_1 + \partial_x p - \mu(\partial_{xx}^2 u_1 + \partial_{yy}^2 u_1) = (F_e[\phi])_x \\ \partial_t u_2 + \partial_y p - \mu(\partial_{xx}^2 u_2 + \partial_{yy}^2 u_2) = (F_e[\phi])_y \\ \partial_x u_1 + \partial_y u_2 = 0 \\ \partial_t \phi + u_1 \partial_x \phi + u_2 \partial_y \phi = 0 \end{cases} \quad (1.104)$$

Pour le cas d'une tension de surface entre deux fluides, c'est-à-dire avec $E(|\nabla\phi|) = \nu_e$, on cherche le modèle linéarisé de (1.104) autour de la position d'équilibre suivante :

$$\begin{cases} \overline{u_1}(x, y) = 0 \\ \overline{u_2}(x, y) = 0 \\ \overline{\phi}(x, y) = y \end{cases} \quad (1.105)$$

On introduit les déformations suivantes :

$$\begin{cases} \begin{pmatrix} u_1(x, y) \\ u_2(x, y) \end{pmatrix} = \begin{pmatrix} v_1(x, y) \\ v_2(x, y) \end{pmatrix} \\ \phi(x, y) = y + \psi(x, y) \end{cases} \quad (1.106)$$

avec (v_1, v_2, ψ) considérés comme petits.

Il reste à calculer la force élastique. Pour obtenir son expression, il faut linéariser les éléments suivants :

$$\begin{cases} \nabla\phi = \begin{pmatrix} \partial_x \psi \\ 1 + \partial_y \psi \end{pmatrix} \\ |\nabla\phi| = \sqrt{(\partial_x \psi)^2 + 1 + 2\partial_y \psi + (\partial_y \psi)^2} = 1 + \partial_y \psi + o(\psi) \\ \frac{1}{|\nabla\phi|} = \frac{1}{1 + \partial_y \psi + o(\psi)} = 1 - \partial_y \psi + o(\psi) \\ \kappa(\phi) = \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|} \right) = \partial_{xx}^2 \psi + o(\psi) \\ \zeta\left(\frac{\phi}{\varepsilon}\right) = \zeta\left(\frac{\overline{\phi}}{\varepsilon}\right) + \frac{\psi}{\varepsilon} \zeta'\left(\frac{\overline{\phi}}{\varepsilon}\right) + o(\psi) \end{cases} \quad (1.107)$$

On obtient alors la force élastique suivante :

$$F_\varepsilon[\phi] = \begin{pmatrix} 0 \\ -\frac{\nu_e}{\varepsilon} \partial_{xx}^2 \psi \end{pmatrix} \quad (1.108)$$

Finalement, le système linéarisé vaut, dans le cas d'une tension de surface entre deux fluides :

$$\begin{cases} \partial_t v_1 + \partial_x p - \mu(\partial_{xx}^2 v_1 + \partial_{yy}^2 v_1) = 0 \\ \partial_t v_2 + \partial_y p - \mu(\partial_{xx}^2 v_2 + \partial_{yy}^2 v_2) = -\frac{\nu_e}{\varepsilon} \partial_{xx}^2 \psi \\ \partial_x v_1 + \partial_y v_2 = 0 \\ \partial_t \psi + v_2 = 0 \end{cases} \quad (1.109)$$

Dans le cadre d'une membrane élastique en position étirée, on linéarise autour de :

$$\begin{cases} \overline{u}_1(x, y) = 0 \\ \overline{u}_2(x, y) = 0 \\ \overline{\phi}(x, y) = \lambda y \end{cases} \quad (1.110)$$

où le paramètre λ correspond à l'étirement de la membrane à l'équilibre.

Et le système linéarisé que l'on obtient vaut :

$$\begin{cases} \partial_t v_1 + \partial_x p - \mu(\partial_{xx}^2 v_1 + \partial_{yy}^2 v_1) = \frac{\nu_e}{\varepsilon} \lambda \partial_{xy}^2 \psi \\ \partial_t v_2 + \partial_y p - \mu(\partial_{xx}^2 v_2 + \partial_{yy}^2 v_2) = -\frac{\nu_e}{\varepsilon} (\lambda - 1) \partial_{xx}^2 \psi \\ \partial_x v_1 + \partial_y v_2 = 0 \\ \partial_t \psi + v_2 = 0 \end{cases} \quad (1.111)$$

On présente dans la suite, l'étude de stabilité pour le cas de la tension de surface. Le raisonnement étant similaire pour le cas de la membrane, on se contentera de présenter les résultats.

1.5.1 Reformulation du système sous une formulation vortex

On reformule le système (1.109) pour obtenir une formulation tourbillon en introduisant le vecteur tourbillon :

$$w = \text{rot}(v) = \begin{pmatrix} 0 \\ 0 \\ \partial_x v_2 - \partial_y v_1 \end{pmatrix} \quad (1.112)$$

Comme on étudie le cas de la dimension 2, w est porté par la troisième coordonnée. Pour simplifier les notations, on se permettra d'identifier w à sa composante selon la troisième coordonnée.

D'une part, en soustrayant la dérivée par rapport à y de la première ligne de (1.109) à la dérivée par rapport à x de la deuxième ligne de (1.109), on obtient :

$$\partial_t \omega - \mu \Delta \omega = -\frac{\nu_e}{\varepsilon} \left((\lambda - 1) \partial_{xxx}^3 \psi + \lambda \partial_{xyy}^3 \psi \right) \quad (1.113)$$

D'autre part, en dérivant par rapport à y la contrainte de divergence nulle et en utilisant la définition de w , on trouve :

$$\Delta v_2 - \partial_x w = 0 \quad (1.114)$$

En injectant ceci dans l'équation d'advection, ψ vérifie l'équation :

$$\partial_t \Delta \psi + \partial_x w = 0 \quad (1.115)$$

Finalement, on trouve le système suivant :

$$\begin{cases} \partial_t \omega - \mu \Delta \omega = -\frac{\nu_e}{\varepsilon} \left((\lambda - 1) \partial_{xxx}^3 \psi + \lambda \partial_{xyy}^3 \psi \right) \\ \partial_t (\Delta \psi) + \partial_x w = 0 \end{cases} \quad (1.116)$$

On va étudier la stabilité du système (1.116) pour un couplage explicite.

1.5.2 Analyse de stabilité du schéma explicite

On étudie la stabilité du schéma explicite par le critère de stabilité de Von Neumann. Il consiste à décomposer les fonctions (w, ψ) en série de Fourier sur la grille cartésienne de différences finies. On considère les fonctions discrétisées représentées par les ensembles de points $(w_{j,l}^n)_{(j,l) \in \mathbb{Z}^2}$, $(\psi_{j,l}^n)_{(j,l) \in \mathbb{Z}^2}$, dont les valeurs sont données par :

$$\begin{cases} w_{j,l}^n = \sum_{(k,h) \in \mathbb{Z}^2} \widehat{w}_{j,l}^n(k,h) e^{2i\pi(kj+hl)\Delta x} \\ \widehat{w}_{j,l}^n(k,h) = \iint_{[0,1]^2} w(t) e^{-2i\pi(kja+hlb)\Delta x} da db \\ \psi_{j,l}^n = \sum_{(k,h) \in \mathbb{Z}^2} \widehat{\psi}_{j,l}^n(k,h) e^{2i\pi(kj+hl)\Delta x} \\ \widehat{\psi}_{j,l}^n(k,h) = \iint_{[0,1]^2} \psi(t) e^{-2i\pi(kja+hlb)\Delta x} da db \end{cases} \quad (1.117)$$

On utilise un schéma d'Euler explicite pour les dérivées temporelles et le schéma explicite associé au système (1.116) s'écrit alors :

$$\begin{cases} \frac{\omega_{j,l}^{n+1} - \omega_{j,l}^n}{\Delta t} - \mu \Delta \omega_{j,l}^{n+1} = -\frac{\nu_e}{\varepsilon} \left((\lambda - 1) \partial_{xxx}^3 \psi_{j,l}^n + \lambda \partial_{xyy}^3 \psi_{j,l}^n \right) \\ \frac{\Delta \psi_{j,l}^{n+1} - \Delta \psi_{j,l}^n}{\Delta t} + \partial_x \omega_{j,l}^n = 0 \end{cases} \quad (1.118)$$

où les dérivées spatiales sont discrétisées comme suit :

$$\begin{cases} \partial_x \widehat{f}_{j,l}^n = \frac{\widehat{f}_{j+1,l}^n - 2\widehat{f}_{j,l}^n + \widehat{f}_{j-1,l}^n}{\Delta x} \\ \Delta \widehat{f}_{j,l}^n = \frac{\widehat{f}_{j+1,l}^n - 2\widehat{f}_{j,l}^n + \widehat{f}_{j-1,l}^n + \widehat{f}_{j,l+1}^n - 2\widehat{f}_{j,l}^n + \widehat{f}_{j,l-1}^n}{(\Delta x)^2} \\ \partial_{xxx}^3 \widehat{f}_{j,l}^n = \frac{(\widehat{f}_{j+2,l}^n - 2\widehat{f}_{j+1,l}^n + \widehat{f}_{j,l}^n) - (\widehat{f}_{j,l}^n - 2\widehat{f}_{j-1,l}^n + \widehat{f}_{j-2,l}^n)}{2(\Delta x)^3} \end{cases} \quad (1.119)$$

La valeur du paramètre ε est souvent choisie égale à plusieurs fois Δx afin d'avoir une épaisseur de frontière de quelques mailles. Pour des raisons de simplicité, on supposera $\varepsilon = \Delta x$. De plus, on notera $s_k = \sin(\pi k \Delta x)$. L'équation (1.118) se réécrit alors :

$$A_{k,h} \begin{pmatrix} \widehat{\omega}_{j,l}^{n+1}(k,h) \\ \widehat{\psi}_{j,l}^{n+1}(k,h) \end{pmatrix} = B_{k,h} \begin{pmatrix} \widehat{\omega}_{j,l}^n(k,h) \\ \widehat{\psi}_{j,l}^n(k,h) \end{pmatrix} \quad (1.120)$$

avec :

$$A_{k,h} = \begin{pmatrix} 1 + \frac{4\mu\Delta t}{(\Delta x)^2}(s_k^2 + s_h^2) & 0 \\ -\frac{\Delta t}{\Delta x}s_{2k}i & \frac{4}{(\Delta x)^2}(s_k^2 + s_h^2) \end{pmatrix} \quad (1.121)$$

et

$$B_{k,h} = \begin{pmatrix} 1 & \frac{4v_e\Delta t}{(\Delta x)^4}s_k^2s_{2k}i \\ 0 & \frac{4}{(\Delta x)^2}(s_k^2 + s_h^2) \end{pmatrix} \quad (1.122)$$

L'étude de stabilité est ramenée à l'étude des valeurs propres de la matrice $A_{k,h}^{-1}B_{k,h}$. Pour simplifier les notations, on pose $\alpha_k = \frac{4s_k^2}{\Delta x^2}$, $\alpha_h = \frac{4s_h^2}{\Delta x^2}$ et $\alpha = \alpha_k + \alpha_h$.

$$\begin{aligned} 0 &= \det(B_{k,h} - rA_{k,h}) \\ &= \begin{vmatrix} 1 - r(1 + \alpha\mu\Delta t) & \frac{v_e\Delta t}{\Delta x^2}s_{2k}\alpha_k i \\ rs_{2k}\frac{\Delta t}{\Delta x}i & (1-r)\alpha \end{vmatrix} \\ &= \alpha \left((1 + \alpha\mu\Delta t)r^2 - \left[2 + \mu\alpha\Delta t - v_e s_{2k}^2 \frac{\alpha_k}{\alpha} \frac{\Delta t^2}{\Delta x^3} \right] r + 1 \right) \end{aligned} \quad (1.123)$$

En posant $a = 1 + \alpha\mu\Delta t$ et $K = \frac{v_e s_{2k}^2 \alpha_k \Delta t^2}{\alpha \Delta x^3}$, on se retrouve à chercher les racines r_1, r_2 du polynôme $aX^2 - (1 + a - K)X + 1$. Par les relations coefficients racines, le produit des deux racines vaut $\frac{1}{a} > 0$, ce qui signifie que les deux racines sont de même signe. De plus, leur somme vaut $\frac{1+a-K}{a}$, donc le signe de cette quantité détermine le signe des racines. Il faut alors distinguer trois cas :

- Si le discriminant Δ est strictement négatif alors les racines sont complexes conjuguées et de même module $\frac{1}{a} < 1$. Ainsi, le schéma est stable dans ce cas.
- Si le discriminant Δ est positif et que les deux racines sont positives, c'est-à-dire si $1 + a \geq K$, on observe que :

$$\Delta = (1 + a - K)^2 - 4a = (a - 1 + K)^2 - 4aK \leq (\mu\alpha\Delta t + K)^2 \quad (1.124)$$

Ainsi :

$$r_i \leq \frac{2 + \mu\alpha\Delta t - K + \sqrt{\Delta}}{2(1 + \mu\alpha\Delta t)} \leq 1 \quad (1.125)$$

Le schéma est également stable dans ce cas-ci.

- Si le discriminant Δ est positif et que les deux racines sont négatives (donc que $1 + a < K$), les conditions limites de stabilité sont atteintes lorsque la plus petite des deux racines vaut -1 :

$$-1 = r_1 = \frac{1 + a - K - \sqrt{\Delta}}{2a} \quad (1.126)$$

D'où, en isolant le terme $\sqrt{\Delta}$ à gauche puis en élevant au carré :

$$(1 + a - K)^2 - 4a = (1 + 3a - K)^2 \quad (1.127)$$

En simplifiant, il résulte que pour que la plus petite racine soit égale à -1 , il faut que $K = 2(1 + a)$. De plus, en regardant l'influence des variations de K sur r_1 (r_1 diminue lorsque K augmente), la condition de stabilité est pour ce cas :

$$K \leq 2(1 + a) \quad (1.128)$$

Or, nous sommes dans ce cas d'un discriminant positif et de $1 + a < K$. Il en résulte que le schéma est toujours instable dans ce cas.

De cette discussion résulte que la condition nécessaire et suffisante pour avoir stabilité est :

$$K \leq 1 + a \quad (1.129)$$

Dans notre cas, cela se traduit par :

$$\frac{v_e s_{2k}^2 \alpha_k \Delta t^2}{\alpha \Delta x^3} \leq 2 + \alpha \mu \Delta t \quad (1.130)$$

En étudiant ce polynôme, on observe qu'il possède deux racines réelles, une négative et une positive. Il en résulte que la condition sur le pas de temps pour obtenir un schéma stable est :

$$\Delta t \leq \min_{(k,h) \in \mathbb{Z}^2} \frac{\mu [s_k^2 + s_h^2]^2 \Delta x}{2v_e s_k^4 c_k^2} + \sqrt{\left(\frac{\mu [s_k^2 + s_h^2]^2 \Delta x}{2v_e s_k^4 c_k^2} \right)^2 + \frac{(s_k^2 + s_h^2) \Delta x^3}{2v_e s_k^4 c_k^2}} \quad (1.131)$$

- Dans le cas de fluide non visqueux (pour lequel $\mu = 0$ ou est très petit) ou ayant un grand nombre de Reynolds / une tension de surface élevée, la condition peut se simplifier comme suit :

$$\Delta t \lesssim \frac{\Delta x^{\frac{3}{2}}}{\sqrt{v_e}} \quad (1.132)$$

On retrouve les conditions de stabilité annoncées par Brackbill et al. dans [BKZ92].

- Dans le cas d'un fluide visqueux ou d'une tension de surface relativement petite, la condition de stabilité se simplifie :

$$\Delta t \lesssim \frac{\mu}{v_e} \Delta x \quad (1.133)$$

On retrouve les conditions de stabilité obtenues par Galusinski et Vigneaux dans [GV07]

On peut réaliser une étude similaire dans le cas d'une membrane élastique étirée décrite par le système (1.111) et obtenir les résultats suivants :

- Pour un fluide non visqueux (pour lequel $\mu = 0$ ou est très petit) ou ayant un grand nombre de Reynolds, on trouve la condition de stabilité suivante :

$$\Delta t \lesssim \frac{\Delta x^{\frac{3}{2}}}{\sqrt{\lambda v_e}} \quad (1.134)$$

- Pour un fluide visqueux ou une tension de surface relativement petite, on obtient la condition de stabilité suivante :

$$\Delta t \lesssim \frac{\mu \Delta x}{\lambda^2 v_e} \quad (1.135)$$

On observe que la condition de stabilité sur le pas de temps peut fortement limiter l'efficacité des simulations, notamment lorsque l'objet élastique immergé est très rigide.

Utiliser un schéma implicite, bien qu'inconditionnellement stable, ne représente pas une alternative satisfaisante. En effet, cela implique la résolution d'un problème couplant une équation de Navier-Stokes avec des termes sources non linéaires et une équation de transport.

La prochaine section est donc consacrée à une solution proposée par Cottet et Maitre [CM16] en proposant un schéma semi-implicite basé sur une prédiction de la position de la fonction *levelset*.

1.6 Premier schéma de prédiction

D'après l'étude précédente, l'utilisation d'un schéma de couplage explicite pour le modèle *levelset* (1.78) peut se révéler insatisfaisante à cause d'une forte limitation dans le choix du pas de temps. Cette restriction est notamment particulièrement handicapante dans le cas d'une constante de raideur λ élevée. De même, l'utilisation d'un schéma de couplage implicite se révèle inenvisageable dans la pratique.

Une solution proposée par Cottet et Maitre dans [CM16] est d'ajouter une équation de prédiction pour obtenir une force élastique plus précise que celle fournie par le schéma explicite. Plus précisément, on prédit avec une équation de diffusion la position de la membrane au temps $n + 1$ et on utilise cette prédiction pour obtenir la force élastique de l'équation de Navier-Stokes. On utilise alors la vitesse du fluide ainsi obtenue pour advecter la fonction *levelset*. En procédant de la sorte, on espère avoir une meilleure expression des forces dans Navier-Stokes et obtenir une meilleure condition CFL de couplage.

En procédant de la sorte, il convient de noter qu'on a bien découplé les équations de Navier-Stokes et d'advection de (1.78). Pour détailler ce raisonnement, on reprend les détails de [CM16] et on dérive une analyse de stabilité du schéma semi-implicite pour le système (1.116).

1.6.1 Dérivation du schéma semi-implicite

On présente dans cette section comment obtenir le schéma semi-implicite à partir d'un couplage implicite des équations et une discrétisation d'Euler en temps :

$$\begin{cases} \rho \left(\frac{u^{n+1} - u^n}{\Delta t} \right) = F_\varepsilon(\phi^{n+1}) + R(u^n, u^{n+1}, \phi^{n+1}) \\ \operatorname{div} u^{n+1} = 0 \\ \frac{\phi^{n+1} - \phi^n}{\Delta t} + u^{n+1} \cdot \nabla \phi^{n+1} = 0 \end{cases} \quad (1.136)$$

où $R(u^n, u^{n+1})$ contient le gradient de pression, le terme d'inertie et le terme de diffusion de l'équation de Navier-Stokes.

En négligeant le term R , on obtient l'approximation \tilde{u}^{n+1} suivante de u^{n+1} :

$$\tilde{u}^{n+1} = u^n + \frac{\Delta t}{\rho} F_\varepsilon(\phi^{n+1}) \quad (1.137)$$

On obtient en prenant le produit scalaire de la force avec $\nabla \phi^{n+1}$:

$$F_\varepsilon(\phi^{n+1}) \cdot \nabla \phi^{n+1} = -\frac{\Delta t}{\rho} E'(|\nabla \phi^{n+1}|) |\kappa(\phi^{n+1})| (|\nabla \phi^{n+1}|)^2 \frac{1}{\varepsilon} \zeta \left(\frac{\phi^{n+1}}{\varepsilon} \right) \quad (1.138)$$

En supposant que ϕ est une fonction distance, on simplifie l'expression grâce à $|\nabla\phi^{n+1}| = 1$ et $\kappa(\phi^{n+1}) = \Delta\phi^{n+1}$:

$$F_\varepsilon(\phi^{n+1}) \cdot \nabla\phi^{n+1} = -\frac{\Delta t}{\rho} E'(|\nabla\phi^n|) \frac{1}{\varepsilon} \zeta\left(\frac{\phi^n}{\varepsilon}\right) \Delta\phi^{n+1} \quad (1.139)$$

En injectant ce terme dans l'équation (1.136), on obtient la discrétisation d'une équation de la chaleur :

$$\frac{\widetilde{\phi^{n+1}} - \phi^n}{\Delta t} - \frac{\Delta t}{\rho} \left[E'(|\nabla\phi^n|) \frac{1}{\varepsilon} \zeta\left(\frac{\phi^n}{\varepsilon}\right) \right] \Delta\phi^{n+1} = -u^n \cdot \nabla\phi^n \quad (1.140)$$

L'équation (1.140) fournit une prédiction de ϕ^{n+1} en fonction de ϕ^n et u^n . Le schéma de couplage semi-implicite s'exprime alors :

$$\begin{cases} \frac{\widetilde{\phi^{n+1}} - \phi^n}{\Delta t} - \frac{1}{\rho} \left[E'(|\nabla\phi^n|) \frac{1}{\varepsilon} \zeta\left(\frac{\phi^n}{\varepsilon}\right) \right] \Delta\widetilde{\phi^{n+1}} = -u^n \cdot \nabla\phi^n \\ \rho(\partial_t u + u \cdot \nabla u) + \nabla p - \mu\Delta u = F_\varepsilon(\widetilde{\phi^{n+1}}) \\ \operatorname{div} u^{n+1} = 0 \\ \frac{\phi^{n+1} - \phi^n}{\Delta t} + u^{n+1} \cdot \nabla\phi^{n+1} = 0 \end{cases} \quad (1.141)$$

Le coût de résolution de l'équation de la chaleur étant négligeable en comparaison de la résolution de l'équation de Navier-Stokes, le coût total d'une itération avec prédiction reste proche d'une itération du schéma de couplage explicite.

On espère qu'en utilisant le système (1.141), la condition CFL du couplage soit meilleure que pour le schéma explicite. Il sera alors possible de prendre un plus grand pas de temps et d'améliorer l'efficacité globale de la simulation.

Remarque 1.6.1. La prédiction $\widetilde{\phi^{n+1}}$ n'est utilisée que pour le calcul des termes de force de Navier-Stokes. L'équation d'advection n'est pas modifiée et on transporte toujours ϕ par une fonction u à divergence nulle, ce qui est essentiel pour assurer la conservation du volume de la levelset.

1.6.2 Analyse de stabilité du schéma semi-implicite

Dans cette partie, on procède à l'étude de stabilité du schéma semi-implicite. Elle reprend à l'identique les étapes de l'analyse de stabilité du schéma explicite. On reprend la méthode décrite précédemment pour dériver un schéma semi-implicite du système (1.116). Pour cela, on part d'une formulation complètement

implicite du système (1.116) :

$$\begin{cases} \partial_t \omega^{n+1} - \mu \Delta \omega^{n+1} = -\frac{v_e}{\varepsilon} \left((\lambda - 1) \partial_{xxx}^3 \psi^{n+1} + \lambda \partial_{xyy}^3 \psi^{n+1} \right) \\ \partial_t (\Delta \psi^{n+1}) + \partial_x \omega^{n+1} = 0 \end{cases} \quad (1.142)$$

En négligeant le terme de diffusion dans la première équation, on obtient une approximation de $\tilde{\omega}^{n+1}$:

$$\tilde{\omega}^{n+1} = \omega^n - \frac{v_e}{\varepsilon} \Delta t \left((\lambda - 1) \partial_{xxx}^3 \psi^{n+1} + \lambda \partial_{xyy}^3 \psi^{n+1} \right) \quad (1.143)$$

En injectant cette expression de $\tilde{\omega}^{n+1}$ dans la deuxième équation, on obtient une prédiction $\tilde{\psi}^{n+1}$ de ψ^{n+1} comme solution de :

$$\Delta \tilde{\psi}^{n+1} - \frac{v_e}{\varepsilon} \Delta t^2 \left((\lambda - 1) \partial_{xxxx}^4 \tilde{\psi}^{n+1} + \lambda \partial_{xxyy}^4 \tilde{\psi}^{n+1} \right) = \Delta \psi^n - \Delta t \partial_x \omega^n \quad (1.144)$$

On utilise l'équation (1.144) comme prédiction et le schéma semi-implicite associé au système (1.116) s'écrit :

$$\begin{cases} \Delta \tilde{\psi}^{n+1} - \frac{v_e}{\varepsilon} \Delta t^2 \left[(\lambda - 1) \partial_{xxxx}^4 \tilde{\psi}^{n+1} + \lambda \partial_{xxyy}^4 \tilde{\psi}^{n+1} \right] = \Delta \psi^n - \Delta t \partial_x \omega^n \\ \frac{\omega^{n+1} - \omega^n}{\Delta t} - \mu \Delta \omega^{n+1} = -\frac{v_e}{\varepsilon} \left[(\lambda - 1) \partial_{xxx}^3 \tilde{\psi}^{n+1} + \lambda \partial_{xyy}^3 \tilde{\psi}^{n+1} \right] \\ \frac{\Delta \psi^{n+1} - \Delta \psi^n}{\Delta t} + \partial_x \omega^{n+1} = 0 \end{cases} \quad (1.145)$$

On procède alors à la même étude que pour le schéma explicite (1.118). On choisit $\varepsilon = \Delta x$ et on utilise les notations suivantes : $\alpha_k = \frac{4s_k^2}{\Delta x^2}$, $\alpha_h = \frac{4s_h^2}{\Delta x^2}$, $\alpha = \alpha_k + \alpha_h$ et $\gamma = \frac{v_e \alpha_k}{\Delta x} [(\lambda - 1)\alpha_k + \lambda \alpha_h]$. Le système linéaire peut ainsi se reformuler sous la forme :

$$A_{k,h} \begin{pmatrix} \tilde{\omega}_{j,l}^{n+1}(k,h) \\ \tilde{\psi}_{j,l}^{n+1}(k,h) \end{pmatrix} = B_{k,h} \begin{pmatrix} \tilde{\omega}_{j,l}^n(k,h) \\ \tilde{\psi}_{j,l}^n(k,h) \end{pmatrix} \quad (1.146)$$

où :

$$A_{k,h} = \begin{pmatrix} 1 + \mu \alpha \Delta t & 0 \\ -\frac{\Delta t}{\Delta x} s_{2k} i & \alpha \end{pmatrix} \quad (1.147)$$

et

$$B_{k,h} = \begin{pmatrix} \frac{\alpha + \gamma \Delta t^2 s_k^2}{\alpha + \gamma \Delta t^2} & \alpha \frac{v_e [(\lambda - 1)\alpha_k + \lambda \alpha_h] \Delta t s_{2k} i}{(\alpha + \gamma \Delta t^2) \Delta x^2} \\ 0 & \alpha \end{pmatrix} \quad (1.148)$$

On cherche les valeurs propres de la matrice $A_{k,h}^{-1} B_{k,h}$ qui vérifient $\det(B_{k,h} - rA_{k,h}) = 0$, ce qui donne :

$$0 = \begin{vmatrix} \frac{\alpha + \gamma \Delta t^2 s_k^2}{\alpha + \gamma \Delta t^2} - r(1 + \alpha \mu \Delta t) & \alpha \frac{v_e [(\lambda-1)\alpha_k + \lambda\alpha_h] s_{2k} i}{(\alpha + \gamma \Delta t^2) \Delta x^2} \\ r s_{2k} \frac{\Delta t}{\Delta x} i & (1-r)\alpha \end{vmatrix}$$

$$= \alpha \left((1 + \alpha \mu \Delta t) r^2 - \left[1 + \mu \alpha \Delta t + \frac{\alpha + \gamma \Delta t^2 s_k^2}{\alpha + \gamma \Delta t^2} - \frac{\gamma \Delta t^2 c_k^2}{\alpha + \gamma \Delta t^2} \right] r + \frac{\alpha + \gamma \Delta t^2 s_k^2}{\alpha + \gamma \Delta t^2} \right) \quad (1.149)$$

En posant $a = \frac{(\alpha + \gamma s_k^2)(1 + \alpha \mu \Delta t)}{\alpha + \gamma \Delta t^2}$ et $K = \frac{\gamma \Delta t^2 c_k^2}{\alpha + \gamma \Delta t^2 s_k^2}$ et en factorisant par $\frac{\alpha + \gamma \Delta t^2 s_k^2}{\alpha + \gamma \Delta t^2}$, on se retrouve à étudier les racines d'un polynôme du type $aX^2 - (1 + a - K)X + 1$. On a déjà été confronté à un polynôme sous cette forme dans le cas explicite. On peut donc reprendre la condition de stabilité (1.129) donnée par la discussion menée précédemment. En la multipliant par $\frac{\alpha + \gamma \Delta t^2 s_k^2}{\alpha + \gamma \Delta t^2}$, cela se traduit :

$$\mu \alpha \gamma \Delta t^3 + 2 \gamma s_k^2 \Delta t^2 + \mu \alpha \Delta t + 2\alpha \geq 0 \quad (1.150)$$

L'inéquation (1.150) est toujours vérifiée pour des Δt positifs car tous les coefficients sont positifs. Ainsi, le schéma semi-implicite est inconditionnellement stable.

1.6.3 Résultats numériques du schéma de prédiction

On présente ici une première étude justifiant l'intérêt d'étudier les schémas de prédiction. On simule le comportement d'une goutte de fluide en suspension dans un autre fluide. L'objet est de vérifier l'amélioration du pas de temps limite de stabilité avec l'équation de prédiction.

On ne rentre pas dans les détails de l'implémentation, qui seront présentés au Chapitre 3.

On considère une bulle de fluide de densité 1 et viscosité 0.01 immergée dans un fluide ayant les mêmes propriétés. La tension de surface est fixée à 10.

L'état initial est une bulle de forme circulaire immergée dans l'autre fluide. Elle va être transportée de gauche à droite en conservant sa forme. Conserver la forme circulaire de la bulle est la difficulté de ce test.

L'intérêt de ce test est que sa stabilité s'observe très facilement, cf les figures 1.16 et 1.15.

On peut évaluer la conservation de la forme circulaire avec la circularité, qui est le taux de remplissage ν en dimension 2. On rappelle que cela correspond au rapport entre l'aire de notre objet et celui du cercle de même périmètre. Comme l'aire et le périmètre sont calculés numériquement, on peut obtenir des valeurs légèrement supérieures à 1.

Les pas de temps limites de stabilité observés sont :



FIGURE 1.15 – Simulation du schéma explicite (1.103) pour un pas de temps $dt = 0.0035$ supérieur au seuil limite. La bulle explose rapidement.

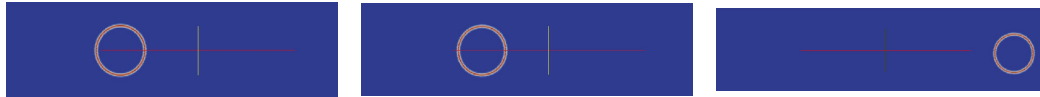


FIGURE 1.16 – Simulation du schéma de prédiction (1.141) pour un pas de temps $dt = 0.0055$ supérieur au seuil limite du schéma explicite. À noter une légère perte de volume.

Temps	Circularité Schéma Explicite	Circularité Schéma Semi-implicite
0	1	1
0.035	1.0009	1.002
0.07	1.0010	1.002
0.105	1.0006	1.002
0.14	1.0001	1.003
0.175	0.998	1.004
0.21	0.993	1.002
0.245	0.978	1.003
0.28	0.953	1.003
0.315	0.898	1.004
0.35	0.878	1.005
0.385	0.853	1.005
0.42	0.830	1.004

TABLE 1.1 – Évolution de la circularité de la bulle toutes les 10 itérations pour les deux méthodes. À gauche, le schéma explicite et à droite le schéma semi-implicite.

- Pour le schéma explicite (1.103) : $dt = 3.10^{-3}$.
- Pour le schéma prédictif (1.141) : $dt = 7.10^{-3}$.

Le nombre de Reynolds de notre simulation est de l'ordre de 100. Dans ce cas, on a vu que la condition CFL pour le schéma explicite :

$$\Delta t \lesssim C\Delta x^{\frac{3}{2}} \quad (1.151)$$

et pour le schéma semi-implicite :

$$\Delta t \lesssim C'\Delta x \quad (1.152)$$

En diminuant par 2 la taille du maillage

- Pour le schéma explicite (1.103) : $dt = 1, 2 \cdot 10^{-3}$. Cela correspond bien à ce qui était attendu car le pas de temps limite doit être divisé par $2^{\frac{3}{2}} \simeq 2.8$.
- Pour le schéma prédictif (1.141) : $dt = 3, 3 \cdot 10^{-3}$. Cela correspond bien à ce qui était attendu car le pas de temps limite doit être divisé par 2.

Conformément à l'étude analytique, on observe une meilleure stabilité du schéma de prédiction par rapport au schéma explicite. De plus, son coût algorithmique est très proche car le temps de calcul lié à la résolution de l'équation de la chaleur est très petit comparé à la résolution de l'équation de Navier-Stokes. Comme il est possible d'utiliser des plus grands pas de temps, on obtient des simulations plus efficaces globalement.

1.6.4 Lien avec les méthodes de mouvements générés par diffusion

Conceptuellement, le schéma de prédiction consiste à prédire la position future de la membrane élastique en résolvant une équation de la chaleur. Son obtention, consistant essentiellement à négliger certains termes physiques de l'équation fluide, peut sembler un peu arbitraire; et une généralisation de la méthode prenant en compte l'effet de la force de courbure n'est pas immédiate.

On va consacrer le deuxième chapitre à l'étude des mouvements générés par diffusion. Cette classe d'algorithmes, introduite par Merriman, Bence et Osher dans [MBO92, MBO94], consiste à générer des mouvements géométriques en alternant résolution d'une équation de la chaleur et étape de seuillage.

Il s'agira alors de voir comment on peut rapprocher l'étape de prédiction de la méthode semi-implicite avec ces méthodes. Ensuite, on verra comment étendre ce principe pour proposer un schéma semi-implicite de prédiction pour le système de globule rouge (1.81).

CHAPITRE 2

MOUVEMENT D'INTERFACE GÉNÉRÉ PAR DIFFUSION

Sommaire

2.1	Introduction : le schéma de Merriman,Bence,Osher	81
2.2	Intuition	83
2.3	Méthode de Convolution-Seuillage	85
2.3.1	Mouvement par courbure moyenne multiphasique avec jonctions	85
2.3.2	Mouvement par courbure moyenne avec conservation de volume	87
2.3.3	Mouvement par flot de Willmore	89
2.3.4	Un défaut majeur : la précision proche de l'interface.	93
2.4	Méthode de Convolution-Redistanciation	94
2.4.1	Développement de la convolution de la fonction distance au voisinage de l'interface	95
2.4.2	Développement de la convolution d'une fonction $f(d)$ au voisinage de l'interface	100
2.4.3	Mouvement par courbure moyenne par Convolution-Redistanciation	101
2.4.4	Vérification numérique de convergence sur le cercle	102
2.4.5	Mouvement par courbure moyenne en dimension 2	103
2.4.6	Extraction de la courbure par diffusion	105
2.4.7	Mouvement par courbure moyenne avec conservation de volume par convolution-redistanciation	109

2.4.8	Flot de Willmore par convolution-redistanciation en dimension 2	110
2.4.9	V�rification num�rique du mouvement par flot de Willmore en dimension 2	111
2.4.10	Flot de Willmore par convolution-redistanciation en dimension 3	112
2.4.11	V�rification num�rique du mouvement par flot de Willmore en dimension 3	114
2.5	Convergence th�orique du mouvement par courbure moyenne	118
2.5.1	Notations	119
2.5.2	L'espace des fonctions � variations born�es $BV(\Omega)$. . .	119
2.5.3	L'approche de Almgren-Taylor-Wang et Luckhaus-Sturzenhecker	120
2.5.4	L'approche de Laux-Otto	124
2.5.5	L'approche de Chambolle	126
2.5.6	Convergence du sch�ma convolution-redistanciation .	129

2.1 Introduction : le sch ma de Merriman, Bence, Osher

Dans [MBO92], Merriman, Bence et Osher ont introduit un algorithme simple   impl menter pour simuler le d placement par courbure moyenne d'une interface Γ . On note Ω l'int rieur de l'objet d limit  par Γ et χ la fonction caract ristique de Ω .

 tant donn  un pas de temps δt , l'algorithme g n re une approximation discr te en temps $(\Gamma_k)_k$ du mouvement par courbure moyenne, o  Γ_k est la position au temps $t = k\delta t$ de l'interface, c'est- -dire que le d placement de l'interface est donn  par la vitesse normale :

$$v = H \quad (2.1)$$

Le principe de l'algorithme est d'alterner les deux  tapes suivantes :

- 1) On diffuse χ^k par l' quation de la chaleur pendant un pas de temps δt :

$$\begin{cases} \partial_t \omega(t, x) - \Delta \chi(t, x) = 0 \\ \chi(0, x) = \chi^k(x) \end{cases} \quad (2.2)$$

On note $\omega = \chi(\delta t, \cdot)$ la fonction ainsi obtenue.

- 2) On obtient Ω_{k+1} en seuillant la fonction ω   la valeur $\frac{1}{2}$:

$$\Omega_{k+1} = \left\{ x : \omega(x) \geq \frac{1}{2} \right\} \quad (2.3)$$

En r p tant l'alternance de ces deux  tapes, on obtient une approximation du mouvement par courbure moyenne de l'interface Γ . Chaque it ration g n re la nouvelle position de l'interface au temps $t^{k+1} = t^k + \delta t$.

Les atouts de cette m thode sont sa simplicit  conceptuelle et son efficacit  en termes de co t num rique. En effet, la premi re  tape consiste   r soudre une  quation de la chaleur pendant un pas de temps δt . La r solution de l' quation de transport classique de la m thode *levelset* implique la r solution d'un probl me non lin aire plus compliqu  ou d'expliciter le terme de transport $H(\phi) \frac{\nabla \phi}{|\nabla \phi|}$ qui implique le calcul de la courbure moyenne.

La r solution de l' quation de la chaleur peut  tre fait tr s efficacement en utilisant par exemple une m thode de "Transform e de Fourier Rapide" sur maillage cart sien. Elle consiste   calculer la solution de (2.2) sous la forme du produit de convolution :

$$\forall x \in \mathbb{R}^d, \omega(x) = \left(G_{\delta t} * \chi^k \right) (x) \quad (2.4)$$

o  $G_t(x)$ repr sente le noyau de la chaleur, d fini comme suit :

$$G_t(x) = \frac{1}{(4\pi t)^{\frac{d}{2}}} e^{-\frac{|x|^2}{4t}}. \quad (2.5)$$

Dans le cadre d'une résolution de l'équation (2.2) par éléments finis, le système est facile à inverser car le schéma approchant l'équation (2.2) est inconditionnellement stable et seules des considérations de précisions sont nécessaires. Si on utilise le même pas de temps, il est possible de réutiliser le même préconditionneur à chaque itération.

Cette première étape, du fait des différentes méthodes pour résoudre (2.2), sera désormais désignée indifféremment comme "étape de convolution" ou "étape de diffusion". C'est lors de cette étape que l'interface est déplacée. La deuxième étape (2.3), appelée "étape de seuillage", permet de revenir à la condition initiale d'une fonction indicatrice. D'un point de vue complexité, elle est linéaire en le nombre d'éléments. On parlera désormais de l'algorithme de "Diffusion-Seuillage", de "Convolution-Seuillage" ou encore MBO (Merriman-Bence-Osher) pour désigner l'algorithme suivant :

Algorithme 2 : MBO : Étant donnés la fonction caractéristique de l'ensemble Ω_0 d'interface Γ_0 , un pas de temps δt et un nombre d'itérations nb_{iter} , générer la suite d'ensembles $(\Omega_k)_k$ décrits par leurs fonctions caractéristiques χ_k .

Pour k de 0 à $nb_{iter} - 1$:

1) Étape de convolution : $\omega_k = G_{\delta t} * \chi_k$.

2) Étape de seuillage : $\chi_{k+1} = \mathbf{1}_{\{\omega_k(x) \geq \frac{1}{2}\}}$

L'algorithme 2.1 : 2 constitue la base des méthodes de convolution-seuillage que l'on va étudier dans ce chapitre. On commence par quelques remarques destinées à se convaincre qu'il déplace effectivement l'interface selon une approximation du mouvement par courbure moyenne.

On s'intéressera ensuite aux différentes extensions existantes : les déplacements par courbure moyenne multiphasiques, le mouvement par courbure moyenne avec conservation de volume et les flots d'ordres supérieurs. On citera également quelques exemples d'applications des méthodes de convolution-seuillage à d'autres phénomènes que le mouvement d'une membrane.

Le choix de représenter le domaine par une fonction caractéristique possède un défaut majeur : on ne peut avoir que 0 ou 1 comme valeur à l'issue de l'étape de seuillage. Si la discrétisation numérique est insuffisamment fine, l'interface peut alors se retrouver "bloquée". On abordera ce problème au paragraphe 1.3.4. Cela se produit si le déplacement est inférieur à la taille d'une maille. En effet, l'étape de seuillage réassignera la valeur de chaque sommet du maillage à sa valeur initiale et l'interface sera immobile, Fig.2.6. Une solution à ce problème consiste à raffiner le maillage au niveau de l'interface. Il est alors nécessaire de remailler à chaque itération pour suivre les mouvements de l'interface et cela peut devenir problématique en 3D.

Une alternative explorée par Esedoglu, Ruuth, Tsai [ERT⁺10] est de diffuser la fonction distance signée à l'interface plutôt que la fonction indicatrice et de

remplacer l' tape de seuillage par une redistanciation. L'avantage est que la fonction distance contient plus d'informations   chaque sommet du maillage que la valeur binaire de la fonction indicatrice. Ainsi, m me pour des faibles d placements, la valeur au niveau des  l ments proches de l'interface est modifi e et l' tape de redistanciation prend en compte ces changements, contrairement   l' tape de seuillage. On s'int ressera alors aux mouvements qu'il est possible d'obtenir en diffusant une fonction distance.

Apr s cela, on pr sentera les  tudes faites sur la convergence th orique de ces m thodes num riques vers le flot continu associ  pour les m thodes de convolution-seuillage. Elles reposent essentiellement sur un principe de comparaison ou une interpr tation de la fonction g n r e par l'algorithme comme minimisation d'une certaine  nergie discr tis e.

2.2 Intuition

On d crit ici les observations faites dans [MBO92] qui ont amen  Merriman, Bence, Osher qui justifient en 2D que l'algorithme 2.1 : 2 simule bien un mouvement par courbure moyenne.

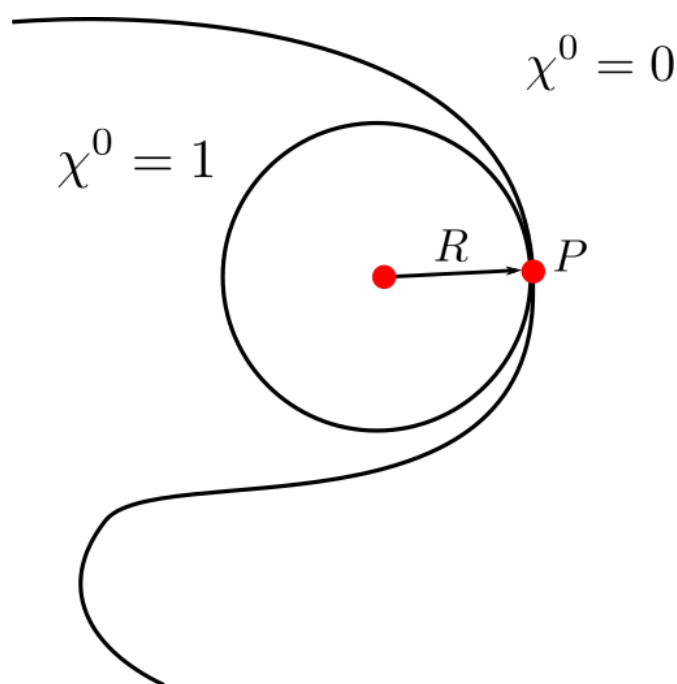


FIGURE 2.1 – Configuration dans un voisinage du point P .

 tant donn  un certain point P localis  sur l'interface, nous  tudions comment il est affect  par une it ration de l' tape de diffusion. On consid re le cercle de courbure associ    la courbe au point P , voir la figure 2.1 et on  crit l' quation

en coordonn es cylindriques. En utilisant la sym trie radiale, on a :

$$\partial_t \chi + \underbrace{\frac{1}{r} \partial_r \chi}_{(1)} - \underbrace{\partial_{rr} \chi}_{(2)} = 0 \quad (2.6)$$

Il s'agit d'une  quation d'avection diffusion 1D selon la direction radiale. Chaque point est d plac  dans la direction normale   la vitesse $\frac{1}{r}$. Ainsi, le point P est d plac  de $\frac{1}{R} = H$.

On peut  galement d crire le comportement de l'algorithme par une vue de profil, cf Fig.2.2.

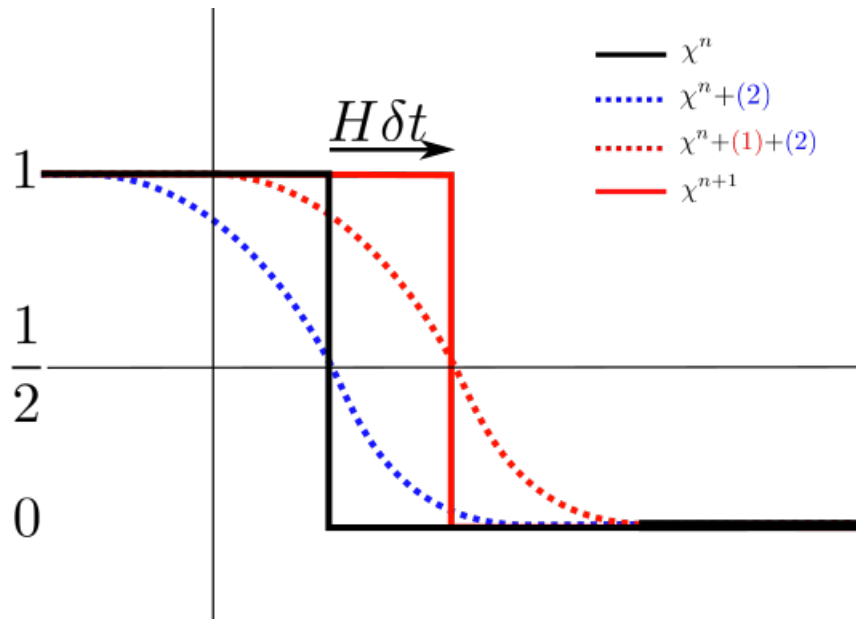


FIGURE 2.2 – Profil d'une  tape de l'algorithme de Diffusion-Seuillage. La courbe noire d crit la fonction caract ristique initiale. La courbe en pointill s bleus repr sente la fonction soumise   une diffusion radiale pure qui ne d place pas l'interface. En pointill s rouges, on obtient le r sultat de l' quation (2.6) et la courbe rouge repr sente le seuillage de cette fonction.

Une autre observation faite par Merriman, Bence et Osher dans leur article [MBO92] permet de justifier intuitivement l'impl mentation du sch ma 2.1 : 2. Une m thode bas e sur un mod le de champ de phases d crit le mouvement par courbure moyenne gr ce   l' quation d'Allen-Cahn :

$$\partial_t \chi - \varepsilon^2 \Delta \chi = \chi(\chi - 1) \left(\chi - \frac{1}{2} \right) \quad (2.7)$$

Il s'agit en fait du flot de gradient \mathbb{L}^2 de l' nergie de Ginzburg-Landau (1.86) vue au chapitre pr c dent modifi e de telle mani re que les phases valent 0 et 1 au lieu de -1 et 1.

Des schémas de splitting alternant une diffusion linéaire et une étape de réaction rapide existent pour résoudre efficacement (2.7). La résolution de l'étape de réaction peut se faire par exemple en résolvant une équation différentielle ordinaire à chaque point du maillage. L'intuition de Merriman, Bence et Osher est que, pour un pas de temps de diffusion suffisamment grand par rapport au paramètre ε , on peut remplacer la résolution de l'équation différentielle de réaction par les valeurs asymptotiques en temps de celle-ci. Cela correspond à faire une étape de seuillage pour séparer les deux phases.

2.3 Méthode de Convolution-Seuillage

Depuis sa genèse en 1992 dans [MBO92], le schéma 2.1 : 2, du fait de sa simplicité conceptuelle et de son efficacité numérique, a rapidement gagné en popularité ; différentes extensions ont vu le jour pour simuler des mouvements géométriques plus complexes tels que le flot de Willmore [GH08, ERT08] ou le mouvement par courbure moyenne avec conservation de volume [RW03]. On commence cette section par présenter les différentes extensions du schéma de convolution-seuillage. On conclura cette section par l'étude du défaut majeur de l'algorithme de convolution-seuillage : la résolution proche de l'interface.

2.3.1 Mouvement par courbure moyenne multiphasique avec jonctions

On s'intéresse ici au cas de plusieurs phases soumises à un déplacement par courbure moyenne formant une partition de l'espace et en tension entre elles, c'est-à-dire soumis à un déplacement normal donné par :

$$v_i = \sigma_{ij} H_i \quad (2.8)$$

Il convient de noter que cela ne s'applique pas vraiment pour le cas de plusieurs globules rouges en suspension dans un fluide. En effet, il s'agit ici d'un réseau d'interfaces qui se déplacent par courbure moyenne. Lors du couplage fluide-structure contenant plusieurs structures élastiques, les différents globules ne sont pas en contact direct les uns avec les autres car ils sont en tension avec le fluide. Cette formulation est surtout utilisée pour modéliser les phénomènes de croissance cristalline [Mul56].

Le cas qui nous intéresse dans cette partie est le cas où l'espace est rempli de différentes phases se déplaçant par courbure moyenne. On se réfère à la figure 2.3 pour une illustration.

La difficulté principale de l'algorithme multiphase est la façon dont on traite le cas d'une jonction entre trois phases ou plus. La condition à respecter, communément appelée condition d'angle d'Herring [Her99], s'exprime : Pour

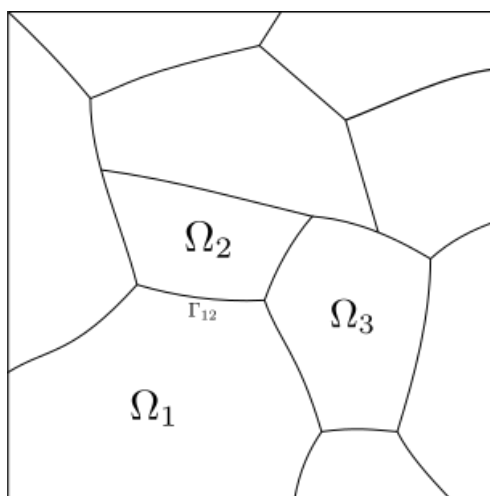


FIGURE 2.3 – Configuration du cas multiphasique. L'espace est recouvert de phases Ω_i se d pla ant par courbure moyenne et en tension les unes avec les autres.

tous x point de jonction entre trois phases, cf Fig.2.4 :

$$\sigma_{12}n_{12}(x) + \sigma_{23}n_{23}(x) + \sigma_{31}n_{31}(x) = 0 \quad (2.9)$$

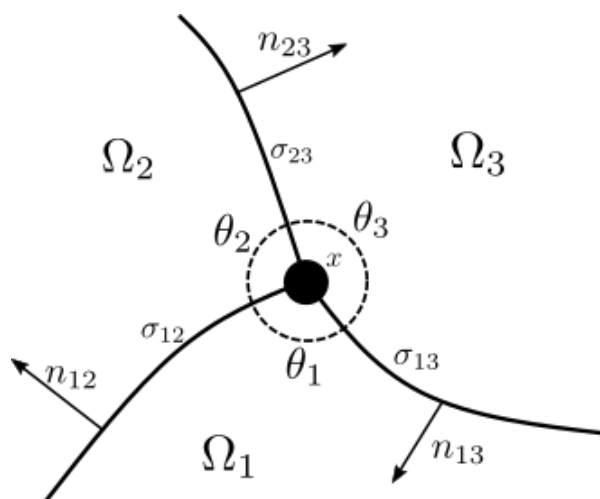


FIGURE 2.4 – Configuration   un point de jonction entre 3 phases. Les angles θ_i sont d termin s par la relation (2.10).

On d duit de cette  quation les angles form s entre chaque phase la condition suivante :

$$\frac{\sin(\theta_1)}{\sigma_{23}} = \frac{\sin(\theta_2)}{\sigma_{31}} = \frac{\sin(\theta_3)}{\sigma_{12}} \quad (2.10)$$

La th orie concernant quatre phases ou plus se rencontrant est encore incompl te

et le comportement de l'algorithme est laiss  libre. Ce cas se produit pourtant m me si la condition initiale n'en contient pas.

L'algorithme 2.3.1 : 3 propos  dans [MBO94, Mas92] utilisant le seuillage (2.11) est valide pour des tensions de surface identiques, mais ne permet pas de mod liser le comportement dans le cas g n ral. Cela s'explique car les diff rents coefficients de tension de surface ne sont pas pris en consid ration dans l' tape de diffusion. La base de l'algorithme multiphase reste la m me que celle de l'algorithme diphasique. Consid rons χ_i, χ_j les fonctions caract ristiques de deux phases dont la fronti re commune Γ_{ij} est non vide. On pose $\omega_i = G_{\delta t} * \chi_i$ et $\omega_j = G_{\delta t} * \chi_j$ les fonctions diffus es de χ_i et χ_j et on propose l' tape de seuillage suivante :

$$\chi_i^{n+1} = \mathbf{1}_{\{\omega_i \geq \omega_k / \forall k \neq i\}} \text{ et } \chi_j^{n+1} = \mathbf{1}_{\{\omega_j \geq \omega_k / \forall k \neq j\}} \quad (2.11)$$

Notre  tape de seuillage consiste   choisir en chaque point la phase dont la valeur de la fonction diffus e est maximale. Ce choix est l gitim  par l'observation suivante : dans le cas biphasique avec $\chi_i = \chi$ et $\chi_j = 1 - \chi$, l' tape de seuillage revient   comparer ω et $1 - \omega$, c'est- -dire d terminer si ω est sup rieur   $\frac{1}{2}$. Le seuillage (2.11) correspond   l'algorithme MBO classique.

Algorithme 3 : Multiphase avec tensions de surfaces  gales :  tant donn s la partition de l'espace form e par les fonctions caract ristiques $\chi_1^0, \chi_2^0, \dots, \chi_p^0$, un pas de temps δt et un nombre d'it rations nb_{iter} , g n rer la suite d'ensembles $(\Omega_k)_k$ d crits par leurs fonctions distances sign es d^k .

Pour k de 0   $nb_{iter} - 1$:

1)  tape de convolution : $\omega_i^k = G_{\delta t} * \chi_i^k$.

2)  tape de seuillage : $\chi_i^{k+1} = \mathbf{1}_{\{\omega_i^k < \omega_j^k / \forall j \neq i\}}$.

Dans [EO15], un algorithme est propos  pour g rer le cas de tensions de surface strictement positives et v rifiant l'in galit  triangulaire suivante :

$$\forall (i, j, k), \sigma_{ij} + \sigma_{jk} \geq \sigma_{ik} \quad (2.12)$$

ces conditions  tant pour assurer le caract re bien pos  du probl me.

L'id e de l'algorithme 2.3.1 : 4 est d'introduire une d pendance en la tension de surface dans l' tape de diffusion et diffuser. On se r f re   [EO15] pour plus de d tails sur les propri t s de l'algorithme 2.3.1 : 4.

2.3.2 Mouvement par courbure moyenne avec conservation de volume

Le mouvement par courbure moyenne avec conservation de volume d'une interface se caract rise par la vitesse normale :

$$v = H - \bar{H} \quad (2.13)$$

Algorithme 4 : Multiphase g n ral :  tant donn s la partition de l'espace form e par les fonctions caract ristiques $\chi_1^0, \chi_2^0, \dots, \chi_p^0$, un pas de temps δt et un nombre d'it rations nb_{iter} , g n rer la suite d'ensembles $(\Omega_k)_k$ d crits par leurs fonctions distances sign es d^k .

Pour k de 0   $nb_{iter} - 1$:

- 1)  tape de convolution : $\omega_i^k = G_{\delta t} * \left(\sum_{j=1}^p \sigma_{ij} \chi_j^k \right)$.
 - 2)  tape de seuillage : $\chi_i^{k+1} = \mathbf{1}_{\{\omega_i^k < \omega_j^k / \forall j \neq i\}}$
-

o  \bar{H} d signe l'int grale de la courbure moyenne sur l'interface : $\frac{1}{A(\Gamma)} \int_{\Omega} H(x) dx$.

L'id e de [RW03] est de modifier la valeur de seuil, c'est- -dire trouver la constante λ telle que $V(\{w_k(x) \geq \lambda\}) = V_0$ et seuiller   cette valeur.

Algorithme 5 : Courbure moyenne avec conservation de volume :  tant donn s la fonction caract ristique χ_0 de l'ensemble Ω_0 d'interface Γ_0 et de volume V_0 , un pas de temps δt et un nombre d'it rations nb_{iter} , g n rer la suite d'ensembles $(\Omega_k)_k$ d crits par leurs caract ristiques χ^k .

Pour k de 0   $nb_{iter} - 1$:

- 1)  tape de convolution : $w_k(x) = \frac{1}{N-1} G_{\delta t} * \chi^k$.
 - 2) Trouver λ tel que $V(\{w_k(x) \geq \lambda\}) = V_0$.
 - 3)  tape de seuillage : $\chi^{k+1} = \mathbf{1}_{\{w_k(x) \geq \lambda\}}$
-

En dimension 2, la valeur de seuillage peut  tre obtenue imm diatement par l'application du th or me de Gauss-Bonnet. Elle relie \bar{H} au p rim tre en fonction du nombre g n rique g de la courbe :

$$\bar{H} = \frac{2\pi g}{A(\Gamma)} \quad (2.14)$$

Ainsi pour une courbe ferm e sans trou, $\lambda = \frac{2\pi}{A(\Gamma^k)}$. En dimension 3, on ne peut plus utiliser cette formule, il faut donc calculer la courbure moyenne et son int grale.

Dans [RW03] λ est obtenue par une m thode de Newton.

Remarque 2.3.1. Comme la fonction $G_{\delta t} * \chi$ est r guli re et positive, la fonction $\lambda \mapsto V(\{G_{\delta t} * \chi \geq \lambda\})$ est continue et varie entre 0 et $V(\Omega)$. Ainsi, λ existe et est unique pour une valeur V_0 appartenant   $[0; V(\Omega)]$, ce qui est le cas en pratique.

2.3.3 Mouvement par flot de Willmore

On rappelle que la motivation de l' tude des sch mas de convolution-seuillage est la suivante : on cherche   pr dire efficacement la position future de la membrane d'un globule pour am liorer le traitement num rique du couplage avec le fluide. Le d placement des globules rouges tend   suivre le gradient de l' nergie d'Helfrich (1.9), dont on rappelle l'expression :

$$W = \begin{cases} \Delta_\Gamma \kappa + \frac{\kappa^3}{2} \\ \frac{1}{2} \Delta_\Gamma \kappa + \frac{1}{2} \kappa (\kappa^2 - 4K) \end{cases} = \begin{cases} \Delta_\Gamma H + \frac{H^3}{2} \text{ en dimension 2} \\ \Delta_\Gamma H + 2H (H^2 - K) \text{ en dimension 3} \end{cases} \quad (2.15)$$

L'objectif est donc d'obtenir ce d placement normal par une m thode de convolution-seuillage. Il s'agit d'un mouvement plus complexe que le mouvement par courbure moyenne car il n cessite de d river deux fois la courbure moyenne et contient un terme non lin aire en elle.

En l' tat, il n'est pas  vident de g n raliser le raisonnement appliqu  au mouvement par courbure moyenne   un mouvement d'ordre sup rieur. La premi re  tape consiste   voir comment le terme de Willmore intervient dans la diffusion d'une fonction caract ristique. Il s'agit en fait du terme d'ordre deux en temps dans le d veloppement limit  de la fonction χ satisfaisant :

$$\begin{cases} \partial_t \chi(t, x) - \Delta \chi(t, x) = 0 \\ \chi(0, x) = \chi^k(x) \end{cases} \quad (2.16)$$

Plus pr cis ment, la proposition suivante fournit un d veloppement limit  de χ en dimensions 2 et 3. Elle est bas e sur les articles [GH08] et [ERT08].

Proposition 2.3.2. *En dimension 2, on a le d veloppement au voisinage de l'interface pour la fonction χ solution de (2.16) :*

$$\chi(x, \delta t) = G_{\delta t} * \chi^k(x) = \frac{1}{2} - \frac{1}{2\sqrt{\pi}} y \delta t^{-\frac{1}{2}} + \frac{1}{2\sqrt{\pi}} H \delta t^{\frac{1}{2}} - \frac{1}{4\sqrt{\pi}} W \delta t^{\frac{3}{2}} + O\left(\delta t^{\frac{5}{2}}\right) \quad (2.17)$$

En dimension 3, on a le d veloppement au voisinage de l'interface pour la fonction χ solution de (2.16) :

$$\chi(x, \delta t) = G_{\delta t} * \chi^k(x) = \frac{1}{2} - \frac{1}{2\sqrt{\pi}} y \delta t^{-\frac{1}{2}} + \frac{1}{\sqrt{\pi}} H \delta t^{\frac{1}{2}} - \frac{1}{4\sqrt{\pi}} W \delta t^{\frac{3}{2}} + O\left(\delta t^{\frac{5}{2}}\right) \quad (2.18)$$

D monstration : Proposition

On pr sente ici la d monstration de [GH08] par souci p dagogique. On effectue

la d monstration dans le cas de la dimension 2 pour plus de simplicit  dans les notations. Le cas de la dimension 3 se traite de mani re similaire.

La fonction χ satisfaisant l' quation de la chaleur (2.16) peut  tre repr sent e sous forme du produit de convolution avec le noyau de la chaleur :

$$\chi(x, t) = G_{\delta t} * \chi^k(x, \delta t) \quad (2.19)$$

On se concentre sur le d placement normal engendr  par (2.16) pour un certain point P de l'interface   l'instant $t^k = k\delta t$. Pour ce faire, on recentre l'espace en ce point. On suppose que l'interface Γ est suffisamment r guli re, au moins C^5 pour les d rivations qui suivront, et peut ainsi  tre repr sent e comme le graphe d'une fonction f au voisinage de P . On r oriente l'espace afin que la courbe soit tangente   l'origine et que la normale sortante soit le vecteur $(0, 1)$. On r f re   la figure 2.5. On peut alors r crire le produit de convolution en fonction de f :

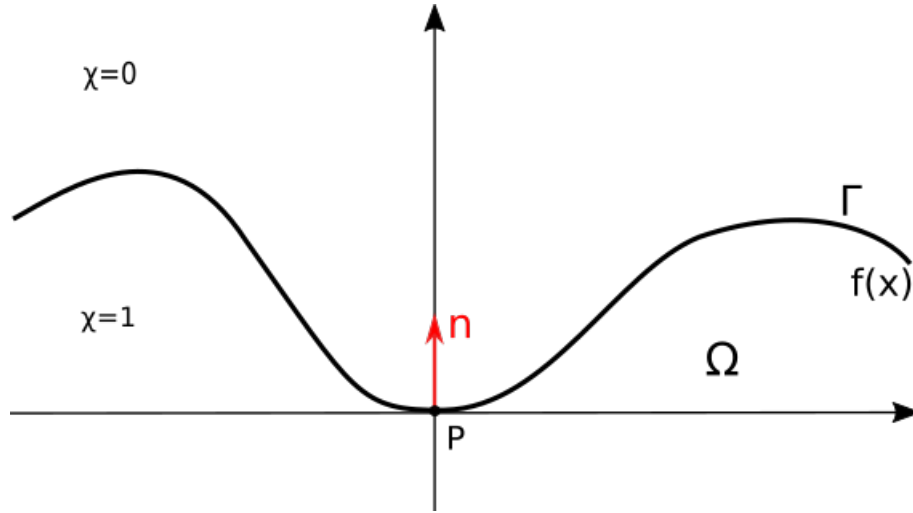


FIGURE 2.5 – Configuration de l' tude.

$$\chi(x, y, t) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{1}{4\pi t} e^{-\frac{(x-\xi)^2+(y-\eta)^2}{4t}} \mathbf{1}_{\{f(\xi) < \eta\}}(\xi, \eta) d\xi d\eta \quad (2.20)$$

On cherche    tudier le d placement normal de l'origine, c' st- -dire le comportement de χ le long de l'axe des y . De plus, on cherche un d placement v rifiant $y = O(t^2)$. Il suffit donc d'exprimer $\chi(0, y, t)$:

$$\begin{aligned} \chi(0, y, t) &= \frac{1}{4\pi t} \int_{-\infty}^{\infty} \int_{-\infty}^0 e^{-\frac{\xi^2+(y-\eta)^2}{4t}} d\eta d\xi + \frac{1}{4\pi t} \int_{-\infty}^{\infty} \int_0^{f(\xi)} e^{-\frac{\xi^2+(y-\eta)^2}{4t}} d\eta d\xi \\ &= \frac{1}{2\sqrt{\pi t}} \int_0^{+\infty} e^{-\frac{\eta^2}{4t}} d\eta - \frac{1}{2\sqrt{\pi t}} \int_0^y e^{-\frac{\eta^2}{4t}} d\eta + \frac{1}{4\pi t} \int_{-\infty}^{\infty} \int_0^{f(\xi)} e^{-\frac{\xi^2+(y-\eta)^2}{4t}} d\eta d\xi \\ &= \frac{1}{2} - \frac{1}{2\sqrt{\pi t}} \int_0^y e^{-\frac{\eta^2}{4t}} d\eta + \frac{1}{4\pi t} \int_{-\infty}^{\infty} \int_0^{f(\xi)} e^{-\frac{\xi^2+(y-\eta)^2}{4t}} d\eta d\xi \end{aligned} \quad (2.21)$$

En d veloppant l'exponentielle sous la premi re int grale du terme de droite de (2.21) :

$$\forall \eta \in [0; y], e^{-\frac{\eta^2}{4t}} = 1 - \frac{\eta^2}{4t} + \mathcal{O}\left(\frac{\eta^4}{t^2}\right) = 1 + \mathcal{O}\left(t^{\frac{5}{2}}\right) \quad (2.22)$$

On a alors :

$$\frac{1}{2\sqrt{\pi t}} \int_0^y e^{-\frac{\eta^2}{4t}} d\eta = \frac{1}{2\sqrt{\pi t}} y + \mathcal{O}\left(t^{\frac{5}{2}}\right) \quad (2.23)$$

Pour la deuxi me int grale, on d veloppe   la fois l'exponentielle dans l'int grale ainsi que la borne $f(\xi)$:

$$\begin{aligned} \int_{-\infty}^{\infty} \int_0^{f(\xi)} e^{-\frac{\xi^2 + (y-\eta)^2}{4t}} d\eta d\xi &= \int_{-\infty}^{\infty} e^{-\frac{\xi^2}{4t}} \int_0^{\frac{1}{2}f''(0)\xi^2 + \frac{1}{6}f'''(0)\xi^3 + \frac{1}{24}f^{(4)}(0)\xi^4} \left(1 - \frac{(y-\eta)^2}{4t}\right) d\eta d\xi + \mathcal{O}\left(t^{\frac{5}{2}}\right) \\ &= \int_{-\infty}^{\infty} e^{-\frac{\xi^2}{4t}} \left[\eta + \frac{(y-\eta)^3}{12t}\right]_{\eta=0}^{\eta=\frac{1}{2}f''(0)\xi^2 + \frac{1}{6}f'''(0)\xi^3 + \frac{1}{24}f^{(4)}(0)\xi^4} d\xi + \mathcal{O}\left(t^{\frac{5}{2}}\right) \end{aligned} \quad (2.24)$$

On rappelle qu'on cherche un $y = \mathcal{O}(t^2)$. D'autre part, les puissances impaires en ξ m neront   une int grale nulle. Ainsi, on peut  liminer la plupart des termes et on trouve :

$$\int_{-\infty}^{\infty} \int_0^{f(\xi)} e^{-\frac{\xi^2 + (y-\eta)^2}{4t}} d\eta d\xi = \int_{-\infty}^{\infty} e^{-\frac{\xi^2}{4t}} \left[\frac{1}{2}f''(0)\xi^2 + \frac{1}{24}f^{(4)}(0)\xi^4 - \frac{1}{96}f''(0)^3\xi^6\right] d\xi + \mathcal{O}\left(t^{\frac{5}{2}}\right) \quad (2.25)$$

En calculant les int grales :

$$\int_{-\infty}^{\infty} \xi^2 e^{-\xi^2} d\xi = \frac{\sqrt{\pi}}{2}, \quad \int_{-\infty}^{\infty} \xi^4 e^{-\xi^2} d\xi = \frac{3\sqrt{\pi}}{4}, \quad \int_{-\infty}^{\infty} \xi^6 e^{-\xi^2} d\xi = \frac{15\sqrt{\pi}}{8} \quad (2.26)$$

Par d finition de la fonction f , on rappelle que l'on a $f(0) = 0$ et $f'(0) = 0$, la courbure de la courbe donn e par $\{x, f(x)\}$ vaut :

$$H(0) = \frac{f''(0)}{(1 + f'(0)^2)^{\frac{3}{2}}} = f''(0) \quad (2.27)$$

De m me, les d riv es tangentielles selon la courbe   l'origine sont les d riv es selon x . L'expression de l'op rateur de Laplace-Beltrami est :

$$\Delta_{\Gamma} H = \frac{f^{(4)}}{[1 + f'^2]^{\frac{3}{2}}} - 9 \frac{f' f'' f'''}{[1 + f'^2]^{\frac{5}{2}}} - 3 \frac{f''^3}{[1 + f'^2]^{\frac{5}{2}}} + 15 \frac{f'^2 f''^3}{[1 + f'^2]^{\frac{7}{2}}} \quad (2.28)$$

En 0, on peut exprimer W en fonction des d riv es de f par :

$$W = \Delta_{\Gamma} H + \frac{H^3}{2} = f^{(4)}(0) - \frac{5}{2} f''(0)^3 \quad (2.29)$$

En combinant ces expressions avec l' quation (2.25), on obtient le d veloppement suivant :

$$G_{\delta t} * \chi^k(x) = \frac{1}{2} - \frac{1}{2\sqrt{\pi}} y \delta t^{-\frac{1}{2}} + \frac{1}{2\sqrt{\pi}} H \delta t^{\frac{1}{2}} - \frac{1}{4\sqrt{\pi}} W \delta t^{\frac{3}{2}} + \mathcal{O}(\delta t^{\frac{5}{2}}) \quad (2.30)$$

En r arrangeant les termes, on obtient la formule recherch e. \square

On peut r arranger l' quation (2.17) comme suit :

$$2\sqrt{\pi\delta t} \left(G_{\delta t} * \chi^k - \frac{1}{2} \right) = -y + H\delta t - \frac{1}{2} W \delta t^2 + \mathcal{O}(\delta t^3) \quad (2.31)$$

En seillant   $\frac{1}{2}$, on obtient alors le d placement normal de l'interface suivant :

$$y = H\delta t + \frac{1}{2} W \delta t^2 + \mathcal{O}(\delta t^3) \quad (2.32)$$

Il est n cessaire d' liminer le terme d'ordre 1 pour obtenir un d placement en W . Cela peut  tre r alis  en faisant une combinaison lin aire de deux convolutions   des temps diff rents :

$$2\sqrt{\pi\delta t} \left(2G_{\delta t} * \chi^k - G_{2\delta t} * \chi^k - \frac{1}{2} \right) = -y + W\delta t^2 + \mathcal{O}(\delta t^3) \quad (2.33)$$

Finalement, pour obtenir une vitesse $W\delta t$ de l'interface, il suffit de remplacer δt par sa racine :

$$2\sqrt{\pi}\delta t^{\frac{1}{4}} \left(2G_{\sqrt{\delta t}} * \chi^k(x) - G_{2\sqrt{\delta t}} * \chi^k(x) - \frac{1}{2} \right) = -y + W\delta t + \mathcal{O}(\delta t^{\frac{3}{2}}) \quad (2.34)$$

On obtient donc un algorithme permettant de simuler un d placement par flot de Willmore au prix de deux convolutions par it ration. Le terme d'erreur est de l'ordre de $\mathcal{O}(\delta t^{\frac{1}{2}})$.

On a r alis  le raisonnement dans le cas de la dimension 2, mais la m me combinaison lin aire permet d'obtenir le r sultat pour la dimension 3 car le terme d'ordre $\frac{3}{2}$ est le m me dans (2.18) que dans (2.17).

Algorithme 6 : Flot de Willmore :  tant donn s la fonction caract ristique χ_0 de l'ensemble Ω_0 d'interface Γ_0 , un pas de temps δt et un nombre d'it rations nb_{iter} , g n rer la suite d'ensembles $(\Omega_k)_k$ d crits par leurs caract ristiques χ^k .

Pour k de 0   $nb_{iter} - 1$:

1)  tape de convolution :

$$A^k(x) = \frac{1}{2\sqrt{\pi}\delta t^{\frac{1}{4}}} \left(2G_{\sqrt{\delta t}} * \chi^k(x) - G_{2\sqrt{\delta t}} * \chi^k(x) \right) \quad (2.35)$$

2)  tape de seuillage : $\chi^{k+1} = \mathbf{1}_{\{A(x) \geq \frac{1}{2}\}}$

2.3.4 Un d faut majeur : la pr cision proche de l'interface.

Un d faut majeur des m thodes de type convolution-seuillage est la n cessit  d'avoir un maillage suffisamment raffin  au voisinage de l'interface. En effet, la fonction caract ristique ne fournit aucune information au niveau sous-maille. En cas d'un d placement trop faible par rapport   la taille du maillage, l' tape de seuillage r assigne   chaque sommet sa valeur initiale. L'interface se retrouve lors coinc e, comme illustr  par la figure 2.6. Une solution propos e dans

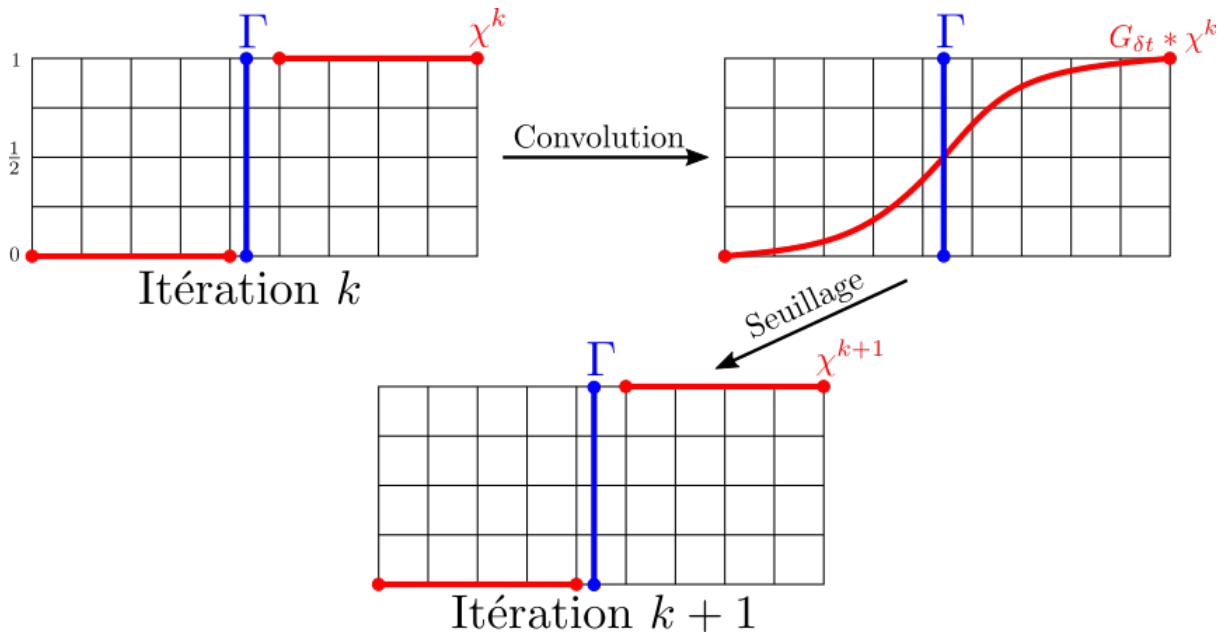


FIGURE 2.6 – Exemple de situation o  la r solution du maillage est trop faible et l'interface reste coinc e dans un cycle infini.

[Ruu98]   ce probl me consiste   imposer une meilleure r solution spatiale au

voisinage de l'interface. Cela implique d'avoir recours à un maillage adaptatif, stratégie qui peut se révéler coûteuse, notamment en 3D.

L'alternative à laquelle on s'intéresse est la suivante : que se passe-t-il si l'on diffuse la fonction distance signée et que l'on remplace l'étape de seuillage par une étape de redistanciation ? L'avantage de la fonction distance signée est qu'elle peut contenir des informations à un niveau inférieur de celui de la taille d'une maille. Même dans le cadre de faibles déplacements, la valeur des sommets va évoluer et on peut éviter le recours obligatoire à l'adaptation de maillage.

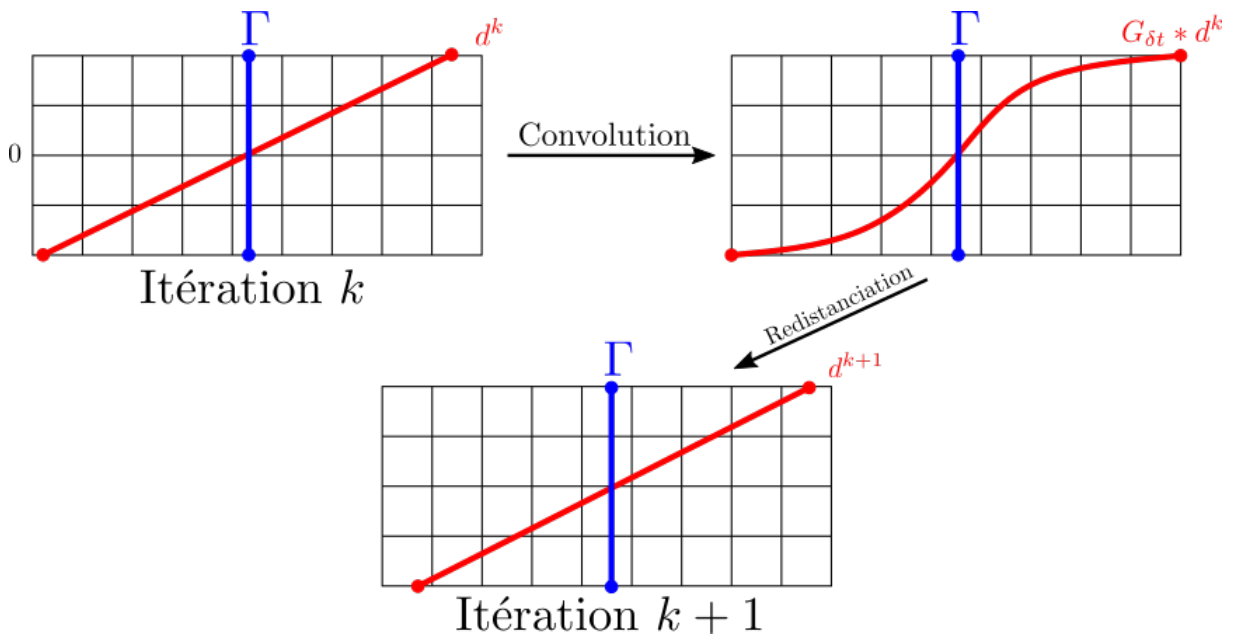


FIGURE 2.7 – La fonction distance contient des informations à un niveau inférieur à la taille des mailles. La configuration à l'étape $k + 1$ a évolué même dans le cas de faibles déplacements.

D'autre part, comme notre modèle (1.81) fait intervenir la distance signée, il semble naturel de se demander quel type de mouvement on peut générer selon le même principe. Le coût d'une étape de redistanciation, fait par une équation d'Hamilton-Jacobi ou par *fast-marching*, est néanmoins plus élevé que l'étape de seuillage.

2.4 Méthode de Convolution-Redistanciation

Dans cette section, on étudie les différents mouvements géométriques que l'on peut générer en alternant la diffusion d'une fonction distance et une étape de redistanciation.

On commence par chercher le d veloppement limit  de la fonction d donn e par :

$$\begin{cases} \partial_t d(t, x) - \Delta d(t, x) = 0 \\ d(0, x) = d^k(x) \end{cases} \quad (2.36)$$

On propose ensuite les variantes des algorithmes de type convolution-seuillage utilisant le d veloppement pr c demment obtenu. Une fois que l'on a  tabli le passage de l'algorithme MBO   l'algorithme de base de convolution-redistanciation, l'adaptation pour les diff rentes extensions est imm diate. On verra n anmoins que dans le cas du flot de Willmore, le terme d'ordre deux en temps n'est pas exactement le bon et qu'un terme de correction est n cessaire. Enfin, on testera la convergence de ces algorithmes pour les cas simples du cercle en dimension 2 et de la sph re en dimension 3 pour valider les erreurs pr dites par les diff rents d veloppements limit s.

2.4.1 D veloppement de la convolution de la fonction distance au voisinage de l'interface

Dans cette section, on cherche un d veloppement de Taylor de la fonction d^{k+1} satisfaisant le syst me (2.36)   l'instant δt . Un tel d veloppement est propos  par Esedoglu, Ruuth et Tsai dans [ERT⁺10] dans le cas de la dimension 2 et reprend le principe de la d monstration du d veloppement de la m thode de convolution seuillage (2.17).

On pr sente ici une m thode diff rente qui a le m rite d' tre plus intrins que et qui se g n ralise ais ment   la dimension 3.

Proposition 2.4.1. *En dimension 2, on a le d veloppement au voisinage de l'interface pour la fonction $G_{\delta t} * d$ solution de (2.36) :*

$$G_{\delta t} * d = d + H\delta t + \frac{1}{2} [\Delta_\Gamma H + H^3] \delta t^2 + \mathcal{O}(\delta t^3) \quad (2.37)$$

*En dimension 3, on a le d veloppement au voisinage de l'interface pour la fonction $G_{\delta t} * d$ solution de (2.36) :*

$$G_{\delta t} * d = d + 2H\delta t + [\Delta_\Gamma H + 4H(H^2 - K)] \delta t^2 + \mathcal{O}(\delta t^3). \quad (2.38)$$

On notera :

$$2W' = \begin{cases} \Delta_\Gamma \kappa + \kappa^3 \\ \Delta_\Gamma \kappa + \kappa(\kappa^2 - 4K) \end{cases} = \begin{cases} \Delta_\Gamma H + H^3 \text{ en dimension 2} \\ \Delta_\Gamma H + 4H(H^2 - K) \text{ en dimension 3} \end{cases} \quad (2.39)$$

Le lemme suivant est bas  sur le lemme 3 de [CRMP17] et les r sultats sont issus du livre de Delfour et Zolesio [DZ11]. Il permet d'exprimer l'op rateur de

Laplace-Beltrami de f en fonction des d riv es d'une extension r guli re de f sur un voisinage de Γ .

Une telle approche permet d' viter l'utilisation de syst mes de coordonn es locales et l'alourdissement des calculs associ s. De plus, il s'agit de propri t s intrins ques dans le sens o  elles ne d pendent pas du choix du prolongement de f consid r . Le prolongement canonique consiste   prendre $f \circ p_\Gamma$, c'est- -dire que l'on affecte   un point du voisinage la valeur prise par la fonction f en sa projection sur Γ . Ce choix est particuli rement pertinent dans notre cas, car on peut exprimer la projection sur Γ avec la fonction distance.

Lemme 2.4.2 (Lemme 3 [CRMP17]). *Pour un domaine Ω de classe C^2 d'interface Γ compacte, il existe un voisinage $S(\Gamma)$ de Γ pour lequel $d \in C^2(S(\Gamma))$.*

De plus, la projection sur la surface Γ est d finie et peut  tre exprim e   l'aide de la fonction distance :

$$p_\Gamma(x) = x - d(x)\nabla d(x) \quad (2.40)$$

Pour une fonction scalaire $f \in C^2(\Gamma)$ et une fonction vectorielle $w \in C^1(\Gamma, \mathbb{R}^N)$, on a les propri t s suivantes :

$$\begin{cases} \nabla_\Gamma f = \nabla (f \circ p_\Gamma)|_\Gamma \\ \operatorname{div}_\Gamma w = \operatorname{div} (w \circ p_\Gamma)|_\Gamma \\ \Delta_\Gamma f = \operatorname{div}_\Gamma \nabla_\Gamma f \end{cases} \quad (2.41)$$

L'op rateur de Laplace-Beltrami appliqu    la courbure moyenne satisfait alors :

$$\Delta_\Gamma \kappa = (N - 1)\Delta_\Gamma H = \Delta [\Delta d (p_\Gamma(x))] |_\Gamma = \Delta [\Delta d (x - d(x)\nabla d(x))] |_\Gamma \quad (2.42)$$

D monstration : Lemme

La preuve du premier point se trouve dans [DZ11] au Chapitre 9, p492. Sachant que $\nabla d(x)_\Gamma$ est la normale unitaire   Γ , l'expression de la projection (2.40) en d coule naturellement, cf Fig.2.8 pour une interpr tation graphique.

Les preuves pour l'expression des d riv es tangentielles se trouvent au Chapitre 9 de [DZ11].

D monstration : Proposition

On  crit le d veloppement limit  en temps de la fonction d satisfaisant le syst me (2.36)   l'instant δt et on utilise l' galit  $\partial_t d = \Delta d$:

$$\begin{aligned} G_{\delta t} * d &= d(0, x) + \delta t \partial_t d(0, x) + \frac{\delta t^2}{2} \partial_{tt} d(0, x) + \mathcal{O}(\delta t^3) \\ &= d(x) + \delta t \Delta d(x) + \frac{\delta t^2}{2} \Delta^2 d(x) + \mathcal{O}(\delta t^3) \end{aligned} \quad (2.43)$$

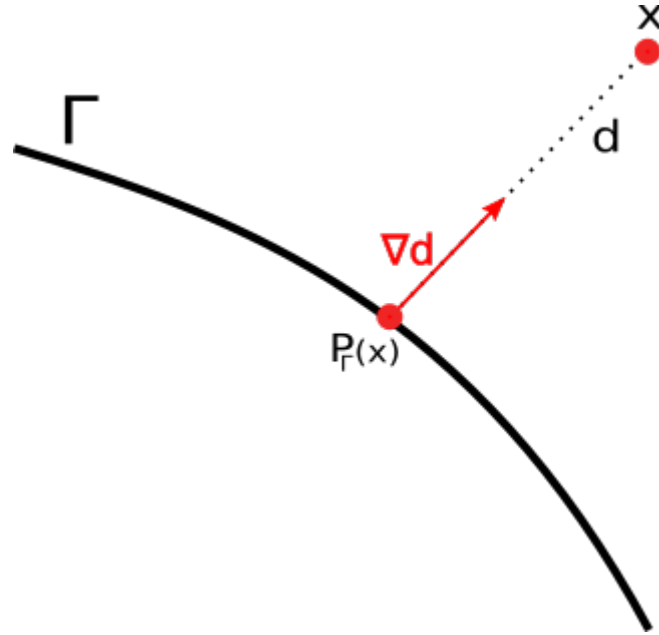


FIGURE 2.8 – Projection sur la courbe utilisant la distance sign e.

D'une part, on a :

$$\Delta d(x) = \kappa = (N - 1)H \quad (2.44)$$

On simplifie les notations en privil egiant l'utilisation de κ par rapport   H pour  viter le terme $N - 1$ li    la dimension. De plus, on omet l'indice k par souci de lisibilit .

On s'int resse maintenant   $\Delta^2 d$. Pour le calculer, on exprime la valeur du Laplace-Beltrami   partir de (2.42). Pour $x \in \Gamma$, on a :

$$\nabla(\Delta d(x - d\nabla d)) = [\mathbb{I} - d\nabla^2 d - \nabla d \otimes \nabla d] \times \nabla \Delta d(x - d\nabla d) \quad (2.45)$$

En prenant la divergence de cette expression, on obtient :

$$\begin{aligned} \Delta(\Delta d(x - d\nabla d)) = & \underbrace{(\mathbb{I} - \nabla d \otimes \nabla d - d\nabla^2 d)^2 : \nabla^2(\Delta d)(x - d\nabla d)}_{(1)} \\ & + \underbrace{\text{div}(\mathbb{I} - \nabla d \otimes \nabla d - d\nabla^2 d) \cdot \nabla \Delta d(x - d\nabla d)}_{(2)} \end{aligned} \quad (2.46)$$

Le terme $(\mathbb{I} - \nabla d \otimes \nabla d)$  tant la projection sur le plan tangent, il est idempotent et on a alors :

$$(\mathbb{I} - \nabla d \otimes \nabla d - d\nabla^2 d)^2 = (\mathbb{I} - \nabla d \otimes \nabla d) + d(d\nabla^2 d \times \nabla^2 d - 2(\mathbb{I} - \nabla d \otimes \nabla d) \times \nabla^2 d) \quad (2.47)$$

En  valuant (1) en $d = 0$, on a :

$$\begin{aligned}
 (1) &= (\mathbb{I} - \nabla d \otimes \nabla d) : \nabla^2(\Delta d) \\
 &= \mathbb{I} : \nabla^2(\Delta d) - \nabla d \otimes \nabla d : \nabla^2(\Delta d) \\
 &= \Delta(\Delta d) - \nabla^2(\Delta d) - \nabla d \otimes \nabla d : \nabla^2(\Delta d) \\
 &= \Delta^2 d - \partial_{nn} \kappa
 \end{aligned} \tag{2.48}$$

En  valuant (2) en $d = 0$, on a :

$$\begin{aligned}
 (2) &= \operatorname{div} (\mathbb{I} - \nabla d \otimes \nabla d) \cdot \nabla \Delta d \\
 &= -\Delta d \nabla d \cdot \nabla \Delta d \\
 &= -\kappa \partial_n \kappa
 \end{aligned} \tag{2.49}$$

En combinant ces deux r sultats et en utilisant la propri t  (2.42), on conclut :

$$\begin{aligned}
 \forall x \in \Gamma, \Delta_\Gamma \kappa &= \Delta (\Delta d(x - d \nabla d))|_\Gamma \\
 &= \Delta^2 d - \nabla d \otimes \nabla d : \nabla^2(\Delta d) + \Delta d \nabla d \cdot \nabla \Delta d \\
 &= \Delta^2 d - \partial_{nn} \kappa + \kappa \partial_n \kappa
 \end{aligned} \tag{2.50}$$

On rappelle que si $\kappa_1, \kappa_2, \dots, \kappa_{N-1}$ d signent les courbures principales de Γ , alors ce sont les valeurs propres de $\nabla^2 d$ sur le plan tangent. La n^{ime} valeur propre de $\nabla^2 d$ est nulle et admet pour vecteur propre la normale (car $\nabla^2 d \cdot \nabla d = 0$).

$$\left\{ \begin{array}{l} \kappa = \sum_{i=1}^{N-1} \kappa_i \\ K = \prod_{i=1}^{N-1} \kappa_i \end{array} \right. \tag{2.51}$$

Comme $\nabla^2 d$ est sym trique r elle, on peut la diagonaliser en base orthonormale et on a alors :

$$\nabla^2 d : \nabla^2 d = \sum_{i=1}^{N-1} \kappa_i^2 \tag{2.52}$$

En dimension 2, cela donne :

$$\nabla^2 d : \nabla^2 d = \kappa^2 \tag{2.53}$$

En dimension 3, cela donne :

$$\nabla^2 d : \nabla^2 d = \kappa_1^2 + \kappa_2^2 = (\kappa_1 + \kappa_2)^2 - 2\kappa_1 \kappa_2 = \kappa^2 - 2K \tag{2.54}$$

On exprime maintenant la d riv e normale de κ :

$$\begin{aligned}
 \partial_n \kappa &= \nabla d \cdot \nabla \Delta d \\
 &= \nabla d \cdot \operatorname{div}(\nabla^2 d) \\
 &= \operatorname{div}(\underbrace{\nabla^2 d \cdot \nabla d}_{=0}) - \nabla^2 d : \nabla^2 d \\
 &= -\nabla^2 d : \nabla^2 d
 \end{aligned} \tag{2.55}$$

Ainsi :

$$\partial_n \kappa = \begin{cases} -\kappa^2 \\ -(\kappa^2 - 2K) \end{cases} = \begin{cases} -H^2 \text{ en dimension 2} \\ -(4H^2 - 2K) \text{ en dimension 3} \end{cases} \tag{2.56}$$

On exprime maintenant la d riv e seconde normale de κ . On utilise (2.55) que l'on d rive   nouveau selon la normale. Les ingr dients sont encore une fois des int grations par parties et la condition $\nabla d \cdot \nabla^2 d = (\partial_j d) (\partial_{ij} d) = 0$ pour une fonction distance :

$$\begin{aligned}
 \partial_{nn} \kappa &= \nabla d \cdot \nabla (\partial_n \kappa) \\
 &= -\nabla d \cdot \nabla (\nabla^2 d : \nabla^2 d) \\
 &= -(\partial_j d) \partial_j [\partial_{ik} d \partial_{ik} d] \\
 &= -2(\partial_j d) (\partial_{ijk} d) (\partial_{ik} d) \\
 &= -2\partial_k \left[\underbrace{(\partial_j d) (\partial_{ij} d) (\partial_{ik} d)}_{=0} \right] + 2(\partial_{jk} d) (\partial_{ki} d) (\partial_{ij} d) + 2 \underbrace{(\partial_j d) (\partial_{ij} d) (\partial_{ikk} d)}_{=0} \\
 &= 2(\partial_{jk} d) (\partial_{ki} d) (\partial_{ij} d) \\
 &= 2 \operatorname{Tr} \left[(\nabla^2 d)^3 \right]
 \end{aligned} \tag{2.57}$$

En consid rant un base de diagonalisation, on a $(\nabla^2 d)^3 = P^{-1} \operatorname{diag} (\kappa_1^3, \kappa_2^3, \dots, \kappa_{N-1}^3, 0) P$ et $\operatorname{Tr} (\nabla^2 d)^3 = \sum_{i=1}^{N-1} \kappa_i^3$. En dimension 2, on trouve :

$$\partial_{nn} \kappa = 2\kappa^3 \tag{2.58}$$

et en dimension 3 :

$$\begin{aligned}
 \partial_{nn} \kappa &= 2(\kappa_1^3 + \kappa_2^3) \\
 &= 2((\kappa_1 + \kappa_2)^3 - 3\kappa_1^2 \kappa_2 - 3\kappa_1 \kappa_2^2) \\
 &= 2(\kappa^3 - 3(\kappa_1 + \kappa_2) \kappa_1 \kappa_2) \\
 &= 2\kappa(\kappa^2 - 3K)
 \end{aligned} \tag{2.59}$$

Ainsi :

$$\partial_{nn}\kappa = \begin{cases} 2\kappa^3 & \text{en dimension 2} \\ 2\kappa(\kappa^2 - 3K) & \text{en dimension 3} \end{cases} = \begin{cases} 2H^3 & \text{en dimension 2} \\ 4H(4H^2 - 3K) & \text{en dimension 3} \end{cases} \quad (2.60)$$

En injectant les valeurs obtenues pour les d riv es normales de la courbure moyenne dans (2.50), on obtient alors l'expression de $\Delta^2 d$:

$$\Delta^2 d = 2W' = \begin{cases} \Delta_\Gamma \kappa + \kappa^3 & \text{en dimension 2} \\ 2[\Delta_\Gamma \kappa + \kappa(\kappa^2 - 4K)] & \text{en dimension 3} \end{cases} = \begin{cases} \Delta_\Gamma H + H^3 & \text{en dimension 2} \\ 2[\Delta_\Gamma H + 4H(H^2 - K)] & \text{en dimension 3} \end{cases} \quad (2.61)$$

Finalement en utilisant (2.61) dans (2.43), on obtient les r sultats annonc s. \square

On retrouve bien avec (2.37) le r sultat obtenu dans la proposition 2 de [ERT⁺10]. Il est important de noter que, contrairement au cas de la fonction indicatrice, on ne trouve pas imm diatement le terme de Willmore   l'ordre 2. Le terme W' que l'on trouve n'a pas le bon rapport entre les deux termes. Pour trouver (2.15), il est donc n cessaire d'ajouter un terme correctif.

2.4.2 D veloppement de la convolution d'une fonction $f(d)$ au voisinage de l'interface

Puisque l'on n'obtient pas le terme de Willmore   l'ordre 2, un terme correctif est n cessaire. S'il s'av rera plut t imm diat en dimension 2, le cas de la dimension 3 sera plus d licat.

On pr sente ici le d veloppement de la convolution de la fonction $f(d)$, o  f est une fonction r guli re quelconque qui permettra d'obtenir le correctif en dimension 3.

La fonction $G_{\delta t} * (f(d))$ v rifiant l' quation de la chaleur suivante :

$$\begin{cases} \partial_t u(t, x) - \Delta u(t, x) = 0 \\ u(0, x) = f(d)(x) \end{cases} \quad (2.62)$$

satisfait un d veloppement proche de l'interface similaire   (2.43) :

$$G_{\delta t} * (f(d)) = f(d) + \Delta (f(d)) \delta t + \Delta^2 (f(d)) \frac{\delta t^2}{2} + \mathcal{O}(\delta t^3) \quad (2.63)$$

On a :

$$\begin{aligned} \Delta (f(d)) &= \operatorname{div}(f'(d)\nabla d) \\ &= f^{(2)}(d) \underbrace{|\nabla d|^2}_{=1} + f'(d)\Delta d \\ &= f^{(2)}(d) + f'(d)\Delta d \end{aligned} \quad (2.64)$$

et :

$$\begin{aligned}
\Delta^2 (f(d)) &= \Delta \left(f^{(2)}(d) + f'(d)\Delta d \right) \\
&= f^{(4)} + 2f^{(3)}\Delta d + f^{(2)}(\Delta d)^2 + 2f^{(2)}\nabla d \cdot \nabla \Delta d + f'\Delta^2 d \\
&= f^{(4)}(d) + 2f^{(3)}(d)\kappa + 2f^{(2)}(d) \left(\frac{\kappa^2}{2} + \partial_n \kappa \right) + 2f'(d)W'
\end{aligned} \tag{2.65}$$

Avec :

$$\frac{\kappa^2}{2} + \partial_n \kappa = \begin{cases} -\frac{\kappa^2}{2} \\ 2K - \frac{\kappa^2}{2} \end{cases} = \begin{cases} -\frac{H^2}{2} \text{ en dimension 2} \\ -2(H^2 - K) \text{ en dimension 3} \end{cases} \tag{2.66}$$

Finalement :

$$G_{\delta t} * f = f + \delta t \left(f^{(2)} + f'\kappa \right) + \delta t^2 \left[\frac{f^{(4)}}{2} + f^{(3)}\kappa + f^{(2)} \left(\frac{\kappa^2}{2} + \partial_n \kappa \right) + f^{(1)}W' \right] + \mathcal{O}(\delta t^3). \tag{2.67}$$

On utilisera (2.67) pour obtenir le terme de correctif du flot de Willmore en dimension 3.

2.4.3 Mouvement par courbure moyenne par Convolution-Redistanciation

On s'int resse maintenant aux algorithmes de type convolution-redistanciation simulant le mouvement par courbure moyenne. Les versions en dimension 2 de ces algorithmes sont pr sents dans [ERT⁺10]. On rappelle que l'on cherche alors une vitesse normale de l'interface donn e par :

$$v = H \tag{2.68}$$

On propose l'algorithme suivant qui est l'adaptation de l'algorithme 2.1 : 2 pour une fonction distance. On diffuse une fonction distance et on remplace l' tape de seuillage par une  tape de redistanciation afin d'obtenir   nouveau une fonction distance et pouvoir it rer le processus.

Algorithme 7 : Courbure moyenne :  tant donn s la fonction distance sign e   l'ensemble Ω_0 d'interface Γ_0 , un pas de temps δt et un nombre d'it rations nb_{iter} , g n rer la suite d'ensembles $(\Omega_k)_k$ d crits par leurs fonctions distances sign es d^k .

Pour k de 0   $nb_{iter} - 1$:

1)  tape de convolution : $A(x) = \frac{1}{N-1} G_{\delta t} * d^k$.

2)  tape de redistanciation : $d^{k+1} = \text{redist}(A)$

Le d veloppement justifie l'algorithme car :

$$\forall x \in \{d^{k+1} = 0\}, \quad d^k(x) = -H^k \delta t + \mathcal{O}(\delta t^2) \quad (2.69)$$

L'algorithme est le m me en dimension 3   la diff rence pr s que l'on doit g rer le facteur 2 pr sent devant le terme de courbure moyenne.

L' quation (2.69) fournit  galement une estimation de l'erreur commise qui est de l'ordre de $\mathcal{O}(\delta t)$.

On peut obtenir un algorithme   l'ordre sup rieur au prix d'une convolution suppl mentaire par  tape. Pour cela on forme une combinaison lin aire qui  limine le terme d'ordre deux du d veloppement. Un noyau convenable pour cette manipulation est :

$$\frac{1}{3(N-1)} \left[4G_{\frac{3}{2}\delta t} - G_{3\delta t} \right] * d^k = d^k + H^k \delta t + \mathcal{O}(\delta t^3) \quad (2.70)$$

On peut donc proposer l'algorithme plus pr cis suivant :

Algorithme 8 : Courbure moyenne am lior e :  tant donn s la fonction distance sign e   l'ensemble Ω_0 d'interface Γ_0 , un pas de temps δt et un nombre d'it rations nb_{iter} , g n rer la suite d'ensembles $(\Omega_k)_k$ d crits par leurs fonctions distances sign es d^k .

Pour k de 0   $nb_{iter} - 1$:

1)  tape de convolution : $A(x) = \frac{1}{3} \left(4G_{\frac{3}{2}t} - G_{3\delta t} \right) * d^k$.

2)  tape de redistanciation : $d^{k+1} = \text{redist}(A)$

Remarque 2.4.3. L'algorithme 2.4.3 : 7 est monotone au sens o  si $A^0 \subset B^0$, alors $A^k \subset B^k$ avec A^k, B^k d signant les ensembles obtenus par l'algorithme   l'it ration k . Cette monotonie est obtenue du fait que le noyau de la chaleur $G_{\delta t}$ est positif.

Pour l'algorithme 2.4.3 : 8 d'ordre sup rieur, le noyau que l'on utilise est le suivant : $4G_{\frac{3}{2}\delta t} - G_{3\delta t}$. Ce noyau n'est pas positif et on ne garantit plus la monotonie de l'algorithme.

L'adaptation de l'algorithme 2.4.3 : 7 au mouvement avec conservation de volume se fait de la m me fa on que pour les algorithmes de convolution-seuillage. Nous y reviendrons dans le Chapitre 3.

2.4.4 V rification num rique de convergence sur le cercle

Dans cette partie, nous allons pr senter les r sultats num riques sur les cas simples du cercle pour les diff rents algorithmes de type convolution-redistanciation

propos s pr c demment. L'int r t de ces exemples est que l'on connait explicitement l' volution du mouvement par courbure moyenne et par flot de Willmore dans ces cas pr cis. On peut ainsi  tudier pr cis ment l'erreur commise et valider la convergence anticip e th oriquement.

Dans cette partie, nous allons v rifier num riquement le d placement par courbure moyenne et le calcul de la courbure moyenne par diffusion.

2.4.5 Mouvement par courbure moyenne en dimension 2

On commence par  tudier la convergence num rique des algorithmes 2.4.3 : 7 et 2.4.3 : 8 dans le cas d'un cercle.

On choisit comme valeur initiale le cercle de rayon $R = 0,25$ qui est soumis au mouvement par courbure moyenne :

$$v = H \quad (2.71)$$

La courbure d'un cercle de rayon R est constante en tout point et vaut $\frac{1}{R}$. Par axisym trie, le cercle va rester un cercle dont le rayon $R(t)$ va diminuer jusqu'   tre r duit   un point. L' quation (2.71) fournit l' quation diff rentielle satisfaite par $R(t)$ tant qu'il n'est pas nul :

$$\frac{DR(t)}{dt} = -\frac{1}{R(t)} \quad (2.72)$$

La solution v rifiant la condition initiale $R(0) = 0,25$ est :

$$R(t) = \sqrt{0,0625 - 2t} \text{ pour } t < 0,125. \quad (2.73)$$

La formule (2.73) est une expression explicite de l' volution du rayon du cercle. On va lui comparer l' volution des rayons num riques pour les algorithmes 2.4.3 : 7 et 2.4.3 : 8. On calcule num riquement le rayon   partir de l'aire du cercle :

$$R_{num} = \sqrt{\frac{\text{Aire}}{\pi}} \quad (2.74)$$

L'erreur relative est calcul e par la formule suivante :

$$\text{Erreur relative} = \frac{|R_{num} - R_{th}|}{R_{th}} \quad (2.75)$$

L'ordre de convergence num rique que l'on consid re est donn  par la formule suivante :

$$\text{Ordre de convergence} = \frac{\log(\text{Erreur}^{k+1}) - \log(\text{Erreur}^k)}{\log(h^{k+1}) - \log(h^k)} \quad (2.76)$$

Les résultats sont présentés dans les tableaux 2.1 et 2.2 et le graphe 2.9. Nos résultats sont similaires à ceux obtenus dans [ERT⁺10]. Il convient de noter que l'on n'obtient pas une convergence à l'ordre 2 avec l'algorithme 2.4.3 : 8 comme suggéré par l'équation (2.70) mais plutôt l'ordre 2 avec une amélioration des erreurs par rapport à celles de l'algorithme 2.4.3 : 7, ce qui était aussi le cas dans [ERT⁺10].

Le pas de temps pour l'équation de la chaleur a été calibré par rapport puis divisé et multiplié par 4 lorsque l'on divise et multiplie le pas de maillage par 2. Ce choix est crucial pour un bon comportement de la méthode. Dans la partie suivante 2.4.6, nous menons une discussion sur la façon de choisir le pas de temps et prescrivons un choix de δt de l'ordre de h^2 , cf (2.86). Nous avons appliqué ce principe pour nos tests de convergence.

h	δt	Erreur relative (%)	Ordre de convergence
0,04	4×10^{-4}	1.1	–
0,02	2×10^{-4}	0.67	0.72
0,01	5×10^{-5}	0.27	1.67
0,005	1.25×10^{-5}	0.091	1.21
0,0025	3.125×10^{-6}	0.042	1.11

TABLE 2.1 – Étude de convergence de l'algorithme 2.4.3 : 7. L'erreur est donnée à $t = 8 \times 10^{-3}$.

h	δt	Erreur relative (%)	Ordre de convergence
0,04	4×10^{-4}	0.88	–
0,02	2×10^{-4}	0.50	0.82
0,01	5×10^{-5}	0.15	1.73
0,005	1.25×10^{-5}	0.061	1.29
0,0025	3.125×10^{-6}	0.027	1.17

TABLE 2.2 – Étude de convergence de l'algorithme 2.4.3 : 8. L'erreur est donnée à $t = 8 \times 10^{-3}$.

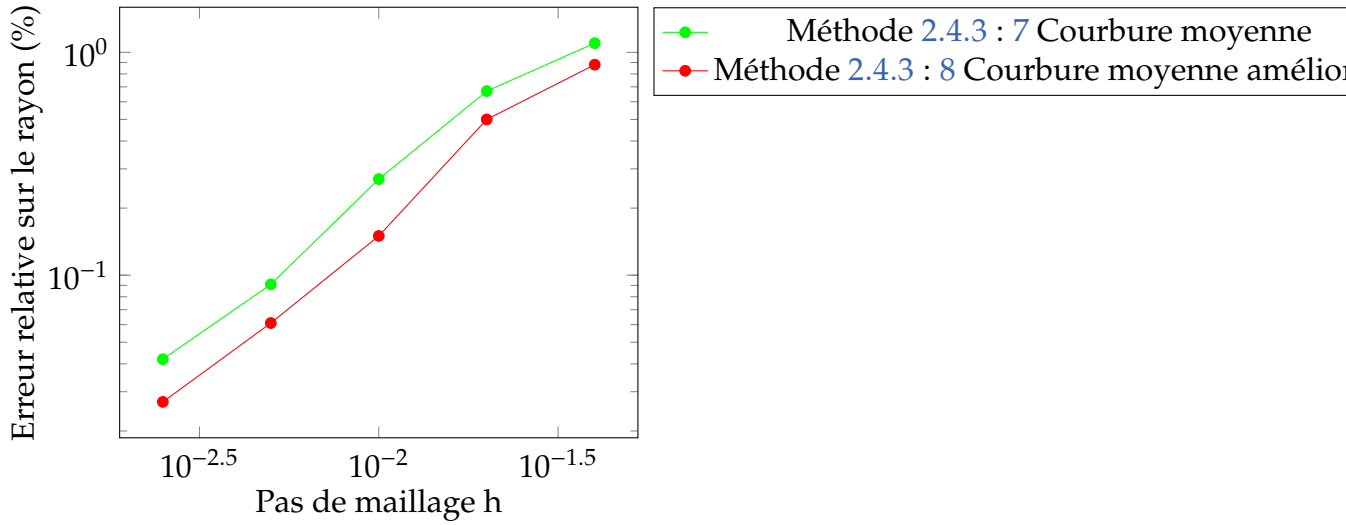


FIGURE 2.9 – Convergence de l’erreur sur le rayon les deux algorithmes de mouvement par courbure moyenne 2.4.3 : 7 et 2.4.3 : 8. On retrouve l’observation faite dans [ERT⁺10] que l’on n’obtient pas l’ordre 2 avec la m thode 2.4.3 : 8 mais plut t une division par deux de l’erreur.

2.4.6 Extraction de la courbure par diffusion

On s’int resse ici   la m thode d’extraction de la courbure suivante, cons quence imm diate de l’algorithme 2.4.3 : 7

$$\kappa = (N - 1)H = \frac{G_{\delta t} * d - d}{\delta t} + \mathcal{O}(\delta t). \quad (2.77)$$

On commence par pr senter les m thodes disponibles sur FEEL++ qui serviront de comparaison. Puis, on  tudie les propri t s de l’algorithme, notamment le choix du pas de temps pour la convolution.

Calcul de la courbure sur FEEL++

La m thode standard pour calculer la courbure de l’interface n cessite deux d rivations successives :

$$\kappa = (N - 1)H = \Delta d \quad (2.78)$$

Lors de simulations avec une approximation polynomiale de faible ordre, typiquement \mathbb{P}^1 , on ne peut pas d river d deux fois. En effet, diff rencier une fonction diminue de un l’ordre. En pratique, il est n cessaire de projeter la d riv e sur un espace d’ordre sup rieur, l’espace original par exemple. Sur FEEL++, on le fait   l’aide de la projection \mathbb{L}^2 suivante :

$$\begin{cases} \text{Trouver } g \in \mathbb{P}^1(\Omega) \text{ tel que :} \\ \forall v \in \mathbb{P}^1(\Omega), \int_{\Omega} g \cdot v = \int_{\Omega} \nabla d \cdot v. \end{cases} \quad (2.79)$$

L'équation (2.79) permet d'obtenir le gradient ∇d sur l'espace \mathbb{P}^1 . On peut alors prendre sa divergence avec deux dérivations/projections successives :

$$\begin{cases} \text{Trouver } \kappa \in \mathbb{P}^1 \text{ tel que } \forall v \in \mathbb{P}^1 : \\ \forall v \in \mathbb{P}^1(\Omega), \int_{\Omega} \kappa \cdot v = \int_{\Omega} \operatorname{div}(g) \cdot v. \end{cases} \quad (2.80)$$

Cependant, la projection \mathbb{L}^2 remplit les informations manquantes, nécessaires pour passer d'un élément \mathbb{P}^0 à \mathbb{P}^1 , avec des valeurs imprévisibles. Ces valeurs n'ont aucune raison d'être régulières, dériver à nouveau l'élément projeté fournit de pauvres résultats.

Une solution à ce problème, proposée dans [Doy14], consiste à ajouter un petit terme diffusif $\eta \Delta \kappa$ pour régulariser ces oscillations :

$$\begin{cases} \text{Trouver } \kappa \in \mathbb{P}^1 \text{ tel que } \forall v \in \mathbb{P}^1 : \\ \forall v \in \mathbb{P}^1(\Omega), \int_{\Omega} \kappa v + \eta \int_{\Omega} \nabla \kappa \cdot \nabla v - \eta \int_{\partial \Omega} \nabla \kappa \cdot n v = \int_{\Omega} \operatorname{div}(g) \cdot v. \end{cases} \quad (2.81)$$

En pratique, pour assurer une bonne convergence de la méthode et éviter un effet de seuil, on choisit η de l'ordre de h^2 .

Calcul de la courbure par diffusion

Dans l'esprit d'un schéma de convolution-seuillage, on peut calculer la courbure en diffusant la fonction distance :

$$\kappa = \frac{G_{\delta t} - d}{\delta t} + \mathcal{O}(\delta t). \quad (2.82)$$

et, au prix d'une convolution supplémentaire, on peut améliorer la précision du calcul en éliminant le premier terme d'erreur :

$$\kappa = \frac{2G_{\delta t} - G_{2\delta t} - \frac{3}{2}d}{\delta t} + \mathcal{O}(\delta t^2). \quad (2.83)$$

L'avantage de ces méthodes est d'être dans le cadre des algorithmes de convolution-seuillage; il ne requiert pas de dérivation et peut donc être utilisé avec une discrétisation \mathbb{P}^1 .

Le choix du pas de temps se révèle être un élément essentiel. Le graphe 2.10 donne le comportement de l'erreur numérique des deux méthodes en fonction du pas de temps.

La méthode (2.82) donne une fonction convexe avec un pas de temps optimal. On observe donc un effet de seuil lorsque le pas de temps est faible. En effet, les erreurs spatiales du terme $G_{\delta t} * d - d$ sont amplifiées par le facteur $\frac{1}{\delta t}$ et ne sont plus négligeables pour un petit δt .

On observe un comportement similaire pour la méthode d'ordre supérieur. La fonction d'erreur est la combinaison de deux fonctions du type de (2.82)

déphasées. Elle comporte alors deux minima. Il convient de constater que la pente d'erreur semble plus forte pour cette méthode. Cependant, l'effet de seuil fait que l'on ne peut pas vraiment attendre une convergence systématique d'ordre 2.

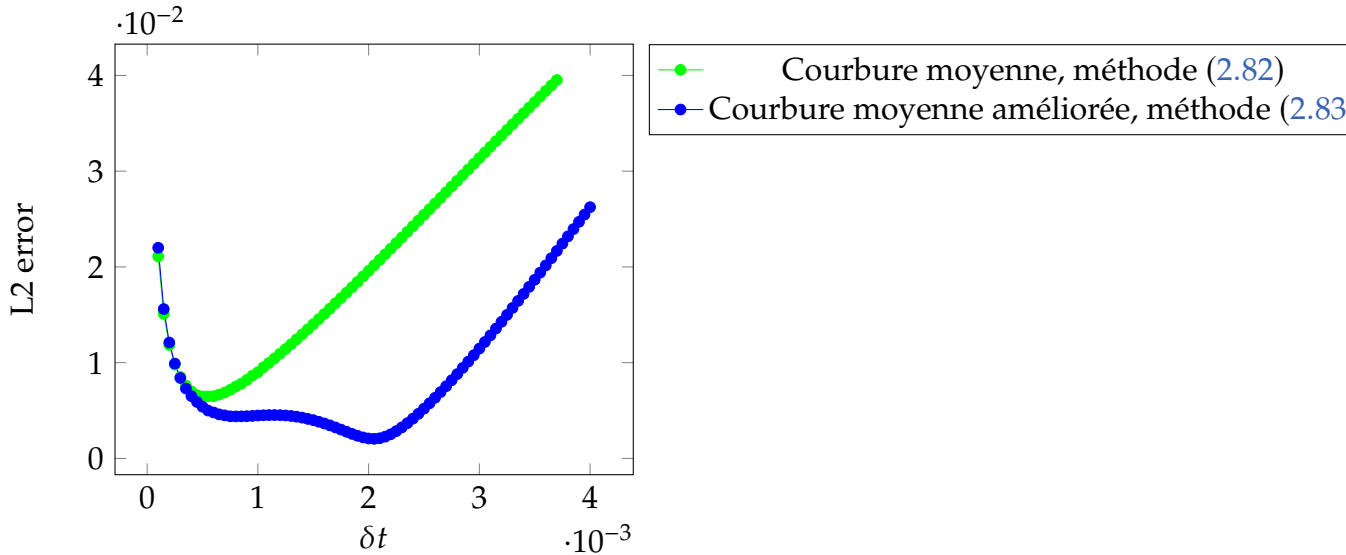


FIGURE 2.10 – Évolution de l'erreur des méthodes (2.82) et (2.83) en fonction du pas de temps. La courbure est calculée en dimension 2 sur un cercle de rayon $r = 0.25$ pour une discrétisation éléments finis avec des éléments de taille $h = 0.01$ et une interface d'épaisseur $1.5h$. L'erreur calculée est : $e_{\mathbb{L}^2} = \frac{\int_{\Omega} [H(x) - H_{th}(x)]^2 dx}{\int_{\Omega} H_{th}(x)^2 dx}$.

Il est nécessaire d'avoir une stratégie pour le choix de δt . Pour une taille de discrétisation h donnée et une fonction *levelset* de courbure locale κ , on propose la condition suivante :

$$\kappa \delta t \sim h \quad (2.84)$$

Un pas de temps donné par (2.84), le terme $\kappa \delta t$ est alors de l'ordre de h . On a déplacé l'interface de quelques éléments lors de la convolution $G_{\delta t} * d$, ce qui semble raisonnable d'un point de vue numérique. Dans l'idéal, il s'agit de déplacer toute partie de la courbe d'au moins un élément pour bien rendre compte des disparités de courbure. Il convient alors de choisir le meilleur compromis.

On explore cette hypothèse dans le cas du cercle. Les tableaux 2.3 illustrent l'évolution du pas de temps optimal de (2.82) en fonction de la finesse du maillage et du rayon (l'inverse de la courbure). Ils vont dans le sens de l'intuition (2.84).

Ainsi, la règle pour le pas de temps est :

$$\delta t \sim \frac{h}{\kappa} \quad (2.85)$$

h	$10^{-3}\delta t_{opt}$	$10^{-3}\delta t_{opt}^{impr}$
0,08	8,5	2,6
0,04	1,7	2,2
0,02	1	2,1
0,01	0,55	1,8
0,005	0,25	1,7

TABLE 2.3 – Évolution du pas de temps optimal en fonction de la taille d'un élément.

R	$10^{-3}\delta t_{opt}$	$10^{-3}\delta t_{opt}^{impr}$
0,4	0,29	2,3
0,3	0,25	0,9
0,25	0,2	0,8
0,2	0,15	1,3
0,15	0,11	0,75
0,1	0,07	0,35

TABLE 2.4 – Évolution du pas de temps optimal en fonction du rayon (c'est-à-dire l'inverse de la courbure).

Numériquement, il est nécessaire de choisir une discrétisation en espace telle que $h \leq \frac{1}{\kappa}$ pour obtenir une représentation de l'allure géométrique de l'interface. En conclusion, la règle déterminant δt est donc :

$$\delta t \sim h^2. \tag{2.86}$$

Le comportement du pas de temps optimal pour la méthode (2.83) est plus difficile à analyser du fait des deux convolutions. Le graphe 2.10 montre qu'il y a un plus grand intervalle de pas de temps donnant un résultat satisfaisant. On voit aussi sur la figure 2.10 qu'on peut prendre des plus grands pas de temps pour cette méthode. Dans la pratique, on prend un δt satisfaisant la même condition (2.86).

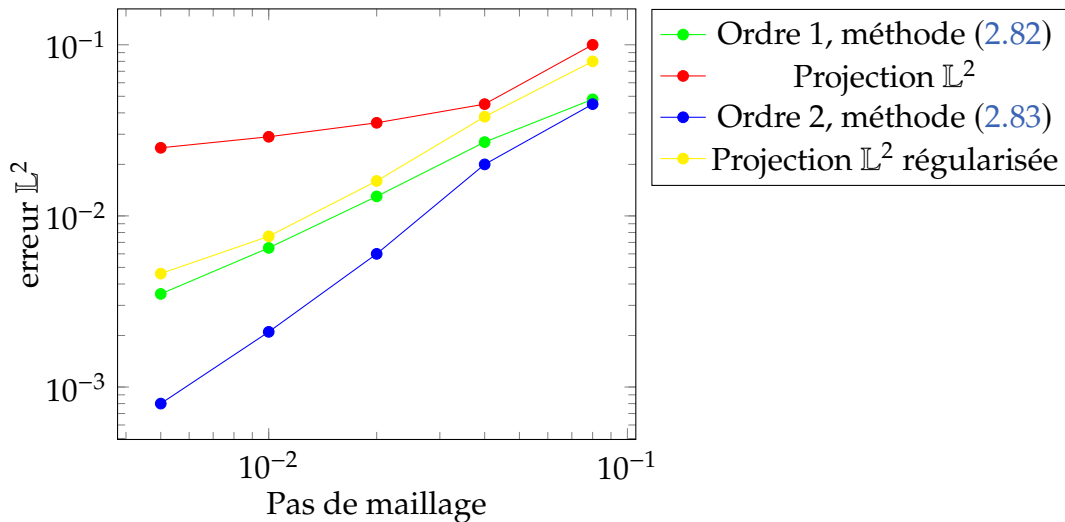


FIGURE 2.11 – Convergence de l'erreur \mathbb{L}^2 pour les différentes méthodes de calcul de la courbure.

h	$e_{\mathbb{L}^2}$	$e_{\mathbb{L}^2 \text{regularis�e}}$	e_{Ordre1}	e_{Ordre2}
0,08	0,1	0,08	0,048	0,045
0,04	0,045	0,038	0,027	0,02
0,02	0,035	0,016	0,013	0,0059
0,01	0,029	0,0076	0,0065	0,0021
0,005	0,025	0,0046	0,0035	0,0008

TABLE 2.5 – Convergence de l'erreur \mathbb{L}^2 des diff rentes m thodes de calcul de courbure.

2.4.7 Mouvement par courbure moyenne avec conservation de volume par convolution-redistanciation

Dans cette section, on s'int resse au mouvement par courbure moyenne avec conservation de volume. On rappelle qu'il consiste   d placer l'interface   la vitesse normale.

$$v = H - \bar{H} \quad (2.87)$$

On a vu   la section 2.3.2 une adaptation des sch mas de convolution-seuillage   ce mouvement. Le principe  tait de modifier la valeur seuil pour trouver la ligne de niveau donnant le bon volume. Une adaptation au contexte des sch mas de convolution-redistanciation est propos e dans [KEF11]. On cherche l'ensemble $\{\phi = \lambda\}$ qui englobe le bon volume. La bonne valeur est $\lambda = \bar{H} = \frac{2\pi}{A(\phi)}$.

Comme pour le cas de l'algorithme de convolution-seuillage, il est possible en dimension 2 de conna tre explicitement la valeur explicite de λ :

$$\bar{H} = \frac{2\pi}{A(\phi)} \quad (2.88)$$

En dimension 3, il est n cessaire de calculer la valeur \bar{H} donc de calculer H puis de l'int grer sur la surface.

Algorithme 9 : Courbure moyenne avec conservation de volume 2D :  tant donn s la fonction distance sign e   l'ensemble Ω_0 d'interface Γ_0 et de volume V_0 , un pas de temps δt et un nombre d'it rations nb_{iter} , g n rer la suite d'ensembles $(\Omega_k)_k$ d crits par leurs fonctions distances sign es d^k .

Pour k de 0   $nb_{iter} - 1$:

1)  tape de convolution : $\frac{1}{N-1}G_{\delta t} * d^k$.

2) Calculer l'aire de la surface $A(\frac{1}{N-1}G_{\delta t} * d^k)$.

3)  tape de redistanciation : $d^{k+1} = \text{redist}\left(\frac{1}{N-1}G_{\delta t} * d^k - \frac{2\pi}{A(\frac{1}{N-1}G_{\delta t} * d^k)}\right)$

Algorithme 10 : Courbure moyenne avec conservation de volume 3D :  tant donn s la fonction distance sign e   l'ensemble Ω_0 d'interface Γ_0 , un pas de temps δt et un nombre d'it rations nb_{iter} , g n rer la suite d'ensembles $(\Omega_k)_k$ d crits par leurs fonctions distances sign es d^k suivant un mouvement par courbure moyenne avec conservation de volume.

Pour k de 0   $nb_{iter} - 1$:

1)  tape de convolution : $\frac{1}{N-1}G_{\delta t} * d^k$.

2) Calculer la courbure moyenne H et sa moyenne $\bar{H} = \frac{1}{A(G_{\delta t} * d^k)} \int_{\{G_{\delta t} * d^k = 0\}} H d\sigma$.

3)  tape de redistanciation : $d^{k+1} = \text{redist} \left(\frac{1}{N-1}G_{\delta t} * d^k - \bar{H} \right)$

On s'int ressera plus en d tail dans le Chapitre 3   la conservation du volume et de l'aire.

2.4.8 Flot de Willmore par convolution-redistanciation en dimension 2

On d crit dans cette section une m thode pour obtenir le d placement par flot de Willmore par convolution-redistanciation en dimension 2. Tous les ingr dients de la m thode se trouvent dans [ERT⁺10]   la diff rence pr s que les auteurs  tudient le d placement pilot  par diffusion de surface.

On rappelle que l'on cherche   obtenir une vitesse normale de l'interface donn e par :

$$W = \Delta_{\Gamma}H + \frac{H^3}{2} \quad (2.89)$$

et que le terme d'ordre 2 dans le d veloppement de la fonction distance est :

$$W' = \Delta_{\Gamma}H + H^3 \quad (2.90)$$

Ainsi, le terme correctif est $B = W' - W = -\frac{H^3}{2}$. On a vu dans la partie pr c dente comment obtenir la courbure   partir de la diffusion, voir (2.82). On peut alors obtenir le flot de Willmore par :

$$\begin{aligned} \left(2G_{\sqrt{\delta t}} - G_{2\sqrt{\delta t}} \right) * d + \frac{1}{2\sqrt{\delta t}} \left(G_{\sqrt{\delta t}} * d - d \right)^3 &= d - W'\delta t + \frac{H^3}{2}\delta t + \mathcal{O}\left(\delta t^{\frac{3}{2}}\right) \\ &= d - W\delta t + \mathcal{O}\left(\delta t^{\frac{3}{2}}\right) \end{aligned} \quad (2.91)$$

On peut  galement d cider d'utiliser (2.83) pour calculer la courbure :

$$\begin{aligned} \left(2G_{\sqrt{2\delta t}} - G_{\sqrt{\frac{\delta t}{2}}}\right) * d + \frac{1}{\sqrt{\delta t}} \left[\frac{G_{2\sqrt{\delta t}} * d - d}{2} \right]^3 &= d - W'\delta t + \frac{H^3}{2}\delta t + \mathcal{O}\left(\delta t^{\frac{3}{2}}\right) \\ &= d - W\delta t + \mathcal{O}\left(\delta t^{\frac{3}{2}}\right) \end{aligned} \quad (2.92)$$

Notons que l'on ne fait que deux convolutions par it ration. En effet, il suffit pour cela d'utiliser les m mes noyaux pour les deux parties de l'algorithme,   savoir le calcul de W' et H . On n'induit donc pas de co t suppl mentaire par rapport   la m thode de convolution-seuillage.

Il est  galement important de constater que l'on est en pr sence d'un terme d'erreur en $\mathcal{O}\left(\delta t^{\frac{3}{2}}\right)$. Cela induit une erreur en la vitesse de l'ordre de $\mathcal{O}\left(\sqrt{\delta t}\right)$.

L'algorithme simulant un d placement par flot de Willmore en dimension 2 est donc le suivant :

Algorithme 11 : Flot de Willmore 2D :  tant donn s la fonction distance sign e   l'ensemble Ω_0 d'interface Γ_0 , un pas de temps δt et un nombre d'it rations nb_{iter} , g n rer la suite d'ensembles $(\Omega_k)_k$ d crits par leurs fonctions distances sign es d^k .

Pour k de 0   $nb_{iter} - 1$:

1)  tape de convolution : Calculer les convolutions : $G_{2\sqrt{\delta t}} * d^k$ et $G_{\sqrt{\delta t}} * d^k$.

2) Former les termes :

$$A^k = 2G_{\sqrt{\delta t}} * d^k - G_{2\sqrt{\delta t}} * d^k$$

$$B^k = \left(G_{\sqrt{\delta t}} * d^k - d^k\right)^3 \text{ ou } \left[\left(2\sqrt{2}G_{\sqrt{\frac{\delta t}{2}}} - \frac{1}{\sqrt{2}}G_{\sqrt{2\delta t}}\right) * d^k - \frac{3\sqrt{2}}{2}d^k\right]^3$$

3)  tape de redistanciation : $d^{k+1} = \text{redist}\left(A^k + \frac{B^k}{2\sqrt{\delta t}}\right)$.

2.4.9 V rification num rique du mouvement par flot de Willmore en dimension 2

Dans cette partie, nous allons v rifier num riquement le d placement selon le flot de Willmore de l'algorithme 2.4.8 : 11. Comme pour le cas du mouvement par courbure moyenne, on se place dans le cas d'un cercle de rayon R et on peut faire les m mes consid rations de sym trie pour justifier que l'on

conserve un cercle. En effet, dans ce cas particulier, le terme $\Delta_\Gamma H$ est nul et il reste un d placement   la vitesse normale $\frac{H^3}{2} = \frac{1}{2R^3}$. Ainsi, $R(t)$ v rifie l' quation diff rentielle :

$$\frac{dR(t)}{dt} = \frac{1}{2R^3(t)} \quad (2.93)$$

En notant la condition initiale R_0 , on obtient :

$$R(t) = \sqrt[4]{R_0^4 + 2t} \quad (2.94)$$

Dans nos simulations, on choisira pour point de d part du test le cercle de rayon 0.25.

La convergence de l'algorithme 2.4.8 : 11 est donn e au tableau 2.6 et dans le graphe 2.12. De la m me fa on que pour le cas des algorithmes de courbure moyenne, on calibre le choix du pas de temps pour les  quations de la chaleur pour $h = 0.01$. Ensuite, on divise ou multiplie par 4 lorsque l'on divise ou multiplie le pas de temps par 2, en suivant notre r gle $\delta t \sim h^2$. On retrouve le taux d'ordre $\frac{1}{2}$ attendu.

h	δt	Erreur relative (%)	Ordre de convergence
0,04	16×10^{-6}	0.89	–
0,02	4×10^{-6}	0.49	0.86
0,01	1×10^{-6}	0.32	0.61
0,005	2.5×10^{-7}	0.24	0.41
0,0025	6.25×10^{-8}	0.17	0.42

TABLE 2.6 –  tude de convergence de l'algorithme 2.4.3 : 7.

2.4.10 Flot de Willmore par convolution-redistanciation en dimension 3

Pour le cas de la dimension 3, on suit le m me principe que pr c demment en extrayant le d placement $W'\delta t$:

$$A = 2G_{\frac{\delta t}{2}} * d - G_{\sqrt{2\delta t}} * d = d - W'\delta t + O\left(\delta t^{\frac{3}{2}}\right) \quad (2.95)$$

o  W' est le terme donn  par (2.61). Cependant, l'obtention du terme correctif est plus compliqu e et n cessite deux diffusions suppl mentaires. En effet, la valeur de la correction B est :

$$B = W' - W = 2H(H^2 - K) \quad (2.96)$$

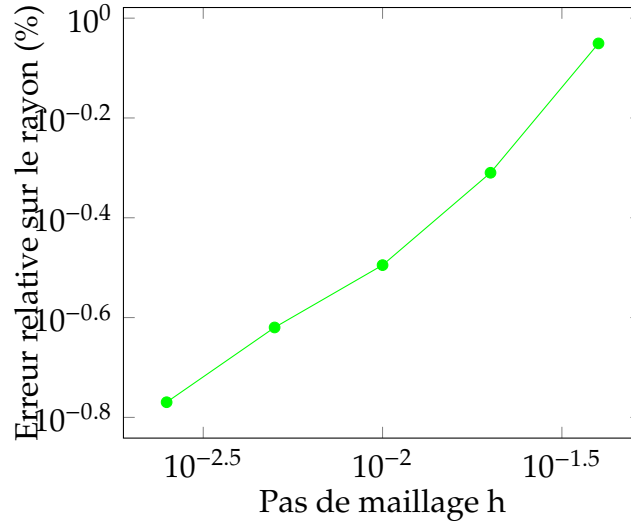


FIGURE 2.12 – Convergence de l'erreur sur le rayon de l'algorithme de mouvement par flot de Willmore en dimension 2 2.4.8 : 11.

On rappelle le d veloppement (2.67) d'une fonction g n rique $f(d)$ proche de l'interface :

$$G_{\delta t} * f = f + \delta t \left(f^{(2)} + f' \kappa \right) + \delta t^2 \left[\frac{f^{(4)}}{2} + f^{(3)} \kappa + f^{(2)} \left(\frac{\kappa^2}{2} + \partial_n \kappa \right) + f' W' \right] + O(\delta t^3). \quad (2.97)$$

Avec :

$$\frac{\kappa^2}{2} + \partial_n \kappa = 2K - \frac{\kappa^2}{2} = -2(H^2 - K). \quad (2.98)$$

Cela sugg re de choisir $f(r) = \frac{r^2}{2}$ de telle sorte que $f^{(4)} = f^{(3)} = 0$ et $f^{(2)} = 1$ et d'obtenir le terme correctif $c = 2(H^2 - K)$:

$$G_{\delta t} * \frac{d^2}{2} = \frac{d^2}{2} + \delta t (1 + d \kappa) + \delta t^2 [-c + d W'] + O(\delta t^3) \quad (2.99)$$

On peut r crire la correction B sous la forme :

$$B = Hc \quad (2.100)$$

On peut donc obtenir B en calculant la courbure moyenne et en extrayant c de (2.99) par une combinaison lin aire de deux convolutions de $\frac{d^2}{2}$:

$$2G_{\sqrt{\frac{\delta t}{2}}} * \frac{d^2}{2} - G_{\sqrt{2\delta t}} * \frac{d^2}{2} = \frac{d^2}{2} - 2\delta t (c + d W') + O\left(\delta t^{\frac{3}{2}}\right). \quad (2.101)$$

En combinant (2.95) et (2.101), on peut obtenir le d placement par flot de Willmore de la fa on suivante :

$$\left[2G_{\sqrt{\frac{\delta t}{2}}} * d - G_{\sqrt{2\delta t}} * d \right] (1 - Hd) + \frac{H}{2} \left[2G_{\sqrt{\frac{\delta t}{2}}} * \frac{d^2}{2} - G_{\sqrt{2\delta t}} * \frac{d^2}{2} \right] + \frac{Hd^2}{2} = d - W\delta t + O\left(\delta t^{\frac{3}{2}}\right). \quad (2.102)$$

Il convient de remarquer que l'on a multipli  l'expression $2G_{\frac{\delta t}{2}} * d - G_{\sqrt{2\delta t}} * d$ par $1 - Hd$ pour  liminer le terme $dW'\delta t$ qui appara t dans (2.101).

De la m me fa on que pour la dimension 2, on peut obtenir la courbure   partir des diffusions $G_{\sqrt{\frac{\delta t}{2}}} * d$ et $G_{\sqrt{2\delta t}} * d$. Au total, on effectue donc 4 convolutions, deux pour d et deux pour $\frac{d^2}{2}$.

De plus, comme on utilise les m mes pas de temps pour d et d^2 , pour la r solution en  l ments finis, on peut r -utiliser les m mes pr -conditionneurs pour chaque pas de temps. Les diff rentes convolutions pouvant  tre r alis es en parall le, on peut conserver le m me temps de calcul en augmentant le nombre de c urs de calcul.

Finalement, l'algorithme s' crit :

Algorithme 12 : Flot de Willmore 3D :  tant donn s la fonction distance sign e   l'ensemble Ω_0 d'interface Γ_0 , un pas de temps δt et un nombre d'it rations nb_{iter} , g n rer la suite d'ensembles $(\Omega_k)_k$ d crits par leurs fonctions distances sign es d^k .

Pour k de 0   $nb_{iter} - 1$:

1)  tape de convolution : Calculer les convolutions : $G_{\sqrt{2\delta t}} * d$, $G_{\frac{\sqrt{\delta t}}{2}} * d$ et $G_{\sqrt{2\delta t}} * d^2$, $G_{\frac{\sqrt{\delta t}}{2}} * d^2$.

2) Calculer la courbure moyenne :

$$H = \left(\frac{G_{\sqrt{\delta t}} * d^k - d^k}{2} \right)^3 \text{ ou } \left[\left(\sqrt{2}G_{\sqrt{\frac{\delta t}{2}}} - \frac{1}{2\sqrt{2}}G_{\sqrt{2\delta t}} \right) * d^k - \frac{3\sqrt{2}}{4}d^k \right]^3 \quad (2.103)$$

3) Former les termes :

$$\begin{aligned} A^k &= 2G_{\sqrt{\delta t}} * d - G_{2\sqrt{\delta t}} * d \\ B^k &= 2G_{\sqrt{\delta t}} * d^2 - G_{2\sqrt{\delta t}} * d^2 \end{aligned} \quad (2.104)$$

4)  tape de redistanciation : $d^{k+1} = \text{redist} \left[\frac{Hd^2}{2} + (1 - Hd)A^k + \frac{H}{2}B^k \right]$.

2.4.11 V rification num rique du mouvement par flot de Willmore en dimension 3

Dans ce partie, nous allons v rifier num riquement le d placement par flot de Willmore en dimension 3 de l'algorithme 2.4.10 : 12. Le cas d'une sph re, pour

laquelle le terme $\Delta_\Gamma H$ est nul et $K = H^2$, donne une vitesse normale nulle :

$$W = \Delta_\Gamma H + 2H(H^2 - K) = 0 \quad (2.105)$$

On pr f re alors s'int resser sur le cas d'un tore r gulier. On note $T_{a,b}$ le demi grand axe a et de demi petit axe b ($T_{a,b}$ est r gulier donc $b < a$) dont la param trisation est donn e par :

$$X(u, v) = \begin{cases} x = (a + b \cos v) \cos u \\ y = (a + b \cos v) \sin u \\ z = b \sin v \end{cases} \quad (2.106)$$

Par sym trie, un tore  voluant selon le flot de Willmore va rester un tore (ou  ventuellement changer de genre topologique, par exemple en *remplissant* son trou). Le th or me 1.2.2 nous dit que l'on devrait converger vers un tore de Clifford dont le ratio entre ses axes vaut $\lambda = \frac{a}{b} = \sqrt{2}$.

On peut exprimer les diff rentes courbures du Tore $T_{a,b}$ comme suit :

$$\begin{cases} H = \frac{1}{2} \frac{a + 2b \cos v}{b(a + b \cos v)} \\ K = \frac{1}{b(a + b \cos v)} \end{cases} \quad (2.107)$$

L' nergie d'Helfrich s'exprime alors :

$$\begin{aligned} E_c(a, b) &= \int_0^{2\pi} \int_0^{2\pi} H(u, v)^2 \|X_u \times X_v\| du dv \\ &= \frac{a^2 \pi^2}{b \sqrt{a^2 - b^2}} \end{aligned} \quad (2.108)$$

Le gradient de E_c s'exprime :

$$\nabla E_c(a, b) = \frac{a\pi^2}{b^2} \frac{2b^2 - a^2}{(2b^2 - a^2)^{\frac{3}{2}}} \begin{pmatrix} -b \\ a \end{pmatrix} \quad (2.109)$$

L' volution selon le flot de Willmore est d crite par l' quation diff rentielle :

$$\partial_t \begin{pmatrix} a \\ b \end{pmatrix} = \nabla E_c(a, b) \quad (2.110)$$

En suivant (2.110), la quantit  $a^2 + b^2$ doit  tre conserv e :

$$\partial_t (a^2 + b^2) = 2(a\partial_t a + b\partial_t b) = 0 \quad (2.111)$$

Ainsi, le tore $T_{a,b}$ doit converger vers le tore de Clifford donn  par les syst me suivant :

$$\begin{cases} a = \sqrt{2}b \\ a^2 + b^2 = a_0^2 + b_0^2 \end{cases} \quad (2.112)$$

Cette convergence num rique est r alis e avec une m thode de champ de phase dans [BMO15].

En pratique, l'algorithme 2.4.10 : 12 calcule une version discr te de l' quation diff rentielle (2.110). En posant $F(a, b) = \frac{a\pi^2}{b^2} \frac{2b^2 - a^2}{(a^2 - b^2)^{\frac{3}{2}}}$ et ρ_n notre pas de descente, on a :

$$\begin{pmatrix} a_{n+1} \\ b_{n+1} \end{pmatrix} = \begin{pmatrix} a_n + \rho_n F(a_n, b_n) b_n \\ b_n - \rho_n F(a_n, b_n) a_n \end{pmatrix} \quad (2.113)$$

La descente de gradient discr te s'exprime alors :

$$a_n^2 + b_n^2 = \prod_{i=0}^{n-1} [1 + \rho_i^2 F(a_i, b_i)^2] (a_0^2 + b_0^2) \quad (2.114)$$

Cela sugg re que l'on n'obtiendra pas une conservation totale de la quantit  $a^2 + b^2$ dans nos simulations.

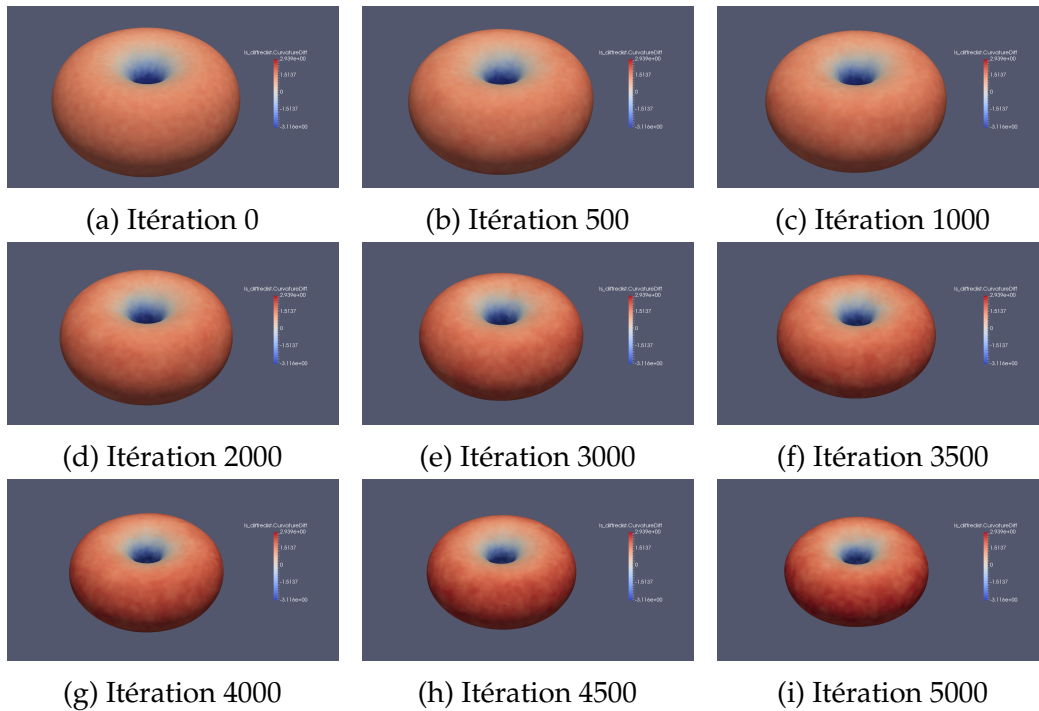


FIGURE 2.13 – Convergence num rique du tore T selon le flot de Willmore vers un tore de Clifford.

On retrouve les valeurs de a et b de $T_{a,b}$ dans les simulations   partir du volume et de l'aire gr ce aux formules suivantes :

$$\begin{cases} a = \frac{1}{8\pi^2} \frac{A^2}{V} \\ b = \frac{2V}{A} \end{cases} \quad (2.115)$$

On pr sente   la figure 2.13 l' volution du tore $T_{a,b}$ selon le flot de Willmore. Le pas de temps choisi est 9×10^{-6} pour un pas d'espace de 0.04 avec un raffinement proche de l'interface pour environ 4,7 millions d' l ments.

L' volution de $a^2 + b^2$ en fonction du temps est repr sent  sur le graphe 2.16. On constate que l'on ne conserve pas $a^2 + b^2$ comme le laissait penser (2.113).

On repr sente  galement l' volution au cours du temps de l' nergie d'Helfrich   la figure 2.14 et du rapport $\frac{a}{b}$   la figure 2.15. On constate que l' nergie et le rapport $\frac{a}{b}$ tendent bien vers ceux de Clifford. Cependant, une fois atteint l'algorithme ne s'arr te pas et d passe l' tat d' quilibre et ne semble pas changer de direction.

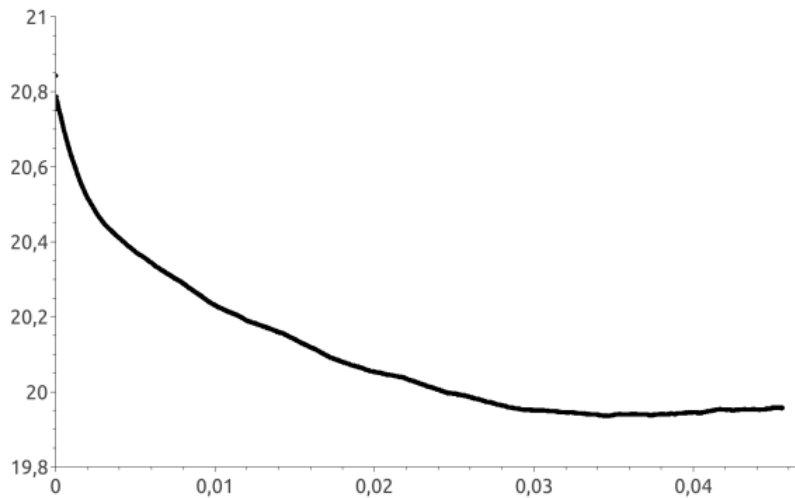


FIGURE 2.14 –  volution de l' nergie d'Helfrich au cours du temps. On observe une d croissance jusqu'  au minimum puis la courbe cro t   nouveau.

La stabilit  num rique de cette simulation est particuli rement d licate. En effet, la m thode est tr s sensible aux asym tries. Si celles-ci s'accumulent, on ne conserve plus la forme d'un tore et l'algorithme explose alors assez rapidement. D'autre part, si l'on arrive   obtenir le tore de Clifford, nous n'arrivons pour l'instant pas   se stabiliser dessus. Des tests plus en profondeur sont en train d' tre men s pour avoir une meilleure compr hension des performances de l'algorithme dans ce cas particulier.

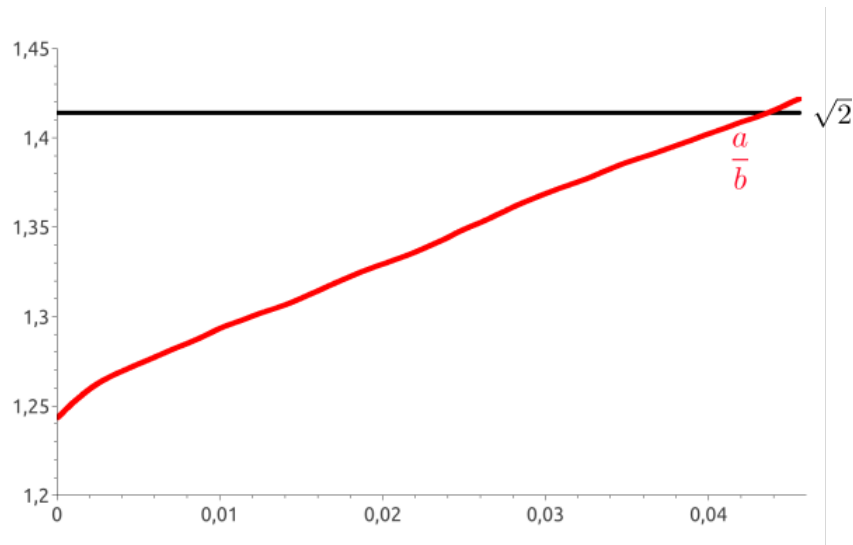


FIGURE 2.15 –  volution du rapport $\frac{a}{b}$ au cours du temps. On observe qu'il augmente vers $\sqrt{2}$. Cependant, on ne se stabilise pas   $\sqrt{2}$.

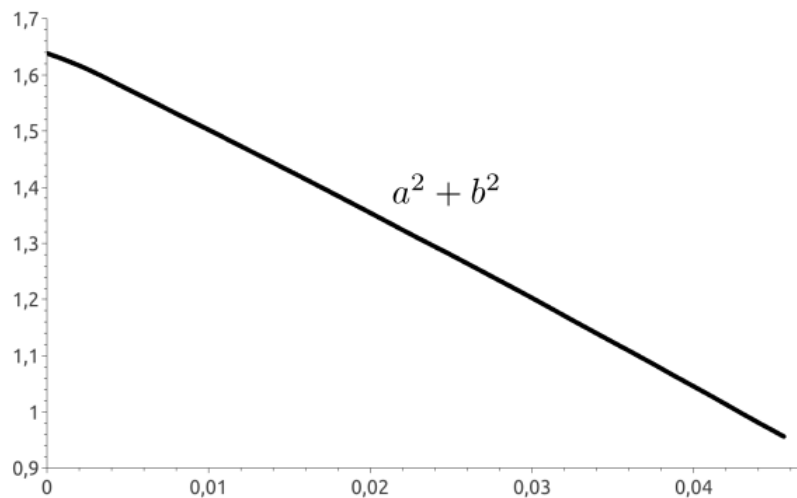


FIGURE 2.16 –  volution de la quantit  $a^2 + b^2$ au cours du temps. On observe une l g re d croissance de celle-ci.

2.5 Convergence th orique du mouvement par courbure moyenne

Cette th ese ayant une vocation plus num rique que th orique, nous exposons bri vement quelques r sultats de convergence [Eva93, BG95] sur l'algorithme de convolution-seuillage. Par souci p dagogique, on pr sentera succinctement les  l ments de d monstration de trois approches : celle de Almgren-Taylor-Wang

et Luckhaus-Sturzenhecker [ATW93, LS95], celle de Laux et Otto [LO16] et celle de Chambolle [Cha04, CN06]. On conclura cette partie par des remarques sur la possibilit  d'adapter ces r sultats   la m thode de convolution-redistanciation.

La convergence th orique de l'algorithme de convolution-seuillage pour le mouvement par courbure moyenne est un sujet qui a  t  beaucoup  tudi  depuis son introduction. Plusieurs preuves ont  t  propos es et nous allons pr senter les diff rents formalismes utilis s. La plupart reposent sur un principe de minimisation d'une certaine  nergie. D'autres preuves reposent sur un principe de comparaison et sur le formalisme des solutions de viscosit .

2.5.1 Notations

On pr sente rapidement les notations que l'on va utiliser dans cette section. On consid re un ouvert A born  de \mathbb{R}^N et $\Omega(0) \subset A$ l'ensemble initial de fronti re $\Gamma(0)$. On  tudiera l' volution de la suite d'ensembles $(\Omega(t))_t$, de fronti re $(\Gamma(t))_t$ et de fonction caract ristique $(\chi(t))_t$. Nous choisissons ces notations afin de nous conformer aux formalismes adopt s dans les articles que nous pr sentons.

2.5.2 L'espace des fonctions   variations born es $BV(\Omega)$

L'espace des fonctions   variations born es est un bon espace de travail pour le cadre de l' tude des algorithmes de convolution-seuillage pour deux raisons. D'une part, il contient les fonctions caract ristiques d'ensembles mesurables born s. D'autre part, il garantit l'existence de la variation totale des ses  l ments et la variation totale d'une fonction caract ristique mesure le p rim tre de l'ensemble.

D finition 2.5.1. Une fonction f de $\mathbb{L}^1(A)$ est   variation born e dans A si :

$$J(f) = \sup \left\{ \int_A f(x) \operatorname{div} g(x) dx / g \in C_c^1(\Omega, \mathbb{R}^2), \|g\|_\infty \leq 1 \right\} < +\infty \quad (2.116)$$

On note

$$BV(A) = \{f \in \mathbb{L}^1(A) / J(f) < +\infty\} \quad (2.117)$$

l'ensemble des fonctions   variations born es.

De plus, pour $f \in BV(A)$, $J(f)$ sera appel e variation totale de f et $|\nabla f|$ est une mesure de Radon.

Cette d finition se base sur une formulation faible de la mesure des variations du gradient de f . En effet, pour une fonction de classe $C^1(A)$, on a :

$$\int_A f \operatorname{div}(\phi) dx = - \int_A \nabla f \cdot \phi dx \quad (2.118)$$

L'intérêt de cet ensemble est qu'il contient les fonctions caractéristiques des ensembles mesurables bornés et ouverts. De plus, la variation totale d'une fonction caractéristique mesure le périmètre de l'interface de l'ensemble qu'elle représente :

$$\text{Per}(\Gamma) = \int_A |\nabla \chi(x)| dx \tag{2.119}$$

2.5.3 L'approche de Almgren-Taylor-Wang et Luckhaus-Sturzenhecker

On présente ici une preuve basée sur un principe de minimisation de flot de gradient. Elle a été établie parallèlement dans [ATW93] et [LS95].

On présente d'abord comment est construit le schéma discret simulant le mouvement par courbure moyenne $(\chi_{\delta t}(t))_t$ étant donné un certain pas de temps δt .

On considère l'énergie, définie sur les ensembles mesurables inclus dans A :

$$E_L(\Omega(t)) = \int_A |\nabla \chi| + \int_A \frac{1}{\delta t} d(x, \Gamma(t - \delta t)) \chi(t) dx \tag{2.120}$$

où $d(x, \Gamma(t - h))$ désigne la distance signée à l'hypersurface $\Gamma(t - h)$. On pose :

$$\Omega_{min}^{E_L} = \min_{\Omega \subset A} E_L(\Omega) \tag{2.121}$$

On définit l'opérateur $\mathcal{T}_L^{\delta t}$ suivant :

$$\begin{aligned} \mathcal{T}_L^{\delta t} : \mathcal{P}(A) &\rightarrow \mathcal{P}(A) \\ \Omega &\mapsto \Omega_{min}^{E_L} \end{aligned} \tag{2.122}$$

On peut alors définir un schéma d'évolution discret en temps initialisé en $\Omega_0 \subset A$ pour un pas de temps $\delta t > 0$:

$$\Omega_{\delta t}(t) = \left(\mathcal{T}_L^{\delta t} \right)^{\lfloor t/\delta t \rfloor} (\Omega_0) \tag{2.123}$$

La notion de flot de courbure moyenne considérée dans [LS95] est la formulation faible pour les fonctions à variations bornées.

Définition 2.5.2. Soient $A \subset \mathbb{R}^d$ un ensemble ouvert borné et $\Omega(0) \subset A$. On pose $A_T = A \times]0, T[$, avec $T \in \mathbb{R}_+$ un temps fixé. On dit qu'une fonction $\chi : A_T \rightarrow \{0, 1\} \in \mathbb{L}^\infty(]0, T[; BV(A))$ décrit un mouvement par courbure moyenne s'il existe une fonction vitesse $v : A_T \rightarrow \mathbb{R} \in \mathbb{L}^1(]0, T[, \mathbb{L}^1(|\nabla \chi|))$ telle que pour tout $\xi \in C^\infty(\overline{A_T}, \mathbb{R}^d)$ vérifiant sur les bords $\xi = 0$ sur $\partial A \times]0, T[$:

$$\int_{A_T} \left[\left(\text{div}(\xi) - \frac{\nabla \chi}{|\nabla \chi|} \nabla \xi \frac{\nabla \chi}{|\nabla \chi|} \right) - v \xi \nabla \chi \right] dx dt = 0 \tag{2.124}$$

et pour tout $\zeta \in C^\infty(\overline{A_T}, \mathbb{R}^d)$ v rifiant $\zeta = 0$ sur $(\partial A \times]0, T[) \cup A \times \{T\}$:

$$\int_{A_T} \chi \partial_t \zeta dt dx + \int_A \chi(0, x) \zeta(0, x) dx = - \int_{A_T} v \zeta |\nabla \chi| dx dt \quad (2.125)$$

Le r sultat de convergence est le suivant :

Th or me 2.5.3. [Th or me 2.3 [LS95]] Soit $\Omega_0 \subset A$ un ensemble de p rim tre fini. On d finit Ω_h la solution discr te donn e par (2.123) et χ_h sa fonction caract ristique. Alors il existe une sous-suite de (χ_h) (que l'on note toujours χ_h par simplicit ) qui converge vers une fonction $\chi \in \mathbb{L}^1(]0, T[, BV(A, \{0, 1\}))$ dans $\mathbb{L}^1(A_T)$.

De plus, si $d \leq 7$ et que l'on a :

$$\lim_{h \rightarrow 0} \int_0^T \int_A |\nabla \chi_h| dt dx = \int_0^T \int_A |\nabla \chi| dt dx \quad (2.126)$$

alors la limite χ d crit un mouvement par courbure moyenne au sens (2.5.2).

Remarque 2.5.4. L'hypoth se (2.126) permet de s'assurer qu'il n'y a pas de perte de surface en passant   la limite $h \rightarrow 0$ pour un t fix .

Id e de la preuve : On pr sente ici un r sum  de la preuve du th or me (2.5.3) dans [LS95]. Elle se d compose essentiellement en deux  tapes. D'abord, il convient de prouver l'existence d'une limite χ de la suite de (χ_h) dans \mathbb{L}^1 lorsque l'on fait tendre h vers 0. Pour cela, une adaptation du th or me de Riesz-Frechet-Kolmogorov est appliqu e. Il est alors n cessaire de montrer la compacit  de la solution discr tis e χ_h en temps et en espace.

- La compacit  en espace d coule imm diatement de la construction de χ_t comme minimiseur de l' nergie (2.120). En effet, en consid rant la fonction χ_{t-h} comme comp titeur de l' nergie (2.120), on obtient la d croissance du p rim tre de l'ensemble Ω_h :

$$\int_A |\nabla \chi| \leq \int_A |\nabla \chi_0| \quad (2.127)$$

En cons quence, on obtient la compacit  en espace. Pour $(s, e) \in \mathbb{R}_+^* \times \mathbb{S}^{d-1}$, on a :

$$\begin{aligned} \int_A |\chi_h(x + se) - \chi_h(x)| dx &\leq s \int_A |\nabla \chi_h(x)| dx \\ &\leq s \int_A |\nabla \chi_0(x)| dx \end{aligned} \quad (2.128)$$

Ainsi, en int grer en temps et en passant   la limite :

$$\lim_{s \rightarrow 0} \int_0^T \int_A |\chi_h(x + se) - \chi_h(x)| dx = 0 \quad (2.129)$$

- La compacit  en temps est plus d licate   prouver. En effet, il faut d montrer :

$$\int_A \int_h^T |\partial_t^{-h} \chi_h| < C \quad (2.130)$$

Cette estimation repose sur l'in galit  isop rim trique ainsi que d'autres consid rations g om triques.

Comme la fonction χ_h est discr te en temps, on peut supposer sans perte de g n ralit  que $\tau = kh$. On a alors :

$$\begin{aligned} \int_0^{T-\tau} \int_A |\chi_h(\cdot + \tau) - \chi_h| &\leq \int_0^{T-\tau} \int_A \sum_{i=1}^k |\chi_h(\cdot + ih) - \chi_h(\cdot + (i-1)h)| \\ &\leq \int_0^{T-\tau} \int_A \sum_{i=1}^k |\partial_t^{-h} \chi_h| h \\ &\leq \int_0^T \int_A |\partial_t^{-h} \chi_h| kh \\ &\leq C\tau \end{aligned} \quad (2.131)$$

Par application du th or me de Riesz-Fr chet-Kolmogorov, l'ensemble $\{\chi_h\}_{h>0}$ est relativement compact dans $\mathbb{L}^1(A_T)$. Cela assure l'existence d'une fonction $\chi \in \mathbb{L}^1(A_T)$ limite d'une sous-suite de χ_h (que l'on notera encore χ_h pour simplifier les notations). On rappelle l' nonc  du th or me donn  dans [Bre83] :

Th or me 2.5.5. Soit Ω un ouvert de \mathbb{R}^N et soit ω une partie ouverte et relativement compacte de Ω . Soit \mathcal{F} un sous-ensemble born  de $\mathbb{L}^p(\Omega)$, avec $1 \leq p < +\infty$. On suppose que :

$$\forall \varepsilon > 0, \exists 0 < \delta < d(\omega, \Omega^c), \text{ tel que} \quad (2.132)$$

$$\|\tau_h f - f\|_{\mathbb{L}^p(\omega)} < \varepsilon, \forall h \in \mathbb{R}^N \text{ avec } \|h\| < \delta \text{ et } f \in \mathcal{F}$$

Alors $\mathcal{F}|_{\omega}$ est relativement compact dans $\mathbb{L}^p(\omega)$.

Soit $G \in \mathbb{L}^1(\mathbb{R}^N)$ une fonction fix e et soit $\mathcal{F} = G * \mathcal{B}$ o  \mathcal{B} d signe un sous-ensemble born  de $\mathbb{L}^p(\mathbb{R}^N)$, avec $1 \leq p < \infty$.

Alors $\mathcal{F}|_{\omega}$ est relativement compact dans $\mathbb{L}^p(\omega)$ pour tout ouvert born  ω de \mathbb{R}^N .

La deuxi me  tape de la preuve consiste   construire une fonction de vitesse $v \in \mathbb{L}^1(0, T, \mathbb{L}^1(|\nabla \chi(t)|))$ satisfaisant (2.125) et prouver ensuite qu'elle v rifie  galement (2.124).

Comme χ_h minimise l' nergie (2.120), elle v rifie l' quation d'Euler-Lagrange associ e :

$$\int_A \left(\operatorname{div}(\xi) - \frac{\nabla \chi_h}{|\nabla \chi_h|} \nabla \xi \frac{\nabla \chi_h}{|\nabla \chi_h|} \right) |\nabla \chi_h| - \frac{1}{h} d(\cdot, \Gamma_{t-h}) \xi \nabla \chi_h = 0 \quad (2.133)$$

On d finit alors la vitesse discr te $v_h = \frac{1}{h} d(x, \Gamma_{t-h})$ et il convient de montrer qu'elle se comporte   la limite comme la vitesse v de l' quation (2.124).

On a, pour tout $\zeta \in C^\infty(\overline{A_T})$ avec $\zeta(T) = 0$, par int gration par parties :

$$\int_{A_T} \chi_h \partial_t^h \zeta + \frac{1}{h} \int_A \int_0^h \chi_0 \zeta = - \int_{A_T} \partial_t^{-h} \chi_h \zeta \quad (2.134)$$

L'estimation (2.130) permet d'assurer la convergence faible d'une sous-suite vers une mesure μ . Pour toute fonction ζ continue sur A_T :

$$\lim_{h \rightarrow 0} \int_{A_T} \left(\partial_t^{-h} \chi_h - \frac{1}{h} d(\cdot, \Gamma_{t-h}) |\nabla \chi_h| \right) \zeta = 0 \quad (2.135)$$

Ainsi, la mesure $v_h |\nabla \chi_h|$ converge faiblement vers μ . Le th or me de Radon-Nikodym garantit l'existence d'une fonction $v \in \mathbb{L}^1(0, T; \mathbb{L}^1(|\nabla \chi(t)|))$ telle que

$$\mu = v |\nabla \chi| \quad (2.136)$$

L' quation (2.125) est alors  tablie et il reste   montrer (2.124). Cela se fait en passant   la limite $h \rightarrow 0$ dans (2.133) en prenant une normale n_ε r guli re et proche de $n = \frac{\nabla \chi}{|\nabla \chi|}$ dans $\mathbb{L}^1(0, T, \mathbb{L}^1(|\nabla \chi(t)|))$. Il faut enfin montrer le passage   la limite sur $\varepsilon \rightarrow 0$.

□

Remarque 2.5.6. Cette preuve a  t   tendue au mouvement par courbure moyenne avec conservation de volume dans [MSS16]. Un terme de p nalisation de la forme $\frac{1}{\sqrt{h}} |\operatorname{Vol}(\Omega) - 1|$ est ajout  au probl me de minimisation (2.120) pour imposer la contrainte de conservation (le volume ayant  t  normalis    1) :

$$E_L(\Omega(t)) = \int_A |\nabla \chi| + \int_A \frac{1}{h} d(x, \Gamma(t - \delta t)) \chi(t) dx + \frac{1}{\sqrt{h}} |\operatorname{Vol}(\Omega) - 1| \quad (2.137)$$

2.5.4 L'approche de Laux-Otto

Dans [LO16], les auteurs montrent la convergence de l'algorithme multiphase 2.3.1 : 4 qui permet des tensions de surfaces diff rentes entre chaque phase. Leur preuve est une preuve de minimisation du m me type que l'approche de la section pr c dente.

Les auteurs utilisent dans l'interpr tation faite par Esedoglu et Otto dans [EO15] de l'algorithme de convolution-seuillage multiphase 2.3.1 : 4 comme minimisant la fonctionnelle suivante :

$$\chi^n \text{ r alise } \min_{\chi \in \{0,1\}} E_h(\chi) - E_h(\chi - \chi^{n-1}) \quad (2.138)$$

o  :

$$E_h(\chi) = \frac{1}{\sqrt{h}} \sum_{i,j} \sigma_{i,j} \int \chi_i G_h * \chi_j dx \quad (2.139)$$

Cela se v rifie ais ment en constatant que :

$$E_h(\chi) - E_h(\chi - \chi^{n-1}) = \frac{2}{\sqrt{h}} \sum_{i=1}^N \int \chi_i \left(\sum_{j=1, j \neq i}^N \sigma_{i,j} G_h * \chi_j^{n-1} \right) dx + T(\chi^{n-1}) \quad (2.140)$$

o  $T(\chi^{n-1})$ d signe un terme d pendant uniquement de χ^{n-1} , et n'intervenant donc pas dans le probl me de minimisation. Pour minimiser cette  quation, il faut donc prendre $\chi_i = 1$ pour l'indice i tel que la quantit  $\sum_{j=1, j \neq i}^N \sigma_{i,j} G_h * \chi_j^{n-1}$, ce qui correspond exactement   l' tape de seuillage de l'algorithme.

Il a  t  montr  dans [EO15] que l' nergie Γ -converge pour la topologie \mathbb{L}^1 vers le p rim tre total de l'interface Γ , donn  par :

$$E(\chi) = \frac{1}{\sqrt{2\pi}} \sum_{i,j} \sigma_{ij} \int_A \frac{1}{2} (|\nabla \chi_i| + |\nabla \chi_j| - |\nabla(\chi_i + \chi_j)|) dt \quad (2.141)$$

On donne maintenant l'adaptation de la d finition de flot par courbure moyenne (2.5.2) au probl me multiphase.

D finition 2.5.7. [D finition (1.1) de [LO16]] Soient un temps $T < +\infty$, une matrice de tension de surface admissible au sens de [LO16] et une donn e initiale $\chi^0 : [0, \Lambda]^d \rightarrow \{0, 1\}^P$ avec $E(\chi^0) < +\infty$. On dit que $\chi = (\chi_1, \dots, \chi_P) : (0, T) \times [0, \Lambda]^d \rightarrow \{0, 1\}^P$ tel que :

$$\begin{cases} \sum_i \chi_i = 1 \text{ presque partout} \\ \sup_t E(\chi(t)) < +\infty \end{cases} \quad (2.142)$$

se d place par courbure moyenne si, pour chaque $i \in [1, P]$, il existe une fonction

vitesse $V_i : (0, T) \times [0, \Lambda]^d \rightarrow \mathbb{R}$ telle que

$$\int_0^T \int V_i^2 |\nabla \chi_i| dt < +\infty \quad (2.143)$$

vérifiant pour tout $\xi \in C_0^\infty((0, T) \times [0, \Lambda]^d, \mathbb{R}^d)$ et pour tout $\zeta \in C^\infty([0, T] \times [0, \Lambda]^d)$, $\zeta(T) = 0$:

$$\begin{cases} \sum_{i,j} \sigma_{ij} \int_0^T \int (\nabla \cdot \xi - v_i \cdot \xi v_i - 2\xi \cdot v_i V_i) \frac{1}{2} (|\nabla \chi_i| + |\nabla \chi_j| - |\nabla(\chi_i + \chi_j)|) dt = 0 \\ \int_0^T \int \partial_t \zeta \chi_i dx dt + \int \zeta(0) \chi_i^0 dx = - \int_0^T \int \zeta V_i |\nabla \chi_i| dt \end{cases} \quad (2.144)$$

Comme dans la section précédente, l'hypothèse suivante est faite pour s'assurer qu'il n'y a pas de perte de surface lors du passage à la limite $h \rightarrow 0$.

$$\int_0^T E_h(\chi^h) dt \rightarrow \int_0^T E(\chi) dt \quad (2.145)$$

Théorème 2.5.8. [Théorème (1.3) de [LO16]] Soient $P \in \mathbb{N}$, une matrice de tension de surface admissible au sens de (2.12), un temps d'horizon $T < \infty$ et une fonction caractéristique χ^0 telle que $E(\chi^0) < \infty$. Alors, si on suppose (2.145), pour toute suite $(h_n)_n$ qui tend vers 0, il existe une sous-suite, encore notée $(h_n)_n$ et une fonction χ telle que $E(\chi(t)) \leq E(\chi^0)$ telles que la suite χ^{h_n} donnée par l'algorithme multiphase (2.3.1 : 4 converge vers χ qui se déplace par courbure moyenne au sens de la définition (2.5.7).

Idée de la preuve : On présente succinctement les différentes étapes de la preuve faite dans [LO16]. Elles suivent essentiellement le même raisonnement que la preuve précédente :

- Une adaptation du théorème de Riesz-Fréchet-Kolmogorov permet de montrer la pré-compacité de $(\chi^h)_h$ dans \mathbb{L}^1 et l'existence de la limite χ . L'argument principal est l'inégalité de dissipation suivante, qui est une conséquence directe de la propriété de minimisation (2.138) :

$$E_h(\chi^n) - \sum_{k=1}^n E_h(\chi^k - \chi^{k-1}) \leq E(\chi^0) \quad (2.146)$$

- Une fois l'existence de χ prouvée, il convient de montrer l'existence des fonctions vitesses V_i . Pour cela, on montre d'abord que $\partial_t \chi$ est une mesure de Radon et chaque $\partial_t \chi_i$ est absolument continue par rapport à

la mesure de p rim tre $|\nabla\chi|dt$, ce qui justifie l'existence de la densit  $V_i \in \mathbb{L}^1(|\nabla\chi_i|dt)$. Il faut  galement montrer que ces vitesses appartiennent   $\mathbb{L}^2(|\nabla\chi_i|dt)$.

L  encore, l'ingr dient principal pour ces propri t s est l'in galit  (2.146) qui permet de borner les int grales consid r es.

- Maintenant que l'existence des diff rents  l ments de la d finition (2.5.7) a  t  prouv e, il reste   montrer le syst me (2.144). Cela est fait en passant   la limite dans l' quation d'Euler-Lagrange associ e au probl me de minimisation dans la direction ξ :

$$\frac{2}{\sqrt{h}} \sum_{i,j} \sigma_{ij} \int \chi_i G_{h^*}(-\nabla\chi_j \cdot \xi) dx + \frac{2}{\sqrt{h}} \sum_{i,j} \sigma_{ij} \int (\chi_i - \chi_i^n) G_{h^*}(\nabla\chi_j \cdot \xi) dx = 0 \quad (2.147)$$

- Le premier terme correspond   la variation $\delta E(\chi, \xi)$ du terme $E(\chi)$ de p rim tre. Il est montr  qu'il converge vers l' nergie de p rim tre :

$$\lim_{h \rightarrow 0} \int_0^T \delta E_h(\chi^h, \xi) dt = \frac{1}{\sqrt{2\pi}} \sum_{i,j} \sigma_{ij} \int_0^T \int (\nabla \cdot \xi - v_i \cdot \nabla \xi v_i) \frac{1}{2} (|\nabla\chi_i| + |\nabla\chi_j| - |\nabla(\chi_i + \chi_j)|) dt \quad (2.148)$$

- Le second terme correspond   la variation $-\delta E_h(\chi - \chi^n, \xi)$ du terme de dissipation. Il est montr  qu'il se comporte asymptotiquement comme le terme de vitesse $\sum_{i,j} \sigma_{ij} \int \xi_i V_j |\nabla\chi_j| dx$ et permet ainsi de r cup rer les termes de vitesses pr sents dans la d finition (2.5.7).

□

Remarque 2.5.9. Cette preuve a  t  adapt e dans [LS16] pour montrer la convergence de l'algorithme 2.3.2 : (5) vers un flot par courbure moyenne avec conservation de volume. Un terme de p nalisation de la forme $\frac{2\lambda-1}{\sqrt{h}} \int \chi dx$ est ajout  au probl me de minimisation (2.138) pour imposer la contrainte de conservation (le volume ayant  t  normalis    1).

2.5.5 L'approche de Chambolle

On s'int resse au formalisme propos  dans [Cha04]. Il s'agit ici encore de r soudre un probl me de minimisation pour construire une solution discr te en

temps. Cependant, la démonstration de la convergence est faite à l'aide de la notion de solution de viscosité. Pour $A \subset \mathbb{R}^d$ ouvert convexe, on considère :

$$\min_{w \in \mathbb{L}^2(\Omega)} \int_A |\nabla w| + \int_A \frac{(w(x) - d_\Omega(x))^2}{2\delta t} dx \quad (2.149)$$

L'équation d'Euler-Lagrange associée justifie le choix de (2.149) :

$$\frac{w - d_\Omega(x)}{\delta t} = \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \quad (2.150)$$

On retrouve en effet un schéma d'Euler de discrétisation du mouvement par courbure moyenne.

Il s'agit d'un problème convexe dont on peut montrer qu'il admet une unique solution notée w_Ω . On définit l'opérateur $\mathcal{T}_C^{\delta t}$ suivant :

$$\begin{aligned} \mathcal{T}_C^{\delta t} : \mathcal{P}(A) &\rightarrow \mathcal{P}(A) \\ \Omega &\mapsto \{w_\Omega < 0\} \end{aligned} \quad (2.151)$$

On peut alors définir un schéma d'évolution discret en temps initialisé en $\Omega_0 \subset A$ pour un pas de temps $\delta t > 0$:

$$\Omega_{\delta t}(t) = \left(\mathcal{T}_C^{\delta t} \right)^{\lfloor t/\delta t \rfloor} (\Omega_0) \quad (2.152)$$

On s'intéresse maintenant à la convergence de cette méthode vers le mouvement par courbure moyenne. Une preuve de consistance est établie dans [CN06] et se base sur un principe de comparaison. On en présentera l'idée générale dans un deuxième temps.

La proposition qui suit est très importante car elle établit l'équivalence entre les schémas (2.152) et (2.123).

Proposition 2.5.10 (Proposition 2.2 [Cha04]). *Soit $\Omega \subset A$, alors $\mathcal{T}_C^{\delta t}(\Omega)$ est solution du problème de minimisation (2.120).*

Les conséquences de cette proposition sont multiples. D'abord, cela implique qu'étudier la convergence du schéma (2.152) ou du schéma (2.123) est équivalent. Ainsi, les résultats de convergence précédemment énoncés s'appliquent à (2.152).

D'autre part, la schéma (2.149) fournit un algorithme pour simuler numériquement le mouvement par courbure moyenne. En effet, étant donné la fonction indicatrice de l'ensemble de Ω_0 , une étape du schéma (2.152) consiste à construire la fonction distance à l'interface Γ_0 puis à résoudre le problème d'optimisation convexe (2.149). Cette méthode numérique peut d'ailleurs être

rapproch e au sch ma de convolution-redistanciation. En effet, l' tape d'optimisation correspond   l' tape de convolution et elle est suivie d'une  tape de seuillage. N anmoins, la r solution num rique de (2.149) requiert des calculs beaucoup plus co teux algorithmiquement que la r solution d'une  quation de la chaleur. Il faut en effet r soudre un probl me d'optimisation convexe.

Il est d'ailleurs int ressant de noter que la r solution d'une  tape de diffusion consiste   minimiser une fonctionnelle proche de (2.149) :

$$\min_w \int_A |\nabla w|^2 + \frac{1}{\delta t} (w(x) - d_\Omega(x))^2 dx \quad (2.153)$$

On pr sente maintenant les r sultats de convergence  nonc s dans [Cha04] et [CN06]. La notion de consistance du sch ma est bas e sur un principe de comparaison avec un sur-flot et sous-flot. On donne ici les d finitions n cessaires et on pr sente les r sultats obtenus dans [Cha04] et [CN06]. La convergence du sch ma est d montr e en utilisant la monotonie de $\mathcal{T}_C^{\delta t}$ et sa consistance.

Proposition 2.5.11 (Monotonie de $\mathcal{T}_C^{\delta t}$).

$$(\Omega_1 \subset \Omega_2) \Rightarrow (\mathcal{T}_C^{\delta t}(\Omega_1) \subset \mathcal{T}_C^{\delta t}(\Omega_2)) \quad (2.154)$$

On passe maintenant   la notion de consistance, d finie ici   l'aide des sur-flots et sous-flots.

D finition 2.5.12. Pour $\Omega(t) \subset \mathbb{R}^N$, $t \in [t_0, t_1]$, on dit que $\Omega(t)$ est un sur-flot s'il existe un ensemble $A \subset \mathbb{R}^N$ et $\delta > 0$ tel que :

$$\forall x \in A \times [t_0, t_1], \partial_t d(x, t) \geq |\nabla d| \operatorname{div} \left(\frac{\nabla d}{|\nabla d|} \right) (x, t) \quad (2.155)$$

On d finit de mani re similaire la notion de sous-flot en inversant l'in galit  (2.155).

On peut alors d finir une notion de consistance d'un sch ma   partir de la comparaison avec les sur-flots et les sous-flots.

D finition 2.5.13. Le sch ma d termin  par (2.152) est dit consistant si :

- Pour tout sur-flot $\Omega(t)$ sur $[t_0; t_1]$, il existe h_0 tel que :

$$\forall t \in [t_0; t_1 - h], \Omega(t + h) \subset \mathcal{T}_C^{\delta t}(\Omega(t)) \quad (2.156)$$

- Pour tout sous-flot $\Omega(t)$ sur $[t_0; t_1]$, il existe h_0 tel que :

$$\forall t \in [t_0; t_1 - h], \Omega(t + h) \supset \mathcal{T}_C^{\delta t}(\Omega(t)) \quad (2.157)$$

Cette définition signifie qu'un sur-flot $\Omega(t)$ se déplace plus rapidement que l'évolution discrétisée pour un pas de temps suffisamment petit.

Proposition 2.5.14 (Proposition 4.1 [CN06]). *Le schéma $\mathcal{T}_C^{\delta t}$ déterminé par (2.152) est consistant au sens de la définition (2.5.13).*

On cite maintenant le résultat de convergence de [CN06], issu d'éléments de la *théorie des barrières* (pour laquelle des références sont proposées dans [CN06])

Théorème 2.5.15 (Proposition 2.3 [CN06]). *Si \mathcal{T}_h définit un schéma monotone et consistant au sens de la définition (2.5.13) et que $\Omega_0 \subset \mathbb{R}^d$ est tel que le mouvement par courbure moyenne est défini de manière unique, alors pour tout $t \geq 0$ où $\Omega(t)$ n'est pas vide, on a la convergence $\Gamma_h(t) \rightarrow \Gamma(t)$ au sens de la distance de Hausdorff.*

Le principe de comparaison de cette méthode est mis en défaut lorsque l'on souhaite étendre ce résultat aux différentes extensions du mouvement par courbure moyenne telles que le mouvement par courbure moyenne avec conservation de volume ou à des mouvements géométriques d'ordre supérieurs.

2.5.6 Convergence du schéma convolution-redistanciation

Il est sans doute possible de transposer les résultats ci-dessus au cas des algorithmes de convolution-redistanciation mais nous avons choisi d'orienter notre travail dans une direction plus numérique dans un premier temps. Nous mentionnons quelques remarques sur l'étude potentielle de la convergence de l'algorithme 2.4.3 : 7. Nous envisageons deux approches pour attaquer ce problème.

La première consisterait à comparer les convolutions $G_{\delta t} * \chi$ et $G_{\delta t} * d$. En effet, on a montré dans les parties 2.2 et 2.4.1 que ces dernières ont le même comportement au premier ordre. Il semble donc raisonnable de chercher à contrôler $\|\mathcal{H}_\varepsilon(G_{\delta t} * d) - G_{\delta t} * \chi\|$ au voisinage de l'interface. On montrerait alors que les déplacements de l'interface calculés par les deux méthodes sont proches l'un de l'autre. On pourrait ainsi espérer réutiliser les résultats sur l'algorithme de convolution-seuillage 2.1 : 2 et les appliquer à l'algorithme de convolution-redistanciation 2.4.3 : 7.

Une deuxième approche pourrait être d'adapter les raisonnements des parties précédentes à notre contexte. Par exemple, on rappelle que dans 2.5.4, les

auteurs considèrent l'énergie suivante :

$$\chi^n \text{ réalise } \min_{\chi \in \{0,1\}} E_h(\chi) - E_h(\chi - \chi^{n-1}) \quad (2.158)$$

où :

$$E_h(\chi) = \frac{1}{\sqrt{h}} \int (1 - \chi)G_h * \chi + \chi G_h * (1 - \chi)dx \quad (2.159)$$

Si l'on remplace χ par d et $1 - \chi$ par $-d$, on obtient alors l'énergie suivante :

$$\tilde{E}_h(d) = \frac{1}{\sqrt{h}} \int -2dG_h * ddx \quad (2.160)$$

Cependant, on peut montrer que ce n'est pas la fonction d^n qui réalise le minimum du problème :

$$\min_{d \text{ distance}} E_h(d) - E_h(d - d^{n-1}) \quad (2.161)$$

L'énergie (2.159) est une approximation de l'aire de surface et il a été montré sa Γ -convergence vers celle-ci. Cela semble moins évident pour (2.160), notamment puisqu'elle ne converge pas vers 0 loin de l'interface.

Cela suggère donc qu'il ne suffit pas de transposer l'énergie au contexte *levelset* mais qu'il s'agit d'en trouver une autre qui conviendrait à la situation.

Ces considérations ne sont que des pistes de réflexion et il serait nécessaire de s'intéresser en détail au sujet pour avoir un avis plus éclairé sur l'approche à suivre afin d'établir une preuve de convergence théorique de l'algorithme de convolution-redistanciation 2.4.3 : 7 similaire aux résultats que l'on a présentés pour l'algorithme de convolution-seuillage.

CHAPITRE 3

FLOT DE WILLMORE SOUS CONTRAINTE DE VOLUME ET D'AIRE, APPLICATION AUX SCHÉMAS SEMI-IMPLICITES

Sommaire

3.1	Méthodes de conservation de volume et d'aire	132
3.1.1	Méthodes de conservation de volume et aire en formulation <i>levelset</i> et pour les méthodes de convolution-seuillage	133
3.1.2	Conservation simultanée du volume et de l'aire	135
3.1.3	Comparaison avec une méthode par multiplicateurs de Lagrange	138
3.1.4	Mouvement par flot de Willmore à volume et aire constants	141
3.2	Obtention numérique de la forme d'équilibre d'un globule rouge	145
3.2.1	Cas de la dimension 2	145
3.2.2	Cas de la dimension 3	150
3.3	Méthode de prédiction dans le cas des globules rouges	155
3.3.1	Minimisation pour une équation de transport discrétisée par le schéma d'Euler explicite	156
3.3.2	Minimisation pour une équation de transport discrétisée par un schéma d'Euler implicite	158
3.3.3	Perspectives de la méthode	159

Dans tout le chapitre, on désignera par volume d'un globule décrit par la fonction *levelset* ϕ , la mesure de Lebesgue de l'ensemble $\{\phi < 0\}$ que l'on notera $V(\phi)$. On appellera aire du globule la mesure de l'hypersurface $\{\phi = 0\}$ que l'on notera $A(\phi)$.

Dans ce chapitre, on s'intéresse à la mise en place du schéma de prédiction dans le cas d'un globule rouge dans l'esprit de la méthode décrite dans la section 1.6. On rappelle que l'idée est de prévoir le déplacement et la forme du globule dans un futur proche pour avoir une meilleure expression des forces dans l'équation fluide afin de relaxer une condition de couplage CFL d'ordre $h^{\frac{3}{2}}$. Il est nécessaire pour cela de savoir déplacer une interface selon le flot de Willmore en imposant un volume et une aire constants. Nous avons vu au Chapitre 2 comment déplacer une interface selon le flot de Willmore en dimension 2 et 3.

On commence alors ce chapitre par la mise en place d'une stratégie pour imposer les contraintes de conservation de volume et d'aire. Après un passage en revue des différentes méthodes existantes dans le contexte *levelset* et de convolution-seuillage, on propose une méthode originale pour obtenir une fonction *levelset* ayant un volume et une aire cibles. L'idée est de prendre comme nouvelle interface l'ensemble $\{\phi = \lambda + \mu H\}$, avec λ et μ deux constantes bien choisies.

En combinant ces deux aspects, on obtient alors un algorithme capable de déplacer l'interface selon le flot de Willmore tout en assurant un volume et une aire constante à la fin de chaque itération. On n'effectue pas une descente de gradient sous contraintes à proprement parler mais plutôt une descente non contrainte suivie d'une correction. Cependant, comme les constantes λ et μ sont une approximation au premier ordre des multiplicateurs de Lagrange de la méthode de descente sous contraintes, on peut espérer ne pas trop augmenter l'énergie lors de cette correction.

On teste le bon comportement de notre algorithme dans le cas d'un globule seul. Il est soumis uniquement à l'énergie de courbure de sa membrane et son déplacement est alors celui simulé par notre algorithme. On vérifie que l'on converge bien vers une forme d'équilibre bi-concave et ceci pour des taux de gonflement variés.

On propose enfin un schéma semi-implicite dans le cas d'un globule rouge qui ouvre la voie vers l'utilisation de cette méthode comme prédicteur de l'algorithme de couplage complet.

3.1 Méthodes de conservation de volume et d'aire

On commence cette partie par mentionner les différentes techniques utilisées en méthode *levelset* pour conserver le volume et l'aire. On propose ensuite une façon d'obtenir une interface ayant un volume et une aire choisie. Puis, on dresse un parallèle de notre approche avec celle par multiplicateurs de Lagrange.

3.1.1 Méthodes de conservation de volume et aire en formulation *levelset* et pour les méthodes de convolution-seuillage

Dans le modèle *levelset* (1.81), la conservation du volume est assurée par la condition $\operatorname{div} u = 0$ et l'équation de transport. En effet :

$$\begin{aligned}
 \frac{d}{dt}V(\phi) &= \int_{\Omega} \partial_t \left(\mathcal{H}_{\varepsilon} \left(-\frac{\phi}{\varepsilon} \right) \right) dx \\
 &= - \int_{\Omega} \partial_t \phi \frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) dx \\
 &= \int_{\Omega} u \cdot \nabla \phi \frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) dx \\
 &= - \int_{\Omega} \operatorname{div} u \mathcal{H}_{\varepsilon} \left(\frac{\phi}{\varepsilon} \right) dx \\
 &= 0
 \end{aligned} \tag{3.1}$$

Dans la pratique des calculs numériques, le fait de régulariser l'interface engendre des pertes de volume. Ces pertes s'accumulant au cours du temps, cela peut avoir des conséquences critiques.

Une méthode, proposée dans [SF99, SFSO98], propose d'ajouter un terme à l'équation d'Hamilton-Jacobi (1.63) lors de la réinitialisation :

$$\partial_t \phi + \operatorname{sign}(\phi_0) (|\nabla \phi| - 1) = \lambda f(\phi) \tag{3.2}$$

La fonction $f(\phi)$ sert à localiser l'interface et correspond à la fonction Dirac δ_{ε} . Le paramètre λ est déterminé pour conserver le volume $\int_{\Omega} \mathcal{H}_{\varepsilon}(-\phi) dx$. Numériquement, cette conservation est assurée sur chaque cellule Ω_{ij} du maillage.

On rappelle qu'une redistanciation par la méthode de *fast-marching* fournit de meilleurs résultats, notamment lors de simulations sur FEEL++ [MCIP18]. On va alors préférer la méthode proposée par Smolianski dans [Smo05] qui a le mérite d'être utilisable quelque soit le choix de la méthode de réinitialisation.

L'idée est de choisir la ligne de niveau $\{\phi = \lambda\}$ qui vérifie la bonne condition de volume. Si ϕ est une fonction distance signée, la forme décrite par $\{\phi = \lambda\}$ est identique à celle de $\{\phi = 0\}$, cf Fig.3.1. Si ce n'est pas le cas, on estime que l'on déforme peu l'objet car la valeur de λ est petite. Le choix de λ est déterminé par le développement suivant :

$$\begin{aligned}
 V(\phi + \lambda) &= \int_{\Omega} \mathcal{H}_{\varepsilon} \left(-\frac{\phi + \lambda}{\varepsilon} \right) dx \\
 &= \int_{\Omega} \mathcal{H}_{\varepsilon} \left(-\frac{\phi}{\varepsilon} \right) dx - \int_{\Omega} \frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) \lambda dx + \mathcal{O}(\lambda^2) \\
 &= V(\phi) - \lambda A(\phi) + \mathcal{O}(\lambda^2)
 \end{aligned} \tag{3.3}$$

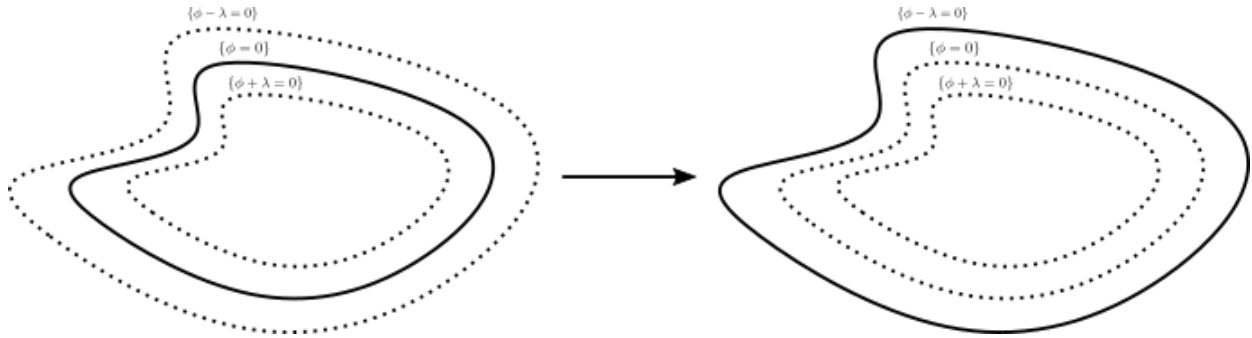


FIGURE 3.1 – Choix de la ligne de niveau $\{\phi - \lambda = 0\}$ qui a le volume choisi, avec ici $\lambda > 0$.

où $V(\phi), A(\phi)$ désignent respectivement le volume et l'aire délimités par ϕ .

Ainsi, si l'on considère une fonction levelset ϕ de volume $V(\phi)$ et d'aire $A(\phi)$, pour retrouver un volume cible V_{cible} , on choisit :

$$\lambda = \frac{V_{cible} - V(\phi)}{A(\phi)} \quad (3.4)$$

Cette méthode ne nécessite que de calculer le volume et l'aire de la membrane. On peut l'utiliser pour (1.81) après l'équation de transport afin de forcer *a posteriori* la conservation du volume.

Il convient de noter que cette méthode est à rapprocher de l'algorithme de convolution-seuillage 2.3.2 : 5 par [RW03] et de son adaptation en convolution-redistanciation de [KEF11] 2.4.7 : 9, 10 . En effet, l'idée de ces algorithmes est de modifier la valeur dans l'étape de seuillage ou redistanciation de manière à conserver le volume. Cela revient à choisir la ligne de niveau adéquate par la méthode de Smolianski.

En ce qui concerne la conservation de l'aire, certains auteurs l'imposent en advectant ϕ par une vitesse u à divergence surfacique nulle. Cela a été fait dans [DGC⁺13, LSM14] par une méthode de multiplicateurs de Lagrange pour imposer la contrainte. On assure une bonne conservation de l'aire mais cela entraîne un mauvais conditionnement du système et augmente fortement le coût algorithmique.

On rappelle que la conservation de l'aire est assurée par pénalisation dans notre modèle (1.81) avec l'introduction de l'énergie élastique E_e . En imposant une grande inextensibilité dans cette énergie, la valeur de $|\nabla\phi|$ est maintenue proche de 1. Cependant, cela contraint le pas de temps pour une discrétisation explicite du problème, [BKZ92, GV07, GV08]

$$\Delta t \lesssim \frac{\Delta x^{\frac{3}{2}}}{\sqrt{\lambda v_e}} \quad (3.5)$$

3.1.2 Conservation simultanée du volume et de l'aire

On propose maintenant une méthode pour conserver simultanément le volume et l'aire. Cette méthode est dans l'esprit de la correction de Smolianski et des changements de valeur seuil de l'algorithme 2.3.2 : 5.

Choisir une nouvelle ligne de niveau pour ϕ permet d'obtenir au choix un volume ou une aire cible, mais il est impossible d'obtenir les deux simultanément. On propose alors d'ajouter à ϕ une fonction non constante c de telle manière que $\{\phi + c = 0\}$ soit de volume V_{cible} et d'aire A_{cible} .

On peut réécrire le développement (3.3) pour une fonction c non constante :

$$V(\phi + c) = V(\phi) + \int_{\Omega} \frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) c dx + \mathcal{O}(c^2) \quad (3.6)$$

On effectue un développement similaire pour l'aire de la surface :

$$\begin{aligned} A(\phi + c) &= \int_{\Omega} \frac{1}{\varepsilon} \zeta \left(\frac{\phi + c}{\varepsilon} \right) |\nabla(\phi + c)| dx + \mathcal{O}(c^2) \\ &= \int_{\Omega} \frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) |\nabla(\phi)| dx + \int_{\Omega} \frac{1}{\varepsilon^2} \zeta' \left(\frac{\phi}{\varepsilon} \right) |\nabla\phi| c dx + \int_{\Omega} \frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) \frac{\nabla\phi \cdot \nabla c}{|\nabla\phi|} dx + \mathcal{O}(c^2) \end{aligned} \quad (3.7)$$

Or :

$$\begin{aligned} \int_{\Omega} \frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) \frac{\nabla\phi \cdot \nabla c}{|\nabla\phi|} dx &= - \int_{\Omega} \operatorname{div} \left(\frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) \frac{\nabla\phi}{|\nabla\phi|} \right) c dx \\ &= - \int_{\Omega} \frac{1}{\varepsilon^2} \zeta' \left(\frac{\phi}{\varepsilon} \right) \frac{\nabla\phi \cdot \nabla\phi}{|\nabla\phi|} c dx - \int_{\Omega} \frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) \operatorname{div} \left(\frac{\nabla\phi}{|\nabla\phi|} \right) c dx \\ &= - \int_{\Omega} \frac{1}{\varepsilon^2} \zeta' \left(\frac{\phi}{\varepsilon} \right) |\nabla\phi| c dx - \int_{\Omega} H \frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) |\nabla\phi| c dx \end{aligned} \quad (3.8)$$

où H est la courbure moyenne.

Finalement, on obtient :

$$A(\phi + c) = A(\phi) - \int_{\Omega} H(\phi) c \frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) |\nabla\phi| dx + \mathcal{O}(c^2) \quad (3.9)$$

Notre premier choix pour c a été de la forme $\lambda + \mu H$, avec (λ, μ) vérifiant le système suivant :

$$\begin{cases} \lambda A(\phi) + \mu \int_{\Omega} H \frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) |\nabla\phi| dx = V_{cible} - V(\phi) \\ \lambda \int_{\Omega} H \frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) |\nabla\phi| dx + \mu \int_{\Omega} H^2 \frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) |\nabla\phi| dx = A_{cible} - A(\phi) \end{cases} \quad (3.10)$$

La solution (λ, μ) du système est :

$$\begin{cases} \lambda = \frac{\overline{H}^2 (V_{cible} - V(\phi)) - \overline{H} (A_{cible} - A(\phi))}{A(\phi)^2 (\overline{H}^2 - \overline{H}^2)} \\ \mu = \frac{A(\phi) (A_{cible} - A(\phi)) - \overline{H} (V_{cible} - V(\phi))}{A(\phi)^2 (\overline{H}^2 - \overline{H}^2)} \end{cases} \quad (3.11)$$

Ce choix s'explique de la façon suivante : il semble en effet naturel d'exprimer c en fonction de deux paramètres et on dispose de méthodes efficaces pour obtenir la courbure. Cette méthode est décrite par l'algorithme 3.1.2 : 13.

Algorithme 13 1ère méthode de correction : Étant donnés une fonction distance ϕ , un volume V_{cible} , une aire cible A_{cible} , trouver la fonction $c = \lambda + \mu H$ telle que la ligne de niveau 0 de $\phi + c$ détermine un ensemble de volume V_{cible} et d'aire L_{cible} .

1) Calculer les intégrales :

$$\begin{cases} A(\phi) = \int_{\Omega} \frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) |\nabla \phi| dx \\ V(\phi) = \int_{\Omega} H \left(-\frac{\phi}{\varepsilon} \right) |\nabla \phi| dx \\ \overline{H} = \frac{1}{A(\phi)} \int_{\Omega} H \frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) |\nabla \phi| dx \\ \overline{H}^2 = \frac{1}{A(\phi)} \int_{\Omega} H^2 \frac{1}{\varepsilon} \zeta \left(\frac{\phi}{\varepsilon} \right) |\nabla \phi| dx \end{cases} \quad (3.12)$$

2) Trouver (λ, μ) satisfaisant le système, si $\overline{H}^2 \neq \overline{H}^2$:

$$\begin{cases} \lambda A(\phi) + \mu A(\phi) \overline{H} = V_{cible} - V(\phi) \\ \lambda A(\phi) \overline{H} + \mu A(\phi) \overline{H}^2 = A_{cible} - A(\phi) \end{cases} \quad (3.13)$$

3) Prendre la ligne de niveau 0 de $\phi + (\lambda + \mu H)$ comme nouvelle levelset.

La seconde méthode, très proche de la première, consiste à découpler l'obtention du volume cible et de l'aire cible. Cela se fait en constatant qu'une correction de $\mu (H - \overline{H})$ peut modifier l'aire tout en conservant le volume. On considère alors la correction $c = \lambda + \mu (H - \overline{H})$ faite en deux temps :

1) Trouver λ tel que $V(\phi + \lambda) = V_{cible}$:

$$\lambda = \frac{V_{cible} - V(\phi)}{A(\phi)} \quad (3.14)$$

2) Trouver μ tel que $A\left((\phi + \lambda) + \mu(H - \bar{H})\right) = A_{cible}$:

$$\mu = \frac{A_{cible} - A(\phi + \lambda)}{\int_{\{\phi+\lambda=0\}} H(\phi + \lambda) \left(H(\phi + \lambda) - \bar{H}(\phi + \lambda)\right) dx} \quad (3.15)$$

Ces méthodes sont efficaces en termes de coût de calculs car elles impliquent uniquement d'obtenir la courbure moyenne et de calculer sa moyenne et sa variance sur la surface. Cependant, il convient de noter que l'on ne conserve pas la forme de l'objet. De par leur expression, λ et μ sont linéaires en la différence de volume/aire à corriger et sont donc petits pour de petites corrections. On estime ainsi que l'on ne déforme pas trop la structure lors de cette étape de correction.

Algorithme 14 2ème méthode de correction: Étant donné une fonction distance ϕ , un volume V_{cible} , une aire cible A_{cible} , trouver la fonction $c = \lambda + \mu(H - \bar{H})$ telle que la ligne de niveau 0 de $\phi + c$ détermine un ensemble de volume V_{cible} et d'aire L_{cible} .

1) Correction du volume :

$$\lambda = \frac{V_{cible} - V(\phi)}{A(\phi)} \quad (3.16)$$

2) Correction de l'aire en conservant le volume, si $\bar{H}^2 \neq \overline{H^2}$:

$$\mu = \frac{A_{cible} - A(\phi + \lambda)}{A(\phi + \lambda) \left(\overline{H^2} - \bar{H}^2\right)} \quad (3.17)$$

3) Prendre la ligne de niveau 0 de $\phi + (\lambda + \mu H)$ comme nouvelle interface dont le volume et l'aire seront V_{cible} et A_{cible} .

Remarque 3.1.1. Ces deux méthodes ne marchent pas dans le cas d'un cercle ou d'une sphère comme $\overline{H^2} = \bar{H}^2$. Dans ce cas, il est impossible de modifier l'une des

deux quantités sans modifier l'autre car elles sont toutes deux déterminées par le rayon.

Dans la suite, on préférera la deuxième méthode qui a le mérite de découpler en deux étapes la correction. Les deux méthodes fournissent néanmoins des résultats similaires en termes de résultats.

3.1.3 Comparaison avec une méthode par multiplicateurs de Lagrange

Le choix de la fonction de correction dans l'algorithme 13 peut sembler arbitraire mais on va montrer qu'il est en adéquation avec l'approche des multiplicateurs de Lagrange. En effet, considérons le lagrangien du problème de minimisation sous contraintes :

$$\left\{ \begin{array}{l} \text{Trouver } (\phi, \lambda, \mu) \text{ minimisant :} \\ \mathcal{L}(\phi, \lambda, \mu) = \frac{1}{2} \int_{\{\phi=0\}} H(\phi)^2 d\sigma + \lambda \left(\int_{\{\phi<0\}} dx - V_0 \right) + \mu \left(\int_{\{\phi=0\}} d\sigma - A_0 \right) \end{array} \right. \quad (3.18)$$

La dérivation du lagrangien par rapport à λ, μ assure la validité des contraintes aux points critiques :

$$\left\{ \begin{array}{l} d_\lambda \mathcal{L}(\phi, \lambda, \mu) = \int_{\{\phi<0\}} dx - V_0 = 0 \\ d_\mu \mathcal{L}(\phi, \lambda, \mu) = \int_{\{\phi=0\}} d\sigma - A_0 = 0 \end{array} \right. \quad (3.19)$$

On exprime la dérivation du volume régularisé par rapport à ϕ dans la direction ψ :

$$\begin{aligned} d_\phi V_\varepsilon(\psi) &= \int_{\Omega} -\mathcal{H}'_\varepsilon(-\phi)\psi dx \\ &= \int_{\Omega} -\frac{1}{\varepsilon} \zeta\left(\frac{\phi}{\varepsilon}\right) |\nabla\phi| \psi dx \end{aligned} \quad (3.20)$$

En faisant tendre ε vers 0, on obtient :

$$d_\phi V(\psi) = \int_{\{\phi=0\}} \psi d\sigma \quad (3.21)$$

On effectue de même pour dériver l'aire régularisée selon ϕ dans la direction ψ :

$$\begin{aligned}
d_\phi A_\varepsilon(\psi) &= \int_\Omega \frac{1}{\varepsilon^2} \zeta\left(\frac{\phi}{\varepsilon}\right) |\nabla\phi| \psi dx + \int_\Omega \frac{1}{\varepsilon} \zeta\left(\frac{\phi}{\varepsilon}\right) \frac{\nabla\phi \cdot \nabla\psi}{|\nabla\phi|} dx \\
&= \int_\Omega \frac{1}{\varepsilon^2} \zeta\left(\frac{\phi}{\varepsilon}\right) |\nabla\phi| \psi dx - \int_\Omega \operatorname{div} \left(\frac{1}{\varepsilon} \zeta\left(\frac{\phi}{\varepsilon}\right) \frac{\nabla\phi}{|\nabla\phi|} \right) \psi dx \\
&= \int_\Omega \frac{1}{\varepsilon^2} \zeta\left(\frac{\phi}{\varepsilon}\right) |\nabla\phi| \psi dx - \int_\Omega \frac{1}{\varepsilon^2} \zeta\left(\frac{\phi}{\varepsilon}\right) \frac{\nabla\phi \cdot \nabla\phi}{|\nabla\phi|} \psi dx - \int_\Omega \operatorname{div} \frac{\nabla\phi}{|\nabla\phi|} \frac{1}{\varepsilon} \zeta\left(\frac{\phi}{\varepsilon}\right) |\nabla\phi| \psi dx \\
&= \int_\Omega H \psi \frac{1}{\varepsilon} \zeta\left(\frac{\phi}{\varepsilon}\right) |\nabla\phi| dx
\end{aligned} \tag{3.22}$$

En faisant tendre ε vers 0, on obtient :

$$d_\phi A(\psi) = \int_{\{\phi=0\}} H \psi d\sigma \tag{3.23}$$

La dérivation par rapport ϕ dans la direction ψ donne alors (en dimension 2 par exemple) :

$$d_\phi \mathcal{L}(\psi) = \int_{\{\phi=0\}} \left(\Delta_\Gamma H + \frac{H^3}{2} \right) \psi d\sigma + \lambda \int_{\{\phi=0\}} \psi d\sigma + \mu \int_{\{\phi=0\}} H \psi d\sigma \tag{3.24}$$

On observe bien dans l'équation précédente que les multiplicateurs de Lagrange suggèrent une correction de type $\lambda + \mu H$ (que l'on a reformulée en $\lambda + \mu(H - \bar{H})$ qui est équivalente). On montre maintenant que nos choix pour λ et μ peuvent être mis en corrélation au premier ordre avec les λ et μ de la présente méthode.

Pour (ϕ, λ, μ) point selle de \mathcal{L} , en évaluant (3.24) en $\psi = 1$, on trouve :

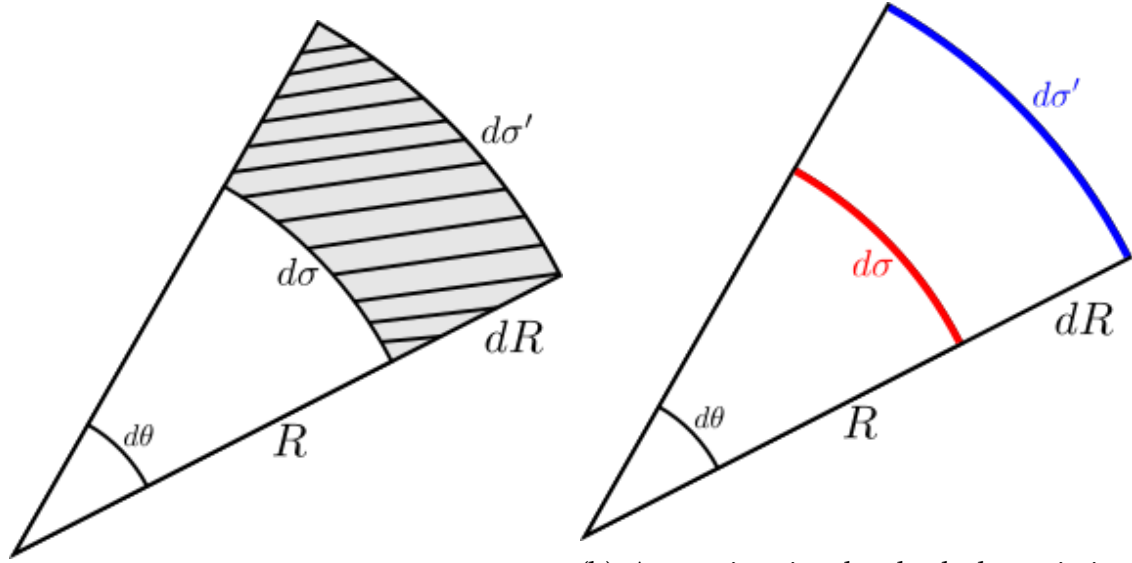
$$\lambda A(\phi) + \mu A(\phi) \bar{H} = - \int_{\{\phi=0\}} \Delta_\Gamma H + \frac{H^3}{2} d\sigma \tag{3.25}$$

Pour $\psi = H - \bar{H}$ dans (3.24), on a :

$$\mu \int_{\{\phi=0\}} H (H - \bar{H}) d\sigma = - \int_{\{\phi=0\}} (H - \bar{H}) \left(\Delta_\Gamma H + \frac{H^3}{2} \right) d\sigma \tag{3.26}$$

On s'intéresse maintenant à donner une approximation au premier ordre des intégrales $\int_{\{\phi=0\}} \left(\Delta_\Gamma H + \frac{H^3}{2} \right) d\sigma$ et $\int_{\{\phi=0\}} (H - \bar{H}) \left(\Delta_\Gamma H + \frac{H^3}{2} \right) d\sigma$. On exprime la variation infinitésimale de volume en se plaçant dans le cercle de courbure associé à un point de la courbe, cf Fig.3.2a :

$$\begin{aligned}
dV &= (R + dR)^2 \frac{d\theta}{2} - R^2 \frac{d\theta}{2} \\
&= R(dR)(d\theta) + (dR)^2 \frac{d\theta}{2}
\end{aligned} \tag{3.27}$$



(a) Approximation locale de la variation de volume créée par le flot de Willmore en dimension 2. L'élément de variation de volume est représenté par la zone hachurée

(b) Approximation locale de la variation d'aire créée par le flot de Willmore en dimension 2. L'élément de variation d'aire est la différence de la longueur de la courbe bleue et de la rouge.

On a d'une part, $d\sigma = R d\theta$. D'autre part, le terme $dR^2 \frac{d\theta}{2}$ est un terme d'ordre supérieur donc négligeable pour notre étude, on a donc :

$$dV \sim dR d\sigma \quad (3.28)$$

Le déplacement dR pendant un temps infinitésimal selon le flot de Willmore est normal d'amplitude :

$$dR = \Delta_{\Gamma} H + \frac{H^3}{2} \quad (3.29)$$

Ainsi, en intégrant sur l'ensemble de la surface, on obtient :

$$V(W) - V(\phi) = \int_{\{\phi=0\}} \Delta_{\Gamma} H + \frac{H^3}{2} d\sigma \quad (3.30)$$

Ainsi, en réutilisant (3.25) on a l'approximation au premier ordre suivante :

$$\lambda A(\phi) + \mu \bar{H} \sim \frac{V(\phi) - V(W)}{A(\phi)} \quad (3.31)$$

Selon le même principe, on peut obtenir une approximation similaire sur l'aire de la surface :

$$\begin{aligned} dA &= d\sigma' - d\sigma = (R + dR)d\theta - R d\theta \\ &= \frac{dR}{R} d\sigma \\ &= H dR d\sigma \end{aligned} \quad (3.32)$$

Sur un temps infiniment court, dR peut être approché par (3.29) et la variation d'aire est approchée par :

$$A(W) - A(\phi) = \int_{\{\phi=0\}} H \left(\Delta_{\Gamma} H + \frac{H^3}{2} \right) d\sigma \quad (3.33)$$

En reportant cette équation dans (3.26), on obtient :

$$\begin{aligned} \mu \int_{\{\phi=0\}} H(H - \bar{H}) d\sigma &\sim - \int_{\{\phi=0\}} H \left(\Delta_{\Gamma} H + \frac{H^3}{2} \right) d\sigma + \bar{H} \int_{\{\phi=0\}} \Delta_{\Gamma} H + \frac{H^3}{2} d\sigma \\ &\sim (A(\phi) - A(W)) + \bar{H} (V(W) - V(\phi)) \\ &\sim (A(\phi) - A(W)) - \lambda A(\phi) \bar{H} - \mu \bar{H} \int_{\{\psi=0\}} H d\sigma \end{aligned} \quad (3.34)$$

Ainsi :

$$\lambda \int_{\{\phi=0\}} H d\sigma + \mu \int_{\{\phi=0\}} H^2 d\sigma = (A(\phi) - A(W)) \quad (3.35)$$

Finalement, le couple (λ, μ) optimal pour cette méthode par multiplicateurs de Lagrange est approché au premier ordre par le système suivant :

$$\begin{cases} \lambda A(\phi) + \mu \int_{\{\phi=0\}} H d\sigma \sim V(\phi) - V(W) \\ \lambda \int_{\{\phi=0\}} H d\sigma + \mu \int_{\{\phi=0\}} H^2 d\sigma = (A(\phi) - A(W)) \end{cases} \quad (3.36)$$

On retrouve alors le système (3.10) dans le sens où $(V(W), A(W))$ correspondent au volume et à l'aire au moment où l'on souhaite appliquer la correction et $(V(\phi), A(\phi))$ sont le volume et l'aire que l'on souhaite récupérer par la correction, c'est-à-dire le volume cible et l'aire cible de la méthode de correction. Cela permet de justifier le choix de notre correction. De plus, il est raisonnable de penser que le déplacement de l'interface suit la direction du gradient du flot avec contraintes de conservation. On vérifiera dans la partie numérique que cette étape de recouvrement de contrainte favorise le déplacement vers la forme d'équilibre.

3.1.4 Mouvement par flot de Willmore à volume et aire constants

Les algorithmes 3.1.4 : 15 et 3.1.4 : 16 pour simuler le déplacement par flot de Willmore avec conservation de volume et d'aire sont obtenus par la combinaison des algorithmes de diffusion par flot de Willmore 11 et 12 et de l'algorithme de correction 3.1.2 : 14.

Ces algorithmes appellent à plusieurs remarques. On a ajouté une renormalisation de la fonction obtenue après le déplacement selon le flot de Willmore.

Comme vu dans la section 1.4.6, cette opération permet à moindre coût de ramener ψ^k proche d'une fonction distance. Bien que l'on ne requiert pas d'avoir une fonction distance pour l'étape de correction, on observe de meilleurs résultats avec cette étape supplémentaire. On n'applique pas de redistanciation car le coût algorithmique serait alors fortement augmenté, notamment en 3D.

Pour des précisions sur la méthode utilisée pour calculer la courbure de l'étape 6), se référer à la partie numérique 3.2.

Il ne s'agit techniquement pas d'une descente de gradient sous contrainte car on n'impose pas les contraintes lors de la descente. On y recolle *a posteriori* en espérant ne pas trop augmenter l'énergie de Willmore. Il conviendra de vérifier que c'est bien le cas lors des test numériques.

On note $\mathcal{P} : \phi \mapsto \tilde{\phi}$ l'opérateur qui, à une fonction *levelset* ϕ , associe sa prédiction $\tilde{\phi}$ obtenue en effectuant une itération de l'algorithme 3.1.4 : 15 en dimension 2 (ou d'une itération de l'algorithme 3.1.4 : 16 en dimension 3).

Algorithme 15 : Flot de Willmore 2D avec conservation de volume et d'aire : Étant donné la fonction distance signée à l'ensemble Ω_0 d'interface Γ_0 de volume V_0 et d'aire A_0 , un pas de temps δt et un nombre d'itérations nb_{iter} , générer la suite d'ensembles $(\Omega_k)_k$ ayant un volume V_0 et une aire A_0 décrits par leurs fonctions distances signées d^k .

Pour k de 0 à $nb_{iter} - 1$:

1) Étape de convolution : Calculer les convolutions : $G_{2\sqrt{\delta t}} * d^k$ et $G_{\sqrt{\delta t}} * d^k$.

2) Former les termes :

$$A^k = 2G_{\sqrt{\delta t}} * d^k - G_{2\sqrt{\delta t}} * d^k$$

$$B^k = \left(G_{\sqrt{\delta t}} * d^k - d^k\right)^3 \text{ ou } \left[\frac{1}{2} \left(4G_{\sqrt{\delta t}} - G_{2\sqrt{\delta t}}\right) * d^k - \frac{3}{2}d^k\right]^3$$

$$\psi^k = A^k + \frac{B^k}{2\sqrt{\delta t}}$$

3) Renormalisation de la fonction ψ^k :

$$\phi^k = \frac{\psi^k}{|\nabla \psi^k|}. \quad (3.37)$$

4) Correction du volume :

$$\lambda = \frac{V_0 - V(\phi^k)}{A(\phi^k)} \quad (3.38)$$

5) Correction de l'aire en conservant le volume :

$$\mu = \frac{A_0 - A(\phi^k + \lambda)}{\int_{\{\phi^k=0\}} H(\phi^k + \lambda) \left(H(\phi^k + \lambda) - \bar{H}(\phi^k + \lambda)\right) |\nabla \phi^k| d\sigma} \quad (3.39)$$

6) Étape de redistanciation :

$$d^{k+1} = \text{redist} \left(\phi^k + \lambda + \mu \left(H(\phi^k + \lambda) - \bar{H}(\phi^k + \lambda) \right) \right) \quad (3.40)$$

Algorithme 16 : Flot de Willmore 3D avec conservation de volume et d'aire : Étant donné la fonction distance signée à l'ensemble Ω_0 d'interface Γ_0 de volume V_0 et d'aire A_0 , un pas de temps δt et un nombre d'itérations nb_{iter} , générer la suite d'ensembles $(\Omega_k)_k$ ayant un volume V_0 et une aire A_0 décrits par leurs fonctions distances signées d^k .

Pour k de 0 à $nb_{iter} - 1$:

- 1) Étape de convolution : $G_{\sqrt{2\delta t}} * d$, $G_{\frac{\sqrt{\delta t}}{2}} * d$ et $G_{\sqrt{2\delta t}} * d^2$, $G_{\frac{\sqrt{\delta t}}{2}} * d^2$.
- 2) Calculer la courbure moyenne :

$$H = \left(\frac{G_{\sqrt{\delta t}} * d^k - d^k}{2} \right)^3 \text{ ou } \left[\left(\sqrt{2}G_{\frac{\sqrt{\delta t}}{2}} - \frac{1}{2\sqrt{2}}G_{\sqrt{2\delta t}} \right) * d^k - \frac{3\sqrt{2}}{4}d^k \right]^3 \quad (3.41)$$

- 3) Former les termes :

$$\begin{aligned} A^k &= 2G_{\sqrt{\delta t}} * d - G_{2\sqrt{\delta t}} * d \\ B^k &= 2G_{\sqrt{\delta t}} * d^2 - G_{2\sqrt{\delta t}} * d^2 \\ \psi^k &= \frac{Hd^2}{2} + (1 - Hd)A^k + \frac{H}{2}B^k \end{aligned} \quad (3.42)$$

- 4) Renormalisation de la fonction ψ^k :

$$\phi^k = \frac{\psi^k}{|\nabla \psi^k|}. \quad (3.43)$$

- 5) Correction du volume :

$$\lambda = \frac{V_0 - V(\phi^k)}{A(\phi^k)} \quad (3.44)$$

- 6) Correction de l'aire en conservant le volume :

$$\mu = \frac{A_0 - A(\phi^k + \lambda)}{\int_{\{\phi^k=0\}} H(\phi^k + \lambda) \left(H(\phi^k + \lambda) - \bar{H}(\phi^k + \lambda) \right) |\nabla \phi^k| d\sigma} \quad (3.45)$$

- 7) Étape de redistanciation :

$$d^{k+1} = \text{redist} \left(\phi^k + \lambda + \mu \left(H(\phi^k + \lambda) - \bar{H}(\phi^k + \lambda) \right) \right) \quad (3.46)$$

3.2 Obtention numérique de la forme d'équilibre d'un globule rouge

Dans la section 1.2, nous avons présenté la physique des globules rouges. Les algorithmes 3.1.4 : 15 et 3.1.4 : 16 permettent de simuler le comportement d'un globule rouge. Nous allons tester l'efficacité de nos algorithmes. On se place dans le cas d'un globule seul laissé au repos dans le sens où il est uniquement soumis à l'énergie de courbure de sa membrane. Nous vérifions ensuite qu'il converge vers une des ses formes d'équilibres, la forme biconcave présentée Fig.1.3 en l'occurrence.

On part d'une ellipse en dimension 2 et d'un ellipsoïde de révolution en dimension 3, en excluant le cas des cercles ou sphères (qui sont des minimiseurs, pour lesquels les algorithmes de correction dégénèrent). Les paramètres de l'ellipse servent à choisir le volume réduit que l'on va tester, dont on rappelle l'expression ici :

$$\nu = \begin{cases} \frac{4\pi V}{A^2} = \frac{V}{\pi \left[\frac{A}{2\pi}\right]^2} & \text{en dimension 2} \\ \frac{6\sqrt{\pi}V}{A^{\frac{3}{2}}} & \text{en dimension 3} \end{cases} \quad (3.47)$$

Les points d'études des simulations sont :

- Diminution de l'énergie d'Helfrich à chaque itération.
- Validité de la conservation du volume et de l'aire.
- Convergence vers la forme d'équilibre annoncée.
- Stabilité de la méthode proche de l'état stationnaire.
- Temps de calculs de chaque élément de la méthode.
- Vérification des formes d'équilibres pour différents taux de gonflements.

On étudie l'ensemble de ces points en détail sur un exemple dont le volume réduit se situe autour de 0.5. Ensuite, on vérifie le dernier point en faisant varier le taux de gonflement et en comparant les résultats avec ceux de référence pour les valider.

3.2.1 Cas de la dimension 2

On s'intéresse ici au comportement de l'algorithme 3.1.4 : 15 dans le cas initial d'une ellipse de semi grand-axe $a = 0.418$ et de semi petit-axe $b = 0.096$ pour un taux de remplissage $\nu = 0.5$.

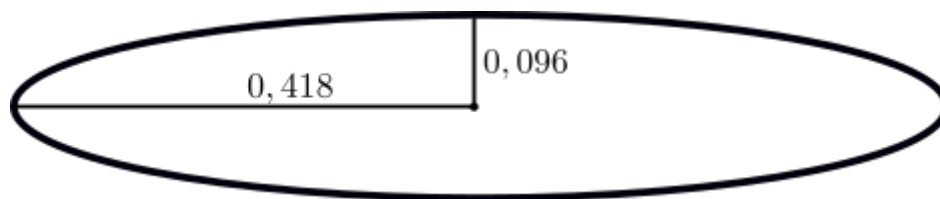


FIGURE 3.3 – Ellipse initiale pour notre simulation avec un taux de remplissage autour de 0.5.

Les paramètres que nous prenons sont les suivants. On considère un maillage sur un domaine rectangulaire de taille 2×1 et un maillage éléments finis de taille $h = 0,001$ pour 43000 éléments. Le paramètre ε qui détermine l'épaisseur de l'interface est fixé à $2h$. Le pas de temps utilisé est $\delta t = 10^{-6}$ de manière à ce que sa racine soit égale à 10^{-3} . Comme la courbure varie entre 2 et 10, on rentre dans le cadre de la règle $\delta \sim \frac{h}{H}$ que l'on a explicitée à la section 2.4.6. On choisit donc d'extraire la courbure nécessaire à l'étape 6) par la méthode de diffusion (2.83) d'ordre 2. Cette approche fournit de bons résultats et est consistante avec l'esprit d'un algorithme par convolution-redistanciation.

Puisque l'on diffuse des fonctions distances signées, les équations de la chaleur sont complétées par des conditions aux bords de type Neumann :

$$\nabla \phi \cdot n = 1 \text{ sur } \partial\Omega \quad (3.48)$$

De plus, la discrétisation temporelle des équations de la chaleur est faite par un schéma de type *bdf* d'ordre 2 en temps. On utilise le pré-conditionneur *LU* de FEEL++ que l'on peut ré-utiliser à chaque itération étant donné que l'on utilise un pas de temps constant, donc la matrice à inverser l'est également.

Convergence de l'algorithme

La convergence de l'algorithme vers la forme d'équilibre biconcave prédite peut être observée à la figure 3.4.

Il convient de constater qu'après obtention de la forme d'équilibre, l'algorithme reste stable dans le sens où la forme n'évolue presque plus et conserve la forme recherchée. La stabilité proche de l'équilibre est un point particulièrement important pour notre algorithme. Ce dernier ayant une étape de correction pour recoller aux contraintes, cela pourrait créer des instabilités proches de l'état d'équilibre.

Lors de la simulation présentée sur la figure 3.4, on observe une conservation de la symétrie radiale du globule. Généralement, la symétrie n'est pas parfaitement conservée à cause de l'accumulation des erreurs numériques. Dans l'exemple de la figure 3.4, la structure particulière du maillage, généré avec *GMSH*¹, favorise artificiellement sa bonne conservation. Comme on travaille

¹<http://gmsh.info/>

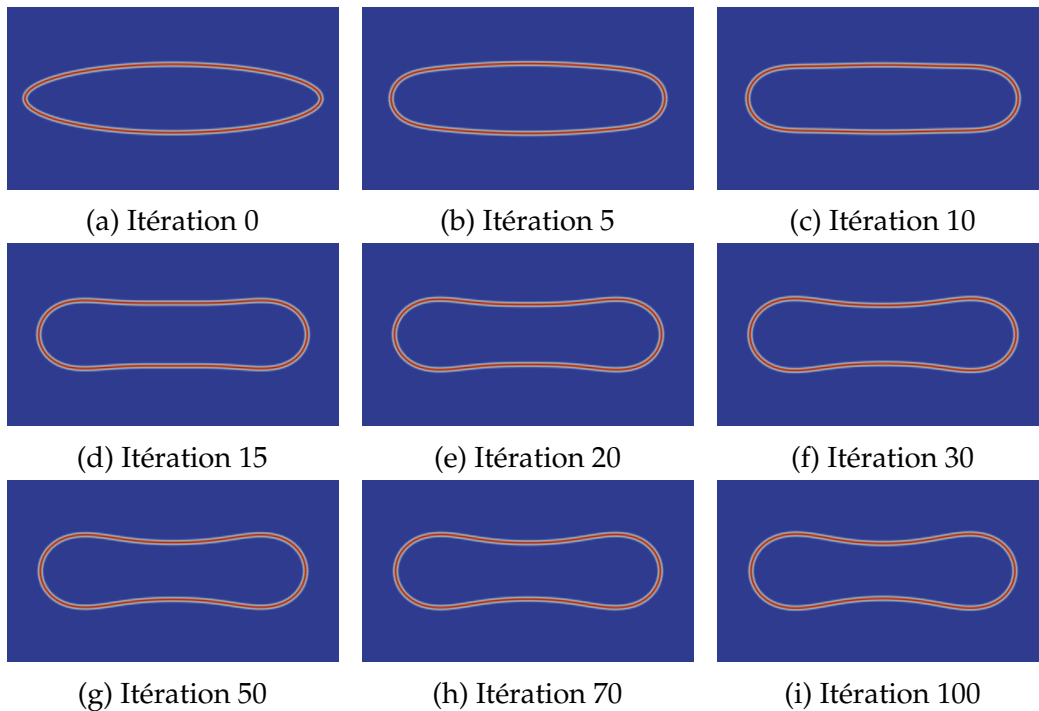


FIGURE 3.4 – Convergence de l'algorithme 3.1.4 : 15 vers la forme d'équilibre

sur un domaine rectangulaire et que l'on n'impose pas de condition particulière lors de la création du maillage, ce dernier aura une structure symétrique par défaut.

Pour une simulation sur un maillage non structuré, voir figure 3.5, dont on a raffiné la partie proche de l'interface, on ne conserve plus parfaitement la symétrie et on observe également un léger déplacement du globule.

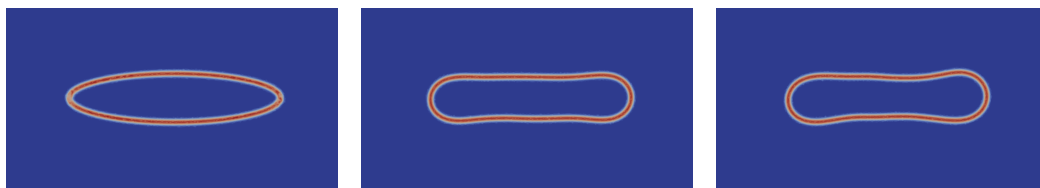


FIGURE 3.5 – Perte de la symétrie et léger déplacement du globule sur un maillage non structuré.

L'évolution de l'énergie d'Helfrich au cours du temps est donnée à la figure 3.6. On observe bien une décroissance de l'énergie, rapide au début puis plus lente proche du point d'équilibre.

Validité de la conservation du volume et d'aire

On s'intéresse maintenant à la conservation du volume et de l'aire. On rappelle

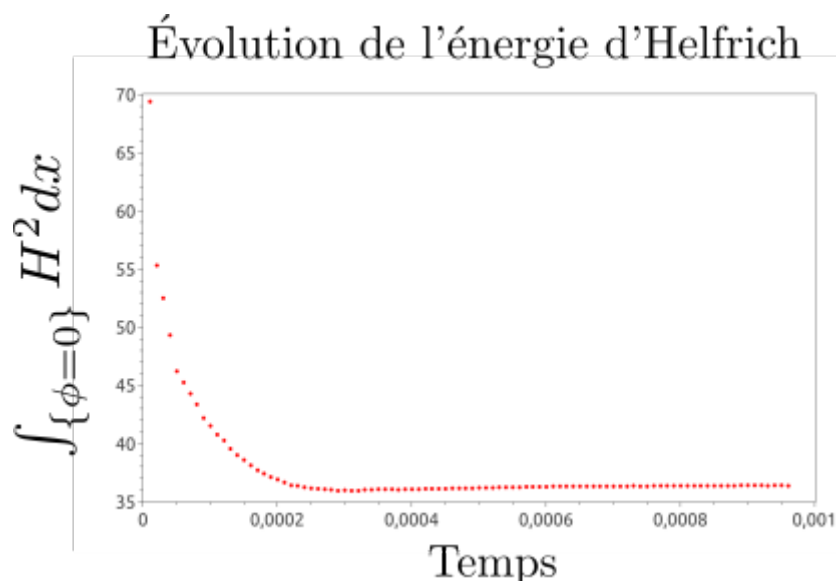


FIGURE 3.6 – Évolution de l'énergie d'Helfrich au cours du temps en dimension 2.

que le déplacement par flot de Willmore n'est pas contraint et que l'on récupère *a posteriori* ces quantités.

Lors de notre simulation, on constate une variation de volume d'environ 2,5% à chaque itération à l'issue de l'étape de diffusion et au début de l'étape de recouvrement des contraintes. L'étape de correction de volume permet de ramener la moyenne d'erreur sur le volume à 0,18%. Dans le pire des cas, l'erreur est de 0,45%. Après l'étape de correction de l'aire, on constate cependant une modification du volume bien que celui soit supposé inchangé par l'opération. On trouve à la fin de l'étape de conservation une erreur moyenne de 0,59% et de 1,35% dans le pire des cas.

En ce qui concerne l'aire de l'interface, on a une diminution d'environ 0,5% à la fin de l'étape de diffusion et de 1,08% après avoir récupéré le bon volume. À la fin de l'étape de conservation, on obtient une erreur moyenne de 0,24% et de 0,43% dans le pire des cas.

De la même manière, les valeurs de λ et μ restent à peu près constantes de l'ordre de $1,5 \cdot 10^{-3}$ et $4,5 \cdot 10^{-4}$ respectivement.

Comparaison des temps de calculs des différentes étapes

La simulation a été lancée sur 16 coeurs (Intel(R) Xeon(R) CPU E5-2670, 2.60GHz) pour 100 itérations sur la machine de calcul *Froggy* et a duré 450 secondes. L'initialisation a pris 50 secondes et chaque itération durait environ 4 secondes. L'étape de diffusion dure 0.15 sec, 0.75 sec pour la redistanciation par *fast-marching* et 1.2 sec pour l'étape de conservation de volume et d'aire. Le temps restant était utilisé pour les différents exports de résultats.

On constate que les deux parties les plus coûteuses de la méthode sont la redistanciation par *fast-marching* et l'étape de recouvrement de volume et d'aire. La version de la *fast-marching* disponible sur FEEL++ n'est pas parallèle et on ne peut augmenter le nombre de nœuds de calcul pour augmenter sa rapidité. L'étape de recouvrement de volume et d'aire nécessite un calcul de courbure ainsi que le calcul de plusieurs intégrales, ce qui prendra du temps avec un grand nombre d'éléments.

Obtention des formes d'équilibres pour différents taux de gonflements et comparaisons avec les autres méthodes.

Nous comparons les formes d'équilibres obtenues avec notre méthode avec celles obtenues dans [KHM11] pour différents taux de gonflements. Dans cet article, les auteurs utilisent la méthode de *Lattice-Boltzmann* et valident leur résultat en comparant avec la méthode des intégrales de frontières [BRS⁺04]. Les résultats sont présentés à la figure 3.7 pour les taux de remplissage égaux à 0.6; 0.7; 0.8 et 0.9. Nous proposons également la forme d'équilibre obtenue avec notre méthode pour $\nu = 0.5$. Il convient de noter que notre méthode ne fonctionne pas pour un taux de remplissage $\nu = 1$ qui correspond au cas interdit du cercle.

Nous proposons à la figure 3.8 les différentes formes d'équilibres obtenues par notre méthode pour différents taux de gonflements.

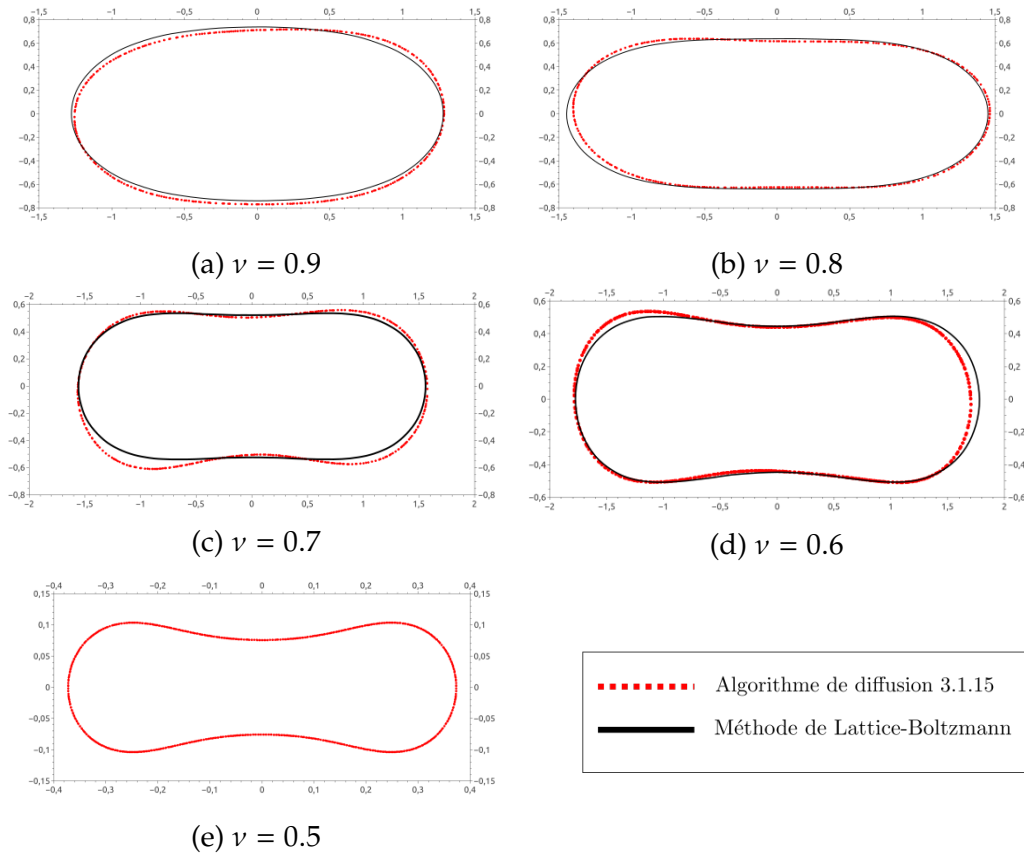


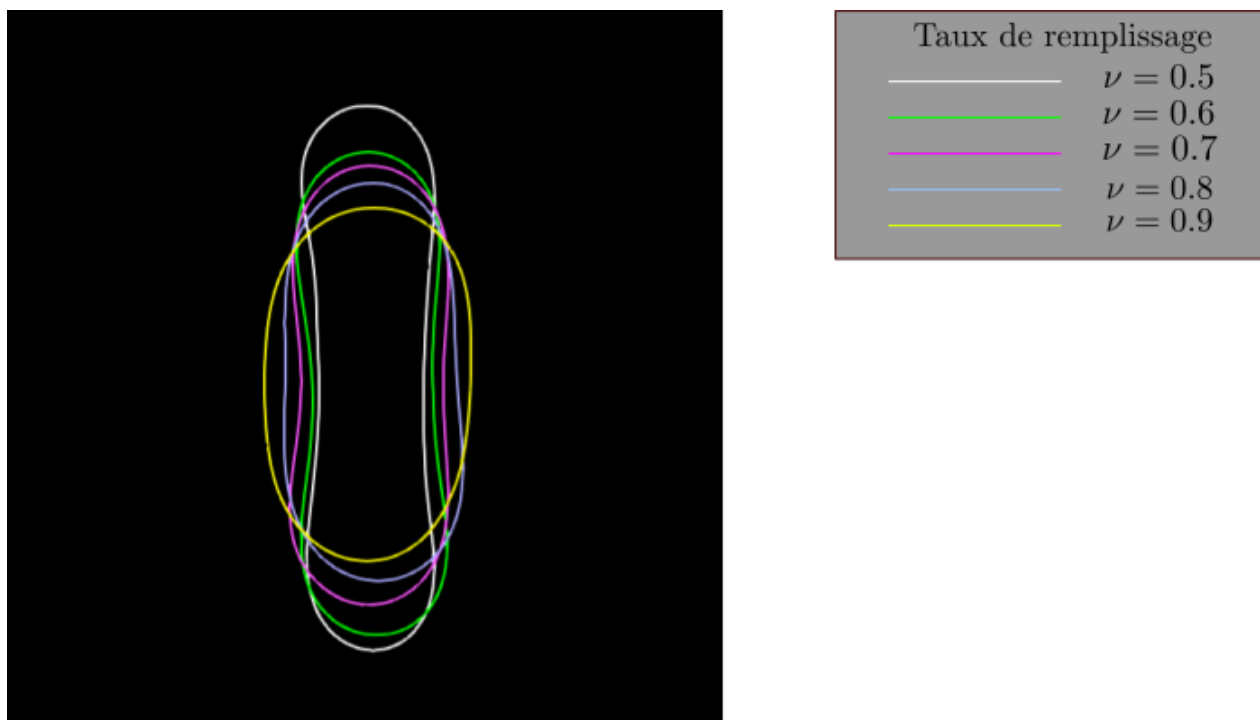
FIGURE 3.7 – Comparaison des formes d'équilibres obtenues par l'algorithme 3.1.4 : 15 avec celles de [KHM11] par la méthode de *Lattice-Boltzmann* pour des taux de gonflements de 0.9; 0.8; 0.7; 0.6. Nous représentons notre résultat pour 0.5.

3.2.2 Cas de la dimension 3

On s'intéresse ici au comportement de l'algorithme 3.1.4 : 16 3D dans le cas initial d'un ellipsoïde de révolution aplati de semi grand-axe $a = 0.418$ et de semi petit-axe $b = 0.096$.

Les paramètres que nous prenons sont les suivants. On considère un maillage sur un domaine parallélépipède de taille $2 \times 2 \times 1$ et un maillage éléments finis de taille $h = 0,008$ pour environ 2 500 000 éléments. Le paramètre ε qui détermine l'épaisseur de l'interface est fixé à $2h$. Le pas de temps utilisé est $\delta t = 10^{-6}$ de manière à ce que sa racine soit égale à 10^{-3} .

La méthode d'extraction de la courbure par diffusion de la section 2.4.6 fournit des résultats non satisfaisants en 3D lorsqu'elle est utilisée aux étapes 7) et 8) de l'algorithme 3.1.4 : 16. En effet, on observe beaucoup d'artefacts et d'irrégularités dans la courbure calculée, voir la figure 3.9. Cela est dû à la multiplication par le facteur $\frac{1}{\delta t}$ qui amplifie les erreurs numériques. De plus,

FIGURE 3.8 – Formes d'équilibres pour différents taux de remplissage ν .

comme on choisit la ligne de niveau $\{\phi + \lambda + \mu(H - \bar{H})\}$, la courbure est utilisée explicitement et les erreurs sont alors directement reportées dans la fonction ϕ^{k+1} . On préfère alors utiliser la méthode de projection \mathbb{L}^2 avec diffusion (2.81) qui donnent des résultats plus lisses.

Puisque l'on diffuse des fonctions distances signées, les équations de la chaleur sont complétées par des conditions aux bords de type Neumann :

$$\nabla\phi \cdot n = 1 \text{ sur } \partial\Omega \quad (3.49)$$

De plus, la discrétisation temporelle des équations de la chaleur est faite par un schéma de type *bdf* d'ordre 2 en temps. On utilise le pré-conditionneur *GASM* de *FEEL++* (se référer à l'annexe pour plus de précisions). Il est nécessaire d'utiliser une méthode itérative pour inverser la matrice car une méthode directe est trop coûteuse en espace mémoire.

Convergence de l'algorithme

La convergence de l'algorithme peut être observée à la figure 3.10 où l'on représente une coupe du profil du globule. On observe la forme bi-concave attendue.

L'évolution de l'énergie d'Helfrich au cours du temps est donnée à la figure 3.11. On observe plus de variations dans l'énergie malgré une décroissance

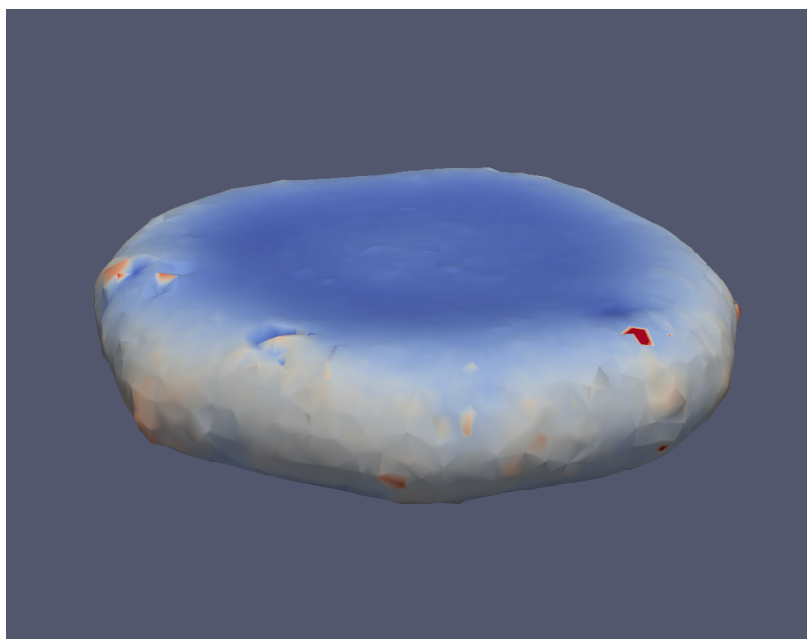


FIGURE 3.9 – Courbure calculée par diffusion pendant l'étape de correction de volume et d'aire présentant plusieurs irrégularités et artefacts qui s'accumulent au cours de la simulation

globale que dans le cas de la dimension 2. Cela est dû à la précision moindre inhérente à la 3D des calculs.

Validité de la conservation du volume et d'aire

Au niveau de la conservation du volume et de l'aire, on obtient un comportement similaire à la version 2D.

Lors de la simulation on constate une variation de volume d'environ 2% à chaque itération à l'issue de l'étape de diffusion, c'est-à-dire au début de l'étape de recouvrement des contraintes. L'étape de correction de volume permet de ramener la moyenne d'erreur sur le volume à 0.26% en moyenne et 0.8% dans le pire des cas.

En ce qui concerne l'aire de la surface, on observe une variation moyenne d'environ 1.3% à l'issue de l'étape de correction de volume. L'étape de correction d'aire permet de ramener cette erreur à une moyenne de 0.45% et de 0.8% dans le pire des cas.

Les valeurs de λ et μ ont des valeurs à peu près constantes de l'ordre respectivement de $5 \cdot 10^{-3}$ et $9 \cdot 10^{-5}$. Le signe de λ est négatif au début de la simulation puis devient négatif lorsqu'on approche la forme d'équilibre.

Comparaison des temps de calculs des différentes étapes

La simulation a été lancée sur 128 cœurs Intel(R) Xeon(R) CPU E5-2670, 2.60GHz

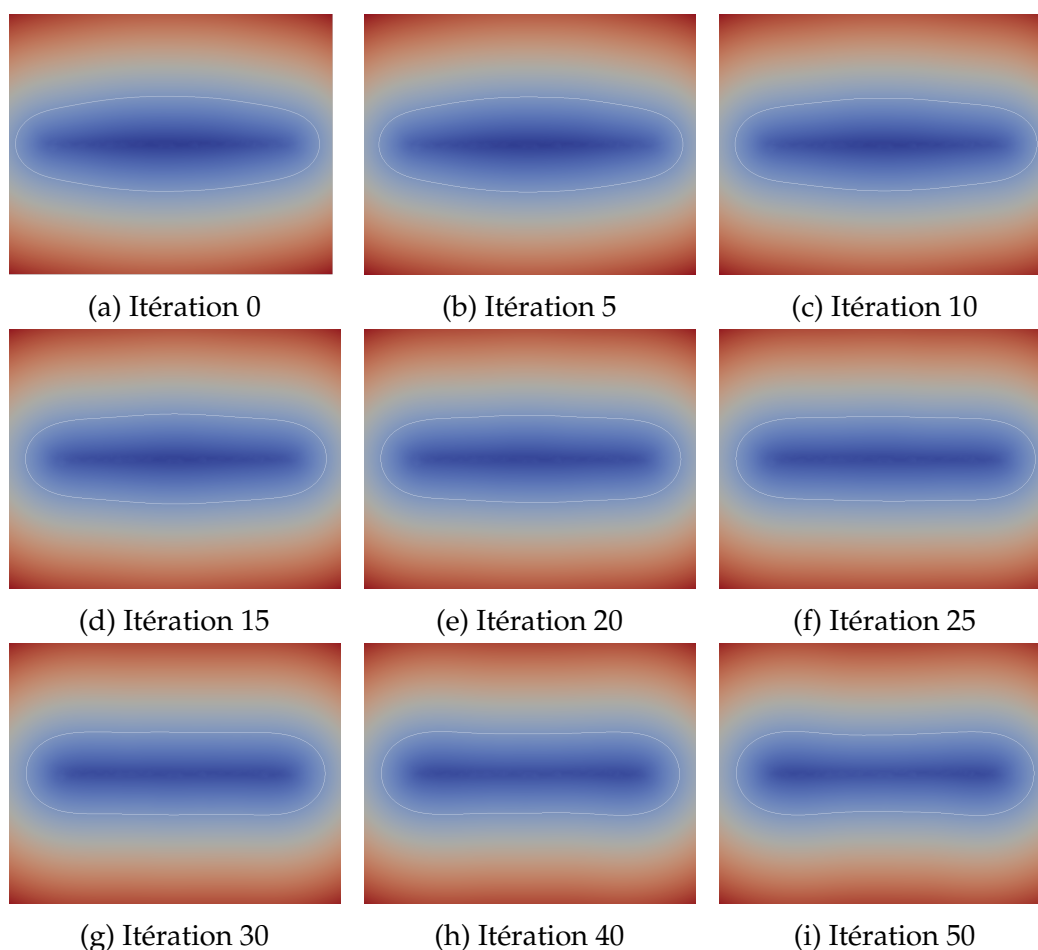


FIGURE 3.10 – Évolution du profil vers une forme d'équilibre biconcave du globule pour la simulation en dimension 3.

pour 200 itérations sur la machine de calcul *Froggy* et a duré 260 minutes, soit 4 heures 20. L'initialisation a duré 5 minutes et chaque itération prend environ 80 secondes. Les principales étapes durent environ : 28 sec pour l'étape de diffusion, 18 sec pour l'étape de *fast-marching* et 26 sec pour le recouvrement du volume et d'aire. Le reste du temps est occupé pour les exports de résultats.

L'efficacité de l'étape de diffusion pourrait être optimisée car les différentes équations de la chaleur sont résolues en séries. Un calcul en parallèle permettrait d'aller quatre fois plus vite si l'on alloue suffisamment de puissance machine.

L'étape de correction des contraintes est coûteuse car il est nécessaire de calculer la courbure moyenne en plus de plusieurs intégrales.

Obtention des formes d'équilibres pour différents taux de gonflements

Nous proposons à la figure 3.12 différentes formes d'équilibres obtenues par notre méthode pour différents des taux de remplissage proches de ceux d'un

Évolution de l'énergie d'Helfrich

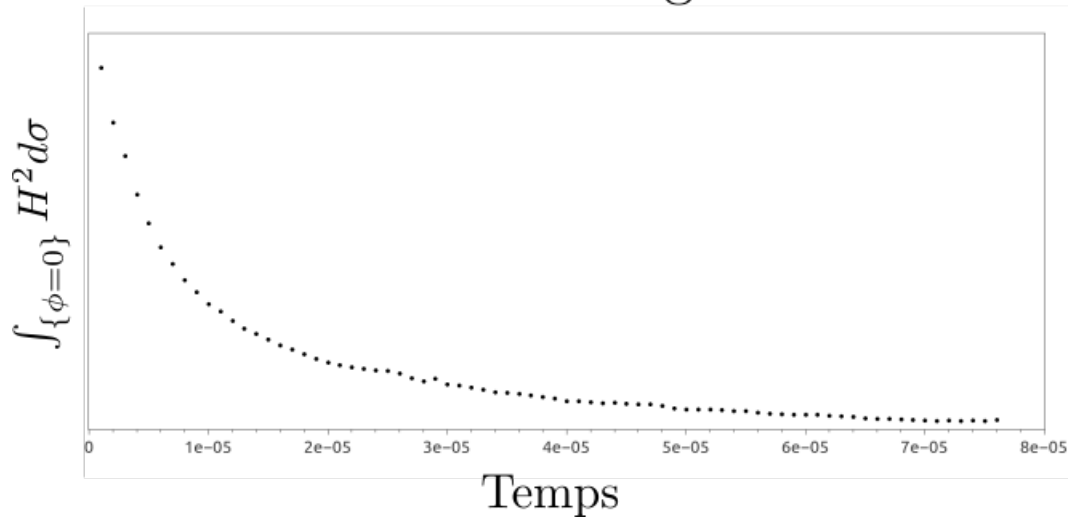


FIGURE 3.11 – Évolution de l'énergie d'Helfrich au cours du temps pour l'algorithme de flot de Willmore 3D avec conservation de volume et d'aire.

globule rouge 0.5 et 0.6 ainsi que pour le cas d'une vésicule proche d'une sphère 0.9.

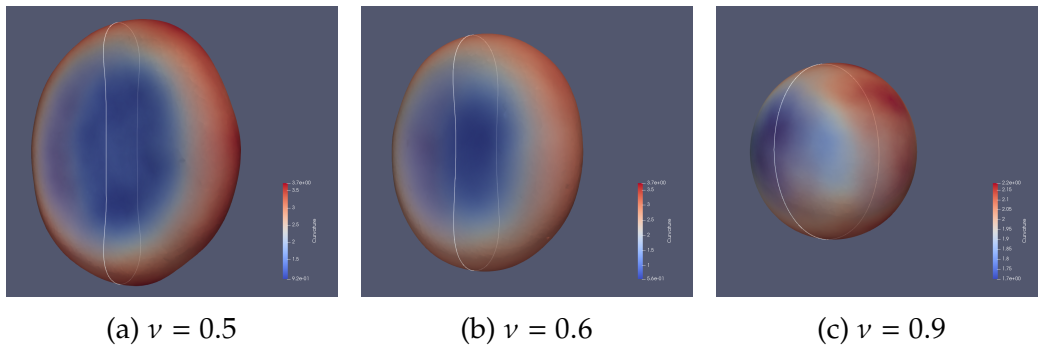


FIGURE 3.12 – Formes d'équilibres d'un globule rouge pour différents taux de remplissage ν en dimension 3.

3.3 Méthode de prédiction dans le cas des globules rouges

Dans la section précédente, on a présenté une méthode efficace pour obtenir le déplacement par flot de Willmore. On revient alors à notre problématique du Chapitre 1, à savoir la mise en place d'un schéma semi-implicite dans le cas d'un globule rouge immergé. On peut désormais utiliser les algorithmes 3.1.4 : 15 et 3.1.4 : 16 comme algorithmes de prédiction de la forme future de l'interface. On présente ici comment dériver le schéma semi-implicite dans ce cas.

On choisit le régime de Stokes de manière à alléger la dérivation de la méthode. Celle-ci est cependant similaire pour un fluide soumis au régime de Navier-Stokes. De même, les conditions aux bords restent à préciser. Dans nos calculs, on se placera dans le cas de conditions de Dirichlet homogènes afin d'annuler les termes de bords.

Considérons le système de Stokes couplé suivant :

$$\begin{cases} \partial_t u - \mu_f \Delta u + \nabla p = (W \nabla \phi) \delta_\varepsilon(\phi) \\ \operatorname{div} u = 0 \\ \partial_t \phi + u \cdot \nabla \phi = 0 \end{cases} \quad (3.50)$$

où le membre de droite correspond à la force de courbure et W est le gradient de l'énergie de Willmore :

$$W = \begin{cases} \Delta_\Gamma H + \frac{H^3}{2} \\ \Delta_\Gamma H + 2H(H^2 - K) \end{cases} \quad (3.51)$$

Comme la force de courbure dérive de l'énergie de courbure, on a :

$$\frac{\kappa_H}{2} \frac{d}{dt} \int_\Omega H^2 d\sigma = -\kappa_H \int_{\{\phi=0\}} W \nabla \phi \cdot u d\sigma \quad (3.52)$$

On pose $(u^k, \phi^k) = (u(t - \delta t), \phi(t - \delta t))$ et on cherche $(u^{k+1}, \phi^{k+1}) = (u(\delta t), \phi(\delta t))$. On reformule le système (3.50) sous la forme du problème de minimisation suivant :

$$(u^{k+1}, \phi^{k+1}) = \arg \min_{(u, \phi)} E(u, \phi) := \frac{1}{2\delta t} \int_\Omega (u - u^k)^2 dx + \frac{\mu_f}{2} \int_\Omega |\nabla u|^2 dx + \frac{\kappa_H}{2} \int_{\{\phi=0\}} H^2 d\sigma \quad (3.53)$$

où les contraintes restent à préciser.

Le gradient de E selon u dans la direction v donne :

$$\begin{aligned} d_u E(v) &= \int_\Omega \frac{u - u^k}{\delta t} \cdot v + \int_\Omega \nabla u : \nabla v dx - \kappa_H \int_{\{\phi=0\}} W \nabla \phi \cdot v d\sigma \\ &= \int_\Omega \left(\frac{u - u^k}{\delta t} - \mu_f \Delta u \right) \cdot v - \kappa_H \int_{\{\phi=0\}} W \nabla \phi \cdot v d\sigma \end{aligned} \quad (3.54)$$

L'équation d'Euler-Lagrange permet de retrouver la discrétisation de l'équation fluide (3.50) :

$$\frac{u^{k+1} - u^k}{\delta t} - \mu_f \Delta u^{k+1} = (W \nabla \phi) \delta_\varepsilon(\phi) \quad (3.55)$$

On discrétise l'équation de transport de la *levelset* de (3.50) soit par un schéma d'Euler explicite :

$$\frac{\phi^{k+1} - \phi^k}{\delta t} + u^{k+1} \cdot \nabla \phi^k = 0 \quad (3.56)$$

soit par un schéma d'Euler implicite :

$$\frac{\phi^{k+1} - \phi^k}{\delta t} + u^{k+1} \cdot \nabla \phi^{k+1} = 0 \quad (3.57)$$

Pour que le système décrive le cas d'un globule immergé dans un fluide de viscosité μ_f soumis au régime de Stokes, il convient d'ajouter les contraintes de conservation de volume et d'aire :

$$\left\{ \begin{array}{l} \int_{\{\phi=0\}} d\sigma = A_0 \\ \int_{\{\phi<0\}} dx = V_0 \end{array} \right. \quad (3.58)$$

Nous allons maintenant procéder à la mise en place de deux schémas semi-implicites qui diffèrent par le choix de la discrétisation de l'équation de transport par (3.56) ou (3.57).

3.3.1 Minimisation pour une équation de transport discrétisée par le schéma d'Euler explicite

Dans cette partie, on discrétise l'équation de transport avec le schéma d'Euler explicite (3.56). On obtient alors le problème de minimisation suivant :

$$\left\{ \begin{array}{l} \inf_{\phi, u} E(\phi, u) = \frac{1}{2\delta t} \int_{\Omega} (u - u_k)^2 dx + \frac{\mu_f}{2} \int_{\Omega} |\nabla u|^2 dx + \frac{\kappa_H}{2} \int_{\{\phi=0\}} H^2 d\sigma \\ \operatorname{div} u = 0 \\ \frac{\phi - \phi^k}{\delta t} + u \cdot \nabla \phi^k = 0 \\ \int_{\{\phi=0\}} d\sigma = A_0 \\ \int_{\{\phi<0\}} dx = V_0 \end{array} \right. \quad (3.59)$$

Pour un couple (ϕ^k, u^k) connu, on cherche (ϕ^{k+1}, u^{k+1}) minimisant l'énergie E sous les contraintes de ce système. On peut réécrire ce problème sous la forme suivante :

$$\left(\phi^{k+1}, u^{k+1}\right) = \arg \min_{(\phi, u) \in C_k} E(\phi, u) \quad (3.60)$$

où $C_k = \left\{ (\phi, u) / \frac{\phi - \phi^k}{\delta t} + u \cdot \nabla \phi^k = 0; \operatorname{div} u = 0; \int_{\{\phi < 0\}} dx = V_0; \int_{\{\phi = 0\}} d\sigma = A_0 \right\}$ désigne l'ensemble des contraintes à l'itération k .

On définit le Lagrangien de ce problème de minimisation :

$$\begin{aligned} \mathcal{L}((\phi, u), \xi, \lambda, \mu, p) = & E(\phi, u) + \int_{\Omega} \left(\frac{\phi - \phi^k}{\delta t} + u \cdot \nabla \phi^k \right) \xi dx + \lambda \left(\int_{\{\phi < 0\}} dx - V_0 \right) \\ & + \mu \left(\int_{\{\phi = 0\}} d\sigma - A_0 \right) - \int_{\Omega} p \operatorname{div} u dx \end{aligned} \quad (3.61)$$

La dérivation de \mathcal{L} selon u dans la direction v donne :

$$\begin{aligned} d_u \mathcal{L}(v) = & \int_{\Omega} \frac{u^{k+1} - u^k}{\delta t} \cdot v dx + \int_{\Omega} \mu_f \nabla u : \nabla v dx + \xi v \cdot \nabla \phi^k - p \operatorname{div} v dx \\ = & \int_{\Omega} \left(\frac{u^{k+1} - u^k}{\delta t} - \mu_f \Delta u + \xi \nabla \phi^k + \nabla p \right) \cdot v dx \end{aligned} \quad (3.62)$$

La solution du problème de minimisation vérifie alors l'équation suivante :

$$\frac{u^{k+1} - u^k}{\delta t} - \mu_f \Delta u^{k+1} + \nabla p^{k+1} = -\xi^{k+1} \nabla \phi^k \quad (3.63)$$

La dérivation de L selon ϕ dans la direction ψ :

$$\begin{aligned} d_{\phi} \mathcal{L}(\psi) = & \kappa_H \int_{\{\phi = 0\}} W \psi dx + \int_{\{\phi^k = 0\}} \frac{1}{\delta t} \xi \psi + \lambda \int_{\{\phi = 0\}} \psi d\sigma + \mu \int_{\{\phi = 0\}} H \psi d\sigma \\ = & \int_{\{\phi = 0\}} [\kappa_H W + \lambda + \mu H] \psi d\sigma + \int_{\Omega} \frac{1}{\delta t} \xi \psi dx \end{aligned} \quad (3.64)$$

La solution du problème de minimisation à l'instant précédent vérifie alors l'équation suivante :

$$\xi^{k+1} = -\delta t [W^{k+1} + \lambda^{k+1} + \mu^{k+1} H^{k+1}] \delta_{\varepsilon}(\phi^{k+1}) \quad (3.65)$$

On rappelle que l'opérateur \mathcal{P} , défini dans la section 3.1.4 à partir des algorithmes 3.1.4 : 15 et 3.1.4 : 16, permet d'obtenir un déplacement normal à la vitesse $W^{k+1} + \lambda^{k+1} + \mu^{k+1} H^{k+1} \delta t$. En approximant $\delta_{\varepsilon}(\phi^{k+1})$ par $\delta_{\varepsilon}(\mathcal{P}(\phi^k))$, on peut donc exprimer ξ^{k+1} sous la forme :

$$\xi^{k+1} = -(\mathcal{P}(\phi^k) - \phi^k) \delta_{\varepsilon}(\mathcal{P}(\phi^k)) \quad (3.66)$$

Les dérivations selon (ξ, p, λ, μ) permettent de récupérer les contraintes de divergence nulle du fluide, de volume et d'aire de la *levelset* ainsi que l'équation d'advection.

Au début de ce chapitre, nous avons montré dans les algorithmes 3.1.4 : 15 et 3.1.4 : 16 qu'on peut obtenir le terme $W^{k+1} + \lambda^{k+1} + \mu^{k+1}H^{k+1}$ ainsi qu'une prédiction $\widetilde{\phi}^{k+1} = \phi^k + [W^{k+1} + \lambda^{k+1} + \mu^{k+1}H^{k+1}] \delta t$ de la forme de la membrane. On rappelle que \mathcal{P} est l'opérateur de prédiction, défini dans la section 3.1.4, qui décrit ce déplacement. Le terme $W^{k+1} + \lambda^{k+1} + \mu^{k+1}H^{k+1}$ est obtenu en même temps que la prédiction $\widetilde{\phi}^{k+1}$ selon nos algorithmes 3.1.4 : 15 en dimension 2 et 3.1.4 : 16 en dimension 3. Ainsi, on obtient le schéma semi-implicite suivant :

$$\begin{cases} \widetilde{\phi}^{k+1} = \mathcal{P}(\phi^k) \text{ et } \xi^{k+1} = (\mathcal{P}(\phi^k) - \phi^k) \delta_\varepsilon(\widetilde{\phi}^{k+1}) \\ \frac{u^{k+1} - u^k}{\delta t} - \mu_f \Delta u^{k+1} + \nabla p^{k+1} = \int_{\Omega} \xi^{k+1} \nabla \phi^k dx \\ \operatorname{div} u^{k+1} = 0 \\ \phi^{k+1} + \delta t u^{k+1} \cdot \nabla \phi^k = \phi^k \end{cases} \quad (3.67)$$

3.3.2 Minimisation pour une équation de transport discrétisée par un schéma d'Euler implicite

Dans cette partie, on discrétise l'équation de transport avec le schéma d'Euler explicite (3.57). On obtient alors le problème de minimisation suivant :

$$\left(\phi^{k+1}, u^{k+1} \right) = \arg \min_{(\phi, u) \in C_k} E(\phi, u) \quad (3.68)$$

où $C_k = \left\{ (\phi, u) / \frac{\phi - \phi^k}{\delta t} + u \cdot \nabla \phi = 0 ; \operatorname{div} u = 0 ; \int_{\{\phi < 0\}} dx = V_0 ; \int_{\{\phi = 0\}} d\sigma \right\} = A_0$ désigne l'ensemble des contraintes.

On définit alors le Lagrangien de ce problème de minimisation :

$$\begin{aligned} \mathcal{L}((\phi, u), \xi, \lambda, \mu, p) &= E(\phi, u) + \int_{\Omega} \left(\frac{\phi - \phi^k}{\delta t} + \delta t u \cdot \nabla \phi \right) \xi dx + \lambda \left(\int_{\{\phi < 0\}} dx - V_0 \right) + \\ &+ \mu \left(\int_{\{\phi = 0\}} d\sigma - A_0 \right) - \int_{\Omega} p \operatorname{div} u dx \end{aligned} \quad (3.69)$$

La dérivation de \mathcal{L} selon u dans la direction v donne :

$$\begin{aligned} d_u \mathcal{L}(v) &= \int_{\Omega} \frac{u^{k+1} - u^k}{\delta t} \cdot v dx + \int_{\Omega} \mu_f \nabla u : \nabla v dx + t v \cdot \nabla \phi \xi - p \operatorname{div} v dx \\ &= \int_{\Omega} \left(\frac{u^{k+1} - u^k}{\delta t} + \xi \nabla \phi - \mu_f \Delta u + \nabla p \right) \cdot v dx \end{aligned} \quad (3.70)$$

La solution du problème de minimisation vérifie alors l'équation suivante :

$$\frac{u^{k+1} - u^k}{\delta t} - \mu_f \Delta u^{k+1} + \nabla p^{k+1} = -\xi^{k+1} \nabla \phi^{k+1} \quad (3.71)$$

La dérivation de \mathcal{L} selon ϕ dans la direction ψ :

$$\begin{aligned} d_\phi \mathcal{L}(\psi) &= \kappa_H \int_{\{\phi=0\}} W \psi dx + \int_{\Omega} \xi \left(\frac{\psi}{\delta t} + u \cdot \nabla \psi \right) dx + \lambda \int_{\{\phi=0\}} \psi d\sigma + \mu \int_{\{\phi=0\}} H \psi d\sigma \\ &= \kappa_H \int_{\{\phi=0\}} [W + \lambda + \mu H] \psi d\sigma + \int_{\Omega} \left(\frac{1}{\delta t} \xi - u \cdot \nabla \xi \right) \psi dx \end{aligned} \quad (3.72)$$

La solution du problème de minimisation vérifie alors l'équation de transport suivante :

$$\xi^{k+1} - \delta t u^{k+1} \cdot \nabla \xi^{k+1} = -\delta t \left(W^{k+1} + \lambda H^{k+1} + \mu^{k+1} \right) \delta_\varepsilon(\phi^{k+1}) \quad (3.73)$$

La dérivation selon ξ permet de récupérer l'équation de transport sur ϕ :

$$\phi^{k+1} + \delta t u^{k+1} \cdot \nabla \phi^{k+1} = \phi^k \quad (3.74)$$

Les dérivations selon (ξ, p, λ, μ) permettent de récupérer les contraintes de divergence nulle du fluide, de volume et d'aire de la *levelset* ainsi que l'équation d'advection. Finalement, on obtient le schéma semi-implicite suivant :

$$\begin{cases} \overline{\phi^{k+1}} = \mathcal{P}(\phi^k) \\ \xi^{k+1} - \delta t u^{k+1} \cdot \nabla \xi^{k+1} = - \left(\overline{\phi^{k+1}} - \phi^k \right) \delta_\varepsilon(\overline{\phi^{k+1}}) \\ \frac{u^{k+1} - u^k}{\delta t} - \mu_f \Delta u^{k+1} + \nabla p^{k+1} = \int_{\Omega} \xi^{k+1} \nabla \overline{\phi^{k+1}} dx \\ \operatorname{div} u^{k+1} = 0 \\ \phi^{k+1} + \delta t u^{k+1} \cdot \nabla \phi^k = \phi^k \end{cases} \quad (3.75)$$

Ce système est très proche du système (3.67) à la différence que l'équation déterminant ξ^{k+1} est maintenant une équation de transport à la vitesse $-u$.

3.3.3 Perspectives de la méthode

Les systèmes (3.67) et (3.75) sont des schémas semi-implicites dans le cas du couplage entre un globule rouge et un fluide. Des tests sont en cours pour évaluer les performances de ces nouveaux schémas par rapport au schéma explicite. Nous sommes en train de mesurer l'augmentation du pas de temps critique de stabilité de la méthode afin de voir l'amélioration du temps de calcul global par rapport au schéma explicite.

CHAPITRE 4

CONCLUSION ET PERSPECTIVES

Dans cette thèse, nous nous sommes intéressés à l'amélioration de schémas numériques pour le couplage fluide-structure, plus particulièrement dans le cas d'une membrane élastique immergée. Nous nous sommes placés dans un contexte purement eulérien du couplage, cette approche permettant notamment l'utilisation d'un maillage fixe unique pour le fluide comme la structure. Il repose sur la méthode *levelset* popularisée par Osher [OS88, OF06] où l'interface est représentée comme ligne de niveau 0 d'une fonction auxiliaire transportée par le fluide.

La stabilité des schémas de couplage constitue une difficulté importante lors de la simulation de problèmes d'interaction fluide-structure. Dans notre cas, l'utilisation d'un schéma implicite est trop coûteuse algorithmiquement car elle nécessiterait la résolution d'un système constitué d'une équation de Navier-Stokes couplée à une équation d'advection avec un terme source non linéaire. Dans le cas d'une membrane très rigide, l'efficacité d'un schéma explicite est grandement limitée par une restriction sur le pas de temps de l'ordre de $\Delta x^{\frac{3}{2}}$ [BKZ92, GV07, GV08].

Une solution à ce problème consiste à prédire la position et la forme de la membrane afin d'avoir une meilleure expression de la force de couplage et permet d'obtenir une stabilité inconditionnelle du couplage. Le prix à payer est la résolution d'une équation de la chaleur supplémentaire. Comme le coût de celle-ci est faible comparé à celui de l'équation de Navier-Stokes, le coût total d'une itération reste proche de celui du schéma explicite. On peut ainsi utiliser des pas de temps plus grands que pour le schéma explicite et il en résulte une meilleure efficacité globale de la méthode semi-implicite.

Cette méthode de prédiction, qui constitue le point de départ de la thèse, a été formulée dans [CM16] dans la cas d'un système de deux fluides en tension ou d'une membrane élastique immergée. L'objectif était de l'étendre au cas d'un globule rouge dans le sang. Il fallait donc prendre en compte l'effet de la force de courbure ainsi qu'imposer un volume et une aire constants lors de cette étape de prédiction. Nous allons décrire les différentes étapes qui nous ont permis d'atteindre cet objectif.

Le Chapitre 1 a été dédié à la présentation générale des modèles mathématiques et des méthodes numériques existants pour le couplage fluide-structure d'une membrane élastique immergée dans un fluide. Nous nous sommes intéressés plus particulièrement à l'approche eulérienne du couplage fluide-structure et aux méthodes *levelset*. Nous avons ensuite présenté l'étude numérique de la méthode de prédiction qui permet de définir le schéma semi-implicite et montré le gain de temps dans les simulations qu'elle pouvait apporter.

Le Chapitre 2 était consacré à la méthode de diffusion introduite par Merriman, Bence et Osher dans [MBO92]. Initialement utilisée pour une fonction indicatrice et le mouvement par courbure moyenne, cette méthode a été généralisée à des mouvements géométriques plus complexes. Elle a notamment été adaptée au contexte *levelset* en utilisant la fonction distance signée à l'interface

et en remplaçant l'étape de seuillage par une étape de redistanciation dans [ERT⁺10]. Nous avons alors proposé des algorithmes pour simuler le flot de Willmore en dimension 2 et 3. Ces algorithmes permettent de simuler un flot d'ordre 4 en ne faisant appel qu'à des résolutions d'équations de la chaleur. Il n'est pas nécessaire de dériver la fonction *levelset*. Ceci est particulièrement crucial puisque l'on choisit généralement une discrétisation \mathbb{P}^1 pour la fonction *levelset* lors des simulations.

Dans le Chapitre 3, on a proposé une méthode numérique efficace permettant, à partir d'une fonction *levelset* donnée, d'obtenir une nouvelle fonction *levelset* ayant un volume et une aire surfacique cibles. Dans [Smo05], Smolianski propose de choisir comme nouvelle interface la ligne de niveau qui englobe le volume cible. Nous avons alors adapté cette idée en choisissant comme nouvelle interface l'ensemble $\{\phi = \lambda + \mu H\}$, où λ et μ sont deux paramètres choisis de manière à obtenir les valeurs cibles de volume et d'aire. Nous avons ensuite proposé un algorithme pour obtenir une forme d'équilibre d'un globule rouge qui nécessite de déplacer l'interface selon le flot de Willmore tout en conservant le volume et l'aire associés. Il s'agit essentiellement de combiner la méthode précédente avec les algorithmes du Chapitre 2. Les codes FEEL++ pour nos simulations seront bientôt mis à disposition sur *Github*. Enfin, on a présenté notre schéma semi-implicite pour le cas d'un globule rouge immergé dans le sang qui était l'objet de la thèse. Il se base sur une prédiction de la position et de la forme de la membrane. Cette prédiction est réalisée à l'aide de notre méthode de déplacement par flot de Willmore sous contrainte de volume et d'aire. La voie est donc ouverte pour l'utilisation de cette méthode en vue de l'amélioration sur le temps de simulation.

Ce travail offre différentes perspectives à la fois numériques et théoriques ainsi que plusieurs pistes d'améliorations. On commence par évoquer deux pistes de travail.

La première piste de réflexion porte sur l'étude de la convergence théorique des schémas de type convolution-redistanciation. La question était de savoir si l'on pouvait montrer des résultats de convergence du même type que ceux existants pour les méthodes de convolution-seuillage. Une première idée a été d'essayer de comparer les deux déplacements et de voir s'il était possible de réutiliser les résultats existants dans le contexte de la fonction distance signée. Une autre idée a été de proposer des énergies à minimiser analogues à celles existant dans la littérature. Nous avons décidé de plus axer notre travail vers la mise en place des méthodes et leur étude numérique que sur ces aspects théoriques. On se reporte à la section 2.5 pour quelques réflexions sur le sujet.

Une autre piste étudiée pendant la thèse a été d'adapter les travaux de Bresch et al. [BDZ15, BCNV16, BJ18] à notre contexte *levelset*. Les auteurs étudient le système d'Euler-Korteweg compressible et proposent d'ajouter une nouvelle inconnue $w = \nabla f(\rho)$ homogène à une vitesse au système. Cet ajout permet d'obtenir des membres de droites antisymétriques dans l'équation fluide et

celle en w . Cette propriété d'antisymétrie permet d'obtenir des résultats de convergence théoriques [BDZ15, BJ18] par dérivations d'estimations *a priori* ainsi que numériques [BCNV16] en considérant les bons schémas. Nous nous sommes alors intéressés à transposer cette idée à notre modèle. Le raisonnement a été de chercher la nouvelle inconnue en fonction de la fonction *levelset* à la place de la densité. Nous avons obtenu un système augmenté en utilisant $w = \frac{2}{3} \frac{\nabla\phi}{\sqrt{|\nabla\phi|}}$ comme nouvelle inconnue. Cette piste n'a pas été poursuivie par choix de se consacrer aux méthodes de diffusion mais reste prometteuse. On propose en Annexe B les embryons de réflexion sur le sujet.

On s'intéresse maintenant aux perspectives qui permettraient de prolonger les résultats obtenus dans la thèse. La première est l'étude précise des performances numériques de notre méthode de prédiction du Chapitre 3 par rapport à un schéma explicite classique. Cette étude est en cours de réalisation. De plus, la méthode est en train d'être implémentée dans le contexte de la classe *levelset* de FEEL++. L'idée est de fournir un outil simple d'utilisation permettant d'exploiter les méthodes développées dans cette thèse en combinaison avec d'autres déjà disponibles dans cette classe.

Une deuxième perspective d'ordre numérique est l'étude des méthodes de convolution-seuillage par diffusion anisotrope, c'est-à-dire considérer une équation avec un coefficient de diffusion non constant. Cela pourrait être particulièrement intéressant pour le calcul de la courbure par diffusion. On pourrait en effet diffuser l'interface plus fortement dans les zones de faibles courbures où les conditions sur le pas de temps sont moins restrictives. Cela permettrait d'améliorer la précision de la méthode dans les cas où la courbure varie fortement sur la surface, sous réserve que son conditionnement du système ne soit pas trop prohibitif.

Une autre perspective possible est l'extension au cas de plusieurs globules rouges, en particulier à la méthode de Jedouaa, Maitre et Bruneau [JBM19] utilisant deux fonctions *levelset* et trois fonctions *label* pour représenter l'ensemble des globules. Il convient de noter que plusieurs problèmes rendent cette extension non triviale. D'une part, on risque un "chevauchement" entre les globules trop proches lors de l'étape de diffusion qui engendrerait des interactions non voulues et un possible échange de matière. On pourrait envisager de faire intervenir la seconde fonction *levelset* qui contrôle la distance au plus proche voisin afin de contrôler le pas de temps de diffusion pour éviter ce genre de problèmes. Il ne faudrait cependant pas que cela impose une contrainte trop restrictive sur le pas de temps de diffusion au risque de retomber sur les écueils du schéma explicite. Une autre piste de solution pourrait être de diffuser les fonctions de *Label* L_1 et L_2 dans l'esprit des algorithmes de convolution-seuillage multiphasique 3 et 4. D'autre part, l'étape de correction de volume et d'aire semble également problématique. Il est en effet impossible de distinguer les différentes interfaces et on risque de grossir certains globules aux dépens d'autres malgré une conservation globale du volume et de l'aire. On peut envisager de

recourir aux fonctions *labels* pour localiser cette étape et l'effectuer pour chaque globule individuellement.

ANNEXE A : TRAVAIL SUR FEEL++

FEEL++, pour FINITE ELEMENT EMBEDDED LIBRARY IN C++, est une librairie d'éléments finis [PCD⁺12] développée en C++ créée par Christophe Prud'homme. Elle permet d'utiliser une grande variété de méthodes de Galerkin ainsi que des méthodes numériques avancées telles que la méthode des domaines fictifs, la méthode de décomposition de domaine ou la méthode des bases réduites pour des éléments finis d'ordre arbitraire.

Nous avons choisi FEEL++ pour sa capacité à faire du calcul haute performance, sa simplicité d'utilisation et son expressivité proche des mathématiques. En effet, cela permet de lancer des calculs massivement parallèles. Le maillage est découpé en sous-maillages et les équations sont résolues en parallèle sur différents coeurs de la machine. De plus, la gestion du parallélisme est effectuée par la librairie et il suffit à l'utilisateur de choisir le nombre de processeurs à utiliser. De la même façon, la définition et la résolution de problèmes différentiels est très simple d'utilisation. Il suffit de donner la formulation variationnelle et d'appeler le solveur. Le fonctionnement de celui-ci est géré directement par FEEL++.

FEEL++ est une librairie expressive dans le sens où la définition des problèmes variationnel est très proche de la formule mathématique. Cela permet de transcrire simplement un algorithme en pseudo-code.

Les fonctions d'export permettent de visualiser les résultats des calculs sur *Paraview*.

Exemple de code FEEL++: l'algorithme de convolution-redistanciation selon le flot de Willmore avec conservation de volume et d'aire en dimension 2

Présentation générale de la librairie FEEL++

Nous donnons dans cette partie un exemple de code FEEL++: l'algorithme de flot de Willmore avec conservation de volume et d'aire en dimension 2 [3.1.4 : 15](#). L'objectif est d'illustrer la similarité d'un code FEEL++ avec la formulation mathématique. On présente dans la double page suivante une version reformulée en pseudo-code de l'algorithme d'un côté et le code FEEL++ correspondant de l'autre.

La partie du code que nous présentons est le coeur de l'algorithme. Il convient de compléter celle-ci avec une phase d'initialisation et une phase d'export des résultats.

Afin de faire tenir le code sur une page, le calcul de la courbure pendant l'étape de correction a été omis. En la calculant par la méthode d'ordre 1 par diffusion, le code correspondant est :

```

1 // Resolution de l'equation de la chaleur
  b1 = integrate(_range = elements(mesh),
3     _expr = (1/sqrt_dt)*idv(phi)*id(psi)
              -0.5*inner(gradv(phi),grad(psi)) );
5 b1 += integrate(_range=boundaryfaces(mesh), _expr=id(psi));
// Definition de l'expression K_expr donnant la courbure moyenne
7 auto K_expr = ( idv(G1) - idv(phi) )/sqrt_dt ;

```

Dans notre code, on appelle K notre courbure moyenne pour ne pas la confondre avec la fonction d'Heaviside H . D'autre part, les expressions correspondant aux fonctions de Dirac et Heavysde régularisées sont définies comme suit :

```

1 // Definition du dirac regularise
auto D_expr = vf::chi( idv(phi)<=-eps ) *vf::constant(0.0)
3     +
      vf::chi( idv(phi)>=-eps ) *vf::chi( idv(phi)<=eps ) *
5     1/(2*eps) * ( 1 + cos(pi*idv(phi)/eps) )
      +
7     vf::chi( idv(phi)>eps ) *vf::constant(0.0);
// Definition du Heaviside regularise
9 auto H_expr = 1-vf::chi( idv(phi)<=-eps ) *vf::constant(0.0)
      -vf::chi( idv(phi)>=-eps ) *vf::chi( idv(phi)<=eps ) *
11     1/2*(1 + idv(phi)/eps + 1/pi*vf::sin( pi*idv(phi)/eps ) )
      -vf::chi( idv(phi)>eps ) *vf::constant(1.0);

```


Algorithme 17 : Flot de Willmore 2D

Pour k de 0 à $nb_{iter} - 1$:

1) Étape de convolution : Calculer les convolutions : $G_{2\sqrt{\delta t}} * \phi$ et $G_{\sqrt{\delta t}} * \phi$.

Calcul de $G_1 = G_{\sqrt{\delta t}} * \phi$ en résolvant $\forall \psi_h$:

$$\frac{1}{\sqrt{\delta t}} \int_{\Omega_h} G_1 \psi_h dx + \frac{1}{2} \int_{\Omega_h} \nabla G_1 \cdot \nabla \psi_h dx = \frac{1}{\sqrt{\delta t}} \int_{\Omega_h} \phi \psi_h dx - \frac{1}{2} \int_{\Omega_h} \nabla \phi \cdot \nabla \psi_h dx + \int_{\partial \Omega_h} \psi_h ds$$

Calcul de $G_2 = G_{2\sqrt{\delta t}} * \phi$ en résolvant : $\forall \psi_h$:

$$\frac{1}{\sqrt{2\delta t}} \int_{\Omega_h} G_2 \psi_h dx + \frac{1}{2} \int_{\Omega_h} \nabla G_1 \cdot \nabla \psi_h dx = \frac{1}{\sqrt{2\delta t}} \int_{\Omega_h} \phi \psi_h dx - \frac{1}{2} \int_{\Omega_h} \nabla \phi \cdot \nabla \psi_h dx + \int_{\partial \Omega_h} \psi_h ds$$

2) Former les termes :

$$A \leftarrow 2G_{\sqrt{\delta t}} * \phi - G_{2\sqrt{\delta t}} * \phi$$

$$B \leftarrow \left(G_{\sqrt{\delta t}} * \phi - \phi \right)^3$$

$$\phi \leftarrow A + \frac{B}{2\sqrt{\delta t}}$$

3) Renormalisation de la fonction ϕ :

$$\phi \leftarrow \frac{\phi}{\sqrt{|\nabla \phi|^2}}$$

4) Correction du volume :

$$\lambda \leftarrow \frac{V_{cible} - V(\phi)}{A(\phi)} \quad (4.1)$$

5) Correction de l'aire en conservant le volume :

$$\mu \leftarrow \frac{A_{cible} - A(\phi + \lambda)}{\int_{\{\phi=0\}} \kappa(\phi + \lambda) (\kappa(\phi + \lambda) - \bar{\kappa}(\phi + \lambda)) |\nabla \phi| d\sigma}$$

6) Étape de redistanciation :

$$\phi \leftarrow \text{redist}(\phi + \lambda + \mu (\kappa(\phi + \lambda) - \bar{\kappa}(\phi + \lambda)))$$

```

// Definition des formes bilineaires des equations de la chaleur
2 a1 = integrate(_range = elements(mesh),
    _expr = (1/sqrt_dt)*idt(G1)*id(psi)
4         + 0.5*inner(gradt(G1),grad(psi)) );
a2 = integrate(_range = elements(mesh),
6         _expr = (1/(2*sqrt_dt))*idt(G2)*id(psi)
            + 0.5*inner(gradt(G2),grad(psi)) );
8 for(int i=0; i<nbIter; i++) {
//Etape 1) Resolution des equations de la chaleur pour G1 et G2
10 b1 = integrate(_range = elements(mesh),
    _expr = (1/sqrt_dt)*idv(phi)*id(psi)
12         -0.5*inner(gradv(phi),grad(psi)) );
b1 += integrate(_range=boundaryfaces(mesh), _expr=id(psi));
14 a1.solve( _solution=G1, _rhs=b1, _name="g1" );
b2 = integrate(_range = elements(mesh),
16         _expr = (1/(2*sqrt_dt))*idv(phi)*id(psi)
            -0.5*inner(gradv(phi),grad(psi)) );
18 b2 += integrate(_range=boundaryfaces(mesh), _expr=id(psi));
a2.solve( _solution=G2, _rhs=b2, _name="g2" );
20 //Etape 2) Former les termes
expr_A = 2*idv(G1)-idv(G2);
22 expr_B = vf::pow( idv(G1) - idv(phi) , 2.0)
phi = vf::project(_space = Xh,_range = elements(mesh),
24         _expr = expr_A + expr_B/(2*sqrt_dt) );
//Etape 3) Renormalisation de la fonction phi
26 phi = vf::project(Xh, elements(mesh),
    idv(phi)/sqrt(gradv(phi)*trans(gradv(phi))) );
28 //Etapes 4) et 5) Calculs de lambda et mu
volume = integrate(_range = elements(mesh),
30         _expr = H_expr ).evaluate() (0,0);
perimeter = integrate(_range = elements(mesh),
32         _expr = Delta_expr ).evaluate() (0,0);
lambda = ( volume-volumeCible )/perimeter;
34 phi = vf::project(Xh, elements(mesh),
    idv(phi)+lambda );
36 interPerimeter = integrate(_range = elements(mesh),
    _expr = Delta_expr ).evaluate() (0,0);
38 intK = 2*pi/interPerimeter;
varK = integrate(_range = elements(mesh),
40         _expr = rescK_expr*(K_expr-intK)*Delta_expr ).evaluate() (0,0);
mu = (interPerimeter-perimeterCible)/varK;
42 phi = vf::project( _space = Xh,
    _range = elements(mesh),
44         _expr = idv(phi) + mu*( rescK_expr-intK ) );
//Etape 6) Redistanciation :
46 phi = vf::project(Xh, elements(mesh),
    idv(phi)/sqrt(gradv(phi)*trans(gradv(phi))) );
48 phi = fmP1->march( phi ); } // fin boucle for

```

ANNEXE B

Cette annexe présente une piste de réflexion envisagée puis abandonnée à la faveur des méthodes de diffusion. Cette piste était de voir s'il est possible d'adapter les travaux de Bresch et al. [BDZ15, BCNV16, BJ18] faits pour des systèmes d'Euler-Korteweg compressibles à notre contexte de couplage fluide-structure.

Dans [BCNV16], il est montré que l'ajout d'une nouvelle inconnue peut améliorer la stabilité numérique du système. Plus précisément, le problème considéré est le système d'Euler-Korteweg compressible suivant :

$$\begin{cases} \partial_t \rho + \operatorname{div} \rho u = 0 \\ \partial_t + \operatorname{div} \rho u \otimes u + \nabla p(\rho) = \operatorname{div} \mathbb{K} \end{cases} \quad (4.2)$$

où $\mathbb{K} = (\rho \operatorname{div}(K(\rho)\nabla\rho) + \frac{1}{2} (K(\rho) - rK'(\rho)) |\nabla\rho|^2) \mathbb{I} - K(\rho)\nabla\rho \otimes \nabla\rho$ est le tenseur de Korteweg et K le coefficient de capillarité.

Ce modèle admet différentes applications telles que les mélanges liquides-vapeurs [JTB02], les superfluides [HAC⁺06] (l'Hélium près du zéro absolu) ou les équations de *shallow water* à petites échelles [LSG95]. Selon l'application le coefficient de capillarité peut être choisi constant ou de la forme $K(\rho) = \frac{c}{\rho}$ dans le cas de *shallow water*.

L'ingrédient pour définir la nouvelle inconnue est une généralisation de l'identité de Bohm $2\rho \operatorname{div}(\frac{\Delta\sqrt{\rho}}{\sqrt{\rho}}) = \operatorname{div}(r\nabla\nabla \log \rho)$ suivante :

$$\rho \nabla \left(\sqrt{K(\rho)} \Delta \left(\int_0^\rho \sqrt{K(s)} ds \right) \right) = \operatorname{div}(F(\rho)\nabla\nabla\phi(\rho)) - \nabla((F(\rho) - F'(\rho)\rho) \Delta\phi(\rho)) \quad (4.3)$$

avec K, F et ϕ déterminées par les équations suivantes :

$$\begin{cases} K(\rho) = \frac{c}{\rho} \text{ avec } c \text{ constante} \\ \sqrt{\rho}\phi'(\rho) = \sqrt{K(\rho)} \\ F'(\rho) = \sqrt{K(\rho)}\rho \end{cases} \quad (4.4)$$

La nouvelle inconnue considérée est alors $w = \nabla\phi(\rho)$. En utilisant les notations précédentes, le système augmenté s'écrit alors :

$$\begin{cases} \partial_t \rho + \operatorname{div} \rho u = 0 \\ \partial_t + \operatorname{div} \rho u \otimes u + \nabla p(\rho) = \operatorname{div} F(\rho \nabla w^T) - \nabla ((F(\rho) - rF'(\rho)) \operatorname{div} w) \\ \partial_t(\rho w) + \operatorname{div} \rho w \otimes u = -\operatorname{div} F(\rho) \nabla u^T + \nabla (F(\rho) - rF'(\rho)) \operatorname{div} u \end{cases} \quad (4.5)$$

Il convient de remarquer que les membres de droite des équations en u et w sont opposés. Il est alors possible d'obtenir une matrice antisymétrique lors de la résolution numérique du système et d'augmenter les conditions de stabilité. On se réfère à [BCNV16] pour les détails sur la dérivation du schéma numérique.

Une idée similaire avait déjà été proposée dans [BGDD07] afin de dériver des estimations *a priori* et montrer le caractère bien posé du système. L'utilisation du système augmenté peut avoir des applications numériques comme théoriques.

On s'intéresse maintenant aux quelques calculs que nous avons faits pour essayer d'utiliser ce raisonnement dans notre contexte *levelset*. On considère le système suivant :

$$\begin{cases} \partial_t u + u \cdot \nabla u + \nabla p - \mu \Delta u = -\operatorname{div} \left(\frac{\nabla \phi \otimes \nabla \phi}{|\nabla \phi|} \right) \\ \nabla \cdot u = 0 \\ \partial_t \phi + u \cdot \nabla \phi = 0 \end{cases} \quad (4.6)$$

Ce système correspond à notre modèle *levelset* (1.78) dans le cas d'une énergie $E'(r) = 1$, c'est-à-dire d'un système de deux fluides en tension avec un coefficient de tension de surface pris à 1. La fonction ϕ va jouer un rôle similaire dans l'obtention du système augmenté à celui de la densité ρ dans le cas compressible. On cherche alors à trouver une fonction $w(\phi)$ homogène à une vitesse qui conviendrait.

En prenant le gradient de l'équation de transport, on a :

$$\partial_t \nabla \phi + \nabla^2 \phi \cdot u + \nabla u^T \cdot \nabla \phi = 0 \quad (4.7)$$

De même, on peut écrire l'équation sur $|\nabla \phi|$:

$$\partial_t |\nabla \phi| + u \cdot \nabla |\nabla \phi| = \frac{\nabla \phi \otimes \nabla \phi}{|\nabla \phi|} : \nabla u \quad (4.8)$$

On déduit de (4.7) et (4.8) l'équation vérifiée par $\frac{\nabla\phi}{\sqrt{|\nabla\phi|}}$:

$$\begin{aligned}
\partial_t \left(\frac{\nabla\phi}{\sqrt{|\nabla\phi|}} \right) &= \frac{\partial_t \nabla\phi}{\sqrt{|\nabla\phi|}} + \partial_t \left(\frac{1}{\sqrt{|\nabla\phi|}} \right) \nabla\phi \\
&= -\frac{\nabla^2\phi \cdot u}{\sqrt{|\nabla\phi|}} - \frac{\nabla u^T \cdot \nabla\phi}{\sqrt{|\nabla\phi|}} + \frac{u \cdot \nabla|\nabla\phi|}{2|\nabla\phi|^{\frac{3}{2}}} \nabla\phi - \frac{\nabla\phi \otimes \nabla\phi : \nabla u}{2|\nabla\phi|^{\frac{5}{2}}} \nabla\phi \\
&= -\operatorname{div} \left(\frac{\nabla\phi}{\sqrt{|\nabla\phi|}} \otimes u \right) - \frac{\nabla u^T \cdot \nabla\phi}{\sqrt{|\nabla\phi|}} - \frac{\nabla\phi \otimes \nabla\phi : \nabla u}{2|\nabla\phi|^{\frac{5}{2}}} \nabla\phi
\end{aligned} \tag{4.9}$$

Le terme $w = \alpha \frac{\nabla\phi}{\sqrt{|\nabla\phi|}}$ avec α constante, vérifie alors l'équation :

$$\partial_t w + \operatorname{div} w \otimes u = -\nabla u^T \cdot w - \frac{w \otimes w : \nabla u}{2|w|^2} w \tag{4.10}$$

On obtient ainsi le système augmenté suivant :

$$\begin{cases}
\partial_t u + \operatorname{div} u \otimes w + \nabla p - \mu \Delta u = -\frac{1}{\alpha^2} \operatorname{div} (w \otimes w) \\
\operatorname{div} u = 0 \\
\partial_t w + \operatorname{div} w \otimes u = -\nabla u^T \cdot w - \frac{1}{2} \frac{w \otimes w : \nabla u}{|w|^2} w \\
\partial_t \phi + u \cdot \nabla\phi = 0
\end{cases} \tag{4.11}$$

Les membres de droite des équations en u et w sont antisymétriques dans le sens suivant :

$$\begin{cases}
\int \left(-\frac{1}{\alpha^2} \operatorname{div} w \otimes w \right) \cdot u = \int \frac{1}{\alpha^2} w \otimes w : \nabla u \\
\int \left(-\nabla u^T \cdot w - \frac{1}{2} \frac{w \otimes w : \nabla u}{|w|^2} w \right) \cdot w = -\frac{3}{2} \int w \otimes w : \nabla u
\end{cases} \tag{4.12}$$

Pour $\alpha = \sqrt{\frac{2}{3}}$, ces deux intégrales sont opposées. On pourrait alors envisager d'écrire des estimations *a priori* ou des schémas utilisant cette propriété.

BIBLIOGRAPHIE

- [ABF99] Philippe Angot, Charles-Henri Bruneau, and Pierre Fabrie. A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik*, 81(4):497–520, 1999.
- [AGS07] Carla Antoci, Mario Gallati, and Stefano Sibilla. Numerical simulation of fluid–structure interaction by sph. *Computers & Structures*, 85(11-14):879–890, 2007.
- [ALV02] Manouk Abkarian, Colette Lartigue, and Annie Viallat. Tank treading and unbinding of deformable vesicles in shear flow: determination of the lift force. *Physical review letters*, 88(6):068103, 2002.
- [ATW93] Fred Almgren, Jean E Taylor, and Lihe Wang. Curvature-driven flows: a variational approach. *SIAM Journal on Control and Optimization*, 31(2):387–438, 1993.
- [AV08] Manouk Abkarian and Annie Viallat. Vesicles and red blood cells in shear flow. *Soft Matter*, 4(4):653–657, 2008.
- [BCNV16] Didier Bresch, Frédéric Couderc, Pascal Noble, and Jean-Paul Vila. A generalization of the quantum bohm identity: Hyperbolic cfl condition for euler–korteweg equations. *Comptes Rendus Mathématique*, 354(1):39–43, 2016.
- [BDZ15] Didier Bresch, Benoît Desjardins, and Ewelina Zatorska. Two-velocity hydrodynamics in fluid mechanics: Part ii existence of global κ -entropy solutions to the compressible navier–stokes systems with degenerate viscosities. *Journal de Mathématiques Pures et Appliquées*, 104(4):801–836, 2015.
- [BG95] Guy Barles and Christine Georgelin. A simple proof of convergence for an approximation scheme for computing motions by

- mean curvature. *SIAM Journal on Numerical Analysis*, 32(2):484–500, 1995.
- [BGDD07] Sylvie Benzoni-Gavage, Raphaël Danchin, and Stéphane Descombes. On the well-posedness for the euler-korteweg model in several space dimensions. *Indiana University Mathematics Journal*, pages 1499–1579, 2007.
- [BH82] Alexander N Brooks and Thomas JR Hughes. Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Computer methods in applied mechanics and engineering*, 32(1-3):199–259, 1982.
- [Bib05] Thierry Biben. Phase-field models for free-boundary problems. *European Journal of Physics*, 26(5):S47, 2005.
- [BIM11] Silvia Bertoluzza, Mourad Ismail, and Bertrand Maury. Analysis of the fully discrete fat boundary method. *Numerische Mathematik*, 118(1):49–77, 2011.
- [BJ18] Didier Bresch and Pierre-Emmanuel Jabin. Global existence of weak solutions for compressible navier–stokes equations: Thermodynamically unstable pressure and anisotropic viscous stress tensor. *Annals of Mathematics*, 188(2):577–684, 2018.
- [BKM05] Thierry Biben, Klaus Kassner, and Chaouqi Misbah. Phase-field approach to three-dimensional vesicle dynamics. *Physical Review E*, 72(4):041921, 2005.
- [BKZ92] JU Brackbill, Douglas B Kothe, and Charles Zemach. A continuum method for modeling surface tension. *Journal of computational physics*, 100(2):335–354, 1992.
- [BMO15] Elie Bretin, Simon Masnou, and Edouard Oudet. Phase-field approximations of the willmore functional and flow. *Numerische Mathematik*, 131(1):115–171, 2015.
- [Bou95] A Bourlioux. A coupled level-set volume-of-fluid algorithm for tracking material interfaces. In *proceedings of the 6th international symposium on computational fluid dynamics, Lake Tahoe, CA*, volume 15, 1995.
- [Bre83] Ham Brezis. Analyse fonctionnelle. théorie et applications.(french)[functional analysis. theory and applications], collection mathématiques appliquées pour la maîtrise.[collection of applied mathematics for the master’s degree], 1983.

- [Bre12] Wim-Paul Breugem. A second-order accurate immersed boundary method for fully resolved simulations of particle-laden flows. *Journal of Computational Physics*, 231(13):4469–4498, 2012.
- [BRS⁺04] J Beaucourt, F Rioual, T Séon, T Biben, and C Misbah. Steady to unsteady dynamics of a vesicle in a flow. *Physical Review E*, 69(1):011906, 2004.
- [Cal94] J-P Caltagirone. Sur l’interaction fluide-milieu poreux; application au calcul des efforts exercés sur un obstacle par un fluide visqueux. *Comptes rendus de l’Académie des sciences. Série II, Mécanique, physique, chimie, astronomie*, 318(5):571–577, 1994.
- [Can70] Peter B Canham. The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell. *Journal of theoretical biology*, 26(1):61–81, 1970.
- [CC08] Mathieu Coquerelle and G-H Cottet. A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies. *Journal of Computational Physics*, 227(21):9121–9137, 2008.
- [CCB⁺08] Philippe Chatelain, Alessandro Curioni, Michael Bergdorf, Diego Rossinelli, Wanda Andreoni, and Petros Koumoutsakos. Billion vortex particle direct numerical simulations of aircraft wakes. *Computer Methods in Applied Mechanics and Engineering*, 197(13-16):1296–1304, 2008.
- [CCCD93] Vicent Caselles, Francine Catté, Tomeu Coll, and Françoise Dibos. A geometric model for active contours in image processing. *Numerische mathematik*, 66(1):1–31, 1993.
- [CDFVM02] Italo Capuzzo Dolcetta, Stefano Finzi Vita, and Riccardo March. Area-preserving curve-shortening flows: from phase separation to image processing. *Interfaces and Free Boundaries*, 4(4):325–343, 2002.
- [Cha04] Antonin Chambolle. An algorithm for mean curvature motion. *Interfaces and free Boundaries*, 6(2):195–218, 2004.
- [Cha13] Vincent Chabannes. *Vers la simulation des écoulements sanguins*. PhD thesis, Université de Grenoble, 2013.
- [Cho67] Alexandre Joel Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of computational physics*, 2(1):12–26, 1967.

- [CKP⁺00] Georges-Henri Cottet, Petros D Koumoutsakos, D Petros, Petros D Koumoutsakos, et al. *Vortex methods: theory and practice*. Cambridge university press, 2000.
- [CL85] Joseph B Collins and Herbert Levine. Diffuse interface model of diffusion-limited crystal growth. *Physical Review B*, 31(9):6119, 1985.
- [CL03] Andrea Colagrossi and Maurizio Landrini. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *Journal of computational physics*, 191(2):448–475, 2003.
- [CM99] Isabelle Cantat and Chaouqi Misbah. Lift force and dynamical unbinding of adhering vesicles under shear flow. *Physical review letters*, 83(4):880, 1999.
- [CM04] Georges-Henri Cottet and Emmanuel Maitre. A level-set formulation of immersed boundary methods for fluid–structure interaction problems. *Comptes Rendus Mathematique*, 338(7):581–586, 2004.
- [CM06] Georges-Henri Cottet and Emmanuel Maitre. A level set method for fluid-structure interactions with immersed surfaces. *Mathematical models and methods in applied sciences*, 16(03):415–438, 2006.
- [CM16] Georges-Henri Cottet and Emmanuel Maitre. A semi-implicit level set method for multiphase flows and fluid–structure interaction problems. *Journal of Computational Physics*, 314:80–92, 2016.
- [CMM08] Georges-Henri Cottet, Emmanuel Maitre, and Thomas Milcent. Eulerian formulation and level set models for incompressible fluid-structure interaction. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(3):471–492, 2008.
- [CN06] Antonin Chambolle and Matteo Novaga. Convergence of an algorithm for the anisotropic and crystalline mean curvature flow. *SIAM journal on mathematical analysis*, 37(6):1978–1987, 2006.
- [COP97] CARLOS Conca, A Osses, and JACQUES Planchard. Added mass and damping in fluid-structure interaction. *Computer methods in applied mechanics and engineering*, 146(3-4):387–405, 1997.
- [CRMP17] Aníbal Chicco-Ruiz, Pedro Morin, and M Sebastian Pauletti. The shape derivative of the gauss curvature. *arXiv preprint arXiv:1708.07440*, 2017.
- [DFG⁺06] Jian Du, Brian Fix, James Glimm, Xicheng Jia, Xiaolin Li, Yuanhua Li, and Lingling Wu. A simple package for front tracking. *Journal of Computational Physics*, 213(2):613–628, 2006.

- [DGC⁺13] Vincent Doyeux, Yann Guyot, Vincent Chabannes, Christophe Prud'Homme, and Mourad Ismail. Simulation of two-fluid flows using a finite element/level set method. application to bubbles and vesicle dynamics. *Journal of Computational and Applied Mathematics*, 246:251–259, 2013.
- [DGH82] Jean Donea, S Giuliani, and Jean-Pierre Halleux. An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. *Computer methods in applied mechanics and engineering*, 33(1-3):689–723, 1982.
- [DHPRF04] Jean Donea, Antonio Huerta, J-Ph Ponthot, and Antonio Rodríguez-Ferran. Arbitrary lagrangian-eulerian methods. *Encyclopedia of computational mechanics*, 2004.
- [DLL07] Qiang Du, Manlin Li, and Chun Liu. Analysis of a phase field navier-stokes vesicle-fluid interaction model. *Discrete and Continuous Dynamical Systems Series B*, 8(3):539, 2007.
- [DLRW05] Qiang Du, Chun Liu, Rolf Ryham, and Xiaoqiang Wang. A phase field formulation of the willmore problem. *Nonlinearity*, 18(3):1249, 2005.
- [DLW04] Qiang Du, Chun Liu, and Xiaoqiang Wang. A phase field approach in the numerical study of the elastic bending energy for vesicle membranes. *Journal of Computational Physics*, 198(2):450–468, 2004.
- [Doy14] Vincent Doyeux. *Modeling and simulation of multi-fluid systems. Applications to blood flows*. Theses, Université de Grenoble, January 2014.
- [DZ11] Michel C Delfour and J-P Zolâsio. *Shapes and geometries: metrics, analysis, differential calculus, and optimization*, volume 22. Siam, 2011.
- [EFFM02] Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational physics*, 183(1):83–116, 2002.
- [EO15] Selim Esedoglu and Felix Otto. Threshold dynamics for networks with arbitrary surface tensions. *Communications on pure and applied mathematics*, 68(5):808–864, 2015.
- [ERT08] Selim Esedoglu, Steven J Ruuth, and Richard Tsai. Threshold dynamics for high order geometric motions. *Interfaces and Free Boundaries*, 10(3):263–282, 2008.

- [ERT⁺10] Selim Esedoglu, Steven Ruuth, Richard Tsai, et al. Diffusion generated motion using signed distance functions. *Journal of Computational Physics*, 229(4):1017–1042, 2010.
- [Eva74] Evan A Evans. Bending resistance and chemically induced moments in membrane bilayers. *Biophysical journal*, 14(12):923–931, 1974.
- [Eva93] Lawrence C Evans. Convergence of an algorithm for mean curvature motion. *Indiana University mathematics journal*, pages 533–557, 1993.
- [FGG01] Charbel Farhat, Philippe Geuzaine, and Céline Grandmont. The discrete geometric conservation law and the nonlinear stability of ale schemes for the solution of flow problems on moving grids. *Journal of Computational Physics*, 174(2):669–694, 2001.
- [FGG06] Miguel A Fernández, Jean-Frédéric Gerbeau, and Céline Grandmont. A projection algorithm for fluid–structure interaction problems with strong added-mass effect. *Comptes Rendus Mathématique*, 342(4):279–284, 2006.
- [FGG07] Miguel Angel Fernández, J-F Gerbeau, and Céline Grandmont. A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid. *International Journal for Numerical Methods in Engineering*, 69(4):794–821, 2007.
- [FLV15] Miguel A Fernández, Mikel Landajuela, and Marina Vidrascu. Fully decoupled time-marching schemes for incompressible fluid/thin-walled structure interaction. *Journal of Computational Physics*, 297:156–181, 2015.
- [FMV15] Miguel A Fernández, Jimmy Mullaert, and Marina Vidrascu. Generalized robin–neumann explicit coupling schemes for incompressible fluid-structure interaction: Stability analysis and numerics. *International Journal for Numerical Methods in Engineering*, 101(3):199–229, 2015.
- [GF00] José Gomes and Olivier Faugeras. Reconciling distance functions and level sets. *Journal of Visual Communication and Image Representation*, 11(2):209–223, 2000.
- [GGG⁺98] James Glimm, Mary Jane Graham, John Grove, Xiao Lin Li, Todd Michael Smith, Dechun Tan, Folkert Tangerman, and Qiang Zhang. Front tracking in two and three dimensions. *Computers & Mathematics with Applications*, 35(7):1–11, 1998.

- [GGL⁺88] James Glimm, John Grove, Brent Lindquist, Oliver A McBryan, and Gretar Tryggvason. The bifurcation of tracked scalar waves. *SIAM Journal on Scientific and Statistical Computing*, 9(1):61–79, 1988.
- [GGL⁺98] James Glimm, John W Grove, Xiao Lin Li, Keh-ming Shyue, Yanni Zeng, and Qiang Zhang. Three-dimensional front tracking. *SIAM Journal on Scientific Computing*, 19(3):703–727, 1998.
- [GGM01] Céline Grandmont, Valérie Guimet, and Yvon Maday. Numerical analysis of some decoupling techniques for the approximation of the unsteady fluid structure interaction. *Mathematical Models and Methods in Applied Sciences*, 11(08):1349–1377, 2001.
- [GH08] Richards Grzhibovskis and Alexei Heintz. A convolution thresholding scheme for the willmore flow. *Interfaces and Free Boundaries*, 10(2):139–153, 2008.
- [GHS93] D Goldstein, R Handler, and L Sirovich. Modeling a no-slip flow boundary with an external force field. *Journal of Computational Physics*, 105(2):354–366, 1993.
- [GHS95] D Goldstein, R Handler, and L Sirovich. Direct numerical simulation of turbulent flow over a modeled riblet covered surface. *Journal of Fluid Mechanics*, 302:333–376, 1995.
- [GM77] Robert A Gingold and Joseph J Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181(3):375–389, 1977.
- [GPH⁺01] Roland Glowinski, Tsorng-Whay Pan, Todd I Hesla, Daniel D Joseph, and Jacques Periaux. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow. *Journal of Computational Physics*, 169(2):363–426, 2001.
- [GPHJ99] Roland Glowinski, T-W Pan, Todd I Hesla, and Daniel D Joseph. A distributed lagrange multiplier/fictitious domain method for particulate flows. *International Journal of Multiphase Flow*, 25(5):755–794, 1999.
- [GV07] Cédric Galusinski and Paul Vigneaux. Level-set method and stability condition for curvature-driven flows. *Comptes Rendus Mathématique*, 344(11):703–708, 2007.
- [GV08] Cédric Galusinski and Paul Vigneaux. On stability condition for bifluid flows with surface tension: Application to microfluidics. *Journal of Computational Physics*, 227(12):6140–6164, 2008.

- [HAC⁺06] MA Hoefler, MJ Ablowitz, I Coddington, Eric A Cornell, P Engels, and V Schweikhard. Dispersive and classical shock waves in bose-einstein condensates and gas dynamics. *Physical Review A*, 74(2):023623, 2006.
- [Hau02] Guillermo Hauke. A simple subgrid scale stabilized method for the advection–diffusion–reaction equation. *Computer Methods in Applied Mechanics and Engineering*, 191(27-28):2925–2947, 2002.
- [Hel73] Wolfgang Helfrich. Elastic properties of lipid bilayers: theory and possible experiments. *Zeitschrift für Naturforschung C*, 28(11-12):693–703, 1973.
- [Her99] Conyers Herring. Surface tension as a motivation for sintering. In *Fundamental Contributions to the Continuum Theory of Evolving Phase Interfaces in Solids*, pages 33–69. Springer, 1999.
- [HFH89] Thomas JR Hughes, Leopoldo P Franca, and Gregory M Hulbert. A new finite element formulation for computational fluid dynamics: Viii. the galerkin/least-squares method for advective-diffusive equations. *Computer methods in applied mechanics and engineering*, 73(2):173–189, 1989.
- [Hug87] Thomas JR Hughes. Recent progress in the development and understanding of supg methods with special reference to the compressible euler and navier-stokes equations. *International journal for numerical methods in fluids*, 7(11):1261–1275, 1987.
- [HW65] Francis H Harlow and J Eddie Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The physics of fluids*, 8(12):2182–2189, 1965.
- [Hym52] Morton A Hyman. Non-iterative numerical solution of boundary-value problems. *Applied Scientific Research, Section B*, 2(1):325–351, 1952.
- [Hym84] James M Hyman. Numerical methods for tracking interfaces. *Physica D: Nonlinear Phenomena*, 12(1-3):396–407, 1984.
- [ILL14] Mourad Ismail and Aline Lefebvre-Lepot. A necklace model for vesicles simulations in 2d. *International Journal for Numerical Methods in Fluids*, 76(11):835–854, 2014.
- [Ism04] Mourad Ismail. *Méthode de la frontière élargie pour la résolution de problèmes elliptiques dans des domaines perforés. Application aux écoulements fluides tridimensionnels*. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2004.

- [JBM19] Meriem Jedouaa, C-H Bruneau, and Emmanuel Maitre. An efficient interface capturing method for a large collection of interacting bodies immersed in a fluid. *Journal of Computational Physics*, 378:143–177, 2019.
- [JM08] D Jamet and C Misbah. Thermodynamically consistent picture of the phase-field model of vesicles: Elimination of the surface tension. *Physical Review E*, 78(4):041903, 2008.
- [JTB02] Didier Jamet, David Torres, and JU Brackbill. On the theory and computation of surface tension: the elimination of parasitic currents through energy conservation in the second-gradient method. *Journal of Computational Physics*, 182(1):262–276, 2002.
- [KBGD16] Bakytzhan Kallemov, Amneet Bhalla, Boyce Griffith, and Aleksandar Donev. An immersed boundary method for rigid bodies. *Communications in Applied Mathematics and Computational Science*, 11(1):79–141, 2016.
- [KEF11] Catherine Kublik, Selim Esedoglu, and Jeffrey A Fessler. Algorithms for area preserving flows. *SIAM Journal on Scientific Computing*, 33(5):2382–2401, 2011.
- [KHM11] Badr Kaoui, Jens Harting, and Chaouqi Misbah. Two-dimensional vesicle dynamics under shear flow: Effect of confinement. *Physical Review E*, 83(6):066319, 2011.
- [KLL⁺04] David Kinderlehrer, Jee Hyun Lee, Irene Livshits, Anthony D Rollett, and Shlomo Ta’asan. Mesoscale simulation of grain growth. In *Materials Science Forum*, volume 467, pages 1057–1062. Trans Tech Publ, 2004.
- [KM99] K Kassner and C Misbah. A phase-field approach for stress-induced instabilities. *EPL (Europhysics Letters)*, 46(2):217, 1999.
- [Kob93] Ryo Kobayashi. Modeling and numerical simulations of dendritic crystal growth. *Physica D: Nonlinear Phenomena*, 63(3-4):410–423, 1993.
- [Kou05] Petros Koumoutsakos. Multiscale flow simulations using particles. *Annu. Rev. Fluid Mech.*, 37:457–487, 2005.
- [KR98] Alain Karma and Wouter-Jan Rappel. Quantitative phase-field modeling of dendritic growth in two and three dimensions. *Physical review E*, 57(4):4323, 1998.

- [KWT88] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.
- [LC15] Injae Lee and Haecheon Choi. A discrete-forcing immersed boundary method for the fluid–structure interaction of an elastic slender body. *Journal of Computational Physics*, 280:529–546, 2015.
- [LO16] Tim Laux and Felix Otto. Convergence of the thresholding scheme for multi-phase mean-curvature flow. *Calculus of Variations and Partial Differential Equations*, 55(5):129, 2016.
- [LOC94] Xu-Dong Liu, Stanley Osher, and Tony Chan. Weighted essentially non-oscillatory schemes. *Journal of computational physics*, 115(1):200–212, 1994.
- [LS61] Ilya M Lifshitz and Vitaly V Slyozov. The kinetics of precipitation from supersaturated solid solutions. *Journal of physics and chemistry of solids*, 19(1-2):35–50, 1961.
- [LS95] Stephan Luckhaus and Thomas Sturzenhecker. Implicit time discretization for the mean curvature flow equation. *Calculus of variations and partial differential equations*, 3(2):253–271, 1995.
- [LS16] Tim Laux and Drew Swartz. Convergence of thresholding schemes incorporating bulk effects. *arXiv preprint arXiv:1601.02467*, 2016.
- [LSG95] Jun Liu, JB Schneider, and Jerry P Gollub. Three-dimensional instabilities of film flows. *Physics of Fluids*, 7(1):55–67, 1995.
- [LSM14] Aymen Laadhari, Pierre Saramito, and Chaouqi Misbah. Computing the dynamics of biomembranes by combining conservative level set and adaptive finite element methods. *Journal of Computational Physics*, 263:328–352, 2014.
- [Luc77] Leon B Lucy. A numerical approach to the testing of the fission hypothesis. *The astronomical journal*, 82:1013–1024, 1977.
- [LXS17] Lei Li, Xiaoqian Xu, and Saverio E Spagnolie. A locally gradient-preserving reinitialization for level set functions. *Journal of Scientific Computing*, 71(1):274–302, 2017.
- [Mas92] Pierre Mascarenhas. *Diffusion generated motion by mean curvature*. Department of Mathematics, University of California, Los Angeles, 1992.

- [Mau01] Bertrand Maury. A fat boundary method for the poisson problem in a domain with holes. *Journal of scientific computing*, 16(3):319–339, 2001.
- [Mau09] Bertrand Maury. Numerical analysis of a finite element/volume penalty method. *SIAM Journal on Numerical Analysis*, 47(2):1126–1148, 2009.
- [MBO92] Barry Merriman, James Kenyard Bence, and Stanley Osher. *Diffusion generated motion by mean curvature*. Department of Mathematics, University of California, Los Angeles, 1992.
- [MBO94] Barry Merriman, James K Bence, and Stanley J Osher. Motion of multiple junctions: A level set approach. *Journal of Computational Physics*, 112(2):334–363, 1994.
- [MCIP18] Thibaut Metivet, Vincent Chabannes, Mourad Ismail, and Christophe Prud’homme. High-order finite-element framework for the efficient simulation of multifluid flows. *Mathematics*, 6(10):203, 2018.
- [MGCM15] Chloe Mimeau, Federico Gallizio, Georges-Henri Cottet, and Iraj Mortazavi. Vortex penalization method for bluff body flows. *International Journal for Numerical Methods in Fluids*, 79(2):55–83, 2015.
- [MI05] Rajat Mittal and Gianluca Iaccarino. Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37:239–261, 2005.
- [MMC⁺09] Emmanuel Maitre, Thomas Milcent, Georges-Henri Cottet, Annie Raoult, and Yves Usson. Applications of level set methods in computational biophysics. *Mathematical and Computer Modelling*, 49(11-12):2161–2169, 2009.
- [MMG12] Anca Morar, Florica Moldoveanu, and Eduard Gröller. Image segmentation based on active contours without edges. In *2012 IEEE 8th International Conference on Intelligent Computer Communication and Processing*, pages 213–220. IEEE, 2012.
- [MMPR12] Emmanuel Maitre, Chaouqi Misbah, Philippe Peyla, and Annie Raoult. Comparison between advected-field and level-set methods in the study of vesicle dynamics. *Physica D: Nonlinear Phenomena*, 241(13):1146–1157, 2012.
- [MN14] Fernando C Marques and André Neves. Min-max theory and the willmore conjecture. *Annals of mathematics*, pages 683–782, 2014.

- [Mon94] Joe J Monaghan. Simulating free surface flows with sph. *Journal of computational physics*, 110(2):399–406, 1994.
- [MSS16] Luca Mugnai, Christian Seis, and Emanuele Spadaro. Global solutions to the volume-preserving mean-curvature flow. *Calculus of Variations and Partial Differential Equations*, 55(1):18, 2016.
- [MSV93] Ravikanth Malladi, James A Sethian, and Baba C Vemuri. Topology-independent shape modeling scheme. In *Geometric Methods in Computer Vision II*, volume 2031, pages 246–259. International Society for Optics and Photonics, 1993.
- [Mul56] William W Mullins. Two-dimensional motion of idealized grain boundaries. *Journal of Applied Physics*, 27(8):900–904, 1956.
- [MVA⁺06] M-A Mader, V Vitkova, M Abkarian, A Viallat, and T Podgorski. Dynamics of viscous vesicles in shear flow. *The European Physical Journal E*, 19(4):389–397, 2006.
- [OF01] Stanley Osher and Ronald P Fedkiw. Level set methods: an overview and some recent results. *Journal of Computational physics*, 169(2):463–502, 2001.
- [OF06] Stanley Osher and Ronald Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science & Business Media, 2006.
- [OS88] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.
- [Pat01] NA Patankar. A formulation for fast computations of rigid particulate flows. *Center for Turbulence Research Annual Research Briefs*, 2001:185–196, 2001.
- [PCD⁺12] Christophe Prud’Homme, Vincent Chabannes, Vincent Doyeux, Mourad Ismail, Abdoulaye Samake, and Gonçalo Pena. Feel++: A computational framework for galerkin methods and advanced numerical methods. In *ESAIM: Proceedings*, volume 38, pages 429–455. EDP Sciences, 2012.
- [Pes77] Charles S Peskin. Numerical analysis of blood flow in the heart. *Journal of computational physics*, 25(3):220–252, 1977.
- [Pes02] Charles S Peskin. The immersed boundary method. *Acta numerica*, 11:479–517, 2002.

- [PMO⁺99] Danping Peng, Barry Merriman, Stanley Osher, Hongkai Zhao, and Myungjoo Kang. A pde-based fast local level set method. *Journal of computational physics*, 155(2):410–438, 1999.
- [PPQ12] Gonçalo Pena, Christophe Prud’homme, and Alfio Quarteroni. High order methods for the approximation of the incompressible navier–stokes equations in a moving domain. *Computer Methods in Applied Mechanics and Engineering*, 209:197–211, 2012.
- [RBCK10] Diego Rossinelli, Michael Bergdorf, Georges-Henri Cottet, and Petros Koumoutsakos. Gpu accelerated simulations of bluff body flows using vortex particle methods. *Journal of Computational Physics*, 229(9):3316–3333, 2010.
- [RK98] William J Rider and Douglas B Kothe. Reconstructing volume tracking. *Journal of computational physics*, 141(2):112–152, 1998.
- [RR02] Yuriko Renardy and Michael Renardy. Prost: a parabolic reconstruction of surface tension for the volume-of-fluid method. *Journal of computational physics*, 183(2):400–421, 2002.
- [RS00] Giovanni Russo and Peter Smereka. A remark on computing distance functions. *Journal of Computational Physics*, 163(1):51–67, 2000.
- [Ruu98] Steven J Ruuth. Efficient algorithms for diffusion-generated motion by mean curvature. *Journal of Computational Physics*, 144(2):603–625, 1998.
- [RW03] Steven J Ruuth and Brian TR Wetton. A simple scheme for volume-preserving motion by mean curvature. *Journal of Scientific Computing*, 19(1-3):373–384, 2003.
- [Sau63] Valerij K Saul’ev. On the solution of some boundary value problems on high performance computers by fictitious domain method. *Siberian Math. J*, 4(4):912–925, 1963.
- [SB96] EM Saiki and S Biringen. Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method. *Journal of Computational Physics*, 123(2):450–465, 1996.
- [Set96] James A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.
- [Set99] James A Sethian. Fast marching methods. *SIAM review*, 41(2):199–235, 1999.

- [SF99] Mark Sussman and Emad Fatemi. An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM Journal on scientific computing*, 20(4):1165–1191, 1999.
- [SFSO98] Mark Sussman, Emad Fatemi, Peter Smereka, and Stanley Osher. An improved level set method for incompressible two-phase flows. *Computers & Fluids*, 27(5-6):663–680, 1998.
- [Smo05] Anton Smolianski. Finite-element/level-set/operator-splitting (felsos) approach for computing two-fluid unsteady flows with free moving interfaces. *International journal for numerical methods in fluids*, 48(3):231–269, 2005.
- [SP05] Nitin Sharma and Neelesh A Patankar. A fast computation technique for the direct numerical simulation of rigid particulate flows. *Journal of Computational Physics*, 205(2):439–457, 2005.
- [SSO94] Mark Sussman, Peter Smereka, and Stanley Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational physics*, 114(1):146–159, 1994.
- [Sus03] Mark Sussman. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics*, 187(1):110–136, 2003.
- [SY17] Drew Swartz and Nung Kwan Yip. Convergence of diffusion generated motion to motion by mean curvature. *Communications in Partial Differential Equations*, 42(10):1598–1643, 2017.
- [SZ03] Ruben Scardovelli and Stephane Zaleski. Interface reconstruction with least-square fit and split eulerian–lagrangian advection. *International Journal for Numerical Methods in Fluids*, 41(3):251–274, 2003.
- [TBE⁺01] Grétar Tryggvason, Bernard Bunner, Asghar Esmaeeli, Damir Juric, N Al-Rawahi, W Tauber, J Han, S Nas, and Y-J Jan. A front-tracking method for the computations of multiphase flow. *Journal of computational physics*, 169(2):708–759, 2001.
- [TT09] Hiroshi Terashima and Grétar Tryggvason. A front-tracking/ghost-fluid method for fluid interfaces in compressible flows. *Journal of Computational Physics*, 228(11):4012–4037, 2009.
- [UT92] Salih Ozen Unverdi and Grétar Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of computational physics*, 100(1):25–37, 1992.

- [Vig07] Paul Vigneaux. *Méthodes Level Set pour des problèmes d'interface en microfluidique*. PhD thesis, Université Sciences et Technologies-Bordeaux I, 2007.
- [VPM09] Petia M Vlahovska, Thomas Podgorski, and Chaouqi Misbah. Vesicles and red blood cells in flow: From individual dynamics to rheology. *Comptes Rendus Physique*, 10(8):775–789, 2009.
- [WCW⁺16] Zhi-Bin Wang, Rong Chen, Hong Wang, Qiang Liao, Xun Zhu, and Shu-Zhe Li. An overview of smoothed particle hydrodynamics for simulating multiphase flow. *Applied Mathematical Modelling*, 40(23-24):9625–9655, 2016.
- [WEO00] Fumihiro Wakai, Naoya Enomoto, and Hiroshi Ogawa. Three-dimensional microstructural evolution in ideal grain growth—general statistics. *Acta Materialia*, 48(6):1297–1311, 2000.
- [Wil96] Thomas James Willmore. *Riemannian geometry*. Oxford University Press, 1996.
- [YJL⁺06] Xiaofeng Yang, Ashley J James, John Lowengrub, Xiaoming Zheng, and Vittorio Cristini. An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids. *Journal of Computational Physics*, 217(2):364–394, 2006.
- [ZMOW98] Hong-Kai Zhao, Barry Merriman, Stanley Osher, and Lihe Wang. Capturing the behavior of bubbles and drops using the variational level set approach. *Journal of Computational Physics*, 143(2):495–518, 1998.