



HAL
open science

Identification de paramètres hydrodynamiques par simulation avec Smoothed Particle Hydrodynamics

Nicolas Gartner

► **To cite this version:**

Nicolas Gartner. Identification de paramètres hydrodynamiques par simulation avec Smoothed Particle Hydrodynamics. Automatique / Robotique. Université de Toulon, 2020. Français. NNT : 2020TOUL0004 . tel-03155087v2

HAL Id: tel-03155087

<https://hal.science/tel-03155087v2>

Submitted on 5 May 2021

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 / UNIVERSITÉ DE TOULON
pour le grade de **DOCTEUR en Automatique, Signal. ,
Productique, Robotique**

École doctorale 548 Mer et Science

présentée par

Nicolas GARTNER

préparée au laboratoire COSMER – EA 7398
COncception de Systèmes Mécaniques et Robotiques

Identification de paramètres hydrodynamiques par simulation avec Smoothed Particle Hydrodynamics

Thèse soutenue à Toulon, le 24 juin 2020,
devant le jury composé de :

Luc JAULIN

Professeur à l'ENSTA Bretagne / *Président*

Guillaume OGER

IR HDR à l'école Centrale de Nantes / *Rapporteur*

Frédéric BOYER

Professeur à l'Institut Mines Télécom Atlantique / *Rapporteur*

Vincent CREUZE

MCf-HDR à l'Université de Montpellier / *Examineur*

Claire DUNE

MCf à l'Université de Toulon / *Examinatrice*

Elisabeth MURISASCO

Professeur à l'Université de Toulon / *Examinatrice*

Carine ROC'H

DGA Techniques navales (invitée) / *Examinatrice*

Mathieu RICHIER

MCf à l'Université de Toulon (co-encadrant) / *Examineur*

Vincent HUGEL

Professeur à l'Université de Toulon / *Directeur de thèse*

Il y a déjà quelque temps que je me suis aperçu que, dès mes premières années, j'avais reçu quantité de fausses opinions pour véritables, et que ce que j'ai depuis fondé sur des principes si mal assurés, ne pouvait être que fort douteux et incertain ; de façon qu'il me fallait entreprendre sérieusement une fois en ma vie de me défaire de toutes les opinions que j'avais reçues jusques alors en ma créance, et commencer tout de nouveau dès les fondements, si je voulais établir quelque chose de ferme et de constant dans les sciences. Mais cette entreprise me semblant être fort grande, j'ai attendu que j'eusse atteint un âge qui fût si mûr, que je n'en pusse espérer d'autre après lui, auquel je fusse plus propre à l'exécuter ; ce qui m'a fait différer si longtemps, que désormais je croirais commettre une faute, si j'employais encore à délibérer le temps qu'il me reste pour agir.

par René Descartes, Les méditations métaphysiques

Remerciements

En premier lieu, je remercie naturellement Guillaume Oger, ingénieur de recherche à l'École Centrale Nantes ayant l'habilitation à diriger des recherches, et Frédéric Boyer, professeur des universités à l'Institut Mines Télécom Atlantique, d'avoir accepté de rapporter cette thèse.

Je remercie également les membres de mon jury, Luc Jaulin, professeur à l'École Nationale Supérieure des Techniques Avancées de Bretagne, Vincent Creuze, maître de conférences à l'université de Montpellier, Claire Dune, maître de conférences à l'université de Toulon, Elisabeth Murisasco, professeur à l'université de Toulon et Carine Roc'h, travaillant à la Direction Générale de l'Armement (DGA), d'avoir accepté d'assister à ma soutenance de thèse et pour leur intérêt dans mon travail de recherche.

Je voudrais également remercier la Direction Générale de l'Armement d'avoir permis l'existence du projet SURFBOT, qui regroupait l'Université de Toulon avec deux entreprises de la région Sud - Provence Alpes Côtes d'Azur, Subsea Tech, une entreprise basée à Marseille et la division Robopec d'IXblue (anciennement Robopec) basée à Six-Fours-les-Plages, et d'avoir ainsi soutenu financièrement mes travaux de recherche. Je remercie toutes les personnes avec qui j'ai pu échanger dans le cadre de ce projet, Yves Chardard, directeur général de Subsea Tech, Pierre Mary, anciennement directeur projets et innovations, Guillaume Russac, ingénieur chez Subsea Tech, Christophe Rousset, anciennement Directeur Général de Robopec, Adrien Barral, ingénieur chez Robopec, Frédéric Shom, de la DGA et Carine Roc'h. Je tiens particulièrement à remercier Bertrand Caré, docteur-ingénieur chez Robopec, pour sa participation au suivi de ma thèse, l'intérêt qu'il a porté à mes travaux et ses conseils.

Je remercie mon encadrant, Mathieu Richier, pour le suivi qu'il a apporté à mes travaux et pour son enseignement de l'art de la présentation, qu'elle soit écrite ou orale. La manière dont il a parfois remis mes travaux en question a pu permettre au sortir de long débats d'avancer dans mes recherches ou d'être plus précis sur certaines notions évoquées de manière floue. Je remercie Vincent Hugel, mon directeur de thèse, pour tout le temps qu'il m'a consacré malgré son emploi du temps que je savais très chargé, pour son analyse attentive de mes travaux et ses retours détaillés. Les discussions que nous avons pu avoir ont toujours été très riches et constructives. Je les remercie tous deux pour l'autonomie qu'ils m'ont accordée dans la direction qu'ont pris mes recherches, qui s'éloigne un peu du domaine de la robotique, qui est leur domaine de prédilection.

Je remercie Niels Montanari, Ingénieur chez Centroid Lab et Ramprasad Sampath, Directeur Général de Centroid Lab, pour leur soutien scientifique et technique qui a grandement participé aux avancées de mes travaux de recherche. Je les remercie également pour leur participation dans la rédaction des publications présentant les résultats de mes recherches. Je remercie tout particulièrement Niels Montanari qui m'a consacré un temps considérable, alors que rien ne le tenait de le faire.

Je remercie tous les membres du laboratoire COSMER, pour leur accueil chaleureux et les

moments que nous avons pu passer ensemble, notamment lors des repas de fin d'année, les galettes des rois, les pots pour chaque occasion qu'il y avait à célébrer, etc. Un laboratoire de bons vivants, en somme. Je remercie en particulier Claire Dune pour l'intérêt qu'elle a porté à ma thèse et le temps qu'elle a consacré à son suivi.

Merci à mes collègues doctorants du laboratoire IMATH et COSMER, avec qui j'ai pu partager mon espace de travail : Jérôme, Matheus, Mohamed Ali, Houssam, Moustoifa, Hadjer, Karol, Pepa, Asma, Yssouf, Hoang Anh, Ornella, Jean-Baptiste, Vladyslav et Manon. Je m'excuse par avance, si j'ai oublié quelqu'un. Votre bonne humeur et nos discussions m'ont beaucoup apportées et vous avez contribué ainsi à mes travaux. Merci à Matheus, Houssam, Hoang Anh, Jean-Baptiste, Ornella, Manon, Vlad et tous les sportifs pour les parties intenses de ping-pong ou de football (qui m'avait coûté une cheville). « 14! » Je n'en dirais pas plus pour Ornella, Manon, Jean-Baptiste, Pepa et Hoang Anh. Merci Hoang Anh de m'avoir fait un peu découvrir le Vietnam à travers tes plats et nos discussions, en espérant pouvoir y aller faire un tour un jour prochain. Merci Jean-Baptiste pour toutes tes idées fédératrices. Merci Ornella pour ta bonne volonté. Merci Manon pour ta sympathie.

Je remercie également ma famille, mes parents, frères et soeurs, tantes et oncles, pour leur soutien infailible pendant ces trois années. Je remercie également mes beaux-parents, Wim et Chantal, qui ont aidé à rendre ces années de thèse moins difficiles.

Je remercie également ceux qui m'ont permis de passer de belles soirées de jeux durant mes week-ends. Merci à François, Oum, Bénédicte, Émile et Simon.

Je tiens particulièrement à remercier, celles sans qui la qualité grammaticale et orthographique de ce document ne serait pas celle qu'elle est aujourd'hui, Chantal et Anna, qui ont eu la patience de relire (plus d'une fois) chacune de ces pages.

Je tiens à remercier plus encore Anna, ma femme, qui a su me supporter pendant ces années. Je tiens à dédier ce travail à notre fils, mon fils, Sacha.

Table des matières

Table des matières	1
1 Introduction générale	5
1.1 Contexte scientifique et industriel	7
1.2 Organisation du mémoire	9
1.3 Généralités sur les véhicules sous-marins et leur environnement	10
1.4 Thématique et axes de recherche	25
1.5 Environnements de simulation : application aux véhicules marins	26
1.6 Simulation de l'interaction dynamique d'un véhicule sous-marin et de l'eau qui l'entoure	37
1.7 Approche développée dans ce mémoire	43
2 Modélisation et estimation des paramètres hydrodynamiques	45
2.1 Modèle analytique des véhicules sous-marins	46
2.2 Description et état de l'art des forces hydrodynamiques	54
2.3 Méthodes d'identification des paramètres hydrodynamiques	64
2.4 Identification expérimentale des paramètres hydrodynamiques	71
2.5 Analyse des résultats obtenus avec l'expérience du balancier	79
2.6 Conclusion	85
3 Simulation hydrodynamique basée SPH	87
3.1 Introduction à la méthode Smoothed Particle Hydrodynamics (SPH)	88
3.2 Interpolation et discrétisation spatiale SPH	93
3.3 Modèles d'écoulement	101
3.4 Validation de la méthode d'interaction fluide-solide	111
3.5 Validation de l'algorithme ISPH	118
3.6 Conclusion du chapitre	123
4 Résultats sur la dynamique des solides immergés avec ISPH	125
4.1 Poussée d'Archimède	126
4.2 Présentation du principe expérimental	129
4.3 Estimation de la masse ajoutée d'une sphère	135
4.4 Estimation des forces de frottement	146
4.5 Discussions	157
4.6 Conclusion du chapitre	164

5 Conclusion générale	165
5.1 Bilan des contributions apportées	166
5.2 Perspectives	169
Annexes	174
A Modélisation cinématique des véhicules sous-marins	175
A.1 Référentiels	175
A.2 Transformation entre VEH et ENU	177
A.3 Transformation entre ENU et ECEF	179
A.4 Transformation entre VEH et COUR	180
B Environnements de simulations et leurs composants principaux	183
B.1 Composants des environnements de simulations	184
B.2 Présentation de plusieurs environnements de simulations existants	188
C Méthodes d'intégration numérique	199
C.1 Méthode d'intégration explicite/progressive	200
C.2 Méthode d'intégration implicite/rétrograde	209
C.3 Conclusion : Méthode explicite à pas variable ou méthode implicite à pas fixe ?	213
C.4 Adaptation dans les moteurs physiques des jeux vidéos	214
D Conception d'un balancier hydrodynamique	217
D.1 Dispositif mécanique	218
D.2 Acquisition des données	220
E Algorithme SPH	221
E.1 Algorithme WCSPH explicite	221
E.2 Algorithme ISPH semi-implicite	222
F Le problème « Tensile Instability » et ISPH	225
F.1 Le problème d'instabilité d'attraction	225
F.2 Tensile Instability Control et ISPH	227
G Équation du mouvement d'un véhicule marin à un point d'origine arbitraire	229
H Calcul de masse ajoutée pour une sphère	231
I Calcul des B-Spline pour le schéma SPH	237
I.1 B-Spline cubique	237
I.2 B-Spline quintique	239
J Méthodes d'implémentations des algorithmes	241
J.1 Organisation des classes	241
J.2 Méthode de stockage de données et de compilation	243
Glossaire	246

<i>Table des matières</i>	3
Bibliographie	246
Table des figures	267

Chapitre 1

Introduction générale

Très tôt, l'Homme, par son goût irréprouvable pour l'aventure, s'est employé à trouver des moyens pour se déplacer sur l'eau. Ainsi, il y a très longtemps, il a inventé le navire. Tellement longtemps qu'il est difficile de le dater. La datation carbone du plus ancien navire que l'on a retrouvé : la Pirogue de Pesse, estime qu'il date d'entre 8040 av. J.-C. et 7510 av. J.-C. Cependant, d'autres navires datant du même millénaire ont été trouvés en Afrique ou en Asie, ce qui laisse penser que ces navires étaient communs. Ces navires lui ont permis, tout d'abord, de voyager sur des lacs et de pêcher. Ils se révélèrent également particulièrement pratiques pour le transport, et notamment le transport de charges lourdes, comme des pierres, comme en atteste la barque monoxyde¹ de Hanson. Il s'agit d'une barque datant de 3500 ans avant notre ère qui a été retrouvée avec son chargement. En faisant évoluer les navires de rivière et de lac plus calme, l'Homme créa des navires plus stables capables d'affronter les vagues et de naviguer dans les zones côtières. Puis, après de longues recherches en astronomie et en optique, il put enfin aller au-devant de la haute mer. En 1492 après J.-C., un certain Christophe Colomb fera le voyage de l'Europe vers l'Amérique en traversant l'océan Atlantique grâce à un navire. Entre les pirogues primitives monoxydes et les porte-avions d'aujourd'hui, beaucoup de progrès ont dû être apportés pour permettre aux navires d'être plus grands, plus résistants, plus rapides, etc.

Après avoir maîtrisé la navigation sur l'eau, l'Homme a cherché aussi à naviguer sous l'eau. Les sous-marins ont été imaginés par Margaret Cavendish dans son roman du XVII^e siècle « The Description of a New World, called The Blazing World », où un vaisseau d'or est tiré par des hommes-poissons. Entre 1620 et 1850, les premiers engins submersibles ont vu le jour. Il s'agissait principalement de cloches de plongée, qui permettaient d'amener des hommes dans les profondeurs et de les remonter à l'aide de systèmes de ballast. Cependant celles-ci ne permettaient pas de naviguer sous l'eau. L'un des premiers sous-marins modernes, capable de naviguer sous l'eau, est le « Nautilus » de l'américain Robert Fulton, qui fit, d'ailleurs, une plongée dans la Seine à Rouen en juillet 1800. Fulton proposait alors aux consuls, dont Napoléon Bonaparte, de s'en servir comme d'une arme contre la marine anglaise. Mais la très faible vitesse du sous-marin en faisait un piètre vaisseau d'attaque s'il s'agissait de poursuivre les navires anglais en fuite. Jules Verne, dans son roman de 1869 « Vingt mille lieues sous les mers », avec le fameux capitaine « Nemo », s'en inspira probablement pour nommer son vaisseau. C'est la première fois qu'un sous-marin est décrit en tant que machine dans une œuvre de fiction. Jules Vernes s'inspira également d'ailleurs du premier sous-marin

1. c'est à dire constituée d'un seul tronç

propulsé par un moteur à air comprimé. Il fut conçu par deux Français : le commandant Siméon Bourgois et l'ingénieur Charles Brun. Ce navire, « Le Plongeur », vu le jour en 1863 et était capable d'embarquer un équipage de sept personnes, même si son autonomie et sa vitesse étaient faibles. Son plus grand défaut était sa grande instabilité. En effet, la navigation sur l'eau est déjà assez complexe, il faut faire face au courant, vent, vagues, etc., mais au moins les navires ne quittent pas la surface de l'eau. Les sous-marins, eux, naviguent en trois dimensions. Pour flotter et naviguer à une profondeur précise, ils doivent faire face aux effets de l'eau présente tout autour. Avec le développement de la mécatronique et de la robotique, de nouveaux véhicules ont vu le jour : les drones de surface, couramment appelés USV « Unmanned Surface Vehicle », les drones sous-marins téléopérés, couramment appelés ROV « Remotely Operated Vehicle » et les drones sous-marins autonomes, couramment appelés AUV « Autonomous Underwater Vehicle ». Ces véhicules ne transportent pas de passager, sont généralement de taille réduite et sont principalement à usage des industries pétrolières, de la recherche en océanographie ou de l'armée. Ces petits véhicules pourraient être fort utiles pour évoluer à proximité des plages, leur taille en fait de très bons candidats. Près de la plage, les vagues et les courants peuvent être très forts, et les obstacles, les rochers, sont très proches. C'est donc un environnement dans lequel il est difficile de naviguer, surtout que plus un navire est petit, plus il subit les effets de la mer. Aujourd'hui, aucun véhicule n'est capable de naviguer de manière autonome dans cette zone, que l'on nomme zone de surf. Dans l'optique de préparer un premier prototype et de faire de premiers essais, il faut permettre une simulation du comportement d'un véhicule dans cette zone. La simulation est une technique qui permet de réduire les coûts lors de la conception de système. Ainsi, l'objet de cette thèse est le suivant : simuler le comportement dynamique d'un véhicule marin dans un milieu aquatique réaliste.

1.1 Contexte scientifique et industriel

Les développements scientifiques conduits dans cette thèse font partie d'un projet, nommé SURFBOT, financé par la Direction Générale de l'Armement (DGA) dans le cadre d'un dispositif « Régime d'Appui pour l'Innovation Duale » (RAPID). Il scelle une collaboration entre différents acteurs de la région Sud, nouvelle appellation de la région Provence-Alpes-Côte d'Azur (PACA) : SUBSEA-TECH, ROBOPEC et le laboratoire COSMER. SUBSEA-TECH est une entreprise basée à Marseille spécialisée dans les solutions d'inspection et d'observation des milieux marins et sous-marins. ROBOPEC est une entreprise basée à Six-Four-les-Plages spécialisée dans le développement de logiciel robotique. Le laboratoire COncEption de Systèmes Mécaniques et Robotiques (COSMER) dépend de l'Université de Toulon et se situe sur le campus de La Garde. L'objectif de ce projet est de développer des techniques de reconnaissance et de cartographie automatiques de fonds marins et de zones côtières adaptées à un robot sous-marin léger et peu coûteux. L'idée de ce projet est donc de concevoir les premières briques logicielles adaptées à un robot sous-marin portable, à longue excursion, c'est à dire, des excursions de plusieurs centaines de mètres. La mission de ce véhicule sous-marin consistera à réaliser la cartographie d'une zone sous-marine proche d'un point d'intérêt vis-à-vis d'une opération amphibie. Ce robot sous-marin devra être capable de réaliser ces tâches de façon autonome. Pour atteindre ces objectifs, il faudra donc mettre en œuvre des moyens expérimentaux permettant de valider ces briques logicielles avec notamment la conception d'un simulateur innovant et la conception d'une plateforme d'acquisition et de tests. La tâche confiée au laboratoire COSMER est la réalisation du simulateur.

Cette plateforme de simulation devra permettre d'accélérer le prototypage du système envisagé et devra donc nécessairement être constitué d'un environnement de simulation suffisamment réaliste pour permettre d'évaluer le système et ses opérations, et notamment ses opérations dans les zones côtières. La plupart des simulateurs existants sont principalement dédiés à la haute mer puisque leurs principaux utilisateurs sont les industries pétrolières. Les contraintes liées aux zones côtières sont également plus difficiles à reproduire en simulation que celles de la haute mer. C'est toute la difficulté de la réalisation de ce simulateur : il faudra qu'il soit capable de modéliser les courants, les vagues et leurs effets sur le véhicule. En parallèle, ce simulateur doit également permettre de simuler les acquisitions des capteurs embarqués par le robot, afin de valider les algorithmes de traitement des données et les lois de commandes envisagées pour le robot. Il faudra donc qu'il soit capable de modéliser toutes les perturbations visuelles et dynamiques engendrées par l'environnement du véhicule sous-marin.

Après l'étude des techniques existantes de simulation, il apparaît qu'une des méthodes les plus intéressantes pour atteindre notre objectif est celle nommée « Smoothed Particle Hydrodynamics » (SPH). Elle utilise un système de particules qui modélise le fluide et permet de reproduire son comportement. SPH est une méthode particulièrement récente et novatrice pour simuler la dynamique du fluide, comme cela sera présenté au chapitre 3. Plusieurs logiciels ou applications utilisant cette méthode sont déjà disponibles sur le marché, notamment pour la réalisation d'animation en trois dimensions d'écoulement de fluide pour les jeux vidéo ou le cinéma. Cette méthode est très pratique car de nombreux calculs peuvent être faits indépendamment les uns des autres, ce qui permet de les réaliser en parallèle. Etant donné le développement des processeurs graphiques parallèles, de premières expérimentations ont été menées avec les solutions de simulation de la marque Nvidia. Cette marque développe des logiciels spécifiques optimisés pour l'utilisation de leurs cartes graphiques et qui sont principalement conçus pour l'animation. De ce fait, le réalisme physique n'était pas leur

première préoccupation voir section 1.5.3, et une solution non issue de l'animation 3D a dû être recherchée. À travers le groupe de recherche international SPHERIC (Smoothed Particle Hydrodynamics Research and Engineering International Community), qui est une organisation visant à rassembler et favoriser les collaborations entre utilisateurs, académiques comme industriels, de la méthode SPH, des échanges ont pu se nouer avec l'entreprise Centroid Lab, petite entreprise comptant moins de 10 employés, basée à Los Angeles. Elle propose ses services pour la simulation de comportement de fluide et développe un logiciel baptisé « Neutrino », qui permet de réaliser des simulations physiquement réalistes avec SPH et qui présente une interface simple à appréhender. L'interface de Neutrino ressemble d'ailleurs à celui de : « Gazebo », l'environnement de simulation robotique le plus courant. Un accord a été signé avec cette entreprise en décembre 2017. Il prévoit un accès au code source pour l'Université de Toulon et l'équipe de recherche du laboratoire COSMER, ce qui permet de travailler directement sur l'implémentation de l'interaction entre le fluide simulé avec la méthode SPH et le solide. C'est notamment grâce à cette collaboration que la plupart des développements scientifiques de cette thèse ont pu être menés.

1.2 Organisation du mémoire

Pour répondre à la problématique du développement d'un simulateur innovant capable de modéliser le comportement de l'eau dans les zones côtières, ainsi que l'interaction entre le robot et l'eau, le mémoire est organisé suivant quatre grands axes :

- Les différentes solutions de véhicules marins sont présentées dans la suite de ce chapitre et les dénominations des différentes zones marines dans lesquels ils sont susceptibles d'évoluer sont détaillées. Cela permet de comprendre précisément les contraintes pour la réalisation de l'environnement de simulation. De ce fait, les différents environnements de simulation existants sont présentés. Leur fonctionnement et leurs limitations par rapport à la zone de surf sont précisés. Une explication est apportée sur la nécessité d'adapter l'existant, ce qui constitue le point de départ de nos travaux de recherche. Enfin, un état de l'art des différentes méthodes de simulation pouvant répondre à la problématique est exposé et débouche sur deux méthodes de modélisation : une modélisation hydrodynamique macroscopique de la dynamique du véhicule, et une modélisation hydrodynamique microscopique/mésoscopique avec la méthode SPH.
- Pour comprendre les différentes forces agissant sur le solide, une modélisation dynamique/hydrodynamique macroscopique du véhicule marin est présentée. Les efforts hydrodynamiques mis en jeu pendant la navigation du véhicule marin sont détaillés et leur origine expliquée, ces efforts étant reliés à différents paramètres, que l'on nommera paramètres hydrodynamiques (cf. Section 2.2). Un état de l'art sur les méthodes d'obtention de paramètres hydrodynamiques est présenté ainsi qu'une méthode expérimentale à moindre coût développée au cours de cette thèse. Cette dernière servira d'expérience témoin pour la validation du simulateur utilisant la méthode SPH.
- Cette méthode SPH est ensuite présentée, et en particulier son évolution ainsi que différentes sous-méthodes sont présentés, selon la manière dont les équations du fluide sont résolues et selon différentes hypothèses. Deux méthodes de résolution du fluide ainsi que deux méthodes d'interaction entre le fluide et le solide qui semblent particulièrement pertinentes pour notre application sont explicitées. Puis des expériences permettant de valider un algorithme capable de déterminer le comportement de l'eau et de gérer l'interaction entre le fluide et le solide dans le cas de déplacements sont proposées.
- Le dernier axe de recherche est la confrontation entre un solide dynamique et le comportement du fluide résolu avec la méthode SPH. Le réalisme physique du comportement du solide dynamique est étudié, et, a minima, un comportement homothétique à la réalité est attendu, en particulier la capacité d'un objet à atteindre une vitesse limite due aux frottements. Les paramètres hydrodynamiques obtenus à l'aide de la simulation SPH sont comparés à ceux obtenus lors des expériences avec le balancier et à celles disponibles dans la littérature scientifique.

L'ensemble de ces développements permet, finalement, de confirmer la faisabilité d'utiliser la méthode SPH pour simuler une mission d'un robot autonome avec une loi de commande dynamique qui évolue dans un environnement aquatique réaliste.

1.3 Généralités sur les véhicules sous-marins et leur environnement

Cette section présente les véhicules sous-marins et leur environnement afin de déterminer les éléments à prendre en compte pour la réalisation du simulateur du comportement dynamique des véhicules sous-marins.

1.3.1 Les véhicules sous-marins

Les véhicules sous-marins sont des véhicules capables d'évoluer librement non seulement à la surface mais également sous l'eau. Ces véhicules se distinguent des navires, qui sont uniquement capables de se déplacer sur l'eau et des bathyscaphes qui se déplacent principalement selon l'axe vertical, dont le principe est représenté sur la figure 1.1. Ces véhicules sous-marins se

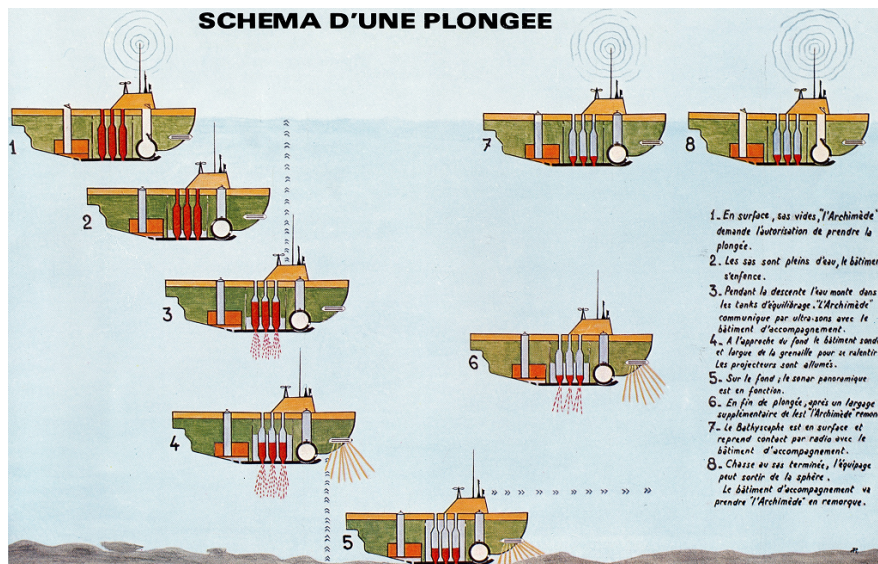


Figure 1.1 – Schéma du fonctionnement d'un bathyscaphe (Source : Georges Houot)

répartissent en trois catégories : les sous-marins habités, que l'on appelle le plus souvent simplement sous-marins et dans lesquels un équipage peut monter à bord, les sous-marins téléopérés, qui sont généralement appelés « Remotely Operated Vehicles » (ROV) et les sous-marins autonomes, qui sont généralement appelés « Autonomous Underwater Vehicle » (AUV). Le principe de ces véhicules sous-marins repose, pour la majorité d'entre eux, sur l'utilisation de ballasts, qui sont des poches remplies d'air lorsque le véhicule doit rester à la surface ou remplies d'eau lorsque le véhicule doit naviguer sous l'eau.

1.3.1.1 Les sous-marins

Les sous-marins habités ne sont pas directement l'objet des recherches présentées ici mais elles pourraient très bien s'y appliquer. Par exemple, les recherches de ce mémoire pourraient permettre d'étudier leur manœuvrabilité. Les sous-marins sont principalement des navires de guerre, comme illustré sur la figure 1.2. Ces navires de guerres sont généralement

de grande taille et embarquent des équipages de plusieurs dizaines de personnes. Les sous-marins sont également utilisés dans le domaine civil par l'industrie pétrolière et la recherche océanographique. De manière plus anecdotique, ils sont utilisés à des fins de transport commercial² et à des fins touristiques.

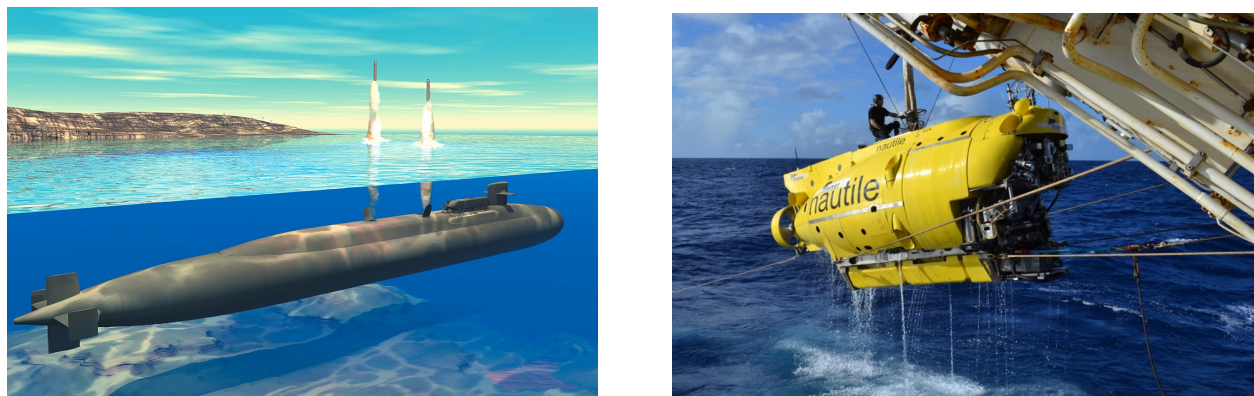


Figure 1.2 – Gauche : Vue d'artiste d'un sous-marin de classe Ohio (américain) en train de lancer des missiles Tomahawk. (Source : United States Navy)
Droite : Récupération du sous-marin Nautilus de l'Ifremer (Source : Ifremer)

Le contexte militaire conduit à un développement acharné d'améliorations techniques et technologiques pour l'adaptation des découvertes les plus récentes à ces navires (batteries plus performantes, piles à combustible, oxygène liquide, réserves d'hydrogène, propulsion nucléaire...). Le coût d'investissement colossal représenté par ces développements limite à un cercle très restreint le nombre de nations concernées. Ce cercle est d'autant plus restreint pour les nations capables de réaliser des plongées de longue durée, avec des sous-marins dit anaérobies, c'est à dire qui ne dépendent pas de l'air extérieur. Ces sous-marins sont généralement capables d'atteindre des profondeurs de plusieurs centaines de mètres, en moyenne 300/400 mètres et des vitesses entre 25 et 30 nœuds (soit entre 46 et 55 km/h), le record de vitesse étant établi par le sous-marin nucléaire russe, code OTAN Classe Papa, à 44.7 nœuds soit 83 km/h. Ces sous-marins sont des atouts stratégiques majeurs, puisqu'ils permettent de créer une menace d'attaque nucléaire mobile capable d'atteindre n'importe quel point sur la surface de la Terre. Certains de ces sous-marins sont conçus pour embarquer des missiles d'une portée de 9000 km, typiquement des missiles M51.

Les sous-marins de recherche océanographique sont utilisés principalement pour l'observation, la recherche et la collecte d'échantillons, les mesures ou pour des missions archéologiques spécifiques, sur des épaves notamment. Parmi ces sous-marins, on peut citer le Nautilus de l'Ifremer représenté sur la figure 1.2, qui est un sous-marin pour l'exploration des grands fonds. Ce sous-marin est capable de descendre jusqu'à 6000 m de profondeur, ce qui est beaucoup plus que les sous-marins militaires et ce qui permet d'explorer 97% des fonds sur la surface de la terre. La grande difficulté pour ces grandes profondeurs étant les fortes pressions auxquelles doivent résister les équipements et la coque qui embarque l'équipage. Sur la figure 1.3, on peut voir une sphère de métal qui est l'habitacle dans lequel est embarqué l'équipage du Nautilus. Le choix de cette forme de sphère correspond à la forme la plus résistante à la pression. Par sécurité, cet habitacle a été conçu pour résister à 900 bars de pression, et pourrait donc

2. Par l'Allemagne au cours de la seconde guerre mondiale

théoriquement plonger jusqu'à 9000 *m*. Pour résister à de telles pressions et également ne pas rendre le véhicule trop lourd, la sphère est réalisée entièrement en titane (TA6 V4). On peut également remarquer que ce sous-marin possède deux bras articulés à l'avant qui lui permettent de manipuler des objets au fond. Il possède un propulseur unique à l'arrière qu'il peut orienter à la manière des nageoires caudales des poissons. Son enveloppe extérieure, construite autour de la sphère, possède d'ailleurs une forme similaire.

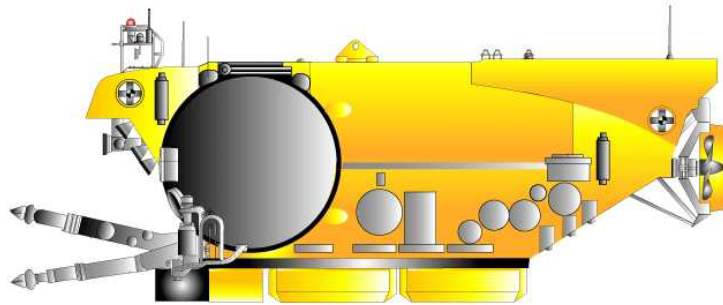


Figure 1.3 – Schéma du sous-marin Nautil de l'Ifremer (Source : Ifremer)

Les sous-marins destinés à l'industrie pétrolière sont utilisés pour permettre la maintenance, l'observation et la prospection du fond. Avec le développement de bras articulés, de nombreuses tâches peuvent être réalisées aujourd'hui avec un véhicule plutôt qu'avec des plongeurs ou des bathyscaphes. Ces tâches sont aujourd'hui partagées avec les véhicules non habités, notamment avec les ROV qui sont beaucoup plus adaptables et performants, car potentiellement plus légers et plus petits. De plus, l'absence de personnel à son bord limite les risques d'accidents. Il faut aussi noter que ces véhicules sous-marins avec habitacle sont généralement extrêmement chers comparés aux véhicules sans habitacle présentés ci-dessous.

1.3.1.2 Marché des véhicules sous-marins non habités

Les véhicules sous-marins non habités, aussi appelés « Unmanned Underwater Vehicles » (UUV), représentent aujourd'hui un marché significatif. Ce marché est estimé à 2.9 milliards de dollars en 2018, dont 430 millions pour les AUV et le reste pour les ROV. (Source : Hydro international, Zion market research). Cette répartition est illustrée sur la figure 1.4. En 1999,

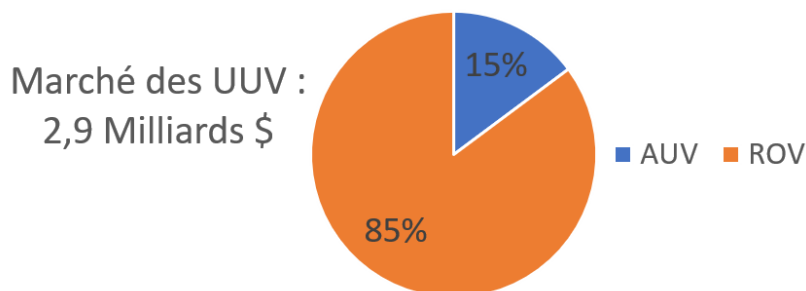


Figure 1.4 – Marché des véhicules sous-marins en 2018

on estimait le nombre de ROV commercialisés autour de 3000 ; il n’y avait alors aucun AUV, et le marché des UUV représentait seulement 183 millions de dollars [WPR01]. Comme on peut le voir, ce marché a subi un énorme boom économique. Il est toujours en pleine expansion au vu des dernières avancées technologiques et de l’intérêt grandissant pour l’exploration et l’exploitation des fonds marins, ainsi que de l’avantage stratégique militaire que les véhicules sous-marins apportent. Comme expliqué précédemment, les deux principaux acteurs du marché des véhicules sous-marins sont historiquement l’industrie pétrolière et l’armement, mais ce marché est en train de s’ouvrir aux loisirs et à d’autres domaines. Le marché des véhicules sous-marins non habités ne représente qu’une infime partie du marché global de l’industrie océanique qui représentait en 2015 1500 milliards de dollars et qui devrait doubler d’ici 2030 (Source : OCDE). Il est possible que le marché des UUV, en pleine expansion, croisse encore plus vite que le marché global, il pourrait donc tripler voire quadrupler entre 2015 et 2030 !

1.3.1.3 ROV : véhicule sous-marin téléopéré

Les ROV sont des véhicules sous-marins contrôlés à distance, reliés par un câble à un opérateur. Ces drones offrent de nombreuses possibilités car ils peuvent être équipés de capteurs et de bras. Notamment, ils sont susceptibles d’évoluer dans des environnements dans lesquels des plongeurs ou des véhicules sous-marins habités plus volumineux en sont incapables. En effet, vu qu’ils n’accueillent pas d’équipage, ils sont capables d’aller dans des environnements extrêmes : eau chaude, froide, polluée (pollution chimique, bactériologique, eau radioactive, etc.), et peuvent subir de fortes pressions. Ainsi ils sont destinés à de nombreuses applications que les sous-marins classiques sont incapables de réaliser : navigation dans des conduites, des égouts, observation et maintenance sur des installations hydroélectriques, côtières, nucléaires ou offshore, exploration de grottes, d’épaves, etc. Ils sont par exemple utilisés pour l’observation de l’état des structures de béton sur les barrages, avec capture d’images acoustiques et télévisuelles pour détecter la présence de défauts, comme par exemple du faïençage³ comme le montre l’article de Girard [GTB⁺18]. La plupart des ROV sont constitués des éléments suivants :

- un système d’actionneurs constitué de propulseurs qui permettent au ROV de se déplacer sous l’eau, de ballasts permettant de gérer la profondeur de plongée du ROV et son équilibre, et éventuellement de bras articulés offrant la possibilité de réaliser des manipulations.
- un système d’alimentation et/ou de stockage de l’énergie : Le ROV peut être relié par un câble et ne nécessite alors aucune énergie stockée. Toutefois, il peut embarquer une batterie.
- un système de communication, via le câble d’alimentation le plus souvent. Effectivement, sous l’eau, la communication sans fil est compliquée à mettre en place⁴. Actuellement, on communique par onde sonore, à l’aide de systèmes appelé « pinger » ou de modems

3. Réseau maillé de fissures superficielles apparaissant à la surface d’un matériau à la suite d’un phénomène de retrait

4. Les ondes électromagnétiques ou ondes radio ont une très faible pénétration dans l’eau. Cette pénétration est d’autant plus faible que la fréquence des ondes est haute. Or plus la fréquence est basse et plus le débit d’informations que l’on peut transmettre est faible. Le son se propage très bien dans l’eau, mais il est difficile de transmettre en haut débit avec des ondes acoustiques et le même problème se pose alors. Par exemple, avec un simple hydrophone et un haut-parleur sous-marin, il paraît difficile de transmettre les données images d’une caméra haute définition pour un affichage en temps réel. De plus, cela crée une forte pollution sonore, ce qui peut perturber la faune.

acoustiques. On ne peut ainsi communiquer qu'avec un débit très réduit d'informations. Il est par exemple impossible de transmettre des données images en temps réel. La transmission d'information sans fil sous l'eau est actuellement un sujet de recherche actif.

- un "cerveau" électronique, constitué d'au moins une carte micro-processeur ou processeur, il permet de communiquer avec la surface et de commander les différents moteurs, capteurs et autres actionneurs.
- des capteurs, le plus souvent au moins une caméra, mais également une centrale inertielle, des gyromètres/accéléromètres, des capteurs de position GPS, des sonars, des « Acoustic Doppler Current Profiler » (ADCP) ou aussi appelé « Doppler Velocity Log » (DVL) qui sont des capteurs qui permettent de mesurer la vitesse dans l'eau par rapport au fond ou à une colonne d'eau, et fonctionnent à l'aide des ondes acoustiques, des capteurs bathymétriques ou encore des appareils de mesure laser (cf. « LIght Detection And Ranging » : LIDAR). Le coût de certains de ces capteurs peut être élevé ; ils sont choisis en fonction de la tâche à accomplir.
- des sources lumineuses ; sous l'eau la visibilité est souvent réduite, la plupart des ROV embarquent donc des LED qui permettent d'éclairer le fond marin.
- une coque, qui permet d'envelopper tous les éléments entre eux et qui peut être conçue pour conférer certaines propriétés dynamiques au véhicule.

La figure 1.5 présente une collection de différents ROV commercialisés ou open-sources.

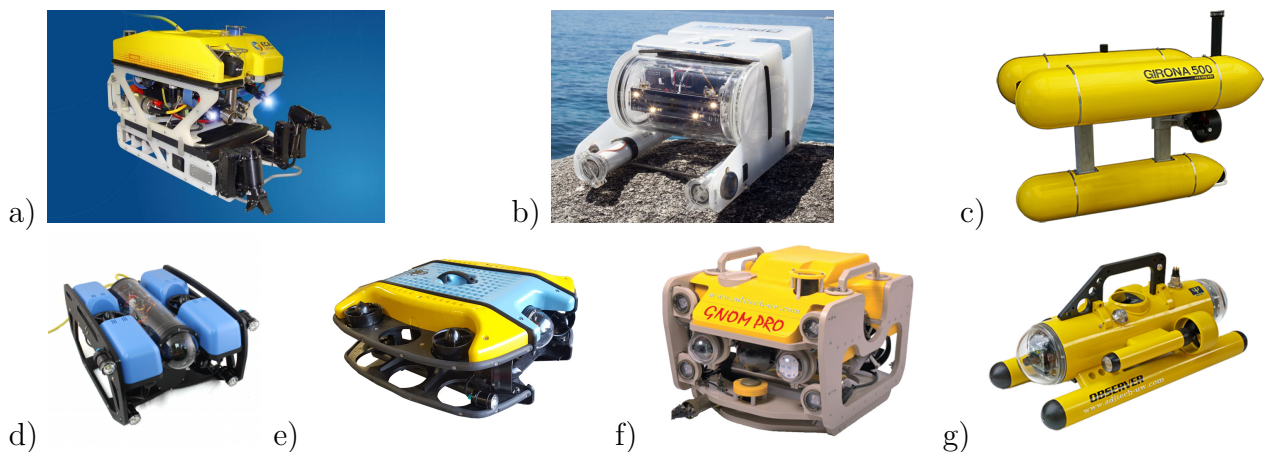


Figure 1.5 – a) H800 de ECA Group b) OpenROV c) Girona 500 de IQUA Robotics d) BlueROV2 de Blue Robotics e) S-ROV de TMI-Orion f) GNOM Pro de GNOM ROV g) Observer de Subsea-Tech (Source : ECA Group, OpenROV, IQUA Robotics, Blue Robotics, TMI-Orion, GNOM ROV, Subsea-Tech)

1.3.1.4 AUV : véhicule sous-marin autonome

Les véhicules sous-marins autonomes se différencient des ROV par leur capacité à être autonomes, tant du point de vue de l'énergie que de la commande. Il faut distinguer deux types d'AUV que sont les « glider », véhicules planeurs sous-marins, et les AUV conventionnels.

Les « glider » sont des véhicules sans motorisation et munis d'ailettes et de ballasts pour gérer leur trajectoire. Ils sont particulièrement intéressants pour des applications ne nécessitant pas de transmission d'informations en temps réel, où il faut parcourir de longues distances et

où il n'est pas nécessaire de suivre une trajectoire très précise. Les applications consistent à cartographier une zone marine au large, récolter des informations bio-géo-chimiques ou biologiques, mesurer des variables physiques (température, etc.). Pour les missions longue distance, l'un des points limitant pour les AUV est le stockage de l'énergie nécessaire à leur propulsion. Les gliders permettent de s'affranchir de cette limitation, puisqu'ils fonctionnent grâce à un système de contrepoids et de ballasts en se laissant planer au gré des courants. En contrepartie, ils sont incapables de réaliser des missions où il est nécessaire de maintenir une position.

En ce qui concerne les AUV conventionnels, parmi les applications à visée commerciale développées à ce jour, rares sont celles qui nécessitent spécifiquement un véhicule autonome : ces véhicules sont plus complexes à concevoir que les ROV et ne fournissent pas de retour temps réel à l'opérateur. Par exemple, si on explore une grotte avec un ROV et que celui-ci cesse de fonctionner, il suffit de suivre le fil qui relie le ROV à la station de commande pour le retrouver, en prenant les précautions pour ne pas casser en emmêler le câble. Dans une grotte, si on utilise un robot autonome, il peut arriver que le robot se perde en cas de panne, ce qui rend la tâche de récupération plus ardue. Lors d'une opération en mer, il n'y a pas de différence entre ces deux classes de véhicules s'ils sont tous rendus flottants⁵, ce qui leur permet dans le cas d'une panne de remonter à la surface et d'émettre un signal de détresse. Le principal avantage des véhicules autonomes par rapport au ROV est la distance qu'ils peuvent parcourir seuls, puisqu'ils ne sont pas reliés. Une grande distance peut être parcourue par un ROV s'il est accompagné par un bateau de surface, mais ceci engendre un coût important. L'AUV est particulièrement intéressant pour descendre à de grandes profondeurs. On trouve les AUV motorisés dans les domaines civils et militaires. Dans le domaine militaire, une application particulière des AUV est la guerre des mines (§1.3.3). Des robots autonomes de type compagnon ont été conçus récemment pour le domaine des loisirs comme le robot iBubble de la société Notilo Plus. Ce robot agit comme un caméraman et peut suivre un plongeur à la trace à l'aide du signal acoustique émis par un bracelet de manière autonome tout en évitant les éventuels obstacles détectés au moyen de la caméra. Il peut ainsi suivre un utilisateur pendant sa plongée. Il est également commandable au joystick. La figure 1.6 présente une collection de différents AUV.

1.3.1.5 Comparaison des ROV et AUV

La figure 1.5 montre que la plupart des ROV ont une forme de prisme, voire une forme de parallépipède rectangle. L'objectif est ici de permettre aux ROV de réaliser des manœuvres précises à faible vitesse. La plupart des AUV présentent une forme similaire (figure 1.6). Ceci n'est pas sans conséquence sur leur comportement dans l'eau. Leur enveloppe extérieure, en forme de cylindre ou de capsule, est allongée dans la direction d'avance principale. Cette forme permet à l'AUV de maintenir facilement son cap dans la direction d'avance principale et de réduire la résistance de l'eau dans cette direction. Pour tous ces robots autonomes, l'une des difficultés lorsqu'ils parcourent de longues distances, est de connaître leur position exacte. Comme évoqué précédemment dans la section 1.3.1.3, les robots sous-marins ne peuvent pas utiliser de GPS. Ils doivent donc utiliser leurs connaissances a priori du terrain ainsi que les capteurs qu'ils embarquent pour connaître leur position en se référant à leur position de départ ou bien être entourés d'un réseau de balises leur permettant de se positionner par triangulation [TDSW11]. L'estimation de la position des véhicules sous-marins autonomes

5. Un objet est considéré flottant lorsqu'il subit une poussée d'Archimède supérieure à son poids.

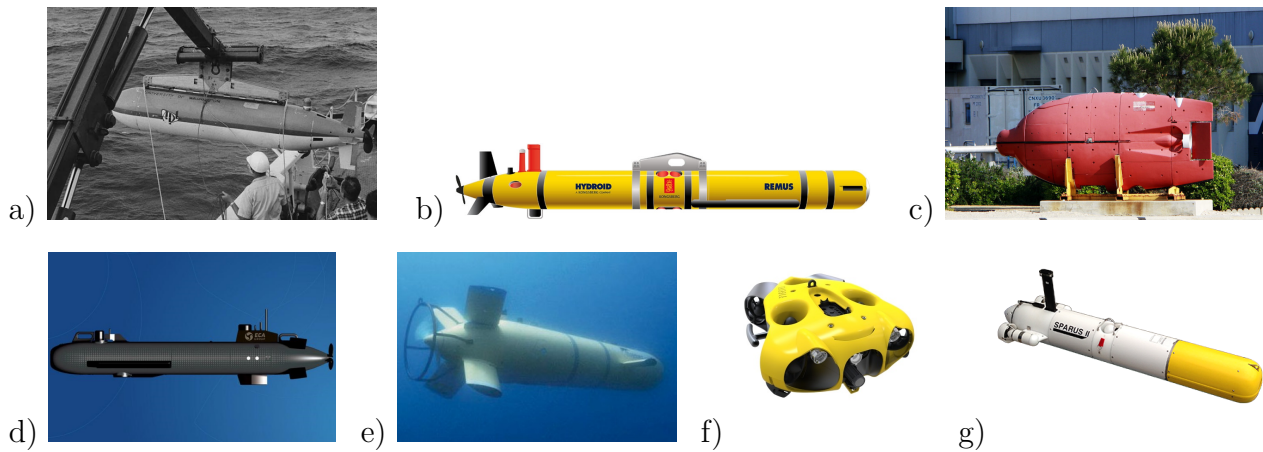


Figure 1.6 – a) Le premier AUV : Special Purpose Underwater Research Vehicle (SPURV) de l’université de Washington (1957) b) Le Remus 600 de Kongsberg c) Le premier AUV français l’ « Épaulard » de ECA Robotics d) A9-M de ECA Group e) Aster X de l’Ifremer f) iBubble de Notilo Plus g) Sparus II de IQUA Robotics (Source : University of Washington, Kongsberg, Ifremer, ECA Group, Rigaud [Rig07], Notilo Plus, IQUA Robotics)

fait partie des verrous scientifiques qui sont en cours d’étude. Les progrès dans ce domaine devraient simplifier la réalisation de véhicules sous-marins autonomes dans un futur proche [TMV⁺18].

1.3.2 Environnement des véhicules sous-marins

Les véhicules sous-marins évoluent dans un environnement aquatique qu’il est nécessaire de décrire afin de comprendre d’où proviennent les efforts appliqués sur leur coque et d’évaluer les intensités de ces efforts. Il existe deux zones principales dans lesquelles ces véhicules sous-marins peuvent évoluer : le large et la zone littorale. La morphologie typique d’une zone littorale est représentée sur la figure 1.7. Cette figure montre le lieu de séparation entre la zone littorale et le large, qui est matérialisée par le début d’une pente douce remontant du plateau océanique vers la côte et faisant débuter la zone dite d’avant-côte. Cette même zone littorale peut être divisée en différentes zones.

1.3.2.1 Le large

D’après la figure 1.7, le large correspond, à la zone au-delà de 2 km de la rive. À cet endroit, les météorologues maritimes peuvent fournir une information et des prévisions sur l’état de la mer. Cet état de la mer est lié à l’influence du vent et de la houle. La « mer de vent » est le système de vagues créé à l’endroit même où souffle le vent. En quittant l’endroit où elles ont été générées, les vagues se régularisent et deviennent la houle, qui peut se propager très loin même en l’absence de vent. Lorsque la houle suit une autre direction que les vagues engendrées par le vent, on parle de mer croisée. La diffraction derrière un détroit ou bien, dans une moindre mesure, la réflexion des trains de vagues sur une côte abrupte peut engendrer également une mer croisée au large. En Europe, le phénomène est connu et redouté dans le golfe de Gascogne et en Méditerranée.

La terminologie associée a été normalisée par les services de météorologie maritime pour

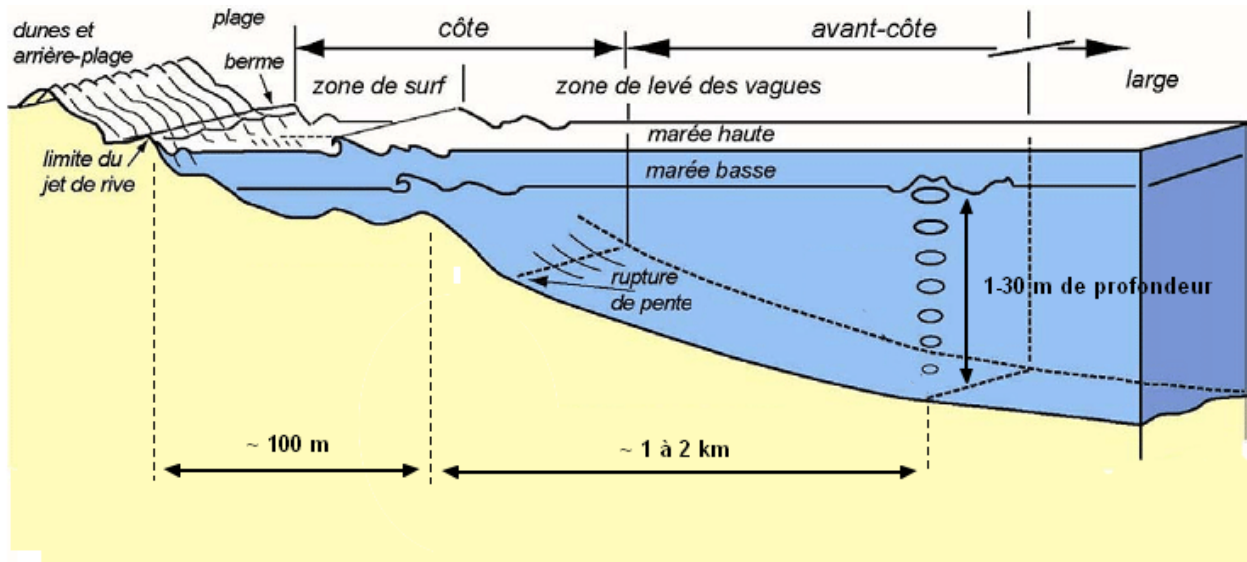


Figure 1.7 – La morphologie typique d'une zone littorale (Source : modifié à partir de [Sho12])

fournir aux navires et aux installations situées en mer une information qui puisse être utilisable. La mesure couramment admise pour l'état de la mer est l'échelle de la mer de Douglas, du vice-amiral Sir Percy Douglas, hydrographe de la Royal Navy dans les années 1920. Ce système a deux codes : l'un pour estimer la hauteur des vagues et l'autre pour décrire la houle. Les tableaux 1.1 présentent l'échelle de Douglas. Pour l'état de la houle, les longueurs d'onde

Etat de la mer de vent			Etat de la houle	
Degré	Hauteur (m)	Descriptif	Degré	Descriptif
0	pas de vagues	calme	0	Pas de houle
1	0 à 0,1	ridée	1	vagues courtes (ou moyennes) et petites
2	0,1 à 0,5	belle	2	vagues longues et petites
3	0,5 à 1,25	peu agitée	3	vagues courtes et modérées
4	1,25 à 2,5	agitée	4	vagues moyennes et modérées
5	2,5 à 4	forte	5	vagues longues et modérées
6	4 à 6	très forte	6	vagues courtes et hautes
7	6 à 9	grosse	7	vagues moyennes et hautes
8	9 à 14	très grosse	8	vagues longues et hautes
9	14 et plus	énorme	9	état confus (indescriptible)

Tableau 1.1 – Description de l'état de la mer selon l'échelle de Douglas

sont :

- vagues courtes : longueur d'onde inférieure à 100 m
- vagues moyennes : longueur d'onde entre 100 et 200 m
- vagues longues : longueur d'onde supérieure à 200 m

Et les hauteurs de houle sont :

- vagues petites : hauteur inférieure à 2 m
- vagues modérées : hauteur entre 2 et 4 m
- vagues hautes : hauteur supérieure à 4 m

En eau profonde, la longueur d'onde λ_{vague} des vagues est directement liée à leur vitesse c_{vague} :

$$c_{vague} = \sqrt{\frac{g\lambda_{vague}}{2\pi}} \quad (1.1)$$

Cet état de la mer de l'échelle de Douglas ne doit pas être confondu avec l'échelle des vents de Beaufort qui est parfois également utilisée dans le domaine maritime. Cette échelle donne la valeur de la vitesse moyenne du vent. Ainsi, elle permet de décrire l'état de la mer de vent mais ne permet pas de décrire la houle, qui provient d'évènements se passant au loin. Un exemple de l'aspect d'une mer calme et d'une mer agitée est présentée sur la figure 1.8.

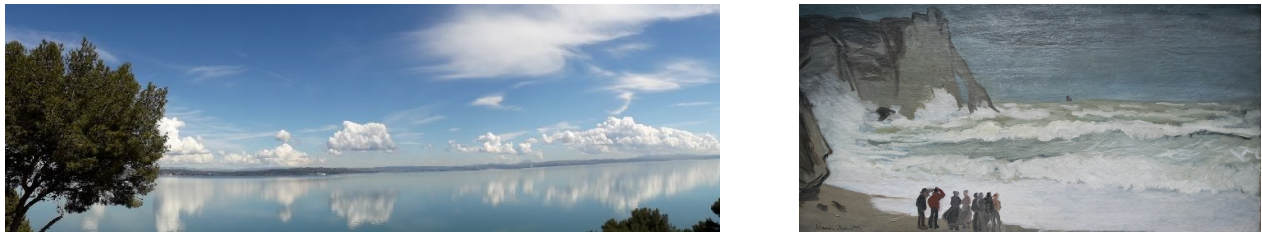


Figure 1.8 – Gauche : Mer calme sur l'étang de Berre. Droite : « Grosse mer à Etretat » de Claude Monet.

Les véhicules sous-marins en mer subissent les effets de la houle et des vagues lorsqu'ils sont proches de la surface. À partir d'une profondeur liée à la hauteur des vagues, les effets de la houle et des vents ne se font plus ressentir. C'est alors que les courants marins entrent en jeu. Un courant marin est un mouvement d'eau de mer régulier, continu et cyclique. Ce type de mouvement est dû aux effets combinés du vent, de la force de Coriolis, et de différences de température, densité et salinité, mais aussi aux contours des continents, aux reliefs de profondeur et à l'interaction entre courants. Aujourd'hui, il n'existe pas de données précises en temps réel ou de prévisions accessibles à tous pour ces courants marins, qui peuvent varier fortement entre différentes profondeurs en termes d'intensité, mais aussi de direction, sur un même lieu géographique (pour un même rayon terrestre).

1.3.2.2 La zone littorale

La zone littorale est aujourd'hui au cœur de l'actualité de la recherche. En effet, avec le réchauffement climatique et la montée du niveau de la mer, l'impact de l'érosion résultante sur les côtes est l'une des principales sources de préoccupations. Il existe différents types de zones littorales :

- Les littoraux meubles, caractérisés par une accumulation de matière, dont on distingue plusieurs catégories :
 - Les vasières littorales, qui sont des étendues basses et faiblement inclinées vers la mer qui se forment dans les lieux où le marnage⁶ est conséquent. Ils apparaissent généralement en fond de baies, aux bords des estuaires ou dans les lagunes. La

6. Distance entre le point le plus haut et le point le plus bas de la mer lors d'une marée



Figure 1.9 – Platier de la falaise des Hachettes en Normandie (Source : Lithothèque de Normandie)

position d’abri de ces sites induit un apaisement des eaux qui permet une sédimentation des particules les plus fines (vase), mais surtout qui empêche l’attaque directe par les houles et donc l’érosion des dépôts. Ces vasières représentent une part anecdotique du littoral.

- Les plages, qui sont des étendues où les vagues et les courants littoraux n’ont plus assez d’énergie pour transporter les sédiments de taille (donc de masse) supérieure à la vase. Les plages représentent 20 % du linéaire côtier mondial.
- Les côtes rocheuses, qui sont caractérisées par des falaises, qui sont des versants continentaux raidis à leur base par l’action de la mer. Ces falaises peuvent être légèrement éloignées de la mer et avoir devant elles un platier rocheux (figure 1.9), qui est une surface plane taillée par les vagues et la pluie, découverte à marée basse. Ces côtes rocheuses représentent la majeure partie du littoral mondial, soit 80 % du linéaire côtier.

La morphologie de ces zones côtières est importante pour la mission des véhicules sous-marins, notamment car elle va être déterminante pour les courants marins mais aussi parce qu’elle va influencer sur la présence et l’abondance de rochers.

Le fond marin de la zone littorale est représenté sur la figure 1.7. La zone d’avant-côte est la zone dans laquelle les vagues se forment. Après avoir passé cette zone, les vagues arrivent vers des profondeurs plus faibles, ce qui comprime l’eau, et puisque l’énergie de la vague est conservée, fait monter la hauteur de la vague et diminuer sa vitesse. A un moment donné, sous l’effet de la pesanteur, la vague va s’effondrer sur elle-même : c’est le déferlement. Cette zone, où les vagues commencent à être déformées par la faible profondeur, est appelée zone de surf. En dehors de la zone de surf, les effets de la mer ressentis par les véhicules sous-marins seront relativement proches de ceux de la haute mer.

1.3.2.3 La zone de surf : un environnement perturbé

La zone de surf est la zone de faible profondeur proche de la zone de jet de rive, où les vagues grandissent puis déferlent en formant une mousse : l’écume. Après la rupture dans la zone de surf, les vagues, réduites en hauteur, continuent à évoluer sur la plage, et elles se dirigent vers le haut sur le front de plage en pente, formant un jaillissement d’eau appelé oscillant. L’eau passe ensuite à nouveau en rétrolavage, repartant vers le large. Généralement la profondeur de la zone de surf à l’endroit le plus au large est entre 5 et 10 m. C’est une

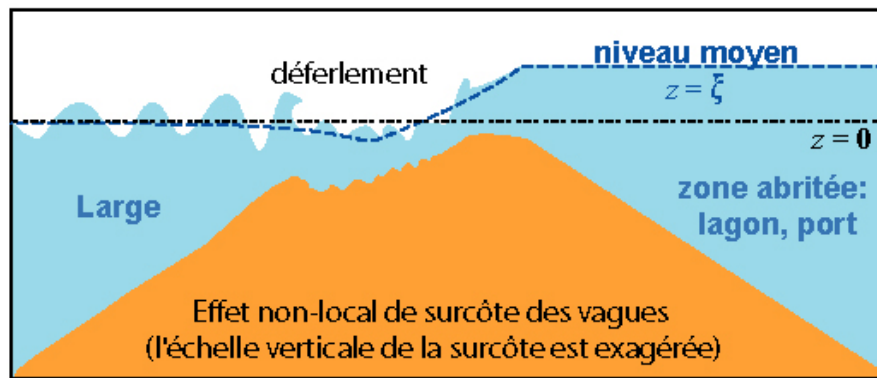


Figure 1.10 – Zone de déferlement ne débouchant pas sur une plage ou une falaise (Source : SHOM)

zone très instable, et il est actuellement très difficile, voire impossible d’y naviguer avec les véhicules sous-marins actuels.

Bien que cette zone donne souvent directement sur une plage ou une falaise, il est aussi possible qu’elle donne sur l’entrée d’un port ou d’un lagon, ce qui est assez fréquent dans le Pacifique. La figure 1.10 montre l’effet de sur-côte, montée du niveau moyen proche de la côte, produit par les vagues lorsqu’elles déferlent avant de toucher terre. Si les vagues déferlent dans un passage peu profond connectant le large à une zone abritée, cette élévation du niveau moyen peut affecter l’ensemble de la zone abritée. Cela est observé lors des grandes tempêtes en Adriatique : le niveau moyen de la lagune de Venise peut ainsi monter de plusieurs dizaines de centimètres lorsque la zone de déferlement s’étend jusqu’au bout des jetées qui connectent la lagune à la mer.

Suite à l’étude de l’environnement des véhicules sous-marins, on peut découper la mer en deux zones distinctes pour le fonctionnement de ces véhicules : la zone de surf et le reste de l’environnement marin. En effet, lors de leur entrée dans la zone de surf, les vagues sont comprimées et leur mouvement est différent, comme représenté sur la figure 1.11. La circulation des particules dans les vagues en dehors de la zone de surf est de forme circulaire ou elliptique. Lors de l’entrée dans la zone de surf, une rotation et une déformation de cette circulation surviennent. Ceci modifie les courants présents dans cette zone. De plus, un courant supplémentaire provenant du ressac s’ajoute. C’est pourquoi cette zone doit être particulièrement distinguée du reste de l’environnement marin pour la simulation de la dynamique de véhicule.

1.3.3 Enjeux de la navigation en zone de surf

Naviguer dans la zone de surf, c’est permettre d’atteindre de nombreux lieux que l’on ne peut pas atteindre pour l’instant avec ces véhicules. Prenons par exemple, les projets pour l’étude et la préservation d’écosystèmes côtiers, comme le projet « T-MEDNet » [TME19] ou encore le projet « Blue Carbon » [Blu]. Le projet « T-MEDNet » est un projet européen, d’observation de l’impact du changement climatique sur la biodiversité de la méditerranée. Le réchauffement climatique impacte notamment la répartition des espèces et peut provoquer une mortalité de masse pour certaines d’entre elles. Pour anticiper et permettre la protection des espèces qui pourraient être menacées dans un futur proche, des mesures de la température

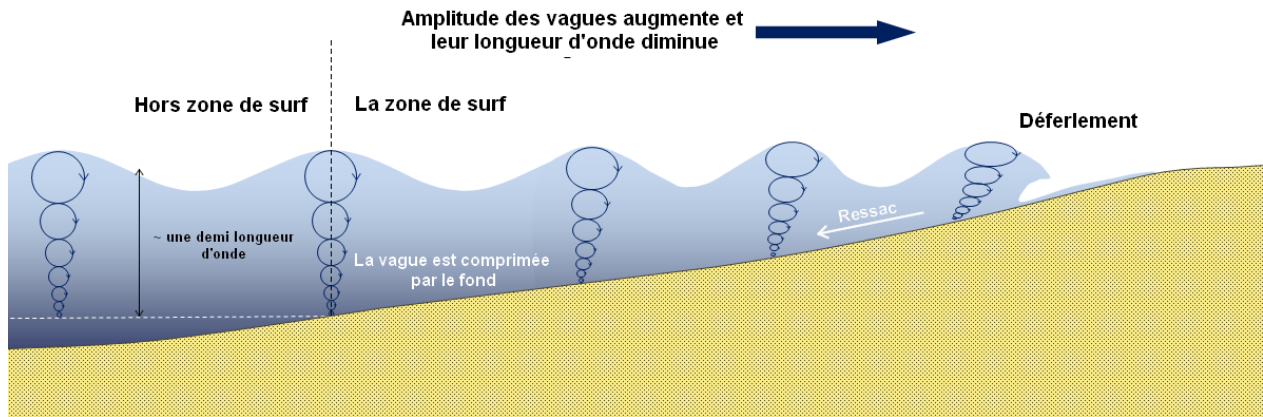


Figure 1.11 – Mouvement des éléments entraînés par les vagues dans la zone de surf (Source : Modifié à partir de *Physical Geology* [Ear18])

de l'eau, ainsi qu'une observation des dégradations notamment sur les coraux sont effectuées (Figure 1.12). Le projet « Blue Carbon » est un projet international de protection des écosystèmes côtiers de mangroves, de vasières et de prairies d'herbiers marins. L'entretien de ces écosystèmes pourrait permettre de s'adapter au changement climatique et de protéger les côtes contre les tempêtes, l'érosion et la montée du niveau de la mer. Ces écosystèmes sont également le refuge de nombreuses espèces marines qui contribuent à leur protection et qu'il faut protéger. De plus, ils stockent d'importantes quantités de « carbone bleu » provenant de l'atmosphère et de l'océan, d'où le nom du projet, et sont maintenant reconnus pour leur rôle dans l'atténuation des changements climatiques. Dans ces deux projets, les premières tâches sont : l'inventaire de la faune présente et l'étude de la topographie des lieux à protéger. Ces tâches sont fastidieuses et elles ne peuvent être réalisées actuellement par un véhicule sous-marin sur toute la surface de la zone côtière, notamment dans les zones rocheuses et la zone de surf.

Dans cette zone, on retrouve également deux applications militaires : la guerre des mines et les opérations de débarquement. Les véhicules sous-marins autonomes sont déjà utilisés pour le déminage à condition que les mines soient placées dans une zone avec suffisamment de fond et de faibles perturbations. L'objectif des véhicules est de détecter les menaces potentielles et de les éliminer. La difficulté réside dans la minimisation de leurs signatures acoustiques et magnétiques afin de ne pas déclencher les mines et de ne pas être détectés. La société ECA Robotics est la société française spécialiste du domaine. Elle propose notamment le K-STER C, qui est un véhicule sous-marin autonome équipé d'une charge creuse pouvant être envoyée dans différentes directions grâce à une tête orientable (figure 1.13). Néanmoins, dans le cadre d'une opération spéciale devant se dérouler dans la zone de surf ou d'une opération de débarquement, on ne peut pour l'instant pas utiliser de véhicule sous-marin.

D'autres applications dans cette zone sont dédiées à l'archéologie marine ou à la plongée de loisir. Parfois les épaves ne se trouvent pas loin des côtes, mais au contraire très proches. C'est le cas, par exemple, de l'épave d'un Spitfire de la seconde guerre mondiale qui se trouve dans le golfe de Lava en Corse à seulement 6 mètres de fond ou encore de l'épave du Junkers 90 Allemand B-1 WK Nr07 du Lufttransportstaffel 290 qui se trouve près de Bastia à seulement 3m de fond. La plongée autour de ces épaves peut être très difficile selon la météo. Lorsque cela n'est pas possible pour l'Homme, l'utilisation d'un véhicule sous-marin pour observer ces

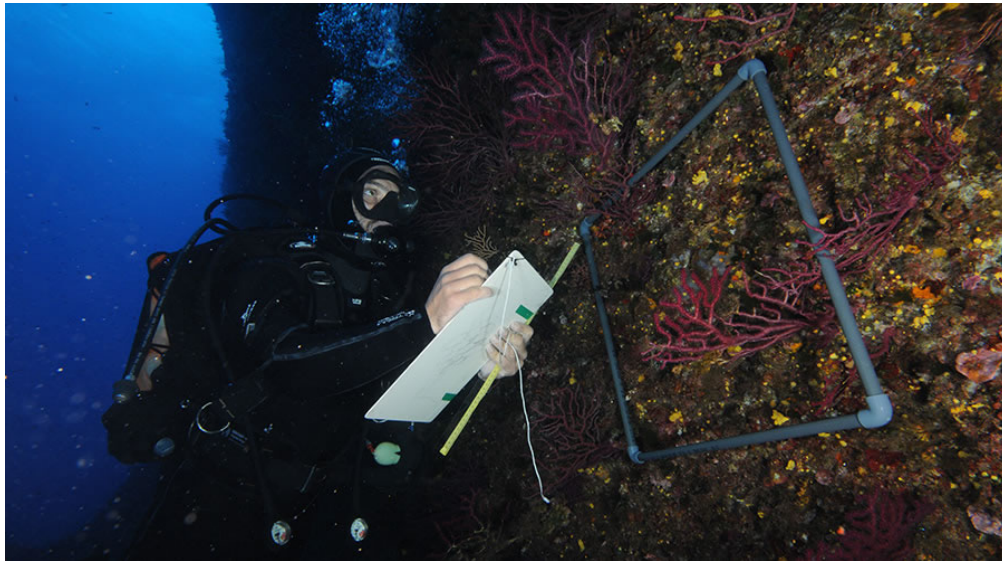


Figure 1.12 – Un plongeur évalue l'état du corail en méditerranée dans le cadre du projet T-MEDNet (Source : T-MEDNet)

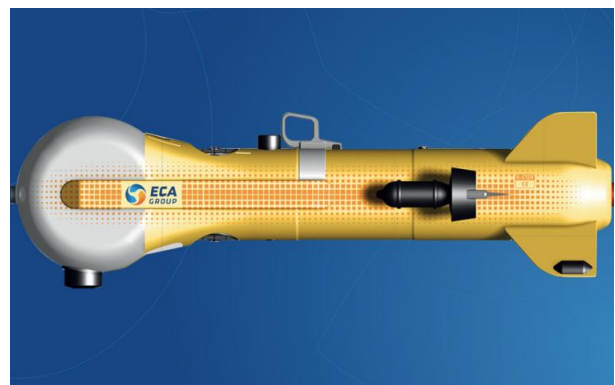
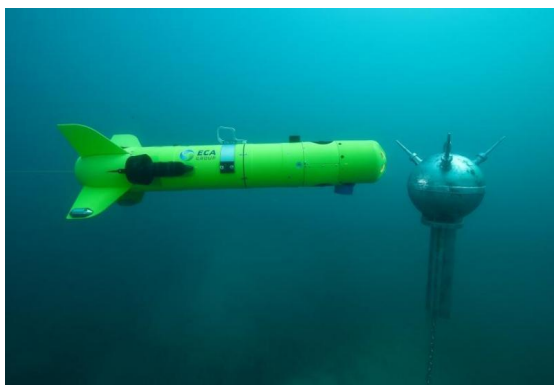


Figure 1.13 – Droite : Un véhicule sous-marins autonome détecte une mine. Gauche : Le véhicule K-STER C avec sa tête orientable pouvant être équipée d'une charge creuse. (Source : ECA Robotics)

épaves n'est pas non plus possible.

C'est également dans cette zone que sont installés certains dispositifs de production d'énergie ou de traitement. Parmi ceux-ci, d'après [MC⁺06] pour le domaine de la production d'énergie, on peut citer les centrales avec systèmes houlo-générateurs, les usines marémotrices, les hydroliennes, les éoliennes offshore et les convertisseurs d'énergies thermiques des mers. Les systèmes houlo-générateurs (an anglais, WEC : wave energy converters), sont des systèmes qui utilisent l'énergie de la houle pour générer un courant électrique. Parmi cette famille, il existe trois grands principes utilisés à ce jour :

- les systèmes à rampe de déferlement, dont le principe est d'utiliser une rampe pour canaliser la houle et ainsi remplir un bassin où le niveau d'eau serait plus haut que le niveau de la mer puis de récupérer l'énergie potentielle stockée à la manière d'un barrage de rivière
- les systèmes à colonne d'eau oscillante, dont le principe est d'exploiter l'énergie de l'oscillation de l'eau de mer à l'intérieur d'une chambre ou d'un creux causée par l'action des vagues.
- les systèmes à corps mus par la houle, dont le principe est d'utiliser des caissons étanches de forme asymétrique et excentrée roulant autour d'un axe et qui, par le mouvement des vagues, actionnent des pompes hydrauliques qui génèrent l'énergie.

Les usines marémotrices utilisent le marnage pour produire de l'électricité en exploitant la différence de hauteur entre deux bassins. Les hydroliennes sont des systèmes de récupération des courants marins qui fonctionnent de la même manière que les éoliennes. Les éoliennes offshore peuvent être flottantes ou fixes, et sont des systèmes qui récupèrent l'énergie du vent. Les centrales « Ocean Thermal Energy Converter » (OTEC) sont fondées sur une machine à turbine utilisant la détente d'un fluide évaporé sous l'effet d'une source chaude (température des eaux de surface : 26 à 30°C) puis condensé grâce à une source froide (température des eaux profondes pompées et ramenées au niveau de la machinerie). Les dispositifs de traitement évoqués peuvent être soit des usines de traitement des eaux qui vont être rejetées en mer, soit des usines de traitement qui permettent de transformer l'eau de mer en eau potable (usine de dessalement).

Ces installations nécessitent un entretien régulier qui pourrait être effectué en partie par des véhicules sous-marins autonomes si ceux-ci étaient capables de naviguer dans la zone de surf.

1.3.4 Conclusion

Cette première section a présenté les différents types de véhicules sous-marins, que l'on différencie en deux groupes, les sous-marins habités et les UUV, qui se décomposent également en deux sous-groupes : les ROV et les AUV. La forme des ROV et des AUV est caractéristique. Cette forme est due à leur fonction, qui est la couverture d'une large zone et il est probable que, pour des tâches nécessitant des mouvements plus précis, une forme de prisme soit préférable, à l'instar des AUV iBubble et des ROV. L'environnement marin dans lequel les UUV évoluent a également été présenté. Cet environnement se découpe en différentes zones distinctes selon les types de perturbations qui peuvent être subies par le véhicule. Actuellement les véhicules marins sont capables de naviguer dans les zones où il n'y a que peu de perturbations, ce qui les limite fortement lorsque l'on s'intéresse à la zone proche de la côte. Or cette zone se révèle particulièrement intéressante pour différents types d'applications : guerre des mines, préservation d'écosystème côtier, entretien d'installations côtières, archéologie sous-marine.

Des avancées techniques sont nécessaires pour permettre au véhicule sous-marin de naviguer de manière autonome dans cette zone. Avant de construire un premier prototype de véhicule marin il est nécessaire d'évaluer la faisabilité de ce projet. Dans de nombreux cas, la réalisation d'une simulation en amont de la conception permet de diminuer les coûts de développement. C'est la raison du développement du simulateur auquel cette thèse contribue. Ce simulateur devra être capable de représenter les effets dynamiques liés aux perturbations du fluide ainsi que l'interaction entre l'eau et le véhicule sous-marin. Les résultats de simulation pourraient également permettre aux concepteurs de véhicules sous-marins d'améliorer les prototypes en optimisant notamment leur forme avant leur fabrication, ou encore leur permettre de concevoir des véhicules reconfigurables. Ainsi, les différentes solutions existantes qui permettent de modéliser les véhicules sous-marins et leur environnement sont présentées dans la suite de ce chapitre.

1.4 Thématique et axes de recherche

Aujourd'hui, il existe un réel engouement pour le développement de véhicules robotiques en milieu sous-marin. La simulation de la dynamique de ces véhicules sous-marins se révèle complexe, notamment du fait de la multitude d'éléments provenant de l'environnement du véhicule à prendre en compte et de la variété de véhicules. La simulation est pourtant nécessaire si on désire dans un futur proche développer des prototypes de véhicules ou créer des algorithmes de commande permettant de rendre les véhicules plus autonomes. **Cette thèse a pour but de déterminer une méthode de simulation de tout type de véhicule sous-marin dans un environnement aquatique réaliste.** Afin de contribuer à cet objectif, les étapes suivantes ont été définies :

- **Caractériser les simulateurs existants.** Cela passe par l'analyse des différentes solutions qui existent : comment sont-elles conçues ? Existe-t-il une solution clef en main répondant aux besoins, générique, et non dédiée à un véhicule particulier ?
- **Mettre au point une méthode permettant de valider le réalisme physique d'un simulateur.** Une démarche par étapes est nécessaire pour valider le réalisme du simulateur. Pour chacune des étapes, des expériences caractéristiques doivent être définies pour vérifier l'apparition des phénomènes attendus à reproduire en simulation, comme par exemple une poussée d'Archimède réaliste.
- **Comprendre les interactions entre, d'une part, le fluide et d'autre part, le solide,** et préciser lesquelles sont à reproduire en simulation. A cet effet, les efforts hydrodynamiques entre fluides et solides doivent être caractérisés et leur modélisation expliquée. Cette étude se révèle particulièrement complexe, car aujourd'hui, les lois de la dynamique des fluides⁷ ne sont pas résolues analytiquement et les effets hydrodynamiques sur les formes solides ne sont connus que de manière empirique.
- **Obtenir un résultat utilisable pour le prototypage dans un temps raisonnable.** Lors de notre étude, il a été montré que l'un des principaux intérêts du simulateur était la préparation de missions et le prototypage de la forme du robot ou de son algorithme de commande. Il est donc souhaitable que ce simulateur fournisse des résultats dans un temps raisonnable par rapport à la préparation d'une mission. L'idéal serait évidemment de pouvoir obtenir un résultat en temps réel.
- **Disposer d'un simulateur adapté à chaque problème.** Le simulateur doit permettre l'étude de différents types de véhicules marins, sans nécessiter d'expérimentations préalables, ni de codage des paramètres hydrodynamiques. Cette adaptabilité offre l'avantage de prévoir les performances de véhicules sous-marins en amont de leur fabrication. Elle permet également l'étude de la conception de véhicules reconfigurables.
- **Conserver un simulateur modulaire et modifiable.** Le simulateur doit offrir la possibilité de modifier ou de personnaliser une partie du logiciel développé pour ajouter des fonctions supplémentaires.

Ces différents points seront pris en compte dans la suite lors de la sélection d'une méthode de simulation correspondant à notre problématique.

7. On parle ici des équations de Navier-Stokes (voir chapitre 3)

1.5 Environnements de simulation : application aux véhicules marins

Cette partie décrit comment sont construits les environnements de simulations en général. Elle présente les principaux environnements de simulation existants permettant de simuler un véhicule et son environnement et se focalise sur leurs insuffisances pour aboutir à notre problématique de recherche.

1.5.1 Environnement de simulation : constitution

Un environnement de simulation est un logiciel qui permet d'accueillir ou créer un modèle, puis d'utiliser celui-ci pour simuler son fonctionnement réel. Pour cela, il faut tout d'abord associer un objet ou un phénomène réel au modèle choisi ou créé spécifiquement, puis appliquer les lois physiques qui s'appliquent à cet objet réel au modèle de la simulation. Les simulations peuvent accueillir un ensemble de modèles et constituer un système de modèles. L'ajout de chaque nouveau modèle rend l'environnement de simulation plus complexe. C'est la complexité qui rend la simulation attractive comparée à l'expérimentation ou au simple calcul. Dans un esprit de simplification, les différents modèles sont généralement regroupés en catégories, comme par exemple tous les objets solides qui sont soumis aux lois de Newton.

Les objectifs de ces environnements de simulation sont [Ebe93] :

- la compréhension de la structure et des rapports à l'intérieur d'un système.
- l'analyse et la prédiction du comportement du système.
- la conception et le dimensionnement de l'équipement
- l'aide à l'expérimentation, qui peut éventuellement finir par ...
- le remplacement de l'expérimentation, souvent très coûteuse, et même quelquefois impossible.
- la validation des systèmes de contrôle.

Dans notre cas, le système à représenter est constitué du véhicule sous-marin et de son environnement (océan, mer, zone de surf). Cet environnement peut être un lieu où le véhicule sous-marin subit un grand nombre de perturbations. Les environnements actuels de simulation, adaptés à la simulation de véhicules, sont tous constitués d'un ensemble d'éléments logiciels similaires, comme représenté sur la figure 1.14. Ils sont constitués :

- d'un programme principal qui permet de gérer tous les sous-éléments du logiciel. Souvent, ce programme principal appelle des bibliothèques externes développées par différents acteurs extérieurs nommés « Application Program Interface » (API).
- d'une bibliothèque graphique, qui s'appelle aussi moteur 3D dans le cas d'un affichage en trois dimensions.
- d'une bibliothèque qui permet le calcul de la dynamique d'objets solides réels, couramment appelé moteur physique. Le moteur physique comprend l'ensemble du code lié à la résolution de la dynamique de tous les corps, y compris les fluides. Aujourd'hui, le développement des moteurs physiques est souvent extérieur au développement des applications elles-mêmes, notamment dans le domaine de l'animation mais aussi dans le domaine de la robotique. Dans ces domaines, on se base sur des moteurs pré-existants comme Nvidia PhysX ou Bullet (voir annexe B).
- d'autres bibliothèques parmi lesquelles on retrouve notamment :
 - une bibliothèque de « post-process », qui consiste à post-traiter l'affichage pour améliorer

le rendu après simulation,

- des bibliothèques permettant la connexion du programme à un réseau ou l'interfaçage de ce programme avec d'autres logiciels, qui peuvent être des middlewares. Les middlewares sont présentés dans l'annexe B. La figure 1.14 montre un exemple d'interfaçage entre le middleware, le système réel, l'environnement de simulation et le logiciel Matlab.
- des bibliothèques permettant l'utilisation de scripts pour donner plus de flexibilité à l'utilisateur.
- des bibliothèques permettant à l'utilisateur de modéliser des objets aussi appelées modélisateurs.

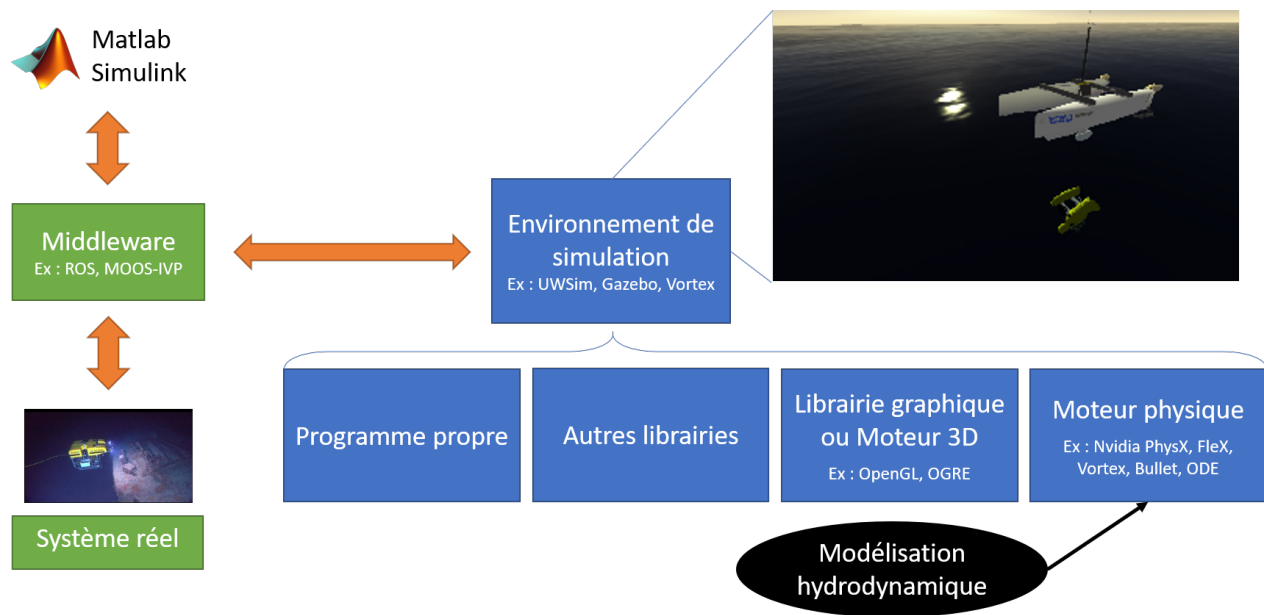


Figure 1.14 – Schéma représentant l'architecture d'un environnement de simulation

Les différents simulateurs existants sont décrits dans l'annexe B. Une partie de cette annexe décrit les éléments suivants : middlewares, bibliothèques graphiques et moteurs physiques. Dans la suite, un récapitulatif de ces solutions actuelles permettra de les distinguer et aidera au choix du simulateur. La modélisation hydrodynamique est la partie la plus importante pour notre application. La comparaison des logiciels de simulation se focalisera donc sur cet aspect.

1.5.2 Récapitulatif des solutions actuelles pour la simulation de la dynamique des véhicules sous-marins

Pour résumer les différents choix possibles de simulateur d'environnement perturbé pour véhicule marin, les simulateurs sont comparés suivant plusieurs critères considérés comme les plus importants pour notre application : leur connectivité, leur caractère personnalisable, leurs domaines principaux d'utilisation et les modélisations du fluide et du solide disponibles. Il a été évoqué que le réalisme physique des simulateurs dédiés à l'animation temps réel n'était pas respecté car il n'est pas forcément recherché. Ce critère sera particulièrement étudié dans la suite de ce chapitre.

1.5.2.1 Connectivité des environnements de simulation présentés

L'utilisation d'un environnement de simulation capable de connexions avec un élément externe (logiciel de calcul, middleware, système réel) est un plus, car cela permet de se servir de données ou d'algorithmes qui ne sont pas présents dans le simulateur. Le tableau 1.2 récapitule les possibilités en termes de connectivité pour une sélection d'environnements de simulation présentés dans l'annexe B. Les deux premières colonnes listent les simulateurs regroupés par catégorie, tel que présentés dans la section précédente. Les colonnes suivantes indiquent si ces logiciels ont une interface dédiée ROS ou au moins un autre middleware orienté robotique, c'est à dire ceux pour lesquels aucun effort de programmation n'est nécessaire pour créer une interface. ROS est un logiciel de robotique open-source très répandu. Il est déjà installé sur les véhicules du laboratoire COSMER, et est de surcroît utilisé par tous les partenaires du projet SURFBOT. Il est possible de trouver des interfaces réalisées par des tiers, comme par exemple un couplage de ROS et Unity [CDFS⁺14], non répertoriés ici.

		Middleware		
	Logiciel	ROS	Autre(s) middleware(s)	Logiciel(s) de calcul
Robotique	Vortex			✓
	Morse	✓	✓	
	UWSim	✓		
	Gazebo	✓	✓	
	V-Rep	✓	✓	
Anim.	Unity			
	Unreal Engine			
	Autodesk Maya			
CFD	ANSYS Fluent			✓
	STAR-CCM+			
	GPUSPH			
	Neutrino			

Tableau 1.2 – Récapitulatif des possibilités de connectivité des environnements de simulation présentés dans l'annexe B.2.

À la lecture du tableau, on peut remarquer que les environnements de simulation dédiés à la robotique sont bien plus facilement intégrables dans un système constitué de plusieurs machines, grâce notamment à l'inclusion de différents middlewares. Seul Vortex sort du lot en proposant une interface directe avec Matlab mais ne possédant pas d'interface avec ROS ou un autre middleware qui pourrait permettre de le connecter à un véhicule réel. Les logiciels dédiés à l'animation ne sont pas dédiés à l'utilisation en tant qu'environnement de simulation pour la robotique sous-marine, par conséquent, ils n'incluent aucune interface vers un middleware ou un logiciel de calcul.

1.5.2.2 Personnalisation des environnements de simulation présentés

Cette partie compare les environnements en fonction de leur possibilité d'interfaçage. La simulation de scènes complexes, ou tout simplement le rajout de fonctionnalités non intégrées

	Logiciel	Script		Open-Source	Interface
		Python	Autre(s)		
Robotique	Vortex	✓			✓
	Morse	✓	✓	✓	
	UWSim	✓		✓	
	Gazebo	✓	✓	✓	✓
	V-Rep	✓	✓		✓
Anim.	Unity	✓	✓		✓
	Unreal Engine	✓	✓		✓
	Autodesk Maya		✓		✓
CFD	ANSYS Fluent		✓		✓
	STAR-CCM+		✓		✓
	GPUSPH		✓	✓	
	Neutrino	✓		*	✓

Tableau 1.3 – Récapitulatif des possibilités de personnalisation des environnements de simulation présentés dans l’annexe B.2. * Accès au code source obtenu grâce à un accord.

à l’environnement de simulation basique nécessitent des options de personnalisation du logiciel. Ceci peut être réalisé à l’aide de scripts ou par la modification du code source du logiciel si c’est un logiciel open-source. En robotique, la présence d’interfaces permettant de modifier la scène avant le début de la simulation est utile à l’environnement de simulation. Les environnements faisant appel à un logiciel externe ou ne proposant que des fichiers paramétrables de description de la scène demandent une plus longue prise en main. Le tableau 1.3 récapitule les possibilités en termes de personnalisation pour les environnements de simulation déjà mentionnés dans le tableau 1.2. Les premières colonnes sont les mêmes que celles du premier tableau 1.2. Les deux colonnes suivantes indiquent s’il est possible d’utiliser des scripts avec ces logiciels, si tel est le cas, le tableau renseigne sur le langage de programmation du script, Python ou un autre langage. Il existe beaucoup de bibliothèques de calcul numérique optimisées utilisant ce langage de programmation. De plus, Python est un langage de programmation interprété, c’est à dire que les programmes Python sont exécutés par un interprète. Un interprète se distingue d’un compilateur par le fait que, pour exécuter un programme, les opérations d’analyse et de traductions sont réalisées à chaque exécution du programme (par un interprète) plutôt qu’une fois pour toutes (par un compilateur). Il est plus pratique de tester et de préparer des scripts en Python. La cinquième colonne indique si le logiciel est open-source, ce qui est un avantage si l’on souhaite modifier directement la gestion des interactions hydrodynamiques dans l’environnement de simulation. Le signe * sur la ligne du logiciel Neutrino signale que l’accès au code source de ce logiciel a été autorisé grâce un accord conclu avec l’entreprise Centroid Lab. Techniquement, ce logiciel n’est pas open-source. La dernière colonne indique s’il existe une interface permettant de modifier les scènes directement dans l’environnement de simulation.

Tous les environnements de simulation permettent l’utilisation de scripts et un grand nombre d’entre eux utilisent python. La plupart des logiciels dédiés aux domaines de la

	Logiciel	Applications	Modélisation physique			
			Solide	Fluide	Surface du fluide	Fluide-solide
Robotique	Vortex	RT, RA, M	✓		✓	✓
	Morse	RT, RA, (M)*	✓		✓*	✓*
	UWSim	M	✓			✓
	Gazebo	RT, RA, (M)*	✓		✓*	✓*
	V-Rep	RT, RA	✓			
Anim.	Unity	Temps réel	✓	✓	✓	
	Unreal Engine	Temps réel	✓	✓	✓	
	Autodesk Maya	Hors ligne	✓	✓	✓	
CFD	ANSYS Fluent	Calcul fluide		✓		✓
	STAR-CCM+	Calcul fluide		✓		✓
	GPUSPH	Calcul fluide	✓	✓	✓	
	Neutrino	Calcul fluide	✓	✓	✓	✓

Tableau 1.4 – Récapitulatif des domaines d’utilisation et des modèles présents à l’intérieur des environnements de simulation présentés dans l’annexe B.2.

M : Ingénierie/Robotique marine et sous-marine, RT : Robotique terrestre, RA : Robotique aérienne. * Dans les branches UWMORSE et UUVSimulator de, respectivement, MORSE et Gazebo

recherche en robotique sont open-source (3 sur 5), contrairement aux autres domaines d’application. Les logiciels qui n’ont pas d’interface pour interagir avec la scène : Morse, UWSim et GPUSPH, sont des logiciels développés par des chercheurs pour la recherche.

1.5.2.3 Domaines d’utilisation et méthode de modélisation des environnements de simulation présentés

Chacun des logiciels présentés dans l’annexe B a été développé pour une application ou un type d’utilisateur, il est donc intéressant de comparer et de prendre en compte cette information, puisque les futurs développements de l’environnement de simulation se feront pour les utilisateurs de ce domaine. Un simulateur dédié à la robotique sous-marine est donc plus adapté à notre cas qu’un environnement de simulation dédié aux jeux vidéo, à l’animation temps réel, à l’animation de fluide pour les effets spéciaux au cinéma qui se fait souvent hors ligne, ou à la stricte simulation de fluide. De plus, selon l’utilisation, les modèles utilisés pour les actions des fluides et des solides sont différents. Le tableau 1.4 récapitule les domaines d’utilisation et les différentes méthodes de modélisation des environnements de simulation présents dans les tableaux précédents (Tableau 1.2, Tableau 1.3). Les premières colonnes sont similaires au tableau 1.2. La deuxième colonne désigne le domaine d’utilisation principale de l’environnement de simulation. Les colonnes suivantes indiquent le type de modélisation physique existant dans l’environnement de simulation : existe-t-il un modèle pour la dynamique des solides ? existe-t-il un modèle pour la dynamique du fluide ? Existe-t-il un modèle pour l’évolution de la surface du fluide ? Existe-t-il un calcul des forces d’interaction entre le fluide et le solide ?

A la lecture de ce tableau, on peut constater qu’aucun des environnements de simulation dédiés à la robotique ne possède de modèle pour la dynamique du fluide, qui est très complexe à obtenir. Seul des logiciels spécifiques au domaine du calcul du fluide et des logiciels d’animation intègrent ce type de calcul. L’absence d’informations sur la dynamique du fluide est rédhibitoire si on veut modéliser la dynamique du véhicule sous-marin dans un environnement perturbé. Néanmoins, certains logiciels dédiés à la robotique marine modélisent la surface de l’océan, c’est à dire les effets de la houle et du vent sur les vagues. Il est également possible de supposer des courants marins s’appliquant sur la surface du véhicule. Les deux types de modélisation, avec ou sans modèle physique pour le fluide, seront présentés dans la section 1.6. Parmi les logiciels intégrant la modélisation de la dynamique du fluide, les logiciels basés sur des maillages, qui sont ici Fluent et STAR-CCM+, permettent difficilement de simuler la surface du fluide. Ceci est plus simple avec des logiciels basés sur des particules comme GPUSPH et Neutrino. Dans GPUSPH le couplage fluide-solide n’est pas encore au point. Dans Neutrino, ce couplage est disponible.

1.5.3 Réalisme physique d’environnements de simulation dédiés à l’animation temps réel

Il existe des modèles pour la dynamique des solides et des fluides dans les logiciels dédiés à l’animation et aux jeux vidéo. Ces logiciels sont capables de délivrer des résultats en temps réel en utilisant la puissance de calcul disponible dans les GPU comme les moteurs physiques Nvidia PhysX et Nvidia FleX. Le logiciel Bullet est présent dans plusieurs logiciels dédiés à la recherche en robotique ainsi que dans Neutrino. Ces trois moteurs physiques, Bullet, PhysX et FleX, ont fait l’objet d’une évaluation à l’aide de quelques cas tests, afin de vérifier s’ils sont bien capables de simuler de manière réaliste la dynamique des solides et la dynamique des fluides.

1.5.3.1 Dynamique des solides avec PhysX et Bullet

La dynamique des solides dans PhysX et Bullet est assez complète. Ils intègrent un modèle pour les collisions, pour les liaisons mécaniques, etc. Afin de vérifier le réalisme de la mécanique du solide dans Nvidia PhysX, un balancier mécanique en oscillation libre est utilisé. Cette expérience a été choisie car elle permet de s’assurer rapidement de la conservation de l’énergie mécanique par l’intégration numérique. Plus précisément un objet est suspendu au bout d’une tige, lié par une liaison encastrement. Cette tige, est en liaison pivot par rapport au bâti. Une masse ponctuelle est fixée à l’extrémité d’une tige sans masse qui oscille sous l’effet de la pesanteur. Soit θ l’angle entre la tige et l’axe vertical, m la masse suspendue, g la constante de gravité et l la longueur de la tige. L’énergie cinétique T et l’énergie potentielle de pesanteur U de ce pendule s’écrivent :

$$T = \frac{1}{2}ml^2\dot{\theta}^2 \quad U = mgl(1 - \cos\theta) \quad (1.2)$$

Ce système est conservatif en l’absence de frottement et de perte d’énergie. En utilisant les équations de Lagrange, avec $L = T - U$, on obtient une équation pour le mouvement du

pendule :

$$\begin{aligned} \frac{d}{dt} \frac{\partial T}{\partial \dot{\theta}} + \frac{\partial U}{\partial \theta} &= 0 \\ ml^2 \ddot{\theta} + mgl \sin \theta &= 0 \\ \ddot{\theta} + \frac{g}{l} \sin \theta &= 0 \end{aligned} \quad (1.3)$$

La solution de cette équation différentielle dans le temps est :

$$\theta(t) = \theta_0 \cos(\omega_0 t) \quad (1.4)$$

$\omega_0 = \sqrt{g/l}$ est la pulsation, et θ_0 l'angle auquel le pendule est lâché. On peut ainsi écrire la trajectoire du point G, centre gravité de la masse placée au bas de la tige (figure 1.15) :

$$x(t) = l \sin(\theta_0 \cos(\omega_0 t)) \quad (1.5)$$

Cette figure représente un balancier simulé avec PhysX ainsi que la courbe de la trajectoire $x(t)$. On constate un déphasage des oscillations au cours du temps et une faible diminution

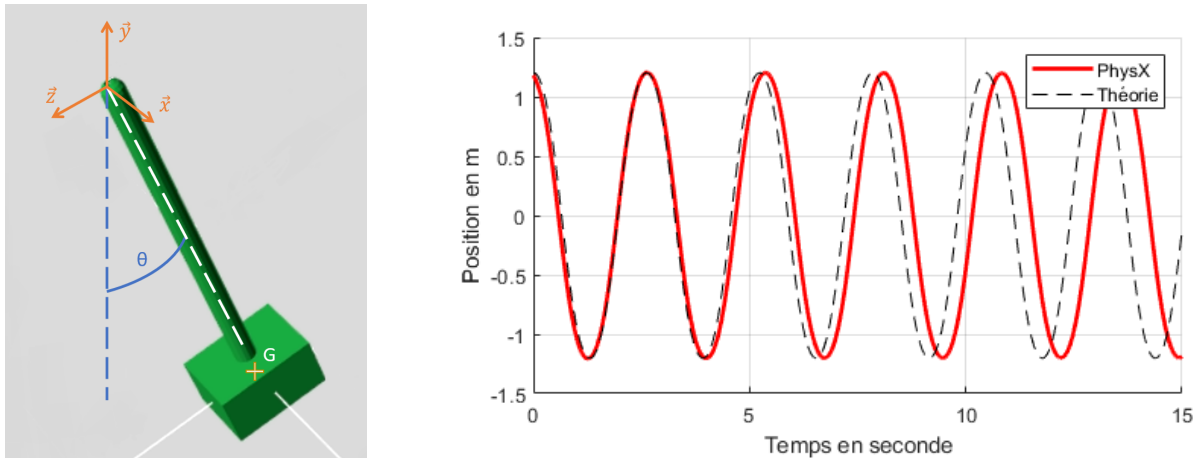


Figure 1.15 – Gauche : Un balancier simulé avec Nvidia PhysX. Droite : Courbe de la trajectoire du point G dans le temps après un lâché à 45°

d'amplitude, qui est de l'ordre de 0.01 m au bout de 15 s . Ceci implique qu'il y a une perte d'énergie mécanique. En effet, le modèle des liaisons de PhysX est en fait un système masse ressort de grande raideur et avec une légère dissipation. De plus, la méthode d'intégration dans PhysX n'est pas symplectique⁸, car cela nécessiterait une analyse complexe des flux d'énergie de chaque système à simuler. Or, l'objectif de PhysX est le rendu visuel réaliste des mécanismes les plus complexes (plusieurs milliers de solides par exemple), beaucoup d'approximations sont donc effectuées afin d'obtenir une simulation plus stable, avec notamment l'ajout de nombreux petits frottements supplémentaires.

Pour éviter une dissipation numérique excessive, les méthodes d'intégration numérique ont été étudiées. Celles-ci sont détaillées dans l'annexe C. S'appuyer sur les moteurs physiques

8. Un intégrateur numérique est dit symplectique s'il conserve la structure hamiltonienne des équations, c'est à dire les expressions des \dot{p}_i et \dot{q}_i (voir chapitre 3.1.2.1)

pour jeux vidéo n'est acceptable que si l'on souhaite obtenir une approximation grossière de la dynamique de la scène. Pour valider nos méthodes de simulation, il faut utiliser d'autres méthodes pour calculer la dynamique des solides et prendre en compte l'énergie mécanique dissipée avec les moteurs physiques.

1.5.3.2 Interaction fluide-solide avec PhysX

Dans Nvidia PhysX, il existe un modèle de la dynamique du fluide qui utilise un système composé de particules. Ce système possède plusieurs paramètres : masse des particules, viscosité, « stiffness », restitution, frottement dynamique et frottement statique. Le paramètre de masse des particules joue sur la densité du système. Le paramètre de viscosité traduit la viscosité des mouvements du fluide. Le paramètre stiffness rend compte de la proximité des particules à l'équilibre, c'est la compressibilité du fluide. Le paramètre restitution gère la force restituée au cours d'une collision. Le paramètre frottement dynamique est lié l'énergie dissipée lors d'un frottement avec un autre objet en fonction de la vitesse. Le paramètre frottement statique ajoute un frottement sec contre le mouvement.

Après une recherche des valeurs de paramètres de stiffness, viscosité et masse des particules conférant au fluide des propriétés proches de celles de l'eau (voir annexes), un test de coulée d'un cube est effectué. Ce test doit permettre de définir s'il existe une interaction entre le fluide et le solide et si cette interaction est suffisamment réaliste. Le cube est un solide de 0.5 m de côté et de densité 7000 kg.m^{-3} . Le test se déroule en 2 étapes, illustrées figure 1.16 :

- une première étape consiste à laisser les 9000 particules se stabiliser. En effet, afin d'initier le simulateur, les particules sont placées selon un quadrillage à une certaine distance à vitesse nulle. Puis, sous l'effet de la gravité, elles se réarrangent pour combler les vides comme décrit sur le schéma de la figure 1.16.
- une seconde étape consiste à faire tomber le cube dans l'eau : un cube est généré au-dessus de la surface, puis tombe et interagit avec les particules qui représentent l'eau.

La figure 1.17 montre les trajectoires de chute du cube obtenues selon deux évolutions de paramètres : l'une avec une évolution de la masse volumique du cube, l'autre avec une évolution de la masse volumique de l'eau. Par défaut, notre référence est la trajectoire d'un cube de densité 7000 kg.m^{-3} dans un fluide de densité 1000 kg.m^{-3} .

Sur cette figure, on constate que la trajectoire des cubes est peu modifiée par l'eau malgré une grande variété de densité. Avec une densité très faible 1 kg.m^{-3} , le cube rebondit sur la surface de l'eau. L'énergie restituée par les particules d'eau est irréaliste car l'altitude finale du cube se trouve au-dessus de son altitude initiale. Le cube de densité 500 kg.m^{-3} coule, ce qui est normal au vu de la vitesse d'entrée, mais il ne remonte pas à la surface. Sa trajectoire change très peu à l'entrée dans l'eau. Effectivement, il n'y a pas de calcul de la poussée d'Archimède sous PhysX. Au vu de la seconde figure, on se rend compte que pour que le fluide freine la coulée du cube il faut choisir une densité immense. Le freinage du cube au cours de sa chute provient de la difficulté de mettre en mouvement des particules d'eau de grande masse, ce qui ressemble à une masse ajoutée (voir chapitre 2), mais ne reflète pas la poussée d'Archimède. Ce logiciel ne permet donc, en aucun cas, de simuler de manière réaliste une interaction fluide-solide.

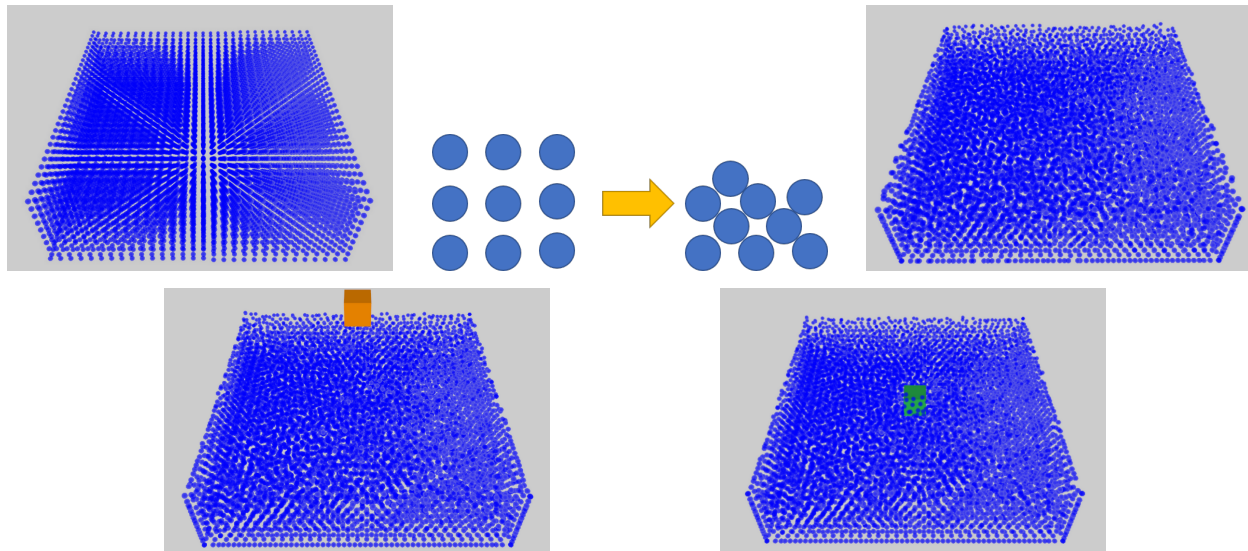


Figure 1.16 – Expérience de chute d’un cube dans Nvidia PhysX. De haut en bas de droite à gauche : Vue de la scène à l’instant initial. Stabilisation des particules. Vue de la scène après stabilisation des particules. Vue de la scène avant la chute du cube. Vue de la scène au cours de la chute du cube.

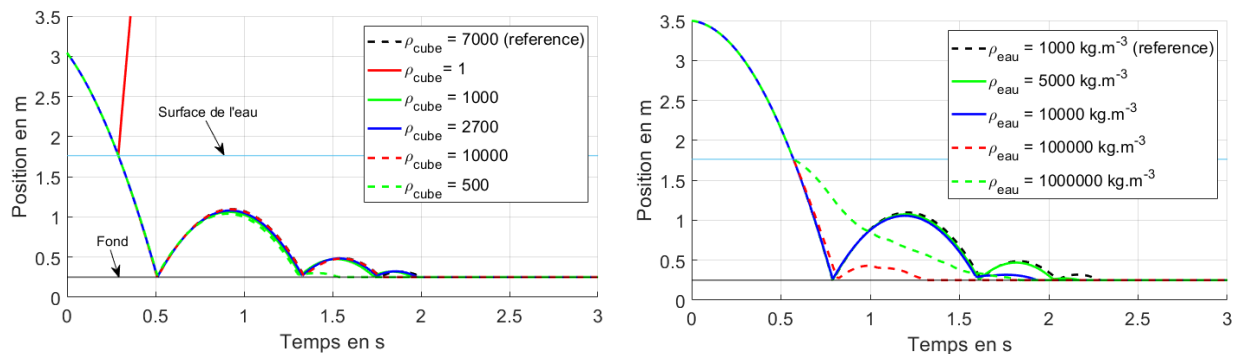


Figure 1.17 – Évolution de l’altitude du cube au cours du temps pour un essai de coulée en fonction de la densité du cube et de l’eau

1.5.3.3 Dynamique des solides et interaction fluide-solide avec FleX

Après PhysX, FleX est le plus récent moteur physique de Nvidia. Il est destiné à la simulation multi-physique complexe et notamment l’interaction entre un fluide et un solide, avec un principe simple : tous les éléments du simulateur sont composés de particules et toutes les interactions se font de particules à particules. Ce logiciel intègre la poussée d’Archimède, ce qui est démontré dans la version de démonstration. Néanmoins, ce logiciel ne permet pas d’ajouter de liaison mécanique. Il y a donc deux options : coder les liaisons mécaniques ou trouver une astuce pour les créer. Pour rajouter une liaison mécanique de type pivot et ainsi recréer le balancier qui avait été codé avec Nvidia PhysX, on peut fixer une ligne de particules dans l’espace. Dans FleX, chaque solide est déformable et il existe une liaison ressort avec une propriété de raideur entre les particules fixes et les autres particules du solide, représenté figure 1.18. La création d’une liaison pivot est par conséquent compliquée et n’est pas forcément réalisable avec les possibilités actuelles, ce qui nécessite de développer une

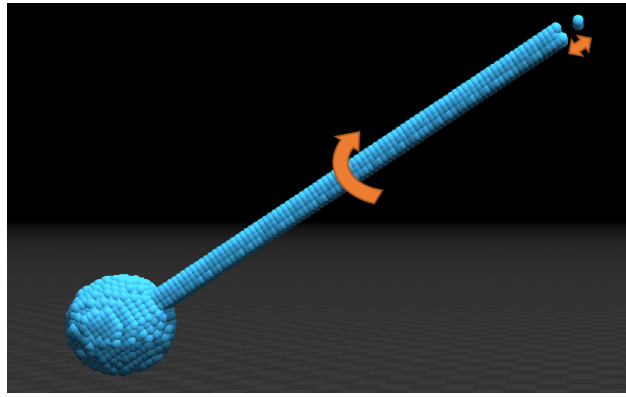


Figure 1.18 – Simulation d'un balancier sous Flex. Illustration du problème lié à la création d'une liaison en fixant les particules d'un solide

nouvelle librairie. Cette dernière option n'a pas été retenue car elle demande un temps de développement chronophage et ne fait pas partie des objectifs de cette thèse.

Un essai de coulée de sphère a également été tenté sous Flex. La poussée d'Archimède fonctionne correctement pour les solides ayant une densité inférieure à celle de l'eau étant donné qu'ils flottent. Cependant, pour les solides dont la densité est supérieure à 1, les trajectoires de coulée ne se rapprochent de la réalité qu'en utilisant des valeurs aberrantes pour la densité des solides (supérieure à 14 fois). La figure 1.19 met en exergue cette situation avec un essai de coulée d'une sphère dans un bassin vertical. La sphère est immergée dans un fluide constitué par des particules. Après stabilisation, la sphère est lâchée et coule. Sa vitesse de chute est observée. Cette simulation permet d'obtenir une vitesse de chute proche de la

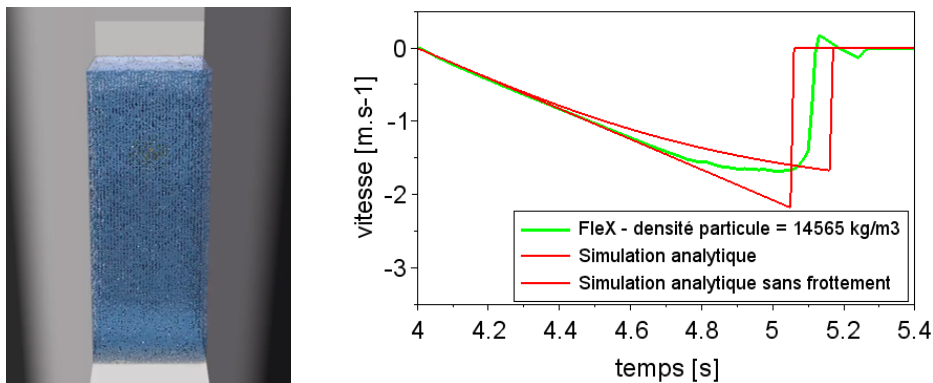


Figure 1.19 – Droite : Vue de la scène de chute d'une sphère simulée avec Flex. Gauche : Courbe de vitesse de coulée de la sphère. Comparaison avec la vitesse théorique de la coulée.

réalité, mais en utilisant des paramètres totalement irréalistes pour la densité du solide.

1.5.4 Conclusion

Cette étude a permis de comparer les logiciels existants pour déterminer le plus adapté pour la simulation de l'interaction dynamique entre un véhicule sous-marin et l'eau qui l'entoure. Les environnements de simulation ont été présentés et quelques détails ont été fournis sur les principaux éléments constitutants : les librairies graphiques, les moteurs physiques et les

middlewares. Ensuite, différents tableaux ont été introduits afin de comparer les différents logiciels existants.

Premièrement, la possibilité d'utiliser un middleware pour inter-connecter différents programmes a été étudiée. Un middleware permettrait, par exemple, de faire une simulation du comportement dynamique dans l'environnement Matlab et d'obtenir un rendu graphique avec Gazebo, Vortex ou Unity. Dans un premier temps cette possibilité est écartée, car elle amènerait des délais de communication et des étapes algorithmiques inutiles.

Deuxièmement, le logiciel choisi doit permettre un certain niveau de personnalisation. Tous les logiciels présentés permettent au minimum d'utiliser des scripts, mais ils ne sont pas tous open-source. Certains d'entre eux ne présentent pas d'interface pour modifier les scènes, ce qui les rend plus difficiles à manier.

Enfin, ces logiciels ont été comparés selon les modèles physiques présents. Dans le domaine de la robotique, aucun logiciel n'intègre un modèle de la dynamique du fluide.

Les moteurs physiques dédiés à l'animation temps réel ne répondent pas à nos besoins, comme cela a été montré dans la partie [1.5.3](#). Ils permettent certes une animation stable en temps réel d'un écoulement, d'un mécanisme ou d'une interaction entre un fluide et un solide, mais au prix d'approximations affectant le réalisme physique. Les logiciels dédiés à la dynamique du fluide et les moteurs hors-ligne sont, a priori, beaucoup plus réalistes mais aussi beaucoup plus lents. Il faudra alors trouver un compromis entre temps de calcul et réalisme physique de la simulation. Une réflexion sur la manière de simuler le comportement dynamique des véhicules marins dans leur milieu naturel est proposée dans la partie suivante.

1.6 Simulation de l'interaction dynamique d'un véhicule sous-marin et de l'eau qui l'entoure

Il existe une myriade de solutions pour la simulation de véhicules sous-marins. Une démarche qui permet d'évaluer le réalisme physique des environnements de simulations sera l'objet de la section 1.6.1. C'est ce réalisme physique qui est le plus important pour notre étude. Ensuite, dans les sections qui suivent, trois solutions seront analysées : utiliser un modèle analytique du véhicule, utiliser un modèle avec un maillage du volume d'eau couplé au modèle du véhicule ou utiliser un modèle avec des particules qui représentent l'eau couplé au modèle du véhicule. L'objectif sera de choisir la méthode qui permettra le meilleur compromis entre le réalisme physique, le temps de calcul et la flexibilité.

1.6.1 Démarche d'évaluation du réalisme physique d'environnement de simulation ayant pour application la robotique sous-marine

Pour rappel, l'objectif de notre simulateur est de simuler de manière fiable la dynamique de véhicules sous-marins. Pour évaluer si un environnement de simulation respecte bien la dynamique, les points suivants seront vérifiés :

1. **Évaluer le comportement du fluide seul dans le simulateur (statique et dynamique).** Cette étape consiste à vérifier si la simulation du fluide s'effectue correctement. Après avoir vérifié que la modélisation du fluide est bien celle d'un d'écoulement et considéré les différentes hypothèses faites pour la simulation de la dynamique du fluide acceptables, il faut vérifier que l'évolution du gradient de pression en fonction de la profondeur est correcte. Cette évolution est donnée dans l'équation :

$$p(z) = \rho_{eau}gz \quad (1.6)$$

Avec p la pression, ρ_{eau} la masse volumique de l'eau, g la constante de gravité et z la profondeur. Il faut également regarder le modèle du fluide et vérifier s'il correspond bien à une équation d'écoulement et regarder les différentes hypothèses faites pour la simulation de la dynamique du fluide.

2. **Évaluer le comportement statique du véhicule immergé.** Avec un gradient de pression correcte, il s'agit maintenant de vérifier que le calcul de la poussée d'Archimède est correct. En faisant un rapide bilan des forces s'exerçant sur le solide, on se rend compte que si l'on arrive à un état d'équilibre à la surface de l'eau alors on a nécessairement équivalence entre la poussée d'Archimède et le poids du solide :

$$\begin{aligned} m \vec{g} &= \rho_{eau} V_{immerge} \vec{g} \\ \rho_{objet} V_{objet} \vec{g} &= \rho_{eau} V_{immerge} \vec{g} \end{aligned} \quad (1.7)$$

Avec ρ_{objet} la masse volumique de l'objet, V_{objet} son volume, \vec{g} le vecteur gravité, $V_{immerge}$ le volume immergé de l'objet. Cette équation 1.7 peut être simplifiée par :

$$\frac{\rho_{objet}}{\rho_{eau}} = \frac{V_{immerge}}{V_{objet}} \quad (1.8)$$

Si l'objet est plus dense que l'eau alors il devra couler. Le simulateur devra donc d'abord valider ce comportement de flottaison avant de passer à l'étape suivante.

3. **Évaluer le comportement dynamique du véhicule immergé.** Un objet plus dense que l'eau sera utilisé. L'objectif est maintenant de vérifier si on est capable de simuler la dynamique d'un véhicule. Cette dernière étape étant très complexe, elle sera découpée en différentes sous-étapes :

- (a) la première sous-étape consiste à valider qu'une vitesse de translation limite due au frottement de l'eau peut être atteinte. Cette vitesse est proportionnelle aux forces de frottement et est atteinte lorsque les forces de frottement compensent la force qui met en mouvement le véhicule. Par exemple, pour la coulée d'un objet, on s'attend à ce que le poids apparent (poids - poussée d'Archimède) devienne à une certaine vitesse égale à la force de frottement :

$$(m_{objet} - \rho_{eau}V_{immerge}g) = K_Q|\dot{z}|\dot{z} \quad (1.9)$$

Avec m_{objet} est la masse de l'objet, K_Q un coefficient de frottement quadratique et \dot{z} la vitesse de coulée. En validant cette étape, on s'assure du caractère homothétique du simulateur.

- (b) Pour la seconde sous-étape, le mouvement obtenu dans le simulateur sera comparé à un essai réel. Notamment, l'évolution temporelle dans le simulateur sera comparée à l'évolution temporelle réelle. Pour cela, un dispositif expérimental sera mis en place : un balancier hydrodynamique (voir Eng [ELLS08]). L'idéal serait de ne pas avoir à entrer de coefficients déterminés empiriquement pour des formes complexes et de pouvoir déterminer ces derniers directement dans notre simulateur.

1.6.2 Méthode utilisant le modèle analytique du véhicule

Pour modéliser l'interaction entre le fluide et le véhicule marin, il est possible d'utiliser un modèle d'équation de mouvement, caractérisant les efforts s'appliquant sur le véhicule. Pour cela il faut faire un bilan de toutes les forces s'appliquant sur le solide, puis les caractériser et analyser la cinétique de ce solide. Le modèle décrit dans le livre de Fossen [Fos11] est un modèle qui permet de modéliser la dynamique d'un véhicule marin ou sous-marin. La dynamique peut être décomposée en deux parties : la cinématique qui est l'étude des effets du mouvement des corps sans que l'on en considère l'origine et la cinétique, qui étudie la mise en mouvement sous l'effet de forces extérieures. C'est d'après l'étude et la combinaison de deux modèles : le modèle cinématique et le modèle cinétique, que Fossen donne une représentation des six degrés de liberté du véhicule dans l'espace en utilisant une notation vectorielle.

$$\underbrace{M_{RB}\dot{\vec{v}} + C_{RB}(\vec{v})\vec{v}}_{\text{Forces du solide}} + \underbrace{M_A\vec{v}_r + C_A(\vec{v}_r)\vec{v}_r + D(\vec{v}_r)\vec{v}_r}_{\text{Forces hydrodynamiques}} + \underbrace{\vec{g}(\vec{\eta}) + \vec{g}_0}_{\text{Forces hydrostatiques}} = \vec{\tau} \quad (1.10)$$

Avec \vec{v} et $\vec{\eta}$ les vecteurs vitesses et positions généralisées du véhicule à 6 degrés de libertés. M_{RB} la matrice d'inertie du solide, $C_{RB}(\vec{v})$ la matrice de l'effet Coriolis de la masse du solide, M_A la matrice des masses ajoutées du solide, C_A la matrice des effets des forces de Coriolis ajoutées, $D(\vec{v}_r)$ les forces de frottement, $\vec{g}(\vec{\eta})$ un vecteur comprenant les forces gravitationnelles et la poussée d'Archimède. Les forces dues aux systèmes de ballast et de réservoir d'eau sont stockées dans le terme \vec{g}_0 . Les forces extérieures sont stockées dans le vecteur τ : cela peut être la force due aux vagues, au vent pour les véhicules de surface, à des chocs, etc. \vec{v}_r est le vecteur vitesse relatif entre le véhicule et un courant irrotationnel :

$$\vec{v}_r = \vec{v} - \vec{v}_c \quad \vec{v}_c = (u_c, v_c, w_c, 0, 0, 0)^T \quad (1.11)$$

Ce modèle est complet et découpe bien les efforts selon 4 types : les forces du solide, c'est à dire les forces dues à l'inertie, les forces hydrodynamiques, qui sont les forces dues au mouvement dans l'eau, les forces hydrostatiques qui sont les forces dues à la poussée d'Archimède, la gravité et les systèmes de flottaison et les autres forces. Il est applicable aussi bien pour les véhicules de surface que pour les véhicules sous-marins, et est utilisé dans différents simulateurs d'environnement marin, comme par exemple Vortex [CML16] ou UWSIM [PPFS12], pour simuler les mouvements des véhicules. Pour les forces du solide, on utilise la masse de l'objet et sa géométrie pour déterminer son inertie et sa force de Coriolis en fonction de sa vitesse. Pour les forces hydrostatiques, on fait de même, on utilise sa masse et sa géométrie pour déterminer son volume, puis sa poussée d'Archimède. Les capacités du système de ballast sont connues. Les forces hydrodynamiques, elles, sont beaucoup plus complexes à modéliser. Les matrices de masses ajoutées et de coefficient de frottement sont le plus souvent obtenues à partir d'expérience réelle, comme dans Eidsvik [ES16] ou Eng [ELLS08] (tel qu'évoqué dans la section précédente 1.6.1). Elles peuvent également être obtenues grâce à des logiciels de calcul par éléments finis comme dans Li [LTS⁺17], où ces logiciels sont utilisés pour valider les expériences réalisées et dans l'article de Gao [GWP⁺18] qui propose une méthode optimisée pour la détermination de ces paramètres avec STAR-CCM+. Pour pouvoir utiliser cette méthode, il faut donc au préalable déterminer des paramètres hydrodynamiques. Ces paramètres et leurs méthodes d'obtention seront détaillés dans le chapitre 2.

1.6.3 Méthode utilisant un modèle avec un maillage de l'eau couplé au véhicule

Un maillage de l'eau entourant le véhicule pourrait être utilisé pour simuler le comportement dynamique des véhicules marins. Il existe deux approches pour la résolution des équations hydrodynamiques : les méthodes eulériennes avec maillage, qui seront développées dans cette partie et les méthodes lagrangiennes sans maillage, qui seront développées dans la prochaine sous-section. Le principe de cette méthode eulérienne est de découper l'espace occupé par le fluide en un maillage de petits éléments, comme illustré sur la figure 1.20. Ceci permet de modéliser des champs (de vitesses, de pression), ce qui est intéressant lorsque l'on cherche à utiliser les équations du fluide comme les équations de Navier-Stokes. En 3 dimensions, il existe

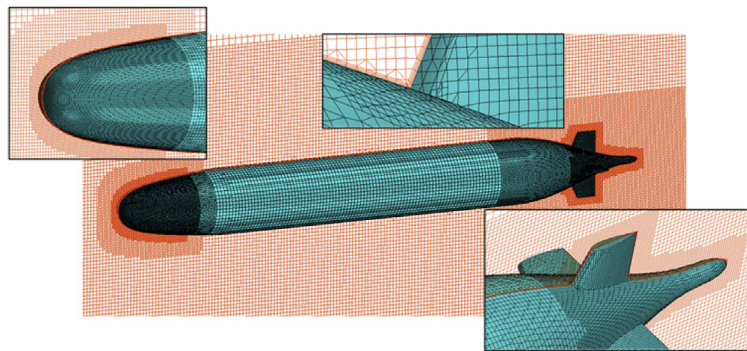


Figure 1.20 – Maillage de l'espace du fluide entourant un véhicule marin pour le calcul de paramètres hydrodynamiques (Source : [GWP⁺18])

différents types de découpage, selon la forme des volumes finis utilisés, comme présentés sur la figure 1.21. Selon ces découpages, le résultat de la simulation peut varier fortement, puisque

certaines petits éléments permettront de mieux prendre en compte la forme de l'objet à mailler. Il sera plus aisé de découper des objets cubiques avec des hexaèdres, mais ces hexaèdres auront plus de mal à épouser les courbes que les tétraèdres ou les polyèdres. Il est donc très important de choisir la forme qui correspond au calcul que l'on veut réaliser. La taille de ces éléments est également très importante car elle est liée à la précision du résultat du calcul. C'est pourquoi l'on voit sur la figure 1.20 que le maillage est raffiné autour des points d'intérêts. La plupart

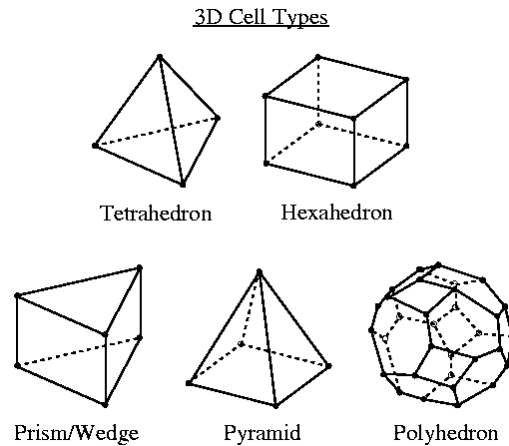


Figure 1.21 – Les différents types de mailles en trois dimensions utilisables pour le maillage (Source : Documentation d'ANSYS 12.0)

du temps, ce principe est utilisé pour calculer un état d'équilibre de l'écoulement d'un fluide. C'est à dire, qu'ils calculent les vitesses et les pressions dans un écoulement qui s'est stabilisé après une période transitoire. Ces calculs sont très efficaces pour ce type de problème. Ils sont néanmoins moins efficaces lorsqu'il s'agit de déterminer une évolution dynamique d'un système. Effectivement, si l'on veut caractériser le mouvement d'un solide et donc son accélération dans le fluide, il faut le déplacer dans l'espace. Actuellement, il existe deux méthodes pour cela : remailler le problème, ou une technique plus récente qui s'appelle overset. La technique la plus ancienne est très couteuse en temps de calcul. La technique la plus récente semble plus intéressante : elle a déjà été utilisée pour l'estimation de paramètres hydrodynamiques pour les AUV dans Phillips [PTF10] ou dans Sukas [SKCG17] pour les véhicules de surface. Dans Phillips, une étude complète des paramètres hydrodynamiques est proposée. Cette étude propose des résultats en 1 mois sur l'obtention des paramètres hydrodynamiques avec entre 10 et 20 % d'erreurs. Dans l'article de Gao [GWP⁺18], une méthode est proposée pour rendre cette estimation moins couteuse en temps. Elle requiert néanmoins plus de 20 heures de calcul. De plus, il est encore plus complexe de simuler la surface libre en plus du véhicule, car cela nécessite l'utilisation de modèles mathématiques avancés, comme c'est le cas dans [Gol18], où le calcul n'est qu'en 2 dimensions et le ratio temps de calcul/temps simulé est de 6000. La simulation avec maillage ne permet pas de simuler une véritable mission ou de tester un algorithme de commande dans un temps raisonnable.

1.6.4 Méthode utilisant un modèle avec des particules représentant l'eau couplé au modèle du véhicule

Pour simuler l'eau qui entoure le véhicule, il est possible d'utiliser un modèle utilisant un système de particules, ce qui correspond à un modèle lagrangien sans maillage du fluide. Elle

consiste à suivre dans le temps les particules fluides le long de leurs trajectoires : c'est une description intuitive de leur mouvement. Elle est assez récente, et date du milieu du xx^e siècle, et de nombreuses études sont menées actuellement pour développer cette technique. C'est une méthode qui est déjà utilisée pour certaines applications, comme l'animation 3D ou les jeux vidéo. Les calculs à réaliser avec cette méthode peuvent être faits en parallèle, comme annoncé dans la section 1.5.3, ce qui permet de diminuer le temps de calcul⁹. Par contre, ce temps réel est souvent atteint au prix de nombreuses approximations et d'une perte du réalisme physique. Certains logiciels spécialisés, où le réalisme physique est important existent, comme Neutrino ou GPUSPH.

Il existe différentes méthodes utilisant des particules :

- Smoothed Particles Hydrodynamics (SPH) est la plus courante.
- Meshless Finite-Mass & Meshless Finite-Volume, voir Hopkins [Hop15]
- Les méthodes de Lattice-Boltzman, où des particules se déplacent le long d'une grille fixe.
- la méthode Particle-In-Cell (PIC) [Har62]
- Fluid-Implicit-Particle (FLIP) [BR86]

SPH sera la méthode utilisée dans cette thèse, les autres méthodes étant trop peu éprouvées pour le moment, et sera détaillée dans le chapitre 3. Cette méthode pourrait s'avérer intéressante, car des applications sur la simulation de vagues dans les zones côtières ont déjà été réalisées et validées, comme dans l'article de Farahani [FDHB14], où toute la circulation dans une zone côtière en fonction de sa morphologie est étudiée. Ceci permet de modéliser les courants internes à la zone côtière, ainsi que les mouvements de particules dans le fluide, comme illustré sur la figure 1.22. Ces simulations sont très encourageantes pour notre application,

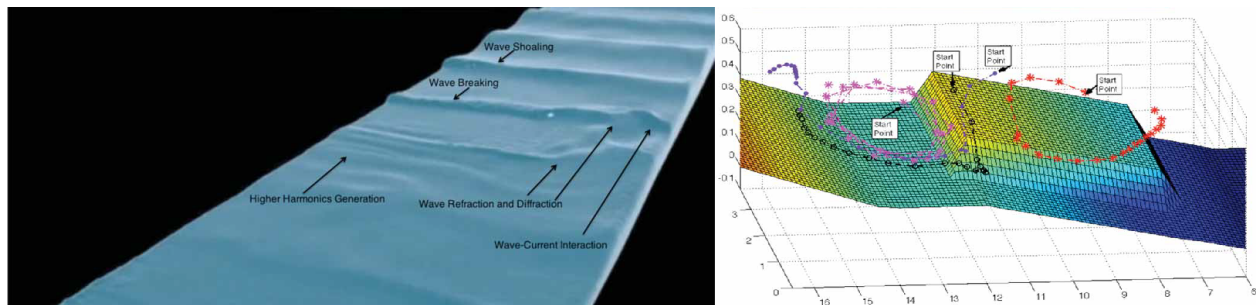


Figure 1.22 – Simulation de la circulation du fluide dans une zone côtière avec un modèle utilisant des particules. Analyse du courant et des vagues. (Source : Farahani [FDHB14], Violeau [VR16])

car elles démontrent qu'il est possible de simuler l'écoulement de l'eau dans toute une zone côtière avec la circulation des vagues.

9. En réalité, l'intérêt du calcul parallèle est de limiter la consommation d'énergie dans le processeur. En effet, le principe des transistors qui composent le processeur est de déplacer des charges électrostatiques entre deux potentiels de tension pour réaliser le calcul. Le travail de ces charges est alors $W = CV^2$, avec C la capacité électrique et V la tension appliquée. La puissance électrique requise pour faire fonctionner le calcul est directement liée à la fréquence à laquelle on déplace ces charges $P = Wf$, f la fréquence. En augmentant le nombre de processeurs et en diminuant leur fréquence, si tant est que le programme est compatible, on diminue la puissance et donc l'énergie nécessaire pour réaliser un même calcul [CPM+95]

1.6.5 Conclusion

Une démarche d'évaluation du réalisme physique du simulateur à concevoir a été proposée. Cette méthode sera appliquée pour valider le fonctionnement de notre simulateur dans les prochains chapitres. Trois méthodes ont été proposées pour notre application de simulation de la dynamique d'un véhicule marin dans un environnement aquatique réaliste : une méthode utilisant le modèle analytique du véhicule (§1.6.2), une méthode utilisant un modèle eulérien avec maillage du fluide (§1.6.3) et une méthode utilisant un modèle lagrangien sans maillage du fluide (§1.6.4). L'objectif de notre simulateur est de permettre une évaluation rapide, suffisamment précise, de la dynamique de prototypes ou d'algorithmes de commande. La méthode utilisant le modèle analytique du véhicule peut permettre la simulation en temps réel du comportement du véhicule marin, mais elle requiert l'obtention de paramètres hydrodynamiques à travers des expériences réelles ou numériques. De plus, elle ne modélise pas le fluide, donc il faudrait utiliser des cartes de courants et prendre en compte les vagues. La méthode utilisant un modèle eulérien avec maillage du fluide permet des calculs très précis, mais est beaucoup trop gourmande en temps de calcul. La méthode utilisant un modèle lagrangien sans maillage du fluide se révèle très prometteuse. Elle permet de simuler le fluide et son interaction avec le solide dans un temps raisonnable. C'est une technique très récente, et pour l'instant, il n'existe pas actuellement d'environnement de simulation complet qui utilise cette méthode. Un accord a cependant été trouvé avec la société Centroid Lab, nous permettant l'accès au code source de calcul du logiciel de simulation développé par la société. Il faut néanmoins valider l'interaction fluide-solide et montrer par exemple que les paramètres hydrodynamiques obtenus avec cette méthode sont valides.

1.7 Approche développée dans ce mémoire

Ce chapitre a expliqué la problématique de cette thèse : **Comment simuler les interactions entre l'eau et un véhicule sous-marin naviguant dans un environnement aquatique réaliste ?** Les différentes contraintes liées à cette problématique ont été décrites : il existe à la fois une grande variété de véhicules sous-marins qu'il faut pouvoir représenter dans le simulateur et de nombreuses solutions pour ce type de simulation. Pour répondre à cette problématique, il existe trois méthodes principales : utiliser le modèle analytique du véhicule (§1.6.2), un modèle eulérien avec maillage du fluide (§1.6.3) ou un modèle lagrangien du fluide (§1.6.4). La solution qui semble être la plus intéressante est le modèle lagrangien sans maillage du fluide avec notamment la technique Smoothed Particles Hydrodynamics (SPH), qui sera détaillée dans le chapitre 3 : Simulation hydrodynamique basée SPH. La capacité de SPH à modéliser des zones côtières et un environnement perturbé a déjà été mise en évidence (§1.6.4). Il faut donc montrer que l'on peut obtenir une simulation réaliste de l'interaction fluide-solide. Notre approche pour confirmer la possibilité d'utiliser ce type de solution pour la simulation de la dynamique de véhicule sous-marin est la suivante, organisée sous forme de chapitres :

- **Étude et identification des paramètres hydrodynamiques à prendre en compte pour la robotique.** Le modèle dynamique des véhicules sous-marins sera présenté dans le chapitre 2 intitulé Modélisation et estimation des paramètres hydrodynamiques. Il décrira les paramètres hydrodynamiques utilisés dans les simulateurs dynamiques classiques. Les effets à l'origine de ces paramètres seront explicités, puis les méthodes permettant l'obtention de ces paramètres seront étudiées. Un dispositif ainsi qu'une nouvelle méthode d'obtention des paramètres hydrodynamiques, nommée NIPF (Numerical Integration Position Fitting), sera proposée. Ceci constituera la première contribution de cette thèse décrite dans ce chapitre.
- **Description du solveur SPH et validation de l'hydrostatique.** La méthode SPH et les solveurs SPH utilisés pour simuler en même temps la dynamique du fluide et celle du solide seront décrits dans le chapitre 3 intitulé Simulation hydrodynamique basée SPH. La démarche adoptée, détaillée dans le paragraphe 1.6.1, consiste à valider tout d'abord le comportement du fluide seul, puis l'interaction du fluide et du solide dans un cas statique et enfin dans un cas dynamique. Les deux premiers points seront vérifiés dans le chapitre 3, tandis que le dernier point fera l'objet du chapitre 4 intitulé Validation du simulateur SPH.
- **Validation du solveur SPH à travers l'obtention de paramètres hydrodynamiques.** Dans le chapitre 4, les paramètres hydrodynamiques simulés avec SPH seront déterminés à l'aide de la méthode NIPF. Une série d'expériences mettra en évidence que ces paramètres sont obtenus, a minima, avec le même ordre de grandeur que ceux du véhicule ou de l'objet réel correspondant. Ceci permettra de valider les capacités du solveur SPH pour la simulation hydrodynamique. Ce solveur pourra alors être utilisé pour vérifier les résultats obtenus dans l'expérimentation détaillée dans le chapitre 2.

Chapitre 2

Modélisation et estimation des paramètres hydrodynamiques

L'évaluation du mouvement des véhicules sous-marins nécessite la compréhension des forces hydrodynamiques.

Dans une première partie, ce chapitre propose d'explicitier l'origine de ces forces intervenant dans la dynamique du véhicule sous-marin (cf. section 1.6.2). Pour cela, le modèle analytique, qui est basé sur une étude cinétique d'un véhicule évoluant sous l'eau, est présenté. Dans ce modèle, les forces s'appliquant sur le solide peuvent être distinguées en deux catégories : les forces hydrostatiques et les forces hydrodynamiques. Si les forces hydrostatiques sont simples à modéliser et dépendent principalement de la géométrie et de la répartition des masses dans le véhicule sous-marin, la modélisation des forces hydrodynamiques, frottements et masse ajoutée, est particulièrement complexe. Ces forces sont modélisées à l'aide de paramètres hydrodynamiques.

La deuxième partie est consacrée aux différents moyens actuels d'estimation des paramètres hydrodynamiques à partir des dispositifs expérimentaux physiques et des moyens numériques variés. Ces moyens sont présentés avec les différentes méthodes d'identification expérimentales qui permettent d'exploiter les résultats obtenus.

La troisième partie est dédiée à la première contribution de la thèse, qui consiste en la mise en œuvre d'une méthode d'identification des paramètres hydrodynamiques à partir des mesures issues d'un balancier hydrodynamique. Les propriétés de ce dispositif et la variabilité des résultats obtenus sont analysées. Ces résultats serviront de base à la validation de notre simulateur basé sur SPH dans le chapitre 4.

2.1 Modèle analytique des véhicules sous-marins

Cette partie présente le modèle cinétique et le modèle dynamique d'un véhicule évoluant en milieu aquatique à partir des actions mécaniques extérieures (forces et moments). Chaque action extérieure est détaillée. Le modèle cinématique des véhicules sous-marins est reporté dans l'annexe A. Le descriptif de cette partie s'inspire fortement de la modélisation présentée dans le livre de Fossen [FS91].

2.1.1 Notations utiles à la modélisation des véhicules sous-marins

Les notations pour les variables qui ont été définies par la Society of Naval Architects and Marine Engineers (SNAME) [oNAS50] sont utilisées pour décrire le mouvement des véhicules marins dans ce manuscrit. Elles sont fixées par rapport à un repère lié au véhicule qui est représenté sur la figure 2.1, et sont résumées dans le tableau 2.1. L'origine de ce repère est au centre du véhicule. L'axe x est orienté vers l'avant du véhicule, l'axe y est orienté vers le coté et l'axe z vers le dessous. Ce repère sera nommé VEH.

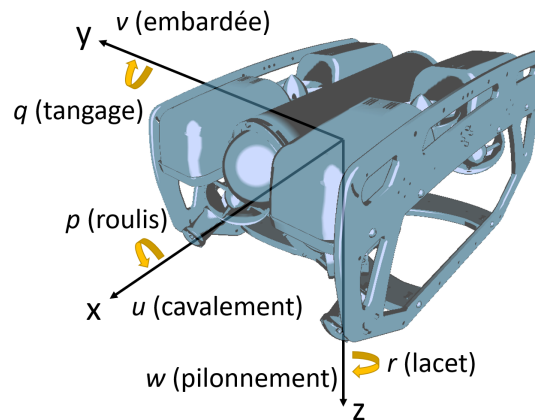


Figure 2.1 – Référentiel lié au véhicule sous-marin

degré de liberté (DDL)	Forces et moments	Vitesses linéaires et angulaires	Positions et angles d'Euler
1 mouvement selon x (cavement)	X	u	x
2 mouvement selon y (embarquée)	Y	v	y
3 mouvement selon z (pilonnement)	Z	w	z
4 rotation autour de x (roulis)	K	p	ϕ
5 rotation autour de y (tangage)	M	q	θ
6 rotation autour de z (lacet)	N	r	ψ

Tableau 2.1 – Les notations de la SNAME pour les véhicules marins

2.1.2 Modèle cinétique des véhicules sous-marins

Afin de décrire le mouvement des véhicules sous-marins soumis aux actions du monde extérieur, notamment hydrostatique et hydrodynamique, il faut étudier leur cinétique. L'objectif de cette section est de montrer que l'on peut simplifier le modèle d'un véhicule sous-marin en une équation comme présenté par Fossen [FS91] :

$$M_{RB} \dot{\vec{v}} + C_{RB}(\vec{v}) \vec{v} = \vec{\tau}_{RB} \quad (2.1)$$

Avec M_{RB} la matrice des masses du solide, C_{RB} la matrice des effets centripètes et de Coriolis due à la rotation du repère du solide VEH autour du repère du centre de gravité G , $\vec{v} = (u, v, w, p, q, r)^T$ le vecteur vitesse généralisé dans le repère du véhicule et $\vec{\tau}_{RB} = (X, Y, Z, K, M, N)^T$ le vecteur des forces et moments généralisé dans le repère VEH.

2.1.2.1 Application des lois de Newton-Euler

Dans le cas d'une masse m constante, la seconde loi de Newton relie cette dernière, l'accélération du solide $\dot{\vec{v}}_{S,G}$ à son centre de gravité G et la somme des forces s'appliquant sur le solide $\sum \vec{f}_{S,G}$ à son centre de gravité G :

$$m \dot{\vec{v}}_{S,G} = \sum \vec{f}_{S,G} \quad (2.2)$$

De la même manière, les travaux d'Euler ont montré que l'accélération angulaire d'un solide autour de son point d'origine $\dot{\vec{\omega}}_{S,G}$ est liée à sa matrice d'inertie calculée au centre de gravité I_G , qui est constante, et à la somme des moments s'appliquant sur son centre de gravité $\sum \vec{m}_{S,G}$.

$$I_G \dot{\vec{\omega}}_{S,G} = \sum \vec{m}_{S,G} \quad (2.3)$$

Pour dériver la vitesse d'un solide dans un référentiel non-inertiel, tel que le référentiel VEH, il faut prendre en compte sa rotation par rapport à un référentiel inertiel, tel que le référentiel Est-Nord-Haut (East-North-Up : ENU, voir annexe A). Ainsi, en notant i le référentiel inertiel et S le référentiel du véhicule, la dérivée de la vitesse du véhicule dans le repère inertiel est :

$$\frac{{}^i d}{dt} \vec{v} = \frac{{}^S d}{dt} \vec{v} + \vec{\omega}_{S/i} \wedge \vec{v} \quad (2.4)$$

Avec $\vec{\omega}_{S/i}$ la vitesse de rotation entre le référentiel inertiel et le référentiel du véhicule. L'effet dû à la rotation de la terre, qui pour rappel tourne à $\omega_T = 7.2921 \times 10^{-5} \text{rad.s}^{-1}$, est tout à fait négligeable par rapport aux efforts hydrodynamiques.

2.1.2.2 Translation du centre de gravité

La figure 2.2 montre un point origine arbitraire O et le centre de gravité CG d'un véhicule sous-marin et leur positionnement dans un référentiel inertiel. A partir de cette figure on peut écrire :

$$\vec{r}_{g/i} = \vec{r}_{S/i} + \vec{r}_g \quad (2.5)$$

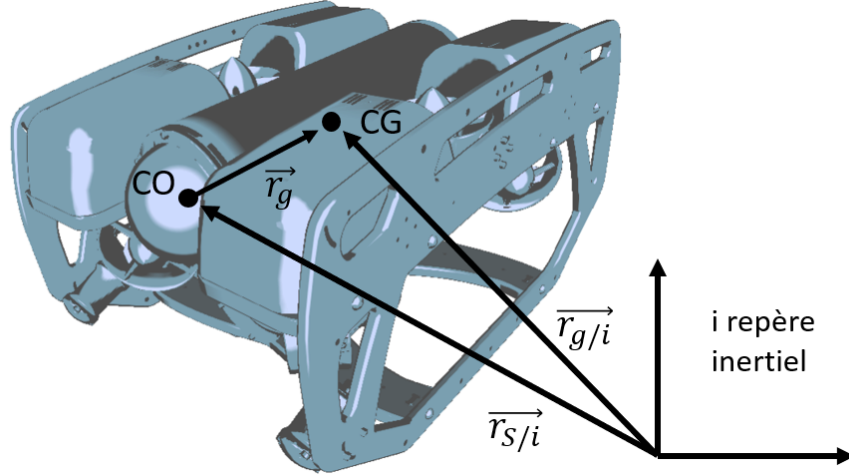


Figure 2.2 – Présentation des repères liés au véhicule marin. Le repère fixe par rapport au centre de gravité (CG) décalé du référentiel arbitraire de centre O (CO).

Avec \vec{r}_g le vecteur qui parcourt la distance entre O et CG. En dérivant l'équation 2.5 et en considérant le repère inertiel :

$$\vec{v}_{g/i} = \vec{v}_{S/i} + \left(\frac{S d}{dt} \vec{r}_g + \vec{\omega}_{S/i} \wedge \vec{r}_g \right) \quad (2.6)$$

Pour un objet rigide la distance entre le centre de gravité CG et l'origine O est fixe donc :

$$\frac{S d}{dt} \vec{r}_g = 0 \quad (2.7)$$

ainsi l'équation 2.6 devient :

$$\vec{v}_{g/i} = \vec{v}_{S/i} + \vec{\omega}_{S/i} \wedge \vec{r}_g \quad (2.8)$$

En reprenant l'équation 2.2 on peut écrire :

$$\begin{aligned} \sum \vec{f}_{S,G} &= \frac{i d}{dt} (m \vec{v}_{g/i}) \\ &= \frac{i d}{dt} (m \vec{v}_{g/i}) + m \vec{\omega}_{S/i} \wedge \vec{v}_{g/i} \\ &= m (\dot{\vec{v}}_{g/i}) + \vec{\omega}_{S/i} \wedge \vec{v}_{g/i} \\ &= m (\dot{\vec{v}}_{g/i}) + S(\vec{\omega}_{S/i}) \vec{v}_{g/i} \end{aligned} \quad (2.9)$$

Avec S l'opérateur qui permet de transformer $\vec{\omega}_{S/i}$ pour écrire le produit vectoriel :

$$S(\vec{\omega}_{S/i}) \vec{v}_{g/i} = \vec{\omega}_{S/i} \wedge \vec{v}_{g/i} = \begin{pmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (2.10)$$

2.1.2.3 Rotation autour du centre de gravité

Pour les mouvements de rotation, une approche similaire à celle utilisée pour les translations peut être suivie. A partir de la seconde loi de Newton ([Équation 2.3](#)), on relie la résultante des moments s'appliquant sur le solide au centre de gravité \vec{M}_g aux vitesses de rotation $\vec{\omega}_{S/i}$:

$$\begin{aligned}\vec{M}_g &= \frac{{}^i d}{dt} (I_G \vec{\omega}_{S/i}) \\ &= \frac{{}^b d}{dt} (I_G \vec{\omega}_{S/i}) + \vec{\omega}_{S/i} \wedge (I_G \vec{\omega}_{S/i}) \\ &= I_G \dot{\vec{\omega}}_{S/i} + \vec{\omega}_{S/i} \wedge (I_G \vec{\omega}_{S/i}) \\ \vec{M}_g &= I_G \dot{\vec{\omega}}_{S/i} - S (I_G \vec{\omega}_{S/i}) \vec{\omega}_{S/i}\end{aligned}\quad (2.11)$$

Avec I_G la matrice d'inertie du solide définie par :

$$I_G = \begin{pmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{pmatrix}, \quad I_G = I_G^T > 0 \quad (2.12)$$

où I_{xx} , I_{yy} et I_{zz} sont les moments d'inertie du solide autour des axes x_b , y_b et z_b , et $I_{xy} = I_{yx}$, $I_{xz} = I_{zx}$ et $I_{yz} = I_{zy}$ sont les produits d'inertie.

2.1.2.4 Équation du mouvement au centre de gravité

En combinant les deux équations [2.9](#) et [2.11](#), on peut représenter le modèle cinétique du véhicule sous-marin au centre de gravité :

$$M_{RB}^{CG} \begin{pmatrix} \dot{\vec{v}}_{g/i} \\ \dot{\vec{\omega}}_{S/i} \end{pmatrix} + C_{RB}^{CG} \begin{pmatrix} \vec{v}_{g/i} \\ \vec{\omega}_{S/i} \end{pmatrix} = \begin{pmatrix} \vec{F}_g \\ \vec{M}_g \end{pmatrix} \quad (2.13)$$

Avec M_{RB}^{CG} la matrice inertielle du solide au centre de gravité et C_{RB}^{CG} la matrice des effets centripètes et Coriolis du solide au centre de gravité. De manière plus détaillée :

$$\underbrace{\begin{pmatrix} mI_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_G \end{pmatrix}}_{M_{RB}^{CG}} \begin{pmatrix} \dot{\vec{v}}_{g/i} \\ \dot{\vec{\omega}}_{S/i} \end{pmatrix} + \underbrace{\begin{pmatrix} mS(\vec{\omega}_{S/i}) & 0_{3 \times 3} \\ 0_{3 \times 3} & -S(I_G \vec{\omega}_{S/i}) \end{pmatrix}}_{C_{RB}^{CG}} \begin{pmatrix} \vec{v}_{g/i} \\ \vec{\omega}_{S/i} \end{pmatrix} = \begin{pmatrix} \vec{F}_g \\ \vec{M}_g \end{pmatrix} \quad (2.14)$$

Parfois il est intéressant d'exprimer la cinétique du véhicule à un autre point arbitraire, ceci est proposé dans l'annexe [G](#).

2.1.3 Modèle dynamique d'un véhicule sous-marin

Au chapitre [1](#), le modèle dynamique du véhicule a été évoqué dans la section [1.6.2](#). Les différentes forces s'appliquant sur un véhicule sous-marin quelconque sont :

- Les forces de frottement
- Les forces de masse ajoutée
- La gravité

- La poussée d'Archimède
- Les effets dus au ballast
- Les forces des vagues
- Les forces dues aux actionneurs

Un modèle simplifié de la dynamique d'un véhicule sous-marin, en suivant la méthode décrite dans Fossen [Fos11] est donc :

$$M \dot{\vec{v}} + C(\vec{v}) \vec{v} + D(\vec{v}) \vec{v} + \vec{g}(\vec{\eta}) + \vec{g}_0 = \vec{\tau} + \vec{\tau}_{vague} \quad (2.15)$$

Avec :

$M = M_{RB} + M_A$	inertie du véhicule incluant celle du solide M_{RB} et celle ajoutée par le fluide M_A .
$C(\vec{v}) = C_{RB}(\vec{v}) + C_A(\vec{v})$	effets Coriolis et centripète du véhicule incluant ceux du solide C_{RB} et ceux ajoutés par le fluide C_A .
$D(\vec{v})$	forces de frottement
$\vec{g}(\vec{\eta})$	forces de pesanteur et de poussée d'Archimède.
\vec{g}_0	forces du système de ballast
$\vec{\tau}$	forces extérieures (exemple : actionneurs)
$\vec{\tau}_{vague}$	forces dues aux vagues

Cette équation peut être réécrite dans la forme montrée dans la section 1.6.2 (Équation 1.10), pour permettre de séparer les différentes forces selon leur type. Pour rappel, cette équation est :

$$\underbrace{M_{RB} \dot{\vec{v}} + C_{RB}(\vec{v}) \vec{v}}_{\text{Forces du solide}} + \underbrace{M_A \vec{v}_r + C_A(\vec{v}_r) \vec{v}_r + D(\vec{v}_r) \vec{v}_r}_{\text{Forces hydrodynamiques}} + \underbrace{\vec{g}(\vec{\eta}) + \vec{g}_0}_{\text{Forces hydrostatiques}} = \vec{\tau} + \tau_{vague}$$

Dans la suite, les forces hydrostatiques, de frottement et de masse ajoutée sont explicitées.

2.1.3.1 Forces hydrostatiques s'appliquant sur un véhicule sous-marin

De manière standard les forces hydrostatiques s'appliquant sur un véhicule sous-marin, sont le poids \vec{P} et la poussée d'Archimède, noté \vec{B} pour « Buoyancy ». Ces forces s'expriment, d'après la SNAME [oNAS50] :

$$\vec{P} = m \vec{g} = P \vec{z} \quad (2.16)$$

$$\vec{B} = -\rho V_{VEH} \vec{g} = B \vec{z} \quad (2.17)$$

Avec ρ la densité de l'eau entourant le véhicule, V_{VEH} le volume d'eau déplacée par le véhicule, m la masse totale du véhicule et $\vec{g} = (0, 0, g)^T$ le vecteur d'accélération gravitationnelle. Ces deux forces ne sont pas toujours appliquées au même endroit sur le véhicule, comme représenté sur la figure 2.3. Le centre de gravité (CG) diffère souvent du centre de poussée d'Archimède noté ici CB. Par commodité, il faut alors déplacer toutes les actions mécaniques dans un même repère inertiel. Ces actions sont exprimées au centre de gravité CG. Le vecteur \vec{r}_b va du point CG vers le point CB. Le vecteur $6 \times 1 \vec{g}(\vec{\eta})$ pour le repère CG s'exprime :

$$\vec{g}(\vec{\eta}) = - \left(\begin{array}{c} \vec{P} + \vec{B} \\ \vec{r}_b \wedge \vec{B} \end{array} \right) \quad (2.18)$$

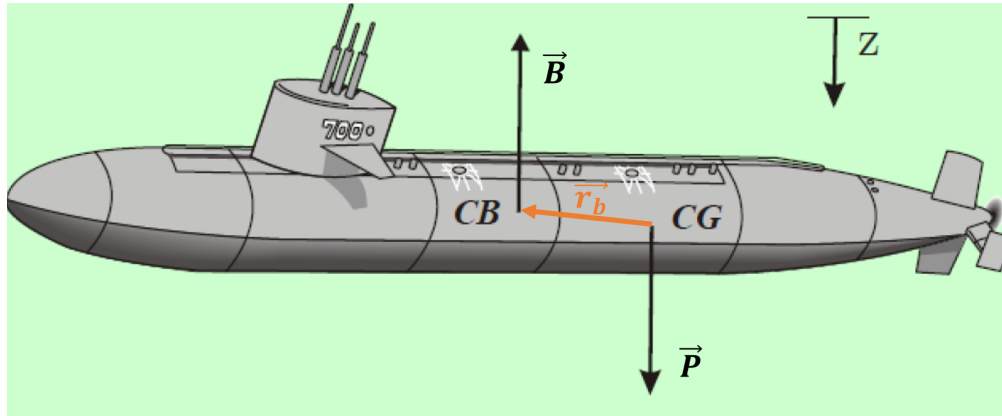


Figure 2.3 – Centre de poussée d’Archimède CB et centre de gravité CG (Source Fossen [Fos11])

2.1.3.2 Force de masse ajoutée

Lorsqu’un corps se déplace dans un fluide, une certaine quantité de fluide doit se déplacer autour de lui. Lorsque le corps accélère, le fluide doit lui aussi accélérer. Ainsi, il faut plus de forces pour accélérer un corps dans un fluide que dans le vide. Puisque la force est égale à la masse fois l’accélération on modélise cette force additionnelle avec un terme de masse ajoutée imaginaire. En utilisant les notations SNAME (voir sous-section 2.1.1), la matrice de masse ajoutée M_A peut être écrite ainsi :

$$M_A = \begin{pmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{pmatrix} \quad (2.19)$$

Cette matrice M_A est semi-définie positive¹. On introduit ici une notation dérivée de celle de la SNAME. Par exemple, pour la force de masse ajoutée Y sur l’axe y , due à l’accélération \dot{u} dans la direction x , on écrit :

$$Y = -Y_{\dot{u}}\dot{u} \implies Y_{\dot{u}} = \frac{\partial Y}{\partial \dot{u}} \quad (2.20)$$

1. Une matrice réelle symétrique est dite semi-définie positive si toutes ses valeurs propres sont positives ou nulles. Une matrice réelle symétrique est dite définie positive si toutes ses valeurs propres sont strictement positives.

La matrice d'effets centripètes et Coriolis ajoutés C_A se déduit directement de la matrice M_A (voir Fossen [Fos11]) :

$$C_A(\vec{v}) = \begin{pmatrix} 0 & 0 & 0 & 0 & -a_3 & a_2 \\ 0 & 0 & 0 & a_3 & 0 & -a_1 \\ 0 & 0 & 0 & -a_2 & a_1 & 0 \\ 0 & -a_3 & a_2 & 0 & -b_3 & b_2 \\ a_3 & 0 & -a_1 & b_3 & 0 & -b_1 \\ -a_2 & a_1 & 0 & -b_2 & b_1 & 0 \end{pmatrix} \quad (2.21)$$

Avec :

$$a_1 = X_{\dot{u}}u + X_{\dot{v}}v + X_{\dot{w}}w + X_{\dot{p}}p + X_{\dot{q}}q + X_{\dot{r}}r \quad (2.22)$$

$$a_2 = Y_{\dot{u}}u + Y_{\dot{v}}v + Y_{\dot{w}}w + Y_{\dot{p}}p + Y_{\dot{q}}q + Y_{\dot{r}}r \quad (2.23)$$

$$a_3 = Z_{\dot{u}}u + Z_{\dot{v}}v + Z_{\dot{w}}w + Z_{\dot{p}}p + Z_{\dot{q}}q + Z_{\dot{r}}r \quad (2.24)$$

$$b_1 = K_{\dot{u}}u + K_{\dot{v}}v + K_{\dot{w}}w + K_{\dot{p}}p + K_{\dot{q}}q + K_{\dot{r}}r \quad (2.25)$$

$$b_2 = M_{\dot{u}}u + M_{\dot{v}}v + M_{\dot{w}}w + M_{\dot{p}}p + M_{\dot{q}}q + M_{\dot{r}}r \quad (2.26)$$

$$b_3 = N_{\dot{u}}u + N_{\dot{v}}v + N_{\dot{w}}w + N_{\dot{p}}p + N_{\dot{q}}q + N_{\dot{r}}r \quad (2.27)$$

Il n'y a donc pas besoin d'identifier de coefficient supplémentaire pour se servir de cette matrice C_A .

Il est possible de simplifier la matrice de masse ajoutée si le corps possède des plans de symétrie. De la même manière que l'on peut simplifier les matrices d'inertie d'un corps lorsque celui-ci possède des symétries. Très souvent, les véhicules ont 3 plans de symétrie et on peut négliger tous les termes non-diagonaux dans la matrice de masse ajoutée. Cette matrice devient alors :

$$M_A = \begin{pmatrix} X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_{\dot{w}} & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{\dot{p}} & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{\dot{q}} & 0 \\ 0 & 0 & 0 & 0 & 0 & N_{\dot{r}} \end{pmatrix} \quad (2.28)$$

2.1.3.3 Force de frottement

Les forces de frottement sont la résultante de nombreux effets hydrodynamiques qui seront expliqués dans la section 2.2.2. En utilisant les notations SNAME (voir sous-section 2.1.1), la matrice d'amortissement $D(\vec{v}_r)$ peut s'écrire ainsi :

$$D(\vec{v}_r) = \begin{pmatrix} D_{X,u} & D_{X,v} & D_{X,w} & D_{X,p} & D_{X,q} & D_{X,r} \\ D_{Y,u} & D_{Y,v} & D_{Y,w} & D_{Y,p} & D_{Y,q} & D_{Y,r} \\ D_{Z,u} & D_{Z,v} & D_{Z,w} & D_{Z,p} & D_{Z,q} & D_{Z,r} \\ D_{K,u} & D_{K,v} & D_{K,w} & D_{K,p} & D_{K,q} & D_{K,r} \\ D_{M,u} & D_{M,v} & D_{M,w} & D_{M,p} & D_{M,q} & D_{M,r} \\ D_{N,u} & D_{N,v} & D_{N,w} & D_{N,p} & D_{N,q} & D_{N,r} \end{pmatrix} \quad (2.29)$$

Avec, par exemple pour le coefficient de frottement $D_{X,v}$ donnant une force X_D selon l'axe x dû à la vitesse v selon l'axe y :

$$X_D = D_{X,v}v = (-X_{L,v} - X_{Q,v} |v|)v \quad (2.30)$$

On adopte ici une notation qui différencie deux types de frottement : ceux qui sont proportionnels à la vitesse qui modélisent au mieux les forces d'amortissement à faible nombre de Reynolds et ceux qui sont proportionnels à la vitesse au carré qui modélisent les frottements à haut nombre de Reynolds, ce qui sera expliqué dans la section 2.2.2. $X_{L,v}$ et $X_{Q,v}$ représentent respectivement les frottements linéaires et quadratiques. Une approximation très courante est de considérer que cette matrice des forces de frottement (Équation 2.29) est diagonale [Fos11]. Cette matrice se réécrit alors :

$$D(\vec{v}_r) = \begin{pmatrix} D_{X,u} & 0 & 0 & 0 & 0 & 0 \\ 0 & D_{Y,v} & 0 & 0 & 0 & 0 \\ 0 & 0 & D_{Z,w} & 0 & 0 & 0 \\ 0 & 0 & 0 & D_{K,p} & 0 & 0 \\ 0 & 0 & 0 & 0 & D_{M,q} & 0 \\ 0 & 0 & 0 & 0 & 0 & D_{N,r} \end{pmatrix} \quad (2.31)$$

2.2 Description et état de l'art des forces hydrodynamiques

La section précédente a mis en évidence l'existence de deux matrices : une qui décrit les forces de frottement et une qui décrit les forces de masse ajoutée. Cette section explique les phénomènes à l'origine de ces forces et donne des valeurs, si possible, théoriques mais le plus souvent empiriques de coefficient applicable pour des formes génériques, notamment pour la sphère.

2.2.1 Forces de masse ajoutée

La section précédente (2.1.3) a montré l'existence de deux matrices liées au phénomène de masse ajoutée : C_A et M_A . L'origine de ce phénomène est ici expliquée. Pour déplacer un solide dans un fluide, il faut également déplacer le fluide qui se trouve autour du solide. En considérant un mouvement de faible amplitude, unidirectionnel et dans un fluide non turbulent, l'apparition de forces de frottement peut être négligée. Une force \vec{F}_A modélise celle qui est nécessaire à la mise en mouvement du fluide :

$$mI_3 \vec{v} = \sum \vec{F} + \vec{F}_A \quad (2.32)$$

Avec $\sum \vec{F}$ la somme des forces hormis \vec{F}_A s'appliquant sur le solide. Cette force supplémentaire \vec{F}_A est directement reliée à la masse du fluide à déplacer. Elle est due au travail de l'énergie cinétique de l'eau à mettre en mouvement. Dans l'air cette force est souvent négligeable, vu que la masse volumique de l'air (en moyenne $1,292 \text{ kg.m}^{-3}$) est négligeable comparée à la masse volumique des solides à déplacer. Dans le cas de l'eau, qui a une masse volumique d'environ 1000 kg.m^{-3} , cette masse à déplacer n'est pas souvent négligeable. Cette masse ajoutée dépend de la forme de l'objet qui se déplace².

Ce coefficient peut être calculé pour tous types de solides. Pour les formes les plus simples on peut le calculer analytiquement, comme cela est décrit dans le rapport de Brennen [Bre82] et comme cela sera décrit dans la suite. Pour les formes plus complexes, on peut se servir de calculs par éléments finis. La méthode utilisée est alors la méthode des éléments finis de frontière, qui s'appelle « Boundary Element Method » (BEM). Cette méthode discrétise la surface du solide et permet d'y calculer des intégrales. Elle est donc particulièrement intéressante dans le cas où le domaine de fluide autour du solide peut être considéré comme s'étendant vers l'infini et est détaillée dans la section 2.3.1.

Dans le cas d'un mouvement de grande amplitude, dans le cas où la circulation du fluide génère des turbulences ou encore dans le cas où le volume de fluide entourant l'objet n'est pas grand ce coefficient n'est pas valable car le travail de l'énergie cinétique de l'eau n'est plus le même. Dans ce cas, cette masse est très difficile à estimer et elle l'est le plus souvent expérimentalement. Dans sa thèse, Berkite [Ber72], étudie l'évolution de cette masse ajoutée pour des objets simples, comme pour les cylindres ou les sphères en fonction de leur proximité avec la surface ou le fond.

Exemple 1 : calcul de la masse ajoutée pour une sphère. Pour calculer la masse ajoutée d'une sphère, on peut se baser sur la théorie des écoulements potentiels. Pour un

2. Mais pas seulement : elle dépend également de l'écoulement dans lequel est placé l'objet

écoulement irrotationnel, on introduit un potentiel d'écoulement ϕ et on utilise l'équation de Laplace :

$$\Delta\phi = 0 \quad (2.33)$$

Il faut alors calculer l'écoulement qui se trouve autour de la sphère si celle-ci se trouve dans un volume infini, en sachant qu'à sa surface la vitesse de l'écoulement est nulle, puis appliquer la formule de Bernoulli en transitoire ([Bat10]) pour obtenir la pression exercée par le fluide sur la sphère :

$$\frac{\partial\phi^*}{\partial t} + \frac{1}{2}u^2 + \frac{p}{\rho} + gz = f(t) \quad (2.34)$$

Cette équation se simplifie en intégrant la fonction $f(t)$ dans le potentiel de vitesse [Bat10] :

$$\phi = \phi^* - \int_{t_0}^t f(\tau)d\tau \quad (2.35)$$

$$\frac{\partial\phi}{\partial t} + \frac{1}{2}u^2 + \frac{p}{\rho} + gz = 0 \quad (2.36)$$

En intégrant cette pression sur la surface, la force que le fluide exerce sur la sphère est obtenue. Ce calcul est détaillé dans l'annexe H. Cette force supplémentaire est souvent considérée comme une masse supplémentaire à déplacer pour le solide :

$$m_{a,Sphere} = \frac{2}{3}\rho\pi R^3\dot{U} \quad (2.37)$$

Ce résultat peut être retrouvé dans la plupart des articles et livres d'hydrodynamique traitant de la masse ajoutée des sphères (ex : [Gra07]), mais :

- Ce résultat n'est valable que dans le cas d'un mouvement dans un fluide parfaitement stationnaire. En effet, si le potentiel d'écoulement ϕ est modifié alors le résultat est modifié, comme cela est également noté par Konstantinidis [Kon13] dans son article. En prenant le cas de l'accélération de la sphère contre un écoulement stationnaire, en reprenant le terme ϕ calculé à l'équation H.16 et l'écoulement H.17, on obtient une force :

$$\begin{aligned} \vec{F}_x &= -2\rho\pi R^2 \int_0^\pi \left(\frac{3R}{2}\dot{U} \cos\theta + \frac{9u^2}{8} \sin^2\theta \right) \cos\theta \sin\theta d\theta \vec{e}_x \\ \vec{F}_x &= -2\rho\pi R^2 \left(\frac{3R}{2}\dot{U} \int_0^\pi \cos^2\theta \sin\theta d\theta + \frac{9u^2}{8} \int_0^\pi \cos\theta \sin^3\theta d\theta \right) \vec{e}_x \\ \vec{F}_x &= -2\rho\pi R^3\dot{U} \vec{e}_x \end{aligned} \quad (2.38)$$

- Ce résultat néglige tout effet de force visqueuse pouvant s'exercer dans le fluide. Ainsi s'il y a une recirculation ou un tourbillonnement du fluide ce résultat n'est plus valable.
- Cette circulation du fluide autour de la sphère ne prend pas en compte les effets de la gravité qui pourraient modifier sa circulation. Ces effets pourraient être pris en compte dans l'équation H.3. On noterait alors :

$$p = \rho \left(\frac{\partial\phi}{\partial t} + \frac{1}{2}|\vec{\nabla}\phi|^2 + gz \right) \quad (2.39)$$

Il faudrait alors prendre en compte la variation de hauteur dans l'écoulement autour de la sphère pour calculer la force de masse ajoutée.

les coefficients $X_{\dot{u}}$, $Y_{\dot{v}}$ et $Z_{\dot{w}}$ de la matrice M_A , décrite dans la section 2.1.3, ont été calculés pour une sphère. Puisque celle-ci est parfaitement symétrique, le coefficient de masse ajoutée est indépendant de l'axe de translation. De même, pour ce qui est des trois coefficients d'inertie ajoutée $K_{\dot{p}}$, $M_{\dot{q}}$ et $N_{\dot{r}}$, puisque la sphère est parfaitement symétrique, il n'y a pas d'inertie ajoutée lors de sa mise en rotation et ces trois coefficients sont nuls.

2.2.1.0.1 Exemple 2 : calcul de la masse ajoutée pour une capsule orientée dans la direction d'avance. La capsule est une forme 3D assez simple constituée d'un cylindre surmonté de deux hémisphères, représentée sur la figure 2.4. Cette forme peut aussi être

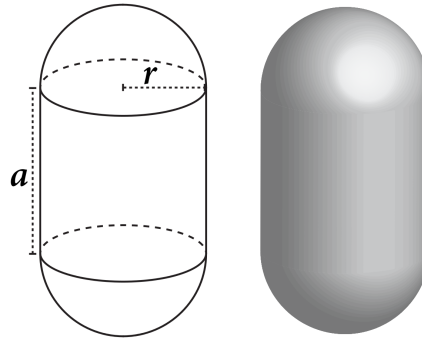


Figure 2.4 – Géométrie de la capsule

appelée « spherocylinder » dans la littérature anglaise, en français : sphérocyllindre³. Il n'existe que peu d'informations sur cette géométrie et aucun calcul théorique de la masse ajoutée, ni de calcul théorique de la circulation du flux d'eau autour de cette forme. La difficulté provient, par exemple, des différents angles d'attaque possibles, qui font certainement varier les paramètres hydrodynamiques, ainsi que des variations du ratio $a+2r/r$, qui influe sur la circulation autour de la capsule. Mishra [MVB11] a évalué la masse ajoutée (et les coefficients de frottement) d'un véhicule sous-marin ayant une forme proche de celle d'une capsule avec un logiciel CAO. On trouve parfois des articles qui traitent de capsule mais la forme utilisée est en fait celle d'ellipsoïde oblong, ce qui ne correspond pas à notre définition de la capsule, comme par exemple dans Lac [LMBB07]. D'autres articles portant sur l'hydrodynamique des sphéroïdes traitent de ce problème d'ellipsoïde oblong, comme Sanjeevi [SP17].

Pour calculer la masse ajoutée d'une capsule allant dans la direction de son axe principal il faut calculer l'écoulement potentiel autour de la capsule. Pour cela il faut calculer la solution de l'équation de Laplace, ce qui entraîne plusieurs complications. La première est que la capsule est un assemblage de deux hémisphères et d'un cylindre et qu'il est impossible d'utiliser en même temps les coordonnées sphériques et les coordonnées cylindriques pour simplifier la résolution du calcul, comme cela est fait pour la sphère en annexe H. La seconde est que ces conditions de frontière, dues à la géométrie de la capsule qu'il faudrait réussir à prendre en compte, ne sont pas unifiables en un polynôme.

3. En français, ce terme est extrêmement rare

2.2.2 Forces de frottement et de portance

La section 2.1.3 a présenté la d'une matrice de force de frottement ou d'amortissements notée $D(\vec{v}_r)$. Ces frottements ou amortissements sont le résultat de différents phénomènes dus au caractère visqueux du fluide. Ces phénomènes sont de différentes natures : frottements surfaciques, amortissements dus à l'écoulement tourbillonnaire (sillage) et forces de portance.

2.2.2.1 Frottements surfaciques

Les frottements surfaciques sont dus à l'écoulement visqueux proche de la surface du solide dans la couche limite. La couche limite est la zone d'interface entre un corps et le fluide environnant lors d'un mouvement relatif entre les deux. Dans cette couche, la vitesse du fluide évolue entre la vitesse du solide, 0 s'il est fixe⁴, et celle du fluide entourant le solide. Les effets se déroulant dans cette couche limite sont caractérisables par le nombre de Reynolds Re . Ce nombre décrit le rapport des forces inertielles aux forces de frottement visqueux dans le fluide. Il a été mis en évidence en 1883 par Osborne Reynolds, et est défini de la manière suivante :

$$Re = \frac{VL}{\nu} = \frac{\text{forces d'inertie}}{\text{forces visqueuses}} \quad (2.40)$$

Avec L la dimension caractéristique en m et V la vitesse caractéristique en $m.s^{-1}$ de l'écoulement, ainsi que ν la viscosité cinématique du fluide en $m^2.s^{-1}$. Cette viscosité cinématique est définie par :

$$\nu = \frac{\mu}{\rho} \quad (2.41)$$

Avec μ la viscosité dynamique du fluide en $Pa.s$ ou $kg.m^{-1}.s^{-1}$ et ρ la masse volumique du fluide en $kg.m^{-3}$. Dans le cas d'écoulement à faible nombre de Reynolds, on peut utiliser des équations simplifiées complétées par des résultats expérimentaux. Autrement, il faut résoudre les équations du fluide visqueux, c'est à dire les équations de Navier-Stokes. L'état de surface du solide a également une forte influence sur cette force de friction surfacique (voir [Ble03]).

2.2.2.2 Amortissements dus à l'écoulement tourbillonnaire

En plus des frottements surfaciques, les phénomènes tourbillonnaires sont à prendre en compte dans la force de frottement. Le paradoxe de d'Alembert illustre cela (Figure 2.5). Celui-ci énonce qu'aucune force hydrodynamique n'agit sur un solide complètement submergé à vitesse constante dans un fluide non-visqueux. Dans les écoulements réels, des tourbillons se forment dans le sillage du solide à cause de la viscosité du fluide et modifient la pression à l'arrière du solide. Ceci génère une force, qui peut être non négligeable, notamment lorsque le nombre de Reynolds est grand et doit alors être pris en compte dans la force de frottement.

2.2.2.3 Effet de portance

En plus des deux effets précédents, la portance doit être prise en compte. La portance est la composante de la force subie par un corps en mouvement dans un fluide qui s'exerce perpendiculairement à la direction du mouvement. Cet effet provient de deux mécanismes. Le

4. À la surface du solide on considère que la vitesse du fluide est égale à la vitesse de la paroi, puisqu'il doit y avoir une continuité spatiale dans le champ de vitesse

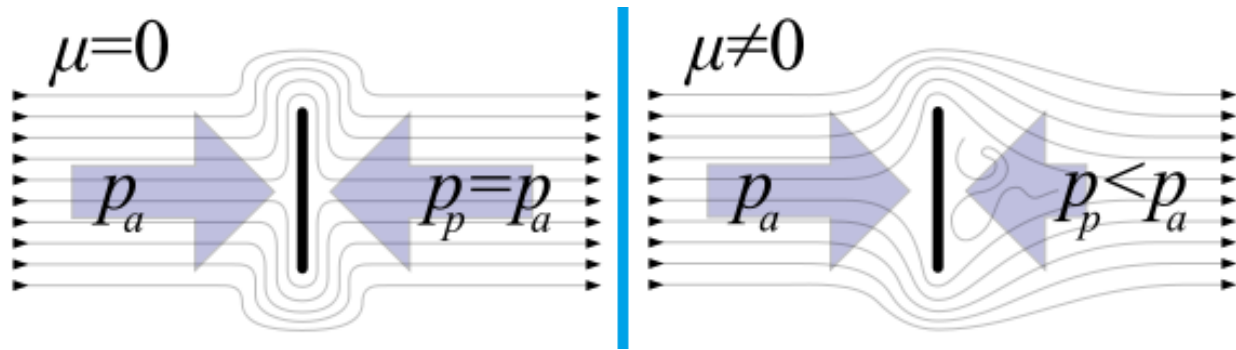


Figure 2.5 – Paradoxe de d'Alembert. Dans le cas d'un fluide sans force visqueuse, il n'y a pas de sillage à l'arrière du solide donc la pression en amont est égale à la pression en aval et aucune force ne s'exerce sur le solide.

premier est dû à la circulation de l'écoulement autour du solide. Si la vitesse à la surface du solide est égale à la vitesse du solide et qu'il y a une plus grande surface sur le solide que sous le solide, alors il y a une différence de vitesse d'écoulement entre les deux surfaces. D'après la formule de Bernoulli (Équation 2.42), si la vitesse est plus grande au-dessus du solide alors une dépression se crée au-dessus du solide. De cette dépression résulte une force qui pousse le solide vers le haut⁵.

$$\frac{1}{2}\rho v^2 + \rho g z + p = \text{constante} \quad (2.42)$$

Le second effet qui crée une force de portance est un effet non-linéaire, qui est souvent appelé « cross flow drag », que l'on pourrait traduire par trainée d'écoulement transversal. Un écoulement transversal, donc perpendiculaire au mouvement du solide, est générée et cet écoulement occasionne des tourbillonnements qui génèrent une perte d'énergie et donc une force supplémentaire sur le solide. Cet effet est très similaire à la dissipation d'énergie dans le sillage de l'objet, à ceci près que sa résultante est perpendiculaire au mouvement.

2.2.2.4 Importance des effets selon l'écoulement

Ces trois effets de dissipation énergétique dans le fluide sont résumés dans une seule matrice $D(\vec{v}_r)$, déjà évoquée dans la section 2.1.3.3 (Équation 2.29). Les éléments de cette matrice peuvent être obtenus par différentes méthodes : par des expériences ou dans certains cas, i.e. fluides idéaux, nombres de Reynolds faibles, géométries très simples, grâce à des formules théoriques. La plupart du temps, pour des véhicules sous-marins réels, des expériences sont réalisées et les coefficients sont déterminés à partir des mesures faites par ces expériences. En général il y a prédominance d'un effet en particulier et l'on choisit de n'utiliser que ce dernier. Ceci dépend du nombre de Reynolds et un seul coefficient est alors utilisé : quadratique ou linéaire.

Le coefficient linéaire est prédominant lorsqu'il est possible de modéliser l'écoulement avec l'équation de Stokes et donc que $Re < 1$. Cette équation de Stokes est une simplification de l'équation de Navier-Stokes pour les écoulements à faible nombre de Reynolds :

$$\mu \Delta \vec{v}_{flu} = \vec{\nabla} p - \rho \vec{f}_{ext} \quad (2.43)$$

5. C'est le principe utilisé pour les ailes d'avions : la forme bombée de l'aile et son orientation créent un écoulement différent au dessus et en dessous de l'aile, ce qui a pour effet de générer une force de portance.

Avec \vec{v}_{flu} la vitesse du fluide et \vec{f}_{ext} une force massique s'exerçant dans le fluide.

Le coefficient quadratique devient important lorsque l'écoulement devient turbulent et qu'il y a apparition des tourbillons, ce que l'on estime arriver lorsque $Re > 2000$. On entre alors dans une phase de transition. Lorsque l'écoulement n'est pas clairement turbulent, tant que $Re < 4000$, le coefficient de frottement peut être une combinaison des deux types de coefficient : linéaire et quadratique.

2.2.2.5 Coefficients de frottement pour des volumes simples

Cette section rassemble des coefficients de frottement empiriques et théoriques pour des formes simples qui serviront dans la suite pour analyser les résultats de nos expériences. Ces coefficients proviennent de [Ble03]. Les coefficients donnés sont adimensionnels et fonction du nombre de Reynolds. La force de frottement peut être obtenue avec :

$$F_{frt} = \frac{1}{2} \rho A C_d u^2 \quad (2.44)$$

Avec u la vitesse relative entre l'objet et le fluide, ρ la masse volumique du fluide, A l'aire caractéristique de l'objet solide et C_d un coefficient de trainée adimensionnel. De la même manière, on peut obtenir une force de portance, due au phénomène décrit précédemment, qui est calculée à partir d'un coefficient de portance C_l :

$$F_{prt} = \frac{1}{2} \rho A C_l u^2 \quad (2.45)$$

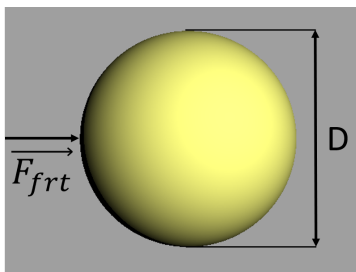
Sphère La sphère est certainement l'objet le plus étudié puisque sa géométrie est particulièrement simple. Grâce à l'équation de Stokes on peut donner une solution pour la force exercée par le fluide pour les très faibles Reynolds ($Re < 1$) :

$$\vec{F} = -3\mu\pi D \vec{v} \quad (2.46)$$

Avec D le diamètre de la sphère. Afin de relier ce résultat pour la sphère avec tous ceux obtenus pour les plus hauts nombres de Reynolds, ce nombre de Reynolds est réutilisé par Wieselsberger [Wie22] pour relier cette formulation à la formulation générale décrite dans l'équation 2.44. Ainsi, pour $Re < 1$ le coefficient C_d vaut [Bat10] :

$$C_d = \frac{24}{Re \left[1 + \left(\frac{3}{16} Re \right) \right]} \quad (2.47)$$

Les coefficients pour des valeurs de Re plus élevées sont souvent représentés sur une échelle logarithmique. Le tableau suivant présente les différentes valeurs des coefficients de frottement pour la sphère et la valeur de l'aire caractéristique de la sphère.



$$A = \frac{\pi D^2}{4}$$

Re	10^2	10^3	10^4	10^5	10^6	5×10^6
C_d	1,0	0,41	0,39	0,52	0,12	0,18

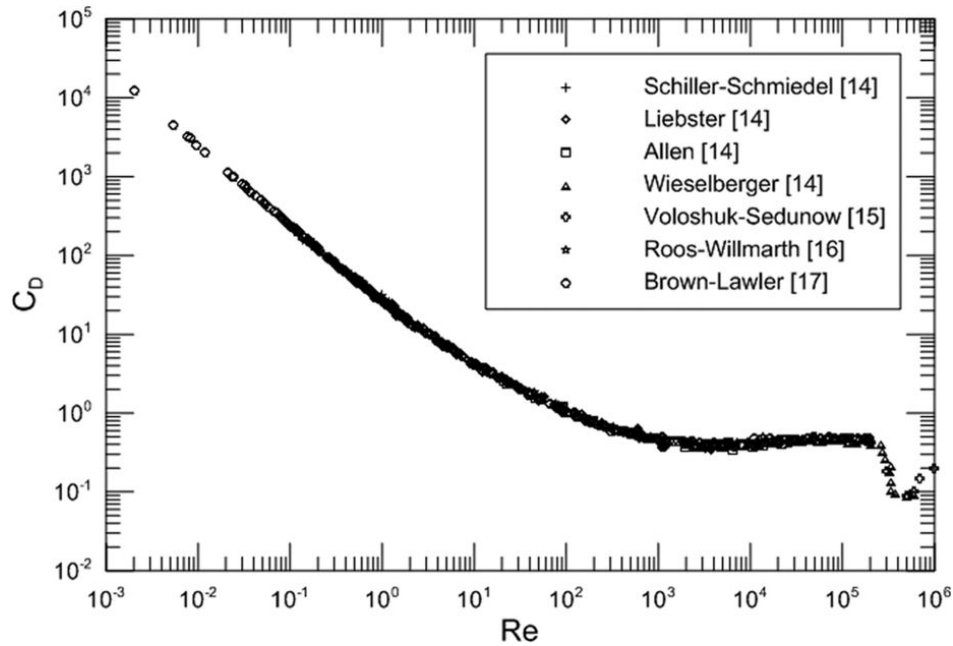


Figure 2.6 – Évolution du coefficient de frottement C_d d'une sphère dans un écoulement de volume infini. Pour $Re < 0.1$ la courbe suit la loi de Stokes. Voir la source pour la provenance des différents ensembles de données. (Source : [DHD15])

La figure 2.6 présente les différents résultats obtenus par la communauté scientifique qui sont résumés dans l'article de Duan [DHD15]. Dans cet article, Duan critique l'utilisation du coefficient C_d et propose une autre constante adimensionnelle $D_C = C_d \cdot Re$ pour mieux décrire les phénomènes physiques présents dans l'écoulement. Il n'y a pas d'explication théorique rigoureuse pouvant expliquer la forme de la courbe de la figure 2.6. Il existe par contre des explications sur les différents phénomènes s'appliquant que l'on retrouve dans [Bea11]. À faible nombre de Reynolds, la force de frottement est due principalement aux forces de frottement surfaciques et C_d diminue avec Re . À partir de $Re = 200$, les perturbations dues à la viscosité et la force due à la dépression dans le sillage de la sphère deviennent prépondérantes. Le coefficient C_d devient alors quasi constant. Il augmente très légèrement pour $10^3 < Re < 3 \times 10^5$ puisque le tourbillonnement de l'écoulement augmente. À environ $Re = 3 \times 10^5$, le coefficient C_d chute brutalement. Cet effet est appelé crise de traînée ou encore paradoxe de Lukyanov–Eiffel ou simplement paradoxe d'Eiffel⁶. Pour les Reynolds plus petits que 3×10^5 , le régime de l'écoulement autour de la sphère est appelé sous-critique, voir schéma 2.7. Le décollement de l'écoulement, c'est à dire l'endroit où se séparent les lignes de courant laminaire et l'écoulement turbulent, se situe en amont de la sphère. Au-delà de 3×10^5 , le régime de l'écoulement devient super-critique, c'est à dire que ce point de décollement passe de l'autre côté de la sphère occasionnant une forte diminution de la taille de la zone de turbulence à l'arrière de la sphère. L'écart entre la pression avant et aval est donc modifié et cette chute de coefficient de frottement survient. La vitesse à laquelle ce décollement a

6. Cet effet a été constaté pour la première fois en 1904 par un étudiant russe G.I.Lukyanov. C'est en 1912 qu'indépendamment les collaborateurs de Gustave Eiffel ont également constaté cet effet en mesurant la traînée des sphères dans la soufflerie qu'il avait inaugurée à la base de sa tour à Paris et déplacée à Auteuil en 1912. Cette soufflerie est aujourd'hui classée Monument Historique.

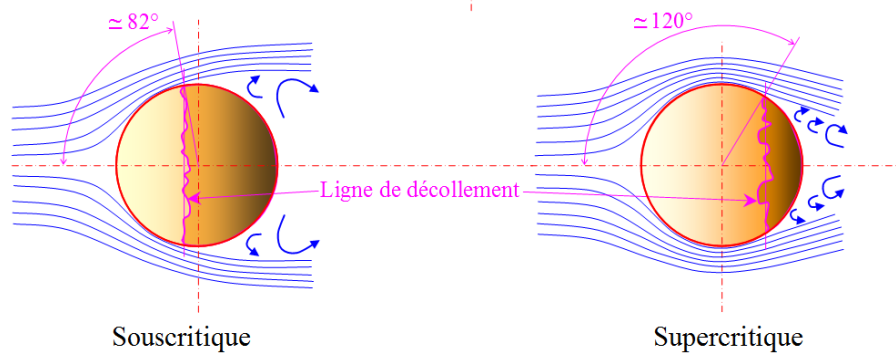


Figure 2.7 – Régime sous-critique et super-critique de l'écoulement autour de la sphère (Source modifiée à partir d'une image de l'Association Go Mars)

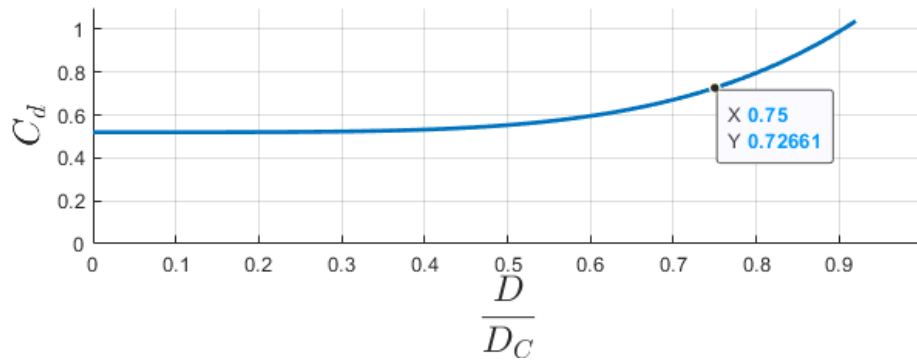
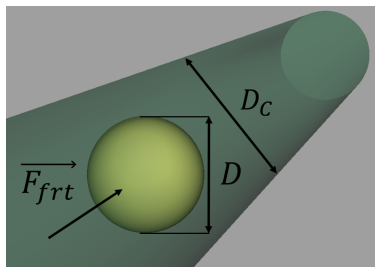


Figure 2.8 – Évolution du coefficient de frottement C_d en fonction du rapport de dimension D/D_0 .

lieu est fonction de l'état de surface de la sphère. Ainsi, pour des sphères plus rugueuses, ce décollement se fait à des nombres de Reynolds plus faibles. [WS06].

On peut également trouver des coefficients pour une sphère placée dans un cylindre dont les parois sont proches de la sphère dans Achenbach [Ach74].



$$A = \frac{\pi D^2}{4}$$

Pour $0 < \frac{D}{D_c} < 0,92$ et $3 \times 10^4 < Re < 2 \times 10^6$

$$C_d = \left[1 + 1.45 \left(\frac{D}{D_c} \right)^{4.5} \right] C_{d,Sphere\infty}$$

Avec $C_{d,Sphere\infty}$ le coefficient du premier tableau ci-dessus : dans le cas où la sphère est dans un fluide infini.

Comme on peut le constater, tout comme dans le cas de la masse ajoutée (§ 2.2.1), l'écoulement de l'eau entourant la sphère est important et toute perturbation de cet écoulement induit une modification du paramètre hydrodynamique. Ainsi le coefficient C_d pour une sphère placée dans un cylindre évolue tel que sur la figure 2.8. Plus la surface de la sphère est proche de la paroi et plus la force de frottement agissant sur la sphère est grande. La variation devient particulièrement importante dès que l'espace entre la sphère et le conduit qui l'entoure s'approche de la longueur du rayon de la sphère. Lorsque l'écart entre la sphère et le cylindre

C_d	$\alpha = 0^\circ$	$\alpha = 45^\circ$	$\alpha = 90^\circ$
$Re = 0,1$	285	335	380
$Re = 10$	4,2	5,8	7,5
$Re = 50$	1,3	2,3	3,2
$Re = 240$	0,5	1,3	1,7
$Re = 300$	0,4	1,15	1,6

Tableau 2.2 – Coefficient de frottement C_d d’une capsule selon 3 angles d’attaque et le nombre de Reynolds. Résultats obtenus par Ouchene [OKT14].

est de l’ordre d’un demi-rayon, la variation est déjà de 40%.

La sphère, de par sa simplicité, est l’objet le plus étudié en hydrodynamique. Pour les autres géométries, hormis le cylindre, bien moins d’informations ont été récoltées à ce jour dans la littérature et leur étude est plus complexe.

Capsule Dans la section 2.2.1, la difficulté de trouver des paramètres hydrodynamiques pour la forme de capsule a déjà été mise en évidence. Certains chercheurs donnent une réponse et des modèles pour la détermination de coefficients de frottement pour n’importe quel sphéroïde [HS08], mais pas pour une forme de capsule. Ouchene [OKT14] a fait des simulations avec un logiciel de calcul de fluide, qui permettent de donner une valeur du coefficient de frottement à appliquer selon l’angle d’attaque de la capsule. Cette méthode sera détaillée dans la section 2.3.1.

Sur la figure 2.9, on représente la capsule ainsi que l’angle d’attaque. Les mesures d’Ouchene

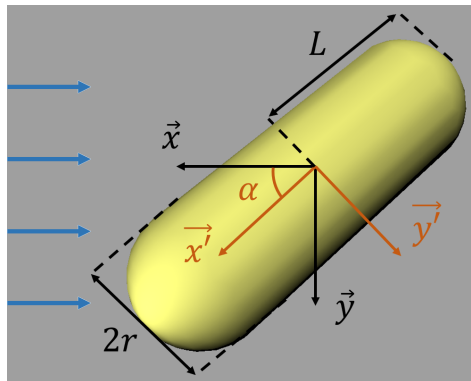


Figure 2.9 – Vue de la capsule face à l’écoulement avec un angle d’attaque α . (x, y) le repère global où l’écoulement se fait selon x et (x', y') le repère local de la capsule.

valent pour une capsule dont le ratio $L/2r = 5$. Le tableau 2.2 récapitule ses résultats pour les angles de 0° , 45° et 90° .

D’autres résultats obtenus sur le même principe peuvent être trouvés dans Yan [YLJ⁺18]. Dans cette étude une comparaison entre les résultats du calcul numérique et une série d’expériences est proposée. Bien que les résultats obtenus avec le calcul numérique et l’expérience qui a été menée ne sont pas les mêmes, l’évolution du coefficient de frottement est tout de même homothétique avec la réalité et avec les résultats d’Ouchene [OKT14].

2.2.3 Conclusion

Les efforts qu'un fluide exerce sur un solide peuvent être divisés en deux catégories : les forces de masses ajoutées et les forces de frottement. Actuellement, des résultats théoriques existent dans des cas particulièrement simples. La plupart des études sur le sujet se basent sur des coefficients empiriques et l'étude de l'hydrodynamique des objets est un sujet actif de recherche. Les résultats empiriques existants restent limités à des formes relativement simples également. Il reste donc deux possibilités pour l'utilisation de formes plus complexes : la détermination par l'expérimentation ou la détermination par un logiciel qui simule la dynamique des fluides.

Puisque l'écoulement est variable lors de la navigation des véhicules marins, il serait intéressant que le simulateur SPH soit capable de reproduire les forces mesurées empiriquement sans nécessiter l'entrée de paramètres hydrodynamiques, grâce à la simulation du fluide.

2.3 Méthodes d'identification des paramètres hydrodynamiques

Cette partie traite des différentes méthodes d'obtention des paramètres hydrodynamiques. Une première solution est l'utilisation d'un logiciel simulant l'écoulement de fluide autour d'un objet. Une seconde solution est la mise en place d'un dispositif expérimental permettant de placer l'objet dans un écoulement. Dans tous les cas, le principe est de déterminer les forces s'exerçant sur l'objet suivant l'écoulement et d'en déduire les paramètres hydrodynamiques.

2.3.1 Méthodes d'identification logicielles

Il existe différents types de logiciels permettant l'estimation de paramètres hydrodynamiques. Tous ces logiciels simulent un écoulement de fluide et font donc partie d'un ensemble de logiciels déjà évoqué dans l'introduction ([chapitre 1](#)) : les logiciels de « Computational Fluid Dynamics » (CFD). Ces logiciels ont un rôle important dans tous les domaines où les fluides sont présents : chimie, génie civil, génie maritime, etc. Seuls les logiciels de simulation de fluide utilisant un maillage sont considérés, car ceux utilisant des particules ne sont pour l'instant pas utilisés pour l'identification de paramètres hydrodynamiques. Il existe différentes méthodes de résolution des équations pour les écoulements, qui peuvent être classées selon leur méthode de modélisation et de résolution des turbulences [RS19]. En les classant du plus au moins précis :

1. La simulation numérique directe, « Direct Numerical Simulation » (DNS), est une méthode de simulation CFD avec laquelle les équations de Navier-Stokes sont résolues numériquement avec toute la gamme des échelles spatiales et temporelles de la turbulence. Donc, ces équations sont résolues sans modèles simplificateurs des phénomènes de turbulences. Cette méthode a pour avantage d'être extrêmement précise, mais est très limitée pour des écoulements rapides. Le nombre de points requis pour obtenir une résolution des turbulences, dans un problème en 3 dimensions, est supérieur ou égal à $Re^{9/4}$ [MM98]. De plus, en considérant que l'on utilise $Re^{9/4}$ point, alors le nombre d'opérations flottantes requises est de l'ordre de Re^3 . La plus grande simulation à ce jour est celle de Yeung [YZS15], où une grille de 8192^3 cellules est utilisée et le nombre de Reynolds simulé est 1300. Cette méthode n'est pour le moment pas applicable à l'ingénierie marine et sert principalement à l'étude de la mécanique des fluides.
2. Pour remédier aux limitations de DNS, on filtre les petites échelles et on calcule directement les grandes échelles de la cascade turbulente. La cascade turbulente est un processus de division des grands tourbillons en tourbillons plus petits. Ceci permet un transfert d'énergie des grandes échelles vers les petites échelles. Les plus petits tourbillons absorbent cette énergie et la dissipe, principalement en énergie thermique. La limite d'échelle spatiale, à partir de laquelle la viscosité permet de dissiper l'énergie cinétique d'un écoulement, est appelée dimension de Kolmogorov. Les équations du modèle « Large Eddy Scale » (LES) sont obtenues en utilisant un filtre passe bas sur les équations aux dérivées partielles qui régissent les champs (pression, densité, etc.) de l'écoulement. Les informations filtrées ne sont cependant pas sans importance et la recherche de modèle précis permettant d'approximer les effets des petites structures de turbulences est un sujet actif de recherche. Cette méthode requiert une grande

précision spatiale au niveau des zones où l'écoulement est à haut Reynolds, par exemple la discrétisation proche des solides doit être très fine.

3. La méthode « Reynolds Averaged Navier Stokes » (RANS) propose une description statistique de l'écoulement. L'idée est d'utiliser la décomposition de Reynolds, selon laquelle une quantité instantanée est décomposée en ses quantités fluctuantes et moyennes dans le temps⁷. On utilise donc une moyenne temporelle des turbulences qui permet de simuler sur des échelles temporelles bien plus grandes que celles qu'il faudrait utiliser pour modéliser les plus grands tourbillons. A partir de cela on obtient l'évolution moyenne de l'écoulement, les effets des turbulences étant pris en compte dans un tenseur de contrainte, appelé contrainte de Reynolds ou contrainte turbulente, qui exprime la fluctuation moyenne des turbulences. La différence entre RANS et LES est que cette contrainte de Reynolds, dans le cas de RANS, est la moyenne de toutes les structures de la turbulence, et dans le cas de LES, est la moyenne des plus petites structures prises en compte dans le calcul des grandes structures de la turbulence. L'approche principale pour résoudre ce problème est basée sur l'utilisation de l'approximation de Boussinesq qui permet de relier le gradient de vitesse moyenne à la contrainte de Reynolds via une viscosité turbulente supplémentaire : provenant des structures de turbulences. Cette viscosité turbulente est appelée en anglais : « turbulent eddy viscosity ». Cette méthode est très souvent réutilisée, car les temps de calculs nécessaires pour l'appliquer sont très faibles. Néanmoins l'approximation de Boussinesq n'est pas toujours valide, e.g. pour des écoulements qui changent brusquement de direction principale. Il faut bien comprendre que la viscosité turbulente n'est pas une propriété du fluide mais une propriété de l'écoulement. De nombreuses méthodes ont été mises au point pour calculer cette viscosité turbulente, avec une, deux ou plus d'équations (Voir Wilcox [Wil93]). Les modèles $k-\epsilon$ et $k-\omega$ étant deux exemples très connus de modèle à deux équations, que l'on rencontre très souvent dans tous les domaines d'ingénierie, marine y compris. On peut interpréter ces termes ainsi : k est l'énergie cinétique turbulente, ω et ϵ sont les taux de dissipation des turbulences. La dissipation turbulente est la vitesse à laquelle les structures de turbulence se dissipent. Ces deux modèles sont similaires. Dans le cas du modèle $k-\omega$, une équation est utilisée pour calculer le taux de dissipation des turbulences au lieu d'utiliser un facteur de dissipation empirique dans le cas du modèle $k-\epsilon$. Une autre méthode fondamentalement différente est décrite dans Launder [LRR75]. Elle est couramment appelée « Reynolds Stress Model » (RSM) et n'utilise pas l'approximation de Boussinesq. Cette méthode ajoute néanmoins un système de six équations aux dérivées partielles, ce qui augmente considérablement le temps de calcul requis, comparé aux autres modèles plus simples. Ces modèles sont donc relativement moins utilisés que les modèles simplifiés à deux équations. Il existe également des modèles hybrides RANS-LES [Sag05] qui ont été développés récemment pour gagner en précision par rapport aux modèles RANS et diminuer le temps de calcul requis pour les modèles LES.
4. Certaines méthodes encore moins précises existent (voir Roelofs [RS19]) mais ne sont pas traitées ici, car ces méthodes sont trop récentes.

Pour les coefficients de frottement, puisqu'ils proviennent des turbulences (voir [sous-section 2.2.2](#)), il faut simuler l'écoulement autour de l'objet pour déterminer le comportement turbulent de l'écoulement. Il faut donc utiliser l'une des quatre méthodes précédemment citées.

7. une idée proposée par Reynolds lui-même [Rey95]

Avec ces méthodes, il est possible de déterminer en parallèle la masse ajoutée si on peut simuler l'accélération du fluide autour du corps ou si l'on peut faire se déplacer la frontière du solide dans le fluide.

En ce qui concerne la masse ajoutée seule, avec l'hypothèse de mouvement de faible amplitude et en négligeant les effets visqueux, on peut utiliser la technique BEM (« Boundary Element Method », méthode des éléments frontières en français).

2.3.1.1 Résolution d'équations d'écoulement pour l'estimation des paramètres hydrodynamiques

Généralement, la technique RANS est utilisée pour la détermination des coefficients de frottement. En 1986, Patel [PC86] faisait déjà l'état de l'art des différentes méthodes pour calculer un écoulement à l'arrière de véhicules axisymétriques, la forme typique des AUV. Sarkar [SSF97] proposait en 1997 de déterminer des coefficients de frottement à l'aide de modèle CFD, en calculant la somme des forces de pression et de frottement visqueux. Ces techniques sont aujourd'hui devenues très courantes et l'utilisation de logiciel CFD pour l'estimation de paramètre hydrodynamique de véhicules marins et sous-marins est devenue standard. Dans les études de Ramirez-Macias [RMBRV16], Eidsvik [ES16], Chin [CLL17], Gao [GWP⁺18], Boe [BRP⁺13], Yoon [YNV17], qui impliquent une expérience physique, les auteurs ont tendance à faire confiance aux paramètres issus des logiciels CFD.

Parmi les logiciels existants couramment utilisés, on peut citer : Solidworks Flow Simulation (e.g. [ES16]), STAR-CCM+ (e.g. [CLL17], [GWP⁺18], [BRP⁺13]) ou encore ANSYS Fluent (e.g. [BRP⁺13], [dDPd08], [YNV17]). Ces logiciels permettent l'utilisation de différents modèles de turbulence. Malheureusement, le modèle de turbulences n'est pas précisé dans tous les documents (e.g. [ES16]), alors que ce modèle devrait avoir un impact significatif sur la précision des coefficients qui ont été déterminés. Le modèle le plus courant semble être le modèle $k-\omega$ Shear Stress Transport (SST), que l'on retrouve par exemple dans Chin [CLL17], de Barros [dDPd08] et Gao [GWP⁺18]. Ce dernier le décrit comme particulièrement intéressant en termes de précision et de temps de calcul. Ce modèle de turbulence date de 1994 et est basé sur des résultats d'expériences physiques [Men94]. Ces expériences ont permis de déterminer certaines constantes, conduisant à une approximation des effets des turbulences. L'objectif de ce modèle est l'obtention rapide de résultats pour des problèmes d'ingénierie classique. En fait, le modèle SST utilise le modèle $k-\omega$ en zone de proche paroi⁸ et le modèle $k-\epsilon$ dans les zones loin des parois à nombre de Reynolds important. $k-\omega$ SST permet donc d'allier les points forts de ces modèles. Gao [GWP⁺18] propose dans son article une méthode où le navire effectue un mouvement composé sur les 3 axes, régi par 3 fréquences caractéristiques indépendantes, ce qui permet de générer un jeu de données suffisamment complet pour permettre la détermination de tous les coefficients au cours d'un même essai.

Le plus souvent, on simule un écoulement constant autour de l'objet que l'on souhaite étudier, ce qui ne permet d'obtenir que $D(\vec{v}_r)$. La simulation d'un écoulement qui accélère permet d'obtenir un paramètre de masse ajoutée et un paramètre de frottement. Il faut donc en général réaliser une série de simulations pour obtenir tous les différents paramètres, vu que chaque simulation réalisée ne permet d'obtenir qu'un paramètre de chaque matrice M_A ou $D(\vec{v}_r)$. Le principal inconvénient de cette méthode, qui consiste à simuler le mouvement d'une interface dans un maillage, est qu'il faut remailler à chaque étape de simulation. Les

8. Ce modèle est peu adapté pour les écoulements complexes (recirculation, anisotropie forte, production négative, etc.).

techniques les plus récentes, comme les techniques d'overset [TNTN97], qui consistent à superposer plusieurs maillages pour éviter les étapes de remaillage, sont depuis peu devenues assez courantes. Elles permettent de réaliser des simulations encore plus complexes, comme dans Coe [Coe13].

2.3.1.2 Détermination de masse ajoutée avec la méthode BEM

Pour calculer la masse ajoutée, on peut utiliser la technique « Boundary Element Method » (BEM). BEM découle de la théorie de l'écoulement potentiel et est particulièrement intéressante lorsque le ratio surface/volume est élevé, ce qui est le cas pour le calcul de masse ajoutée (volume fluide infini). La technique est décrite dans la thèse de Floc'H [Flo11] ; le domaine n'est pas entièrement discrétisé, mais uniquement la frontière du domaine. Le principe est celui que l'on a déjà présenté dans le paragraphe 2.2.1 : on considère un domaine de fluide V entouré par les frontières du solide S_W et du fluide à l'infini S_∞ . En supposant que le fluide est incompressible, que l'on néglige son caractère visqueux et que l'on considère l'écoulement irrotationnel, les vitesses dérivent du gradient d'une fonction scalaire ϕ appelée potentiel d'écoulement. A partir de l'équation H.4 et de la définition de ϕ on obtient alors l'équation de Laplace (Équation 2.33). Cette équation est l'équation principale du problème. Dès lors que l'on obtient la répartition du potentiel d'écoulement, on utilise l'équation de Bernoulli 2.36 pour obtenir la pression du fluide sur la surface. Pour rappel, ces deux équations à résoudre sont :

$$\Delta\phi = 0$$

$$p = \rho \left(\frac{\partial\phi}{\partial t} + \frac{1}{2} |\vec{\nabla}\phi|^2 \right)$$

Pour résoudre le problème du Laplacien on utilise les conditions aux limites pour ajouter des équations supplémentaires :

- A la surface du solide, puisque l'on néglige les effets visqueux, on applique une **condition de glissement**. Ainsi en définissant les vecteurs \vec{n} normaux à la surface du solide S_W pointant vers l'extérieur du fluide, on a :

$$\vec{v} \cdot \vec{n} = \vec{\nabla}\phi \cdot \vec{n} = 0 \text{ sur } S_W \quad (2.48)$$

- Le champ de potentiel ϕ peut être décomposé en deux : ϕ_∞ le potentiel dû à la vitesse de l'écoulement en amont de l'objet et ϕ_W le potentiel de perturbation de cet écoulement dû à la présence de l'objet. Cette perturbation doit s'atténuer lorsque l'on s'éloigne du corps. Ainsi :

$$\lim_{r \rightarrow \infty} \vec{\nabla}\phi = 0 \quad (2.49)$$

Avec r la distance entre le point considéré et le corps. Cette condition est appelée **condition d'atténuation** ou **condition de régularité à l'infini** [Flo11].

- Il est nécessaire d'ajouter une dernière condition pour déterminer l'écoulement autour des corps dont la forme induit une force de portance, comme c'est le cas pour les profils d'ailes d'avions par exemple. Les frottements visqueux, qui sont ici négligés, sont liés au phénomène de portance (§ 2.2.2). La **condition de Kutta** permet de modéliser la portance. Quand un corps se déplace dans un fluide il y a deux points où la vitesse du fluide est nulle sur chaque section du corps. Ces deux points sont le bord d'attaque,

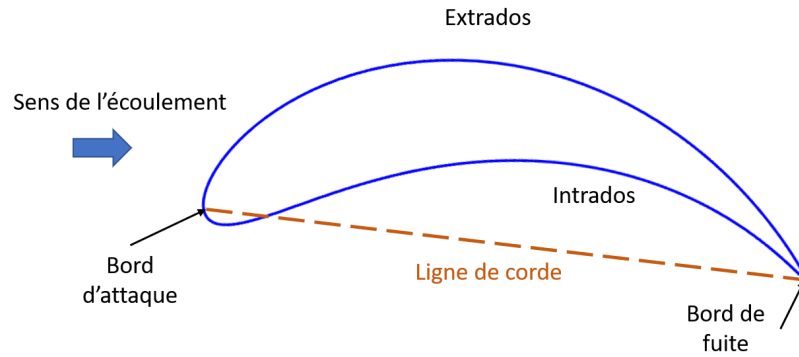


Figure 2.10 – Schéma d'un profil de Joukowski avec les points bord d'attaque et bord de fuite

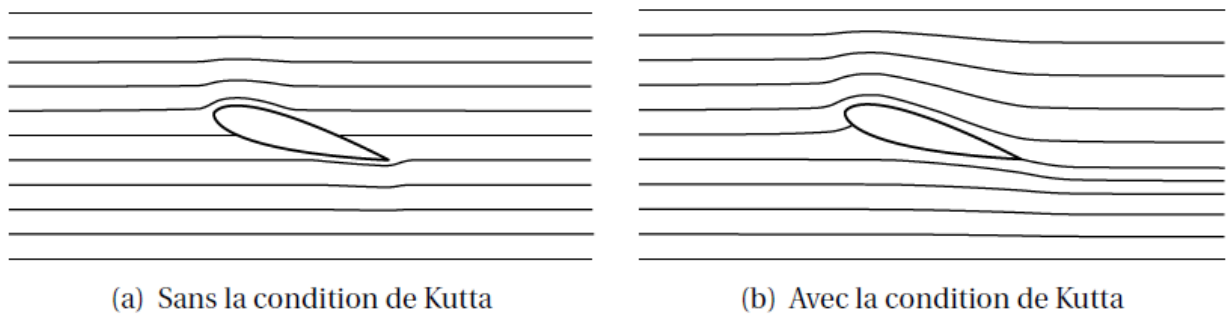


Figure 2.11 – Effet de la condition de Kutta sur l'écoulement de fluide parfait autour d'un aileron (Source : [Flo11])

point le plus en amont du corps et le bord de fuite, point le plus en aval du corps et sont reliés par une droite que l'on appelle ligne de corde. Dans le cas d'un corps symétrique, comme un cylindre, la ligne de corde est alignée avec l'axe principal de l'écoulement. Il n'y a alors aucune portance. La figure 2.10 montre l'emplacement de ces points sur un profil de Joukowski. Sur un tel profil, la ligne de corde n'est pas alignée avec l'axe de l'écoulement. La condition de Kutta dit qu'au bord de fuite il doit y avoir une continuité des champs de pression et vitesse. Sans prendre en compte cette condition, le point où la vitesse de l'écoulement serait nul en aval serait aligné avec le bord d'attaque selon l'axe de l'écoulement, voir figure 2.11.

A l'aide de ces conditions, on résout le système d'équations et, à partir de la variation de pression sur la surface lors d'une accélération, on obtient la masse ajoutée. Différents articles se basent sur ce genre de logiciel pour l'estimation de paramètres hydrodynamiques pour véhicules marins, e.g. Pétilion [PYDSM19], qui utilise cette technique pour étudier les effets de proximité avec le fond, Perrault [PBOW03], qui effectue une étude de variabilité des paramètres de masse ajoutée grâce à cette technique, validée par des moyens expérimentaux, ou encore la thèse de master de Severholt [Sev17], qui utilise cette technique pour déterminer les paramètres hydrodynamiques d'une forme d'AUV.

2.3.2 Méthodes d'identification expérimentales

On peut classer les différents moyens expérimentaux selon les catégories suivantes :

- Ceux qui utilisent un système de mesure externe et qui ne nécessitent aucune installation sur le corps ou sur le véhicule que l'on souhaite étudier. Il existe des systèmes sans actionneur, comme le balancier hydrodynamique décrit dans Eng [ELLS08], Yi [YAQ17] et Langmajer [LB16a]. Il y a également des systèmes avec actionneur, qui peuvent être divisés en deux catégories principales : les installations à bras rotatifs, aussi appelées « Rotating arm » en anglais, et les mécanismes à mouvement plan, qui se nomment « Planar Motion Mecanism » (PMM) en anglais. Les installations à bras rotatifs mettent en mouvement à l'aide d'actionneur le véhicule marin en lui faisant faire une translation circulaire. Les mécanismes à mouvement plan permettent eux d'effectuer des translations sur un plan, le plus souvent horizontal. Une méthode systématique pour la mise en place d'expériences avec ces deux mécanismes a été proposée dans Sutulo [SK98] en 1998 pour les véhicules de surface. Ces mécanismes sont donc aujourd'hui connus et assez courants. Le mécanisme de PMM est utilisé, par exemple, par Eidsvik [ES16] pour la mesure des paramètres d'un ROV, voir figure 2.12. Eidsvik [ES16] cherche à obtenir des valeurs approchées des paramètres hydrodynamiques en utilisant un modèle consistant en un assemblage de sous-ensembles. Ces sous-ensembles étant des objets 3D très simples dont les paramètres sont connus.

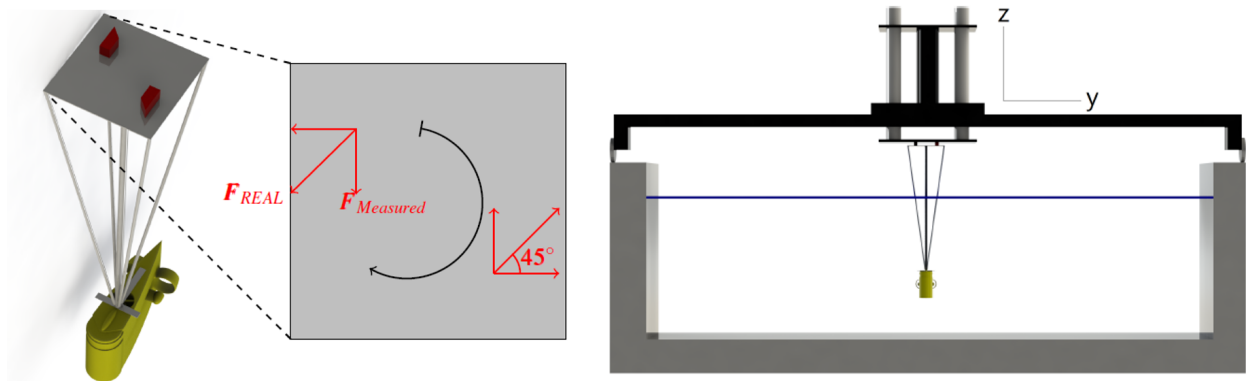


Figure 2.12 – Schéma d'un ROV placé sur un mécanisme à mouvement plan (PMM) (Source : [ES16])

- Ceux qui utilisent en même temps des systèmes externes et des équipements présents sur le véhicule pour la mesure ou pour la mise en mouvement. Par exemple, Caccia [CIV00] utilise le compas du véhicule pour déterminer son orientation, le capteur de profondeur du véhicule et un capteur d'écho-sondage qui mesure la position par rapport au mur de la piscine. Ces capteurs permettent l'obtention de la trajectoire du véhicule en réponse à des efforts imposés avec le moteur. L'auteur note que ce type d'expérimentations peut être particulièrement intéressant lorsque l'on possède un véhicule mais pas d'équipement spécifique pour effectuer les mesures de paramètres. Harris [HPW18] propose une méthode qui utilise un gyrocompas, un capteur de profondeur et un modem acoustique pour se positionner par rapport à un véhicule de surface ou une balise flottante équipée d'un GPS. Cette méthode permettrait de déterminer tous les paramètres hydrodynamiques du véhicule en une seule fois. Il

précise que l'obtention de ces paramètres permet de compenser une absence de DVL⁹. Ridaou [RTEF⁺04] se sert des capteurs externes : une caméra pour mesurer la position du véhicule et un capteur d'effort sont utilisés. Les moteurs du véhicule sont utilisés pour la mise en mouvement. SmallWood [SW03] estime les paramètres directement grâce à un observateur d'état¹⁰.

2.3.3 Conclusion

Les différentes méthodes logicielles et expérimentales existantes pour l'obtention de paramètres hydrodynamiques ont été détaillées. Dans la grande majorité des cas, le principe est de mesurer l'effort du fluide sur la surface externe du véhicule. Ces techniques ont un tel niveau de fiabilité que l'on considère que les résultats obtenus servent de réalité terrain. Les logiciels utilisés par les chercheurs en robotique et en ingénierie marine sont généralement basés sur la méthode RANS. L'utilisation de cette méthode requiert une bonne connaissance des modèles de turbulences utilisés.

9. Doppler Velocity Log, voir section 1.3.1.3

10. En automatique et en théorie de l'information, un observateur d'état est une extension d'un modèle, qui permet de reconstruire l'état à partir d'un modèle du système dynamique et des mesures d'autres grandeurs. C'est à dire d'obtenir une estimation de l'évolution temporelle d'une grandeur physique non mesurable

2.4 Identification expérimentale des paramètres hydrodynamiques

Parmi les différents dispositifs permettant l'obtention de paramètres hydrodynamiques pour des véhicules sous-marins présentés en section 2.3.2), le balancier hydrodynamique a été choisi, car il est peu onéreux, facile à mettre en place et utilisable en bassin peu profond. Une sphère, dont les paramètres sont bien connus (§ 2.2), constitue l'objet d'étude. Celle-ci servira aussi à la validation de notre simulateur SPH, décrit dans le chapitre 3. Les choix de conception du balancier hydrodynamique sont détaillés dans l'annexe D. Le dispositif expérimental mis en place dans le bassin d'essai est représenté sur la figure 2.13. La position angulaire du balancier est mesurée à l'aide d'un codeur incrémental haute résolution (voir annexe D).

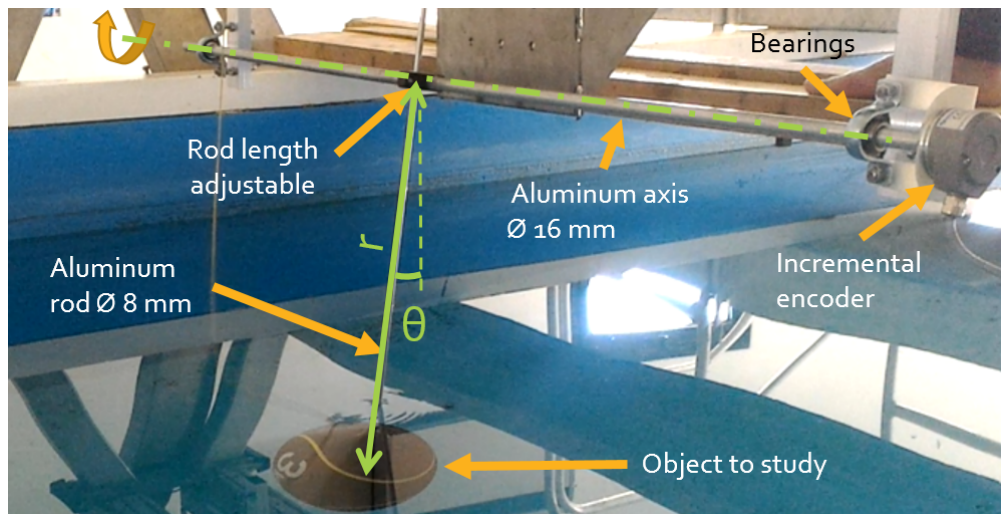


Figure 2.13 – Le balancier dans le bassin à houle de l'Université de Toulon

Les essais ont été effectués à l'intérieur d'un bassin avec une médecine-ball de masse $m_S = 6.4 \text{ kg}$ ¹¹, de rayon $R_S = 0.1085 \text{ m}$ et un moment d'inertie sur l'axe pour le plus petit rayon du pendule $J_{\Delta}^{Sphere} = 5 \text{ kg.m}^2$. L'inertie de la sphère est significativement plus grande ($\text{min.} \times 10^2$) que le reste de la structure. Ces valeurs ont été choisies de manière à ce que son poids équivalent en eau et en volume soit proche d'un mini-ROV rectangulaire de 10 kg et 39.2 dm^3 (LxDxH : 457mm x 338mm x 254mm). L'eau est restée calme avant et pendant les essais. Le tableau 2.3 liste les tests qui ont été effectués. Pour chaque expérience, l'objet est

N° d'essai	$m_S(\text{kg})$	$R_S(\text{m})$	$r \text{ (m)}$	$\theta_0(^{\circ})$	$v_{max}(\text{m.s}^{-1})$
1, 2 & 3	6.4	0.1085	1.10	18	0.4
4, 5 & 6	6.4	0.1085	0.63	11	0.2
7, 8 & 9	6.4	0.1085	0.84	16	0.3

Tableau 2.3 – Expériences dans l'eau

amené à une position de départ, puis relâché et laissé en oscillation libre.

11. Densité : 1200 kg.m^{-3}

2.4.1 Équation du mouvement d'un balancier hydrodynamique

Le balancier de notre étude est schématisé sur la figure 2.14. Les paramètres hydrodynamiques

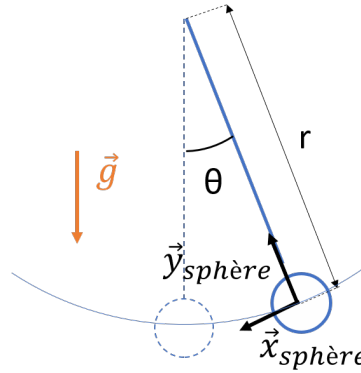


Figure 2.14 – Schéma du balancier étudié pour la détermination des paramètres hydrodynamiques

miques à estimer avec le balancier hydrodynamique sont la masse ajoutée selon l'axe \vec{x}_{Sphere} du repère de la sphère : $X_{\ddot{u}}$, les frottements linéaires $X_{L,u}$, proportionnels à la vitesse, et les frottements quadratiques $X_{Q,u}$, proportionnels au carré de la vitesse, toujours selon l'axe \vec{x}_{Sphere} , selon les notations utilisées dans la section 2.1.3. Les différentes forces appliquées sur l'objet sont énumérées ci-dessous :

- Poids $\vec{P} = -m_S \vec{g}$
- Flottabilité de l'objet $\vec{B} = \rho_{fluid} V_S \vec{g}$
- Force de frottement linéaire $\vec{F}_L = -X_{L,u} u \vec{x}_{Sphere}$
- Force de frottement quadratique $\vec{F}_Q = -X_{Q,u} u^2 \vec{x}_{Sphere}$
- Force de masse ajoutée $\vec{F}_{Ma} = -X_{\ddot{u}} \ddot{u} \vec{x}_{Sphere}$

Où u et \ddot{u} sont la vitesse et l'accélération de la sphère sur l'axe \vec{x}_{Sphere} , obtenues par dérivation de la position x_{Sphere} : abscisse curviligne de la sphère sur le cercle décrit par le mouvement du balancier (Figure 2.14), \vec{g} , le champ gravitationnel, m_S , la masse du solide, V_S , le volume du solide, ρ_{fluid} , la densité du liquide. Considérant $x_{Sphere} = r\theta$, et en projetant les forces s'appliquant sur l'axe \vec{x}_{Sphere} , on a :

$$\ddot{\theta} = \frac{P_{eq}}{Mr} \sin \theta - \frac{X_{L,u}}{M} \dot{\theta} - \frac{X_{Q,u} r}{M} |\dot{\theta}| \dot{\theta} \quad (2.50)$$

Avec le poids équivalent $P_{eq} = \|\vec{B}\| - \|\vec{P}\|$ et l'inertie équivalente $M = m_S + X_{\ddot{u}}$. Par souci de simplification de lecture, la notation suivante sera utilisée dans la suite de ce chapitre :

$$m_a = X_{\ddot{u}} \quad K_l = X_{L,u} \quad K_q = X_{Q,u} \quad (2.51)$$

2.4.2 Méthodes numériques d'identification de paramètres

Plusieurs types de méthode d'identification des paramètres hydrodynamiques existent. Selon l'article de Harris [HPW18], les méthodes d'identification appartiennent principalement à l'une des catégories suivantes :

- estimation adaptative basée sur un observateur d'état,
- variantes du filtre de Kalman,
- techniques d'apprentissage automatique, ou en anglais « Machine Learning »,
- utilisation de réseaux neuronaux, ou en anglais « Neural Network », une technique qui est aussi liée au « Deep Learning »,
- méthodes des moindres carrés.

Ces méthodes sont adaptées à l'identification de paramètres hydrodynamiques avec ou sans dispositif expérimental. Dans tous les cas, elles pourraient toutes être adaptées à l'utilisation d'un dispositif expérimental même si pour certaines, cela n'a jamais été fait.

2.4.2.1 Estimation adaptative ou modèle d'observation

L'estimation adaptative repose sur l'utilisation d'un modèle d'estimation ou d'observation, comme décrit dans Smallwood [SW03]. L'intérêt de cette méthode est qu'elle peut être utilisée au cours du fonctionnement du véhicule et peut permettre d'affiner l'estimation des paramètres au cours de la mission du véhicule, par exemple, suivant l'état de la mer. Cette méthode a principalement été développée et utilisée sous la supervision de Louis L. Whitcomb [KW07], [TW13], [SW17] qui l'a étendue à la mesure d'erreur de capteurs.

La méthode décrite dans Smallwood [SW03] utilise donc la vitesse $v(t)$ mesurée et une vitesse estimée $\hat{v}(t)$. L'estimation est basée sur l'équation :

$$\dot{\hat{v}}(t) = \hat{\xi}^T \hat{f}(t) - K \Delta v(t) \quad (2.52)$$

Avec $\hat{f}(t)$ le vecteur contenant les informations de vitesses qui permettent à partir des paramètres $\hat{\xi}$ à estimer d'obtenir une accélération estimée. $K > 0$ est un gain. Les erreurs d'estimation peuvent être écrites :

$$\begin{aligned} \Delta v(t) &= \hat{v}(t) - v(t) \\ \Delta \dot{v}(t) &= \dot{\hat{v}}(t) - \dot{v}(t) \\ \Delta \xi(t) &= \hat{\xi}(t) - \xi(t) \end{aligned} \quad (2.53)$$

En faisant l'hypothèse que ces paramètres ne varient pas dans le temps :

$$\dot{\xi}(t) = 0 \implies \Delta \dot{\xi}(t) = \dot{\hat{\xi}}(t)$$

La loi permettant de mettre à jour le vecteur de paramètre ξ est :

$$\Delta \dot{\xi}(t) = -\Lambda \Delta v(t) \hat{f}(t) \quad (2.54)$$

Avec $\Lambda = \text{diag}(\lambda_i)$, pour $i = 1..size(\xi)$, et $\lambda_i > 0$ des gains permettant d'ajuster la vitesse de convergence selon chaque paramètre. La stabilité de cet observateur peut être démontrée par la théorie de Lyapunov [SW03].

De plus, différentes méthodes utilisant le filtre de Kalman¹² existent pour déterminer les paramètres hydrodynamiques d'un véhicule marin. La méthode que l'on retrouve le plus souvent pour ce domaine d'application est le filtre de Kalman étendu (FKE). On la retrouve par

12. Le filtre de Kalman est un filtre à réponse impulsionnelle infinie qui estime les états d'un système dynamique à partir d'une série de mesures incomplètes ou bruitées. Le filtre a été nommé d'après le mathématicien et informaticien américain d'origine hongroise Rudolf Kalman.

exemple dans Shi [SZPS09], Sabet [SSZ14] [SDFA18], Ernani [ZBE15] et Alotta [ACC⁺15]. Ce dernier l'utilise directement dans l'algorithme de navigation. Une autre méthode est l'utilisation du filtre de Kalman « unscented », comme cela est proposé dans Tiano [TSLN07], Sabet [SSZ14] ou Alotta [ACC⁺15]. Sabet [SSZ14] propose une bonne comparaison de ces deux méthodes.

L'estimation adaptative est appropriée pour l'estimation de paramètre en direct au cours d'une mission. Mais elle est probablement moins précise que d'autres méthodes plus simples comme celle des moindres carrés, présentée dans le paragraphe 2.4.2.3.

2.4.2.2 Techniques d'apprentissage automatique et réseaux neuronaux

Les techniques d'apprentissage automatique sont les précurseurs des techniques utilisant les réseaux neuronaux. Le principe de ces techniques est de rechercher statistiquement la meilleure solution pour permettre aux données mesurées de correspondre à un modèle prédéfini. Globalement ces méthodes permettent d'obtenir et d'affiner des paramètres pour faire fonctionner les schémas de commande des véhicules sous-marins. Ces paramètres sont parfois les paramètres hydrodynamiques que l'on souhaite déterminer. Par exemple, la méthode de régression à vecteurs de support, en anglais « Support Vector Regression », est utilisée par Wehbe [WHK17] [WK17] [WHK19] ou Xu [XZYC13]. D'autres méthodes sont utilisées, comme la méthode « kernel ridge regression » ou la méthode « gaussian process regression » que l'on retrouve dans Wehbe [WHK17]. L'apprentissage automatique, pourtant plus ancien que les méthodes récentes de réseaux neuronaux, est pourtant moins courant, probablement parce qu'il est plus complexe à mettre en place.

Un réseau de neurones, est un système artificiel dont la conception est à l'origine schématiquement inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques. Le réseau de neurones est toujours utilisé en parallèle du schéma de contrôle ou du modèle dynamique pour contrôler le véhicule. Van de Ven [vdVJS⁺07] nous rapporte que les techniques utilisant des réseaux de neurones qui permettent d'estimer les paramètres hydrodynamiques peuvent être distingués en trois catégories :

- **Apprentissage et contrôle simultanés** : le véhicule apprend à améliorer ses performances en cours de mission. Cette technique est employée, par exemple, par Ishii [IFU95], Wang [WLY] [WL03], Labonte [Lab], et plus récemment Cui [CYLS17] et Peng [PWW19].
- **Apprentissage et contrôle séparés** : l'apprentissage du véhicule se fait avant d'appliquer ce que le réseau de neurones a appris pour le contrôle. Cette technique est employée par Ishii [IU00].
- **Contrôle augmenté** : Ajout d'un réseau de neurones à un schéma de commande non-linéaire classique, ce qui lui assure une stabilité minimale du contrôle et permet d'augmenter ses performances.

Ces méthodes semblent très lourdes à mettre en place et sont probablement plus adaptées pour l'estimation de paramètres hydrodynamiques dans des cas plus complexes.

2.4.2.3 Les méthodes des moindres carrés

Les méthodes des moindres carrés consistent à comparer une série de données expérimentales à un modèle mathématique censé représenter ces données. Le principe de ces méthodes est de minimiser l'impact des erreurs expérimentales à partir d'un maximum de données.

En général, les problèmes traités pour l'obtention de paramètres hydrodynamiques ont des modèles non-linéaires. L'une des méthodes les plus courantes pour traiter ce genre de problèmes est d'utiliser un algorithme de Gauss-Newton, comme décrit dans Gartner [GRH18] et probablement assez proche de ce qui est utilisé par Eidsvik [ES16], Caccia [CIV00], Ridao [RTEF⁺04] et Eng [ELLS08]. L'algorithme de Gauss Newton est un algorithme itératif qui considère un set de paramètres $\vec{\xi}$ à estimer avec, par exemple dans le cas du balancier $\vec{\xi} = [m_a, K_l, K_q]^T$, et de minimiser un résidu $\vec{R}(\vec{\xi})$. Ce résidu correspond à l'erreur restante entre les valeurs mesurées et les valeurs obtenues en prenant en compte la valeur actuelle des paramètres estimés pour correspondre à un modèle défini. Dans le cas du balancier, le résidu est :

$$\vec{R}(\vec{\xi}) = \vec{\theta}^m(t) - \vec{\Phi}(\vec{\theta}^m(t), \dot{\vec{\theta}}^m(t), \vec{\xi}) \quad (2.55)$$

Avec $\vec{\theta}^m$ le vecteur $N \times 1$ position angulaire mesurée, $\dot{\vec{\theta}}^m$ et $\ddot{\vec{\theta}}^m$, respectivement les première et seconde dérivée de $\vec{\theta}^m$ après filtrage, le cas échéant. N est le nombre de points mesurés par le capteur. Le vecteur accélération estimée de dimensions $N \times 1$, dans le cas du balancier, a pour expression :

$$\vec{\Phi}(t) = \frac{P_{eq}}{Mr} \sin \vec{\theta}^m(t) - \frac{K_l}{M} \ddot{\vec{\theta}}^m(t) - \frac{K_q r}{M} |\dot{\vec{\theta}}^m(t)| \dot{\vec{\theta}}^m(t) \quad (2.56)$$

La prochaine valeur de ξ est obtenue à l'aide du résidu :

$$\vec{\xi}_{n+1} = \vec{\xi}_n - K_\xi \vec{\nabla}_R^+(\vec{\xi}_n) \vec{R}(\vec{\xi}_n) \quad (2.57)$$

Avec K_ξ un gain fixe utilisé pour réguler le taux de convergence de l'algorithme de Gauss-Newton et $\vec{\nabla}_R^+(\vec{\xi}_n)$ la pseudo inverse de la matrice jacobienne $\vec{\nabla}_R(\vec{\xi}_n)$ qui est calculée comme suit dans le cas du balancier :

$$\vec{\nabla}_R(\xi_n) = - \left[\frac{\partial \vec{\Phi}}{\partial M_a} \quad \frac{\partial \vec{\Phi}}{\partial K_l} \quad \frac{\partial \vec{\Phi}}{\partial K_q} \right] \quad (2.58)$$

Cette méthode permet d'atteindre les mêmes résultats que la méthode des moindres carrés classique dans le cas d'un modèle linéaire. Le choix de la première valeur des paramètres du vecteur ξ est important car il peut mener après itération à un minimum local et non global, ou mener à un minimum local se trouvant en dehors des valeurs admissibles pour les paramètres, i.e. un coefficient de frottement ou de masse ajoutée négatif.

2.4.3 Solution numérique proposée (NIPF)

La méthode Numerical Integration Position Fitting (NIPF), particulièrement adaptée au mécanisme du balancier hydrodynamique, a été développée. La première étape consiste à intégrer numériquement l'équation 2.50 en utilisant la méthode Runge-Kutta d'ordre 4. (voir annexe C) Les paramètres sont estimés grâce à l'algorithme points intérieurs [BGN00, WMNO05] en utilisant la fonction de coût normalisée suivante :

$$J = \frac{(f_s^P - f_m^P)^2}{(f_m^P)^2} G_f + \frac{(\varphi_s^P - \varphi_m^P)^2}{4\pi^2} G_\varphi + \frac{\overline{\Delta\theta} \Delta\theta^T}{\overline{\theta}^m \overline{\theta}^m T} G_\Delta \quad (2.59)$$

Avec f_s^P et f_m^P les fréquences principales des signaux simulés et mesurés, φ_s^P et φ_m^P les phases reliées aux fréquences de ces deux signaux. G_f , G_φ et G_Δ sont des gains choisis pour équilibrer

les effets des éléments de la fonction de coût. $\overrightarrow{\Delta\theta} = \overrightarrow{\theta^s} - \overrightarrow{\theta^m}$ avec $\overrightarrow{\theta^s}$ le signal obtenu par intégration numérique. $\overrightarrow{\theta^m}$ est le signal non filtré mesuré. La fonction de coût J utilise la phase et la fréquence pour rendre la convergence plus rapide, plus régulière et plus précise. La phase et la fréquence sont obtenues grâce à la technique FFT (Fast Fourier Transform).

Cette fonction de coût a été choisie pour améliorer la précision de l'estimation de masse ajoutée. L'inertie a une grande influence sur la fréquence d'oscillation, donc cette fonction de coût permet de découpler l'estimation de l'inertie et du coefficient de friction. Chaque intervalle de paramètres est borné par des valeurs réalistes. Tous les paramètres doivent être positifs et la limite supérieure est définie avec la valeur empirique maximale connue.

L'ensemble des paramètres initiaux choisis pour l'identification sont les paramètres théoriques et empiriques :

$$\overrightarrow{\xi_{init}} = [m_a^{th}, 0, C_d^*] \quad (2.60)$$

Avec $C_d^* = 0.41$ un coefficient de frottement adimensionnel moyen pour la sphère pour la plage de 10^3 - 10^5 Reynolds, qui est décrit dans le chapitre 2.2.2.5 et m_a^{th} la valeur de masse ajoutée théorique pour notre sphère. Nous considérons qu'en théorie il n'y a pas de frottement linéaire en parallèle du frottement quadratique.

2.4.4 Validation de la méthode NIPF

Une façon de vérifier la qualité de l'identification par la méthode NIPF est de comparer cette méthode avec une méthode classique de Gauss Newton en utilisant une solution analytique du modèle dynamique. En fait, en considérant seulement l'amortissement linéaire ($K_q = 0$) et en considérant de petites oscillations ($\sin \theta \approx \theta$), la solution de l'équation 2.50 est :

$$\theta(t) = \left(\theta_0 \cos \beta t + \frac{-\alpha \theta_0 + \dot{\theta}_0}{\beta} \sin \beta t \right) e^{\alpha t} \quad (2.61)$$

Avec $\alpha = \frac{-K_l}{2M}$, $\beta = \frac{\sqrt{-4rP_{eq}M - K_l^2 r^2}}{2rM}$.

En appliquant un algorithme de Gauss Newton sur l'ensemble des paramètres $\overrightarrow{\xi} = [\alpha, \beta]$, le $R(\overrightarrow{\xi}) = \overrightarrow{\theta^m(t)} - \overrightarrow{\Phi(t, \overrightarrow{\xi})}$ est actualisé avec l'équation 2.57. Ensuite, pour comparer les résultats, la fonction de coût (Équation 2.59) est appliquée sur le signal reconstruit $\overrightarrow{\theta^s}$ en utilisant les coefficients estimés. Cette méthode est appelée Analytical Integration Position Fitting (AIPF), et est basée sur la méthode introduite par Langmajer [LB16a].

Les résultats comparés sont affichés sur le tableau 2.4 où les méthodes sont appliquées sur les 5 premières périodes de chaque signal pour un essai à chaque vitesse maximale (voir Tableau 2.3). Les résultats entre les deux méthodes sont très proches (moins de 3% de différence), ce qui confirme que la méthode d'intégration numérique peut être utilisée pour l'identification des paramètres. Comme il n'est pas toujours possible de résoudre analytiquement des équations différentielles non linéaires, la méthode d'intégration numérique sera privilégiée pour l'identification des paramètres. Cette méthode est comparée à une méthode classique d'ajustement des dérivées dans la section suivante.

2.4.5 Justification de la méthode NIPF

L'algorithme de Gauss Newton classique (GN) décrit dans la section 2.4.2.3 requiert une double dérivation de l'angle mesuré. Cette double dérivation conduit à une accélération très bruitée. (Figure 2.15).

v_{max} ($m.s^{-1}$)	Method	$m_a(kg)$		K_l		Cost
		Value	Var	Value	Var	Value
0.4	AIPF	1.23		1.19		3.31
0.4	NIPF	1.19	-3%	1.18	-1%	3.64
0.3	AIPF	1.60		1.05		0.21
0.3	NIPF	1.57	-2%	1.05	0%	0.19
0.2	AIPF	1.85		1.26		1.29
0.2	NIPF	1.83	-1%	1.25	-1%	1.29

Tableau 2.4 – Comparaison des résultats avec les méthodes AIPF et NIPF

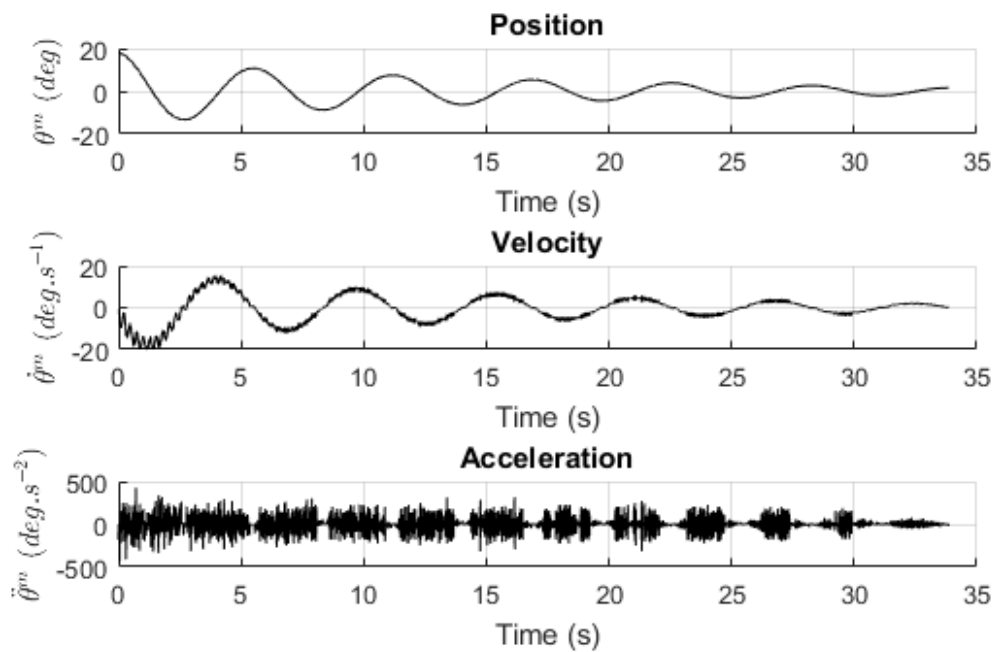


Figure 2.15 – Signal angulaire mesurée et ses dérivées sans filtre

Pour réduire le bruit, un filtrage numérique avec compensation du déphasage¹³ est appliqué après la première dérivation pour s’assurer qu’il n’y a pas de distorsion de phase pendant le filtrage. Un filtre numérique de Butterworth passe-bas d’ordre 3 est utilisé. Comme le pendule a une fréquence de $0,2 Hz$, la fréquence de coupure doit être d’au moins $0,4 Hz$.

Six filtrages différents sont comparés dans le tableau [Tableau 2.5](#) en utilisant un modèle dynamique impliquant un amortissement linéaire et un amortissement quadratique. Sans filtrage ou avec un filtre haute fréquence, l’algorithme ne converge pas. Pour les basses fréquences de coupure ($0,4 Hz$), qui sont proches de la fréquence pendulaire ($0,2 Hz$), le signal est déformé en raison de la trajectoire oscillante. Les fréquences de coupure autour de $2 Hz$ (10 fois la fréquence d’oscillation du pendule) sont de bons compromis pour couper les oscillations perturbatrices sans déformer le signal de trajectoire. Malgré la convergence, les valeurs estimées entre l’algorithme de Gauss Newton et NIPF varient (au moins 3% pour

13. La fonction filtfilt de Matlab

Tableau 2.5 – Comparaison des résultats avec différents filtres - Modèle de frottement LQ

Filter	Cutoff (Hz)	Method	$m_a(kg)$		K_l		K_q		Cost function
			Value	Var	Value	Var	Value	Var	
none		NIPF	1.57		0.67		1.75		0.06
none		GN	2.12	35%	15.7	> 100%	-130	NA	170
0.4		GN	1.68	7%	0.74	10%	3.98	> 100%	26
1		GN	1.51	-4%	0.78	17%	0.87	-50%	0.08
2		GN	1.53	-3%	0.83	24%	0.55	-68%	0.08
4		GN	1.51	-4%	0.78	16%	0.86	-51%	0.09
20		GN	1.47	-6%	1.82	171%	-6.36	NA	12.5

m_a , 16% pour K_l et 50% pour K_q), ce qui montre la forte influence du filtrage. La technique NIPF est un bon moyen d'éviter un filtrage qui a un impact non négligeable sur les résultats.

2.4.6 Conclusion

Parmi les différents moyens expérimentaux existants, le balancier hydrodynamique a été sélectionné. La première contribution de cette thèse a été le développement de la méthode NIPF, publiée dans l'article [GRH18], pour exploiter les données récoltées. NIPF permet d'éviter l'utilisation de filtrage et sa robustesse par rapport à la méthode de Gauss Newton a été mis en évidence. La partie suivante analyse les résultats obtenus grâce à cette méthode en matière de variabilité ainsi que leur évolution au regard de la vitesse de la sphère.

2.5 Analyse des résultats obtenus avec l'expérience du balancier

2.5.1 Influence du nombre de périodes du balancier pour NIPF

Les résultats des 9 essais décrits dans le tableau 2.3 utilisant la méthode NIPF et l'équation avec le modèle de frottement linéaires et le modèle quadratique sur 2, 3 et 4 périodes de temps sont présentés sur la figure 2.16.

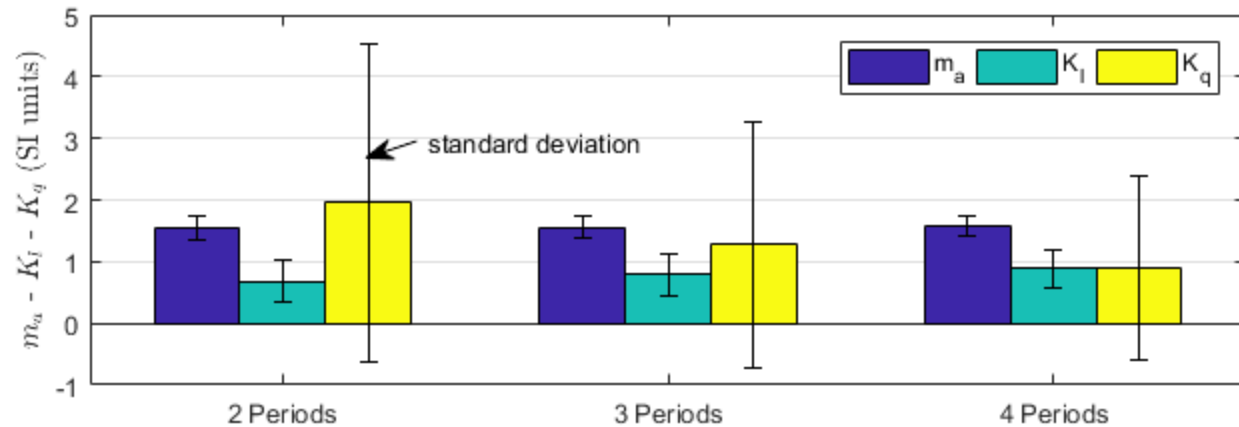


Figure 2.16 – Identification des paramètres (9 essais) sur 2, 3 et 4 périodes. LQ : Modèle de frottement linéaires et quadratiques.

En ce qui concerne la masse ajoutée, le nombre de périodes d'oscillations du balancier a peu d'influence. En effet, seule la masse ajoutée influence la fréquence du pendule et cette dernière est prise en compte dans la fonction de coût (Équation 2.59). Par contre, les résultats pour les deux coefficients de frottement sont moins précis, avec un écart-type supérieur à la moitié de la valeur moyenne pour K_l et 2 fois la valeur moyenne pour K_q . Par conséquent, la méthode NIPF se révèle efficace pour déterminer le coefficient de masse ajoutée, mais il semble difficile d'obtenir des coefficients pour le modèle de frottement LQ, ce qui sera discuté par la suite.

2.5.2 Variabilité de la masse ajoutée en fonction du modèle de frottement choisi

Pour vérifier l'influence du modèle de frottement sur la masse ajoutée, la méthode NIPF est utilisée en considérant dans le modèle dynamique (Équation 2.50), soit uniquement le coefficient de frottement linéaire, ce modèle est nommé L ($K_q = 0$), soit uniquement le coefficient de frottement quadratique, ce modèle est nommé Q ($K_l = 0$). Ces deux modèles sont comparés au modèle complet LQ avec les 9 essais décrits dans Tableau 2.3. Pour obtenir une évolution en fonction de la vitesse, l'essai est découpé en petites plages de 1 s de longueur, qui se décalent de 0.25 s. Certaines plages se chevauchent donc. Une estimation est faite sur chacune des plages. Les résultats sont affichés sur la figure 2.17 en fonction de la vitesse maximale enregistrée dans la plage sélectionnée.

Cette figure montre que les valeurs de la masse ajoutée déterminées avec la méthode NIPF sont cohérentes, quel que soit le modèle d'amortissement. La moyenne des valeurs est

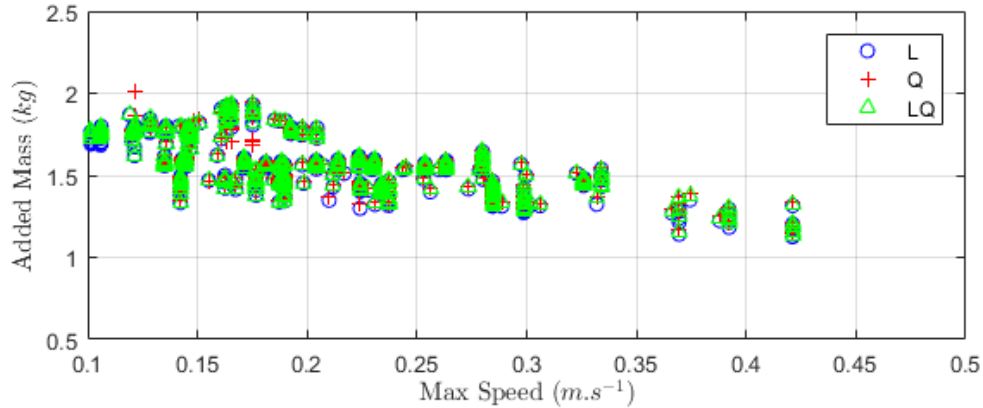


Figure 2.17 – Masse ajoutée en fonction de la vitesse maximale au cours des 9 essais. L, Q, LQ : Modèles de frottement Linéaires, Quadratiques, and Linéaires et Quadratiques

la même pour les trois modèles : $\overline{M_a} = 1,56 \text{ kg}$ avec un écart type de 0,16, ce qui est faible devant la moyenne et justifie une bonne précision. Si on essaie de faire passer une droite par l'ensemble des points on obtient pour les 3 ensembles la même équation de droite avec une pente légèrement décroissante :

$$m_a(v_{max}) = -0.1v_{max} + 1.56 \quad (2.62)$$

Pour vérifier la convergence de l'algorithme NIPF, les variations de la fonction de coût par rapport à une large variation des coefficients m_a , K_l et K_q pour les deux modèles d'amortissement simples L et Q sont présentées sur la figure 2.18. Sur les figures en trois dimensions, on remarque que les surfaces possèdent un unique minimum. Les figures de droite se concentrent sur toutes les valeurs dont le coût est inférieur à 5. Ceci correspondrait à des couples candidats valides pour les coefficients avec une erreur totale sur trajectoire inférieure à 20% : une marge acceptable pour ce type d'identification de paramètre. Le point rouge met en évidence le point de convergence de l'algorithme NIPF. On constate que l'algorithme converge correctement vers un couple valide de valeurs pour chacun des deux modèles. Cependant, l'algorithme NIPF est plus sensible au paramètre de masse ajoutée qu'au paramètre de frottement. Sur les figures où sont représentées les zones dont le coût est inférieur à 5, le paramètre coefficient de frottement a moins d'influence sur la fonction de coût que la masse ajoutée.

En ce qui concerne la détermination des coefficients d'amortissement, utilisons le modèle LQ et concentrons-nous sur les variations de la valeur de coût par rapport aux coefficients d'amortissement en utilisant une masse ajoutée prédéterminée. Ces variations sont représentées sur la Figure 2.19. En fait, il y a beaucoup de couples correspondants (K_l, K_q) à cause de la surface plate près du coût minimum. Ceci explique pourquoi l'algorithme des points intérieurs de la méthode NIPF ne converge pas systématiquement vers le même ensemble de valeurs, ce qui conduit à des écarts-types importants pour K_l et K_q comme indiqué sur la Figure 2.16.

2.5.3 Comparaison des résultats avec des valeurs de références

Pour comprendre la répartition entre écoulement laminaire et turbulent autour de la sphère, le nombre de Reynolds est calculé en fonction de la vitesse de la sphère à chaque instant. Pour

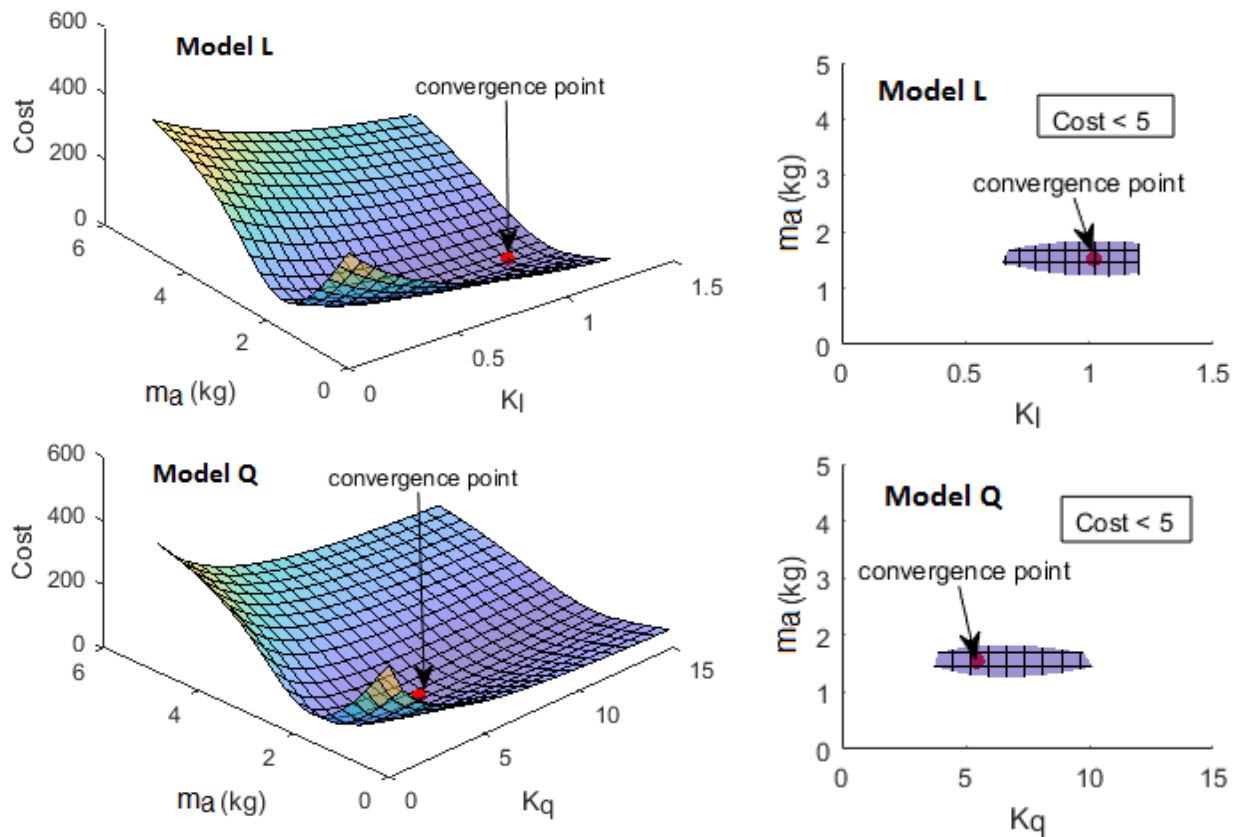


Figure 2.18 – Variations des coûts par rapport aux coefficients pour l'essai 7.
L : frottements linéaires, Q : frottements quadratiques

les vitesses qui ont été mesurées : 100% des points de mesure correspondent à un écoulement turbulent ($Re > 1$). Le nombre de Reynolds de l'écoulement est compris entre $Re = 10^2$ et $Re = 10^5$. On imagine alors que le coefficient de frottement quadratique de la sphère doit évoluer entre $C_d = 1,0$ et $C_d = 0,41$ (§ 2.2.2.5). Ainsi, à l'aide de l'équation 2.44, on calcule que le coefficient quadratique est compris entre $7,6 N.m^2.s^{-2} < K_q < 18,5 N.m^2.s^{-2}$. Ces coefficients sont représentés sur la figure 2.20.

Les valeurs estimées pour K_q à vitesse faible se situent dans la fourchette des valeurs données par Blevins [Ble03], tandis que les valeurs à vitesse élevée sont légèrement inférieures de 20 %. Cela peut être dû à l'effet de sillage causé par le mouvement oscillatoire lorsque l'objet revient dans le sillage de l'élan précédent. Cela reste à analyser.

La masse ajoutée théorique estimée pour la sphère avec l'équation 2.37 est de 2,7 kg, soit 69 % de plus que la valeur moyenne estimée de 1,6 kg. Les hypothèses de la section 2.2.1 concernant le calcul de la masse ajoutée théorique ne sont pas valables ici, puisque le flux autour de la sphère ne peut être considéré uniforme et doit être considéré visqueux. Yang [YCM⁺15] a comparé la masse ajoutée d'une sphère estimée avec un calcul CFD et théorique. Les valeurs estimées sont similaires parce qu'elles sont issues du même modèle à petite échelle : Navier-Stokes avec les mêmes hypothèses. Dans le cas de notre mouvement qui n'est pas rectiligne et qui est oscillatoire dans un fluide réel cette valeur théorique est fautive. L'utilisation d'un dispositif expérimental tel que le pendule à décroissance libre permet dans ce cas d'obtenir des paramètres valables et ceux-ci peuvent être intéressants car cela ressemble aux conditions de navigation d'un ROV soumis à la houle. Une analyse plus approfondie des

effets de la trajectoire sera réalisée en simulation dans le chapitre 4.

2.5.4 Discussion sur les modèles de frottement

La méthode NIPF montre que le modèle de friction LQ n'est pas bien adapté puisque l'algorithme d'optimisation ne permet pas de séparer définitivement les termes de friction K_l et K_q car ils peuvent se compenser mutuellement. En effet, dans la section 2.5.2, il a été constaté qu'un large éventail de couples de coefficients (K_l, K_q) permet d'obtenir une faible valeur de coût, ce qui compromet la convergence de l'algorithme NIPF. Ceci est peut-être dû à la méthode expérimentale. Par conséquent, on peut se poser la question : doit-on utiliser un seul coefficient de frottement - K_l ou K_q ?

En comparaison avec le modèle d'amortissement quadratique, la figure 2.21 présente les résultats obtenus avec le modèle d'amortissement linéaire. Pour les basses vitesses, inférieures à $0,18 \text{ m/s}$, la variabilité plus élevée de K_l s'explique par la présence de frottement dans les roulements à billes qui réalisent la liaison pivot de la structure. La détermination d'un coefficient d'amortissement linéaire est intéressante, car ce coefficient pourrait être utilisé pour créer une commande prédictive complexe, en tirant parti de la linéarité du modèle dynamique. L'utilisation d'un unique coefficient de frottement semble simplifier la détermination. Cette expérience ne permet pas de conclure sur le choix entre coefficient de frottement quadratique ou linéaire.

La méthode NIPF doit prendre en compte au moins une période d'oscillation allant de 0 à la vitesse maximale avant ou arrière, la fonction de coût dépendant de la fréquence et de la phase d'oscillation. La valeur de la vitesse maximale est choisie comme référence, car elle est plus représentative que la vitesse moyenne de la trajectoire. En effet, les coefficients résultant du processus d'identification ne doivent pas être considérés comme se rapportant à une vitesse donnée. Ils doivent être considérés comme un coefficient moyen pour une plage de vitesse déterminée.

Pour expliquer la difficulté de déterminer le coefficient de frottement, les variations des forces de traînée (modèle Q) sur la sphère sont représentées en fonction du temps sur Figure 2.22. Dans plus de 70% des cas, la force de masse ajoutée est supérieure à la force de frottement. (Courbe pointillée au-dessus de la courbe pleine) En conclusion, le pendule à oscillation libre et la méthode NIPF se sont révélés efficaces pour déterminer la masse ajoutée,

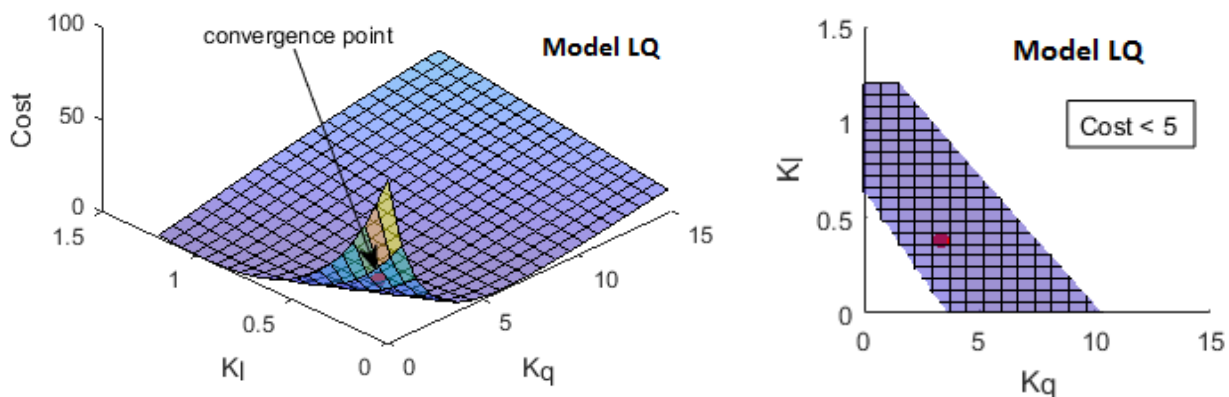


Figure 2.19 – Variation des coûts inférieure à 1 en ce qui concerne les coefficients K_l et K_q - essai 7 - $m_a = 1.57 \text{ kg}$. LQ : Modèle de frottement linéaire et quadratique.

mais comme le coefficient d'amortissement varie avec la vitesse, un dispositif expérimental plus approprié pourrait être utilisé pour déterminer les coefficients de friction sur une plus grande plage de vitesse.

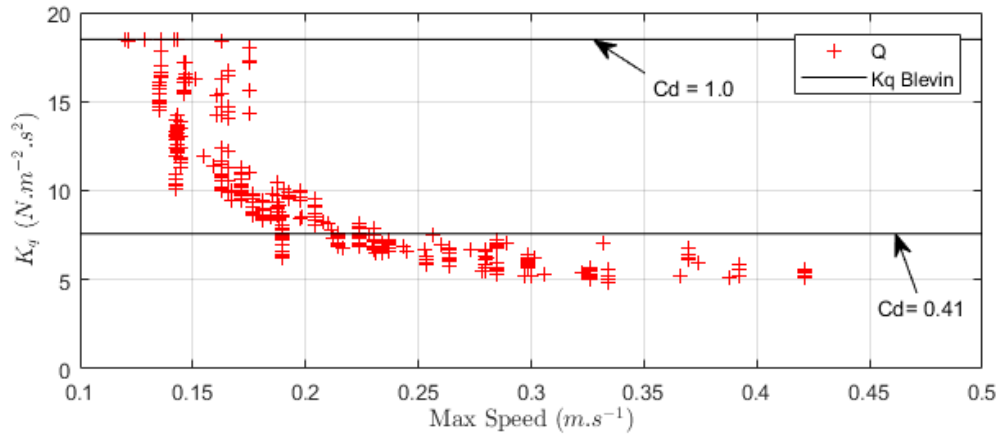


Figure 2.20 – Évolution du frottement quadratique en fonction de la vitesse la plus élevée dans la plage glissante au cours des 9 essais comparée à deux C_d qui délimitent les plages de Reynolds caractéristiques

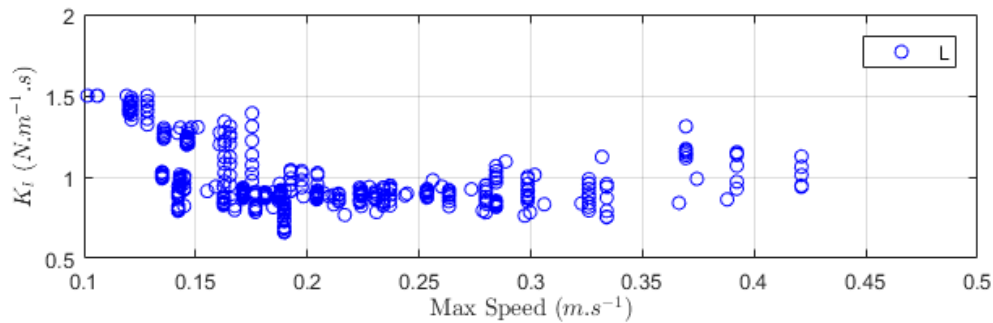


Figure 2.21 – Évolution du frottement quadratique en fonction de la vitesse la plus élevée dans la plage glissante au cours des 9 essais

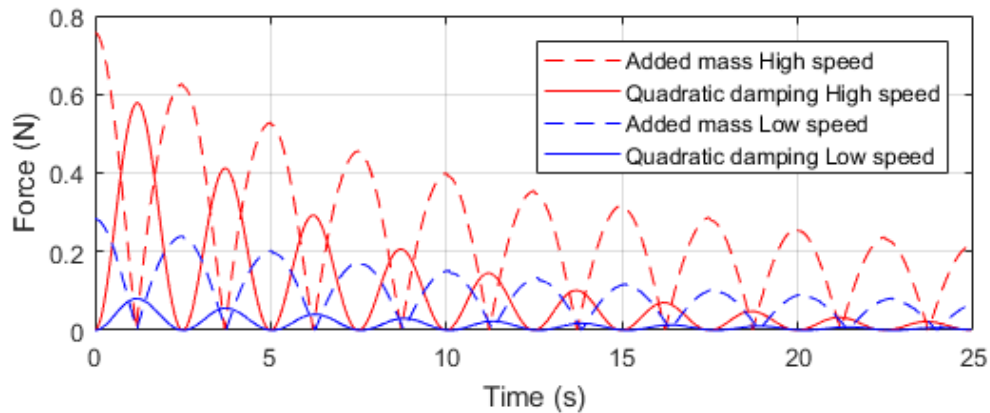


Figure 2.22 – Force de masse ajoutée et force de traînée appliquées sur la sphère pendant l'oscillation à une vitesse maximale donnée. (Modèle Q).

2.6 Conclusion

La première contribution de cette thèse a été présentée : une nouvelle méthode numérique d'identification des paramètres hydrodynamiques d'un objet immergé exploitant les mesures de position angulaire issues d'un balancier hydrodynamique. Cette contribution a été publiée dans l'article [GRH18]. La méthode Numerical Integration Position Fitting (NIPF) a été comparée avec la méthode de Gauss Newton. Elle se révèle plus robuste et plus précise. Grâce à cette méthode, il est possible de dissocier l'estimation de la force de masse ajoutée des forces de frottement. Trois modèles de frottement ont été utilisés : un modèle linéaire, un modèle quadratique et un modèle combinant ces deux derniers. S'il est possible d'écarter le modèle combiné, il reste difficile de choisir entre le modèle linéaire et le quadratique, bien que le régime de l'écoulement soit turbulent. De plus, comme attendu, ces coefficients sont variables en fonction de la vitesse mesurée.

La plupart du temps, les simulateurs utilisés pour la robotique se servent d'un coefficient de frottement constant. Du fait de la variation constatée dans nos mesures, l'utilisation d'un coefficient constant apparaît mal adaptée. Dans le simulateur proposé dans le cadre de cette thèse, ces variations en fonction de la vitesse sont effectives grâce à la simulation de l'écoulement via SPH. Ceci rend son utilisation plus pertinente pour les simulations de la dynamique du véhicule.

Le prochain chapitre (3) présente la méthode SPH permettant de simuler la dynamique du fluide ainsi que les premières étapes de la démarche de validation du simulateur évoquées dans la section 1.6.1, avec notamment, les étapes qui concernent l'hydrostatique et la dynamique du fluide seul. Le chapitre 4 est consacré à la validation des simulations dynamiques du véhicule avec SPH.

Chapitre 3

Simulation hydrodynamique basée SPH

Après une étude des solutions provenant des domaines d'application de l'animation et des jeux vidéo, notamment Nvidia PhysX et Nvidia FleX (§ 1.5.3), et une analyse des différentes possibilités de simulation des véhicules sous-marins (§ 1.6), il apparaît que l'une des solutions les plus prometteuses pour simuler leur comportement en milieu aquatique agité est l'utilisation de la technique SPH (Smoothed Particle Hydrodynamics). Cette méthode permet la simulation simultanée de la dynamique du fluide et du solide, ce qui devrait être plus précis que l'utilisation de paramètres hydrodynamiques constants. De plus, dans le domaine de l'animation, les calculs avec SPH sont réalisés en temps réel.

La première partie de ce chapitre est consacrée à un historique du développement de cette méthode et explique ses grands principes, notamment le fonctionnement de l'interpolation et de la discrétisation spatiale, détaillé dans le paragraphe 3.2.

La deuxième partie de ce chapitre est dédiée à la modélisation de l'écoulement. Il existe différentes manières de résoudre les équations de Navier-Stokes. La nature de l'écoulement permet de choisir entre différentes catégories d'algorithmes de résolution de ces équations. Deux classes d'algorithmes, Incompressible SPH (ISPH) et Weakly Compressible SPH (WCSPH) sont sélectionnées pour le simulateur. Ces deux algorithmes sont présentés dans le paragraphe 3.3. La manière dont le fluide et le solide interagissent, qui est le cœur de cette thèse, est particulièrement détaillée dans la section 3.4. La seconde contribution de cette thèse consiste à valider l'utilisation d'une méthode de modélisation de l'interaction fluide-solide au sein des deux algorithmes SPH.

Finalement, la troisième partie consiste à évaluer le simulateur en suivant la démarche de validation décrite dans la section 1.6.1. Le comportement du simulateur dans un cas hydrostatique et un cas hydrodynamique est analysé. La troisième contribution de cette thèse réside dans le choix entre deux algorithmes SPH : ISPH et WCSPH, grâce à une étude du ratio temps simulé/temps de calcul et de l'analyse des résultats dans un cas hydrodynamique (solide statique).

3.1 Introduction à la méthode Smoothed Particle Hydrodynamics (SPH)

Smoothed Particle Hydrodynamics (SPH) est une méthode de simulation particulièrement adaptée à la modélisation des mouvements d'un fluide. Contrairement aux méthodes classiques, cette méthode n'utilise pas un maillage mais un ensemble de particules de fluide qui se déplacent avec l'écoulement en respectant les lois de la dynamique des fluides et dont les interactions sont pondérées par une fonction de lissage, dite kernel. Cette première partie de chapitre présente un historique de l'évolution de l'utilisation de SPH puis ses fondements scientifiques.

3.1.1 Évolution historique des applications de la méthode SPH

La méthode SPH a été introduite en 1977 par deux équipes d'astrophysiciens : Gingold et Monaghan [GM77] d'une part et Lucy [Luc77] d'autre part. Elle était destinée au départ aux calculs de la dynamique des fluides pour des objets célestes, où la densité du fluide peut varier énormément dans le temps et l'espace en l'absence de frontière de domaine, comme par exemple, les supernova [HP99] [TC01], les explosions de naines blanches [GSBGW99] ou même pour la prédiction de l'évolution de l'univers [Mon90]. Au fil du temps, son utilisation a été étendue à différents domaines, ce qui est représenté sur la figure 3.1 :

- L'énergie : où elle est utilisée, par exemple, pour le calcul de l'écoulement autour d'une turbine génératrice hydraulique [KAP11] [MLCP10] ou des turbines génératrices au gaz [BHKB13], ou encore pour évaluer les performances de générateurs à colonne d'eau oscillante [CAD⁺17].
- L'ingénierie civile : pour l'évaluation de structure côtière, comme par exemple, l'étude de l'interaction des vagues avec une structure brise-lames [ACR⁺14] ou également l'étude d'impact de vagues sur des bâtiments [GGD04] [SMA⁺16] [SGNTS14].
- Les méthodes de fabrication : pour l'étude de moulage [CHAN02] [Cle98] [BK00], de la formation de copeaux [ASAA08] [SEF13] ou encore du forgeage [CPDH12].
- Le nucléaire ou la physique des particules : pour l'étude de collisions relativistes de particules nucléaires [AKOH00] [AGHQ12]
- La chimie : l'étude de réaction ayant lieu dans un milieu poreux [TMSW07], l'étude de réaction ayant lieu dans un brûleur à jet d'air/eau [CFL⁺03], la fabrication de polymère [RKF⁺13], l'étude de réaction chimique [GGBK09] ou encore l'étude des effets de mélanges [RCM08].
- L'animation 3D : où elle est utilisée pour générer une animation d'un écoulement de fluide. La rapidité de cette technique, comparée à des techniques de calcul de fluides classiques, et la possibilité d'utiliser le calcul parallèle l'ont rendue très attractive pour ce milieu. L'objectif pour cette application est donc particulier : il faut permettre une simulation de fluide qui semble réaliste, mais dont on ne vérifiera pas la réalité physique, en optimisant le temps de calcul [MCG03]. Des développements ont été menés pour :
 - la parallélisation du calcul [IABT10] [ICS⁺14]
 - des techniques mélangeant fluide et maillage [CIPT14] [CMK15]
 - des méthodes de simplification d'algorithme comme, par exemple, pour la frontière solide [AIA⁺12]
 - des méthodes de simulation multi-physique, avec des interactions fluide-solide, fluide-vêtements [MMCK14]

Certains de ces développements ont pris la forme de logiciels commerciaux : Nvidia PhysX et Nvidia FleX.

- La géophysique : pour l'étude de l'érosion [BMHDGH03], les écoulements à travers les milieux poreux [ZFM99] [ZF02] [TM06], l'étude de glissement de terrain [MH04], l'étude des mouvements du sol [BFSO08] ou encore l'étude de désastres géologiques (mouvement de plaque tectonique, occurrence de séisme, etc.) [HD14]
- L'océanographie : comme par exemple, l'étude des écoulements proches des plages en fonction de leur topologie [FDHB14] [MK99] ou encore l'étude des mouvements des sédiments [ON16] [TDPTK17].
- Enfin, depuis récemment, l'ingénierie océanique avec différentes applications :
 - La protection de l'environnement : comme par exemple, pour le calcul de l'efficacité de barrage flottants [VBAMdN07] [YL13].
 - L'armement : l'étude d'explosion sous-marine [LLLZ03], l'étude d'explosion pour la fragmentation des roches [PD14]
 - L'architecture navale, comme par exemple, le flux généré par une hélice de bateau et le largage d'une structure dans l'eau [ULR13], l'étude de cuve anti-roulis [SIDPRAP06], l'étude des effets de l'eau sur une éolienne marine [LB16b] ou encore la simulation de l'interaction d'une vague (forte) avec une plate-forme de forage pétrolier en mer [DCVB⁺13]. Les premières études pour ce domaine furent menées par Colagrossi [CL03].

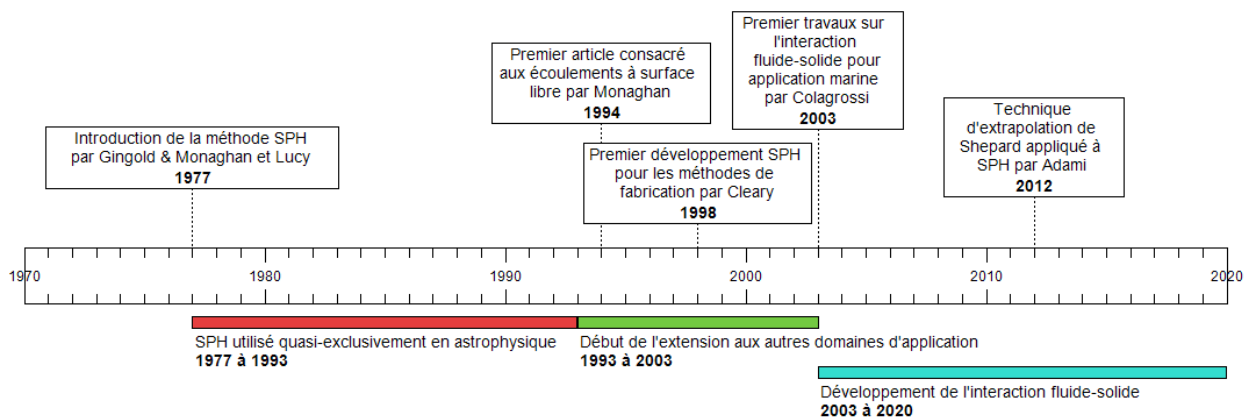


Figure 3.1 – Chronologie du développement des applications de la méthode SPH.

Différents articles états de l'art avec un focus plus ou moins large peuvent être trouvés pour cette méthode : Monaghan [Mon05], Liu [LL10], Monaghan encore [Mon12], Shadloo [SOT16] et Violeau [VR16]. L'article de Violeau traite uniquement les problèmes d'écoulement à surface libre, ce qui est particulièrement intéressant. Dans le cas d'un écoulement biphasique, avec de l'eau et de l'air, dans lequel l'une des phases n'est pas significativement impactée par l'autre, l'interface fluide-fluide peut être vu comme une simple frontière, ce qui permet d'utiliser un modèle monophasique de l'écoulement. La surface libre modélise parfaitement les effets des vagues et de surface, en l'absence de phénomènes créant des mélanges d'eau et d'air ou de l'écume.

Comme on peut le constater il existe de très nombreuses applications à SPH, puisque dès qu'il est question de simulations de fluide, que ce soit d'un liquide ou d'un gaz, il est possible

d'utiliser cette méthode. Évidemment, la technique utilisée au départ en astrophysique a beaucoup évolué et a dû être adaptée pour être applicable dans tous ces différents domaines.

3.1.2 Fondements scientifiques

Cette partie résume le lien entre la mécanique hamiltonienne, le milieu continu et SPH. Ces liens sont rigoureusement démontrés dans le livre de Violeau [Vio12] et ne seront qu'énoncés ici. En effet, l'objectif de cette section est de comprendre les bases scientifiques à l'origine de la méthode SPH.

3.1.2.1 La mécanique Hamiltonienne

Les lois de la mécanique classique, introduites par Newton en 1687 sont reformulées par Lagrange en 1788. La méthode de Lagrange relève de la mécanique analytique. Les équations de Lagrange sont basées sur le principe de moindre action, introduit par Mautpertuis en 1744. Ce principe, pour en donner un exemple mécanique, stipule qu'un corps prend la direction qui lui permet de dépenser le moins ou d'acquérir le plus d'énergie (ici mécanique) dans l'immédiat, c'est à dire lorsque l'on étudie un pas de temps dont la durée tend vers 0. Pour un système à N degrés de liberté décrit par N coordonnées généralisées q_i et lorsque le système est conservatif (c'est à dire que la variation de l'énergie mécanique est nulle ou que les forces non-conservatives ne travaillent pas), on peut écrire ces équations de Lagrange :

$$\forall i, \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = 0 \quad (3.1)$$

Avec L une fonction réelle que l'on nomme le lagrangien. L permettant nécessairement de relier l'état du système conservatif à un instant t à un instant $t + dt$ et est fonction des variables q_i et \dot{q}_i . Pour un système conservatif, L est constant dans le temps.

On appelle les moments conjugués ou encore impulsions généralisées, p_i :

$$p_i = \frac{\partial L}{\partial \dot{q}_i} \quad (3.2)$$

A partir de ces notations on peut écrire les équations d'Hamilton, qui servent à décrire un système :

$$\frac{\partial H}{\partial t} = - \frac{\partial L}{\partial t}$$

$$\forall i, \frac{\partial H}{\partial p_i} = \dot{q}_i \quad (3.3)$$

$$\frac{\partial H}{\partial q_i} = -p_i \quad (3.4)$$

Avec H le Hamiltonien. On peut montrer que H est égal à l'énergie totale du système¹ et donc si H varie dans le temps, le système n'est pas conservatif. Les notations de la mécanique Hamiltonienne sont particulièrement utiles pour simplifier les équations servant à décrire des systèmes complexes et permettent de montrer facilement si le système est conservatif ou non.

1. Ceci est vrai dans le cas où la transformation est scléronome et que les énergies potentielles ne dépendent pas des vitesses généralisées.

3.1.2.2 Le fluide du milieu continu

Pour modéliser le milieu continu, il faut comprendre qu'aujourd'hui aucune solution complète et précise n'existe. Seuls des modèles de solution existent, puisqu'un véritable milieu continu peut sembler être calme et avoir des molécules, parmi celles qui le composent, qui se déplacent à très grande vitesse. Un modèle statistique précis pourrait être utilisé pour modéliser le fluide, notamment grâce aux lois de la physique statique². Mais pour SPH, un modèle macroscopique plus intuitif est utilisé. Pour cela, le fluide est modélisé par des entités spatiales, suffisamment grandes par rapport à la distance intermoléculaire pour permettre de donner une moyenne statistique de toutes les quantités physiques utiles à notre calcul (vitesse, masse, ...). Ces entités sont appelées particules. Généralement la taille choisie pour les particules est de l'ordre de $10^{-4} m$, cette taille peut être choisie plus grande. Dans certains cas particuliers, micro-fluide ou gaz raréfié, cette valeur est à adapter. Cette distance doit être choisie plus grande que le libre parcours moyen d'un élément microscopique qui compose la particule pour conserver la validité du modèle. La physique statistique a démontré rigoureusement que lorsque les particules contiennent un nombre suffisamment grand d'éléments microscopiques, la vitesse moyenne varie peu [LL80]. Néanmoins, il est évident qu'il faut conserver une taille suffisamment petite pour voir apparaître l'effet que l'on veut observer. Soit \vec{r} la position du centre de masse de chaque particule de masse m composant le fluide. En considérant un champ de vitesse $\vec{u}(\vec{r})$, localisé aux points \vec{r} , les propriétés d'un domaine de fluide continu, e.g. sa masse M , son énergie cinétique E_k , sa quantité de mouvement \vec{P} et son moment cinétique \vec{M} peuvent alors être calculées ainsi :

$$\begin{aligned}
 M &= \int_{\Omega} \rho d\Omega \\
 E_k &= \int_{\Omega} \rho e_k d\Omega \\
 \vec{P} &= \int_{\Omega} \rho \vec{u} d\Omega \\
 \vec{M} &= \int_{\Omega} \rho \vec{r} \wedge \vec{u} d\Omega
 \end{aligned} \tag{3.5}$$

Avec Ω le domaine de fluide de volume V considéré, $d\Omega$ une partie infinitésimale de Ω , e_k l'énergie cinétique massique de $d\Omega$, et ρ la densité du fluide. A partir de ces grandeurs, on peut expliquer la notion de flux, de tenseur de contraintes ou encore bâtir les équations du milieu continu (Navier Stokes). A cela il faudrait ajouter la théorie du chaos pour bien décrire la turbulence dans le milieu fluide. Ce sont ces outils qui permettent la résolution d'écoulement avec la méthode SPH.

3.1.2.3 Principe de la méthode SPH

Ce qui rend la méthode SPH particulièrement attractive pour la simulation d'environnement marin, c'est la possibilité de calculer l'écoulement dynamique d'un fluide sans maillage, à partir d'un ensemble de particules en mouvement avec l'écoulement. Dans son article de 2005, Monaghan [Mon05] paraphrase Von Neumann [VN44] :

2. i.e. la statistique de Maxwell-Boltzmann, la loi de distribution des vitesses de Maxwell, la statistique de Bose-Einstein

La méthode utilisant des particules n'est pas seulement une approximation des équations du fluide dans le continuum, mais elle permet une résolution rigoureuse des équations pour un système de particules qui se rapproche du système moléculaire sous-jacent, et est plus fondamentale que celle des équations du continuum.

Le milieu continu est recréé à partir de deux interpolations successives, une fonction "kernel" d'interpolation du milieu continu, puis une discrétisation en particules de ce milieu représentant un volume fini de fluide. Lorsque les équations du mouvement sont dérivées selon le principe SPH à partir de la mécanique Lagrangienne ou Hamiltonienne, les quantités de mouvement linéaire et angulaire sont conservées. En dehors de l'influence de phénomènes dissipatifs, l'énergie l'est également. De plus, les particules étant des points matériels dotées d'une masse donnée et invariante (en tout cas dans les méthodes SPH standard), la masse est aussi conservée. La possibilité de concevoir des schémas numériques parfaitement conservatifs est un atout majeur de la méthode SPH, bien que, en pratique, l'emploi de termes de stabilisation dans les équations SPH du mouvement fasse perdre certaines des propriétés de conservation, en particulier, de l'énergie. De récents développements ont été menés pour s'assurer de la consistance³ de cette méthode [OMTdL16], ainsi que de ses propriétés de convergence⁴ [Mou06], rendant cette technique de plus en plus attractive. Le choix de la fonction kernel est particulièrement important pour garantir ses propriétés : consistance, convergence et conservatisme⁵.

3. En analyse numérique, la consistance d'un schéma numérique est une propriété locale de l'algorithme qui en découle. Elle concerne essentiellement la capacité du schéma à représenter une solution régulière satisfaisant localement les équations aux dérivées partielles, ceci lorsque les pas de discrétisation (discrétisation en temps, en espace, etc.) tendent tous vers 0. La consistance est une propriété distincte de la convergence, cette dernière étant de portée globale.

4. La convergence d'un algorithme est la convergence de la suite des itérés générés par l'algorithme. Elle peut être locale si elle n'a lieu que pour certains paramètres initiaux. Sinon elle est globale. La convergence d'un algorithme peut être vérifiée si après de nombreuses itérations, le résultat obtenu ne varie plus. En pratique, on définit un critère d'arrêt admettant qu'en dessous d'une certaine variation du résultat entre deux itérations l'algorithme a convergé.

5. Conservatisme, ici, sous-entend que les propriétés de l'écoulement dans l'espace et dans le temps sont conservées lors de l'intégration temporelle à travers l'algorithme SPH

3.2 Interpolation et discrétisation spatiale SPH

La discrétisation pour le modèle SPH se fait en deux étapes. Tout d'abord, la fonction d'interpolation kernel, qui doit permettre de modéliser fidèlement le milieu continu, est introduite. Puis, le système discret de particules est mis en place. Cette section a également pour objectif d'expliquer le choix des fonctions kernel, plus simplement appelées kernel par la suite, qui sont utilisées dans cette thèse.

3.2.1 La fonction kernel : première étape de discrétisation

3.2.1.1 Interpolation et fonction kernel

L'objectif de la fonction kernel est de permettre de modéliser les champs dans le continuum spatial. A titre d'illustration, prenons un champ scalaire A arbitraire qui subit un produit de convolution avec une distribution de Dirac δ :

$$\begin{aligned} A(\vec{r}, t) &= (A * \delta)(\vec{r}, t) \\ &= \int_{\Omega} A(\vec{r}', t) \delta(\vec{r} - \vec{r}') d^n \vec{r}' \end{aligned} \quad (3.6)$$

Avec n la dimension spatiale du problème (dans cette thèse uniquement 3, mais on retrouvera souvent des problèmes de dimension 2 et parfois des problèmes de dimension 1). Ici Ω décrit tout le milieu continu et \vec{r}' les points différents de \vec{r} appartenant à Ω . L'équation 3.6 provient directement de la définition du Dirac :

$$\forall A, A = A * \delta \quad (3.7)$$

Pour des raisons pratiques, ce Dirac δ est approché par une fonction d'interpolation, le kernel $w(\vec{r} - \vec{r}')$. Cette fonction est une fonction régulière⁶, qui a des valeurs non-nulles sur un support compact de taille h_t ou infini. Ce kernel peut être, par exemple, une fonction Gaussienne avec un support infini, ou encore une fonction de type Gaussienne avec un support fini. Ainsi, ceci constitue la première étape de discrétisation :

$$A(\vec{r}, t) \approx \int_{\Omega_{\vec{r}}} A(\vec{r}', t) w(\vec{r} - \vec{r}', \iota) d^n \vec{r}' \quad (3.8)$$

Avec $\Omega_{\vec{r}}$ le domaine fluide centré autour de \vec{r} et un paramètre ι qui définit généralement la longueur de lissage. Ceci est expliqué dans la prochaine section, qui détaille la seconde étape de discrétisation. Une discussion sur le choix du paramètre ι est décrite dans [BRP05]. La taille du support h_t , lorsqu'il est fini, est de l'ordre de grandeur du paramètre ι . Dans ce cas, le support du kernel est invariant par rotation et prend donc la forme d'une sphère de rayon h_t . Cette première étape de discrétisation est représentée en dimension 1 sur la figure 3.2. La fonction kernel est alors choisie pour satisfaire différentes conditions, en notant Ω_0 le domaine fluide copie de $\Omega_{\vec{r}}$ translaté sur l'origine :

$$\int_{\Omega_0} w(\vec{r}_{dif}, \iota) d^n \vec{r}_{dif} = 1 \quad (3.9)$$

6. Une fonction régulière est une fonction définie et continue, sans point de singularité, sur son domaine de définition.

et

$$\int_{\Omega_0} w(\overrightarrow{r_{dif}}, \iota) \overrightarrow{r_{dif}} d^n \overrightarrow{r_{dif}} = 0 \quad (3.10)$$

Avec $r_{dif} = \overrightarrow{r} - \overrightarrow{r^j}$. L'équation 3.9 dit que la moyenne du kernel doit être égale à 1, comme pour un dirac. L'équation 3.10 met une condition sur le kernel pour garantir la symétrie centrale du kernel, c'est à dire :

$$w(\overrightarrow{-r_{dif}}, \iota) = w(\overrightarrow{r_{dif}}, \iota) \quad (3.11)$$

Avec $\overrightarrow{r_{dif}} = \overrightarrow{r} - \overrightarrow{r^j}$. Ce qui amène directement à garantir :

$$\nabla w(\overrightarrow{-r_{dif}}, \iota) = -\nabla w(\overrightarrow{r_{dif}}, \iota) \quad (3.12)$$

avec ∇w le gradient du kernel. Ces conditions stipulent que le moment à l'ordre zéro du kernel doit être égal à un et que le moment au premier ordre du kernel doit valoir zéro. Elles sont nécessaires pour que l'approximation SPH continue soit consistante au premier ordre en ι . En validant les conditions précédemment énoncées on s'assure de (voir Violeau [Vio12]) :

$$\begin{aligned} \int_{\Omega_{\overrightarrow{r}}} A(\overrightarrow{r^j}, t) w(\left| \overrightarrow{r} - \overrightarrow{r^j} \right|, \iota) d^n \overrightarrow{r^j} &= A(\overrightarrow{r}, t) + O(\iota^2) \\ \int_{\Omega_{\overrightarrow{r}}} A(\overrightarrow{r^j}, t) \overrightarrow{\nabla} w(\left| \overrightarrow{r} - \overrightarrow{r^j} \right|, \iota) d^n \overrightarrow{r^j} &= \overrightarrow{grad} A(\overrightarrow{r}, t) + O(\iota^2) \\ \int_{\Omega_{\overrightarrow{r}}} \overrightarrow{A}(\overrightarrow{r^j}, t) w(\left| \overrightarrow{r} - \overrightarrow{r^j} \right|, \iota) d^n \overrightarrow{r^j} &= \overrightarrow{A}(\overrightarrow{r}, t) + O(\iota^2) \\ \int_{\Omega_{\overrightarrow{r}}} \overrightarrow{A}(\overrightarrow{r^j}, t) \cdot \overrightarrow{\nabla} w(\left| \overrightarrow{r} - \overrightarrow{r^j} \right|, \iota) d^n \overrightarrow{r^j} &= div \overrightarrow{A}(\overrightarrow{r}, t) + O(\iota^2) \end{aligned} \quad (3.13)$$

Ceci donne une information de précision potentielle pour nos calculs. Le choix de ι est donc déterminant puisqu'il permet de majorer l'erreur.

3.2.1.2 Choix des fonctions kernel

Pour des questions de temps de calculs et au vu de la taille des simulations qu'il faut effectuer (simulations qui comporteront au minimum une dizaine de milliers d'éléments et jusqu'à des millions d'éléments), l'utilisation de kernel à support infini est écartée. D'après Violeau [Vio12], les fonctions kernel qui répondent aux différents critères précédemment énoncés (Équation 3.9, Équation 3.10) sont définies comme ceci :

$$w(\overrightarrow{r_{dif}}, \iota) = \frac{\alpha_W}{\iota^n} f(q^*) \quad (3.14)$$



Figure 3.2 – Valeur approchée d'un dirac en dimension 1 avec la fonction kernel.

Où f est une fonction réelle définie positive, suffisamment régulière pour être au moins deux fois dérivable et α_W un coefficient adimensionnel qui dépend de la fonction kernel et de la dimension n de l'espace. q^* se calcule comme suit :

$$q^* = \frac{|\overrightarrow{r_{dif}}|}{\iota} \quad (3.15)$$

Pour déterminer des fonctions kernel adimensionnelles le changement de variable suivant est utilisé :

$$q = \frac{\iota}{h_t} q^* = \frac{|\overrightarrow{r_{dif}}|}{h_t} \quad (3.16)$$

On obtient alors la fonction kernel :

$$w(\overrightarrow{r_{dif}}, h_t) = \frac{\alpha_W}{h_t^n} f(q) \quad (3.17)$$

α_W est calculée pour s'assurer de la condition de normalisation (Équation 3.9), que l'on peut réécrire ainsi :

$$\alpha_W \int_{\Omega_{\vec{q}}} f(q) d^m \vec{q} = 1 \quad (3.18)$$

Avec q la norme de \vec{q} et $\Omega_{\vec{q}}$ le domaine défini par toutes les valeurs possibles de \vec{q} . Ainsi, f étant positive α_W est nécessairement positif. Pour la dimension 3, les petits éléments de volumes $d^3 \vec{q} = q^2 dq d^2 \omega$ avec $d^2 \omega$ l'angle solide infinitésimal de la sphère. La condition de l'équation 3.18 donne :

$$\alpha_W = \left[S \int_0^{R_f} f(q) q^2 dq \right]^{-1} \quad (3.19)$$

Avec $R_f = \frac{h_t}{\iota}$ le rayon de la sphère et la limite du support de la fonction kernel et $S = 4\pi$ la surface de la sphère de rayon unitaire.

Parmi les fonctions à support fini populaires qui répondent à ce problème, les fonctions B-spline, généralisation des courbes de Bézier, de degrés M sont les plus courantes pour la méthode SPH. Ces fonctions, réelles définies positives, sont définies par :

$$f_M(q) = \frac{1}{M!} \sum_{k=0}^{M+1} (-1)^k \binom{M+1}{k} \left(\max \left\{ \begin{array}{l} 0 \\ q + \frac{M+1}{2} - k \end{array} \right\} \right)^M \quad (3.20)$$

Les coefficients entre parenthèses $\binom{M+1}{k}$ dénotent un coefficient binomial.

Il a été démontré par de Boor [dB01] que cette fonction par morceaux f_M est la plus régulière parmi les fonctions polynomiales pour un support fini. C'est à partir des équations 3.19 et 3.20 que le kernel pour l'algorithme SPH utilisé dans cette thèse est défini.

3.2.2 Fonctions utilisées dans cette thèse

Dans ce manuscrit deux fonctions Kernel ont été choisies pour la simulation du fluide : une fonction B-spline cubique (de degré $M = 3$, Équation 3.20) et une fonction B-spline quintique (de degré $M = 5$, Équation 3.20).

3.2.2.1 B-spline cubique

La fonction B-spline cubique $f_{W,3}$ est l'une des fonctions les plus utilisées pour la simulation SPH. C'est une fonction très simple mais plutôt imprécise (Violeau [Vio12]), puisque seulement trois fois dérivable. Cette fonction s'obtient en suivant les calculs de l'annexe I :

$$\alpha_{W,3} = \frac{16}{\pi h_t^3}$$

$$f_{W,3}(q) = \begin{cases} \frac{1}{2} - 3q^2(1-q) & 0 \leq q \leq \frac{1}{2} \\ (1-q)^3 & \frac{1}{2} \leq q \leq 1 \\ 0 & q \geq 1 \end{cases} \quad (3.21)$$

La figure 3.3 représente la fonction $f_W\left(\frac{\vec{r}_{dif}}{h_t}\right)$ du B-spline cubique (Équation 3.21) et sa dérivée, qui représente le gradient. Le signe de $\frac{\vec{r}_{dif}}{h_t}$ est pris en compte pour montrer la continuité à 0 de la fonction Kernel et du gradient. Le gradient SPH est orienté selon le vecteur unitaire $\vec{e}_{r' \rightarrow r}$ dans la direction de $r' \rightarrow r$, avec r' la particule voisine et r la particule que l'on considère.

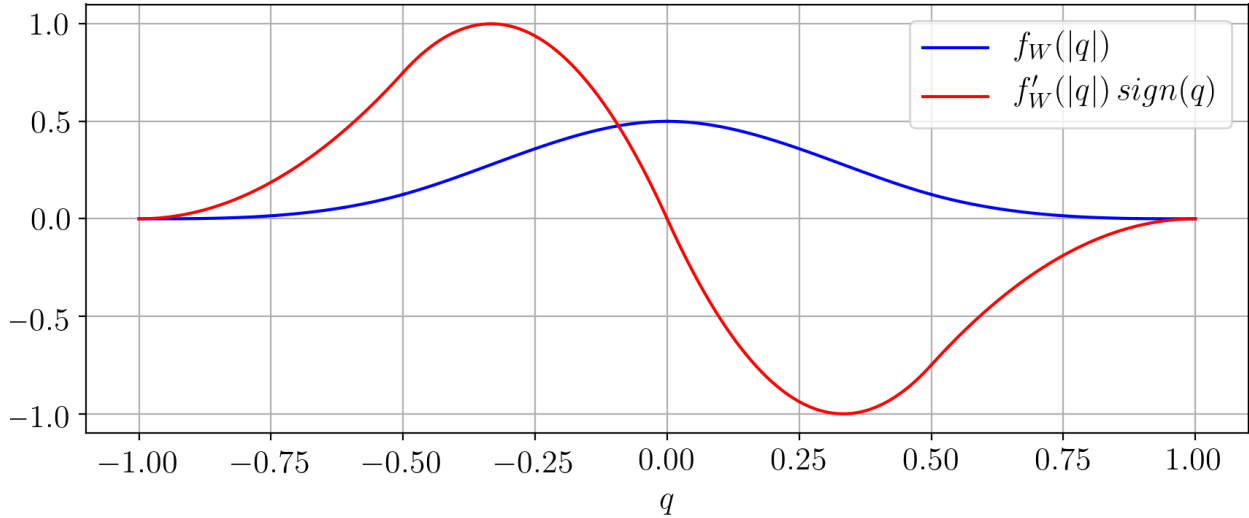


Figure 3.3 – Représentation des fonctions $f_W(|q|)$ et $f'_W(|q|) \text{sign}(q)$ pour le B-spline cubique.

3.2.2.2 B-spline quintique

La fonction B-spline quintique doit permettre une interpolation plus précise que la B-spline cubique pour la simulation SPH. Elle permet notamment d'étendre la portée de la fonction kernel, comparée à la fonction cubique. Ainsi, l'effet d'un plus grand nombre de particules est pris en compte dans les calculs de champ. Cette limitation sur l'étendue de la portée du kernel est due au signe de la transformée de Fourier du kernel. Les limitations de portée du kernel liés au degré de la fonction utilisée sont expliquées dans Dehnen [DA12]. Pour les kernels B-splines, au delà d'une certaine taille de rayon, le signe de la transformée de Fourier change. Une instabilité de formation de paires ("pairing instability" en anglais) survient, avec le même type d'effets que l'instabilité "tensile". Cette instabilité tensile est discutée dans l'annexe F et est liée à la prise en compte de valeurs négatives dans les calculs de pression et notamment pour le gradient de pression.

Dans l'article de Dehnen [DA12], les fonctions de Wendland [Wen95] sont également présentées et réputées permettre l'utilisation d'un nombre illimité de particules sans causer d'instabilité. D'après Dehnen, ces fonctions sont particulièrement intéressantes lorsque l'on souhaite utiliser un kernel avec un plus grand nombre de particules. Dans le cas où il y a un faible nombre de particules, et par extension un rayon du kernel court, ces fonctions sont moins précises que les B-Spline. Puisque nos calculs doivent être réalisés dans un temps minimal, ces fonctions ne sont pas utilisées. L'utilisation de ces fonctions pourrait être une piste de recherche future.

Cette fonction s'obtient, comme pour la cubique, en suivant les calculs de l'annexe I :

$$\alpha_{W,5} = \frac{2187}{40\pi h_t^3}$$

$$f_{W,5}(q) = \begin{cases} 10 \left(\frac{11}{405} - q^2 \left(\frac{2}{9} - q^2(1-q) \right) \right) & 0 \leq q \leq \frac{1}{3} \\ (1-q)^5 - 6 \left(\frac{2}{3} - q \right)^5 & \frac{1}{3} \leq q \leq \frac{2}{3} \\ (1-q)^5 & \frac{2}{3} \leq q \leq 1 \\ 0 & q \geq 1 \end{cases} \quad (3.22)$$

La figure 3.4 représente, de la même manière que pour le B-spline cubique, la fonction $f_W \left(\frac{\vec{r}_{dif}}{h_t} \right)$ du B-spline quintique (Équation 3.22) et sa dérivée, qui représente le gradient.

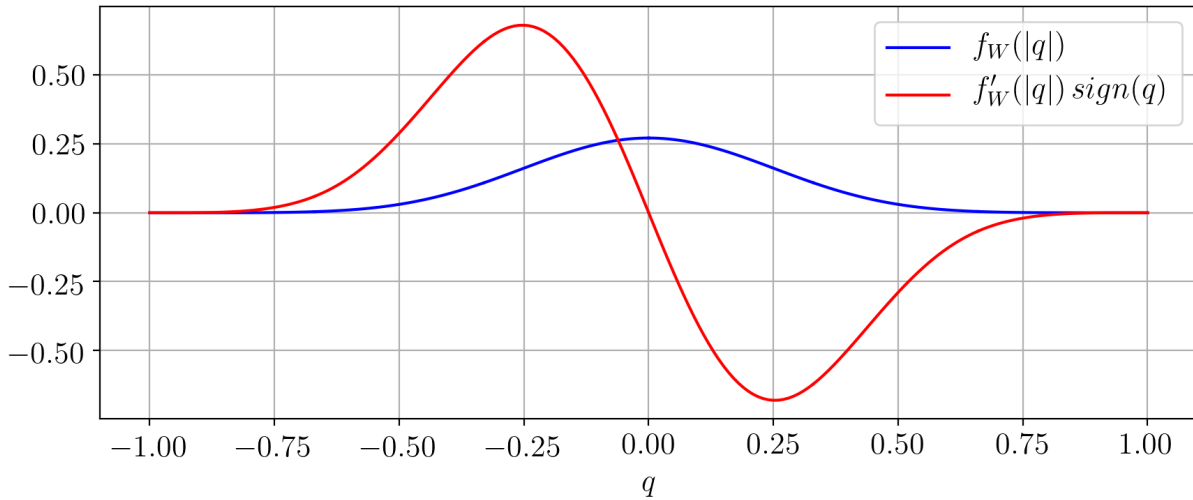


Figure 3.4 – Représentation des fonctions $f_W(|q|)$ et $f'_W(|q|) \text{sign}(q)$ pour le B-spline quintique.

3.2.3 Discrétisation spatiale SPH : seconde étape de discrétisation

À l'origine, la méthode SPH a été créée pour modéliser les milieux continus avec un ensemble de particules, comme déjà décrit dans le chapitre 3.1.2. Du point de vue de la dynamique, les particules se comportent comme des masses ponctuelles, pourtant ce sont bien des entités macroscopiques, ayant donc des propriétés thermodynamiques (pression, etc.) et cinématiques (vitesse, etc.) évoluant dans le temps. On considère en particulier \vec{r}_a comme

étant la position du centre de masse du groupe de molécules qui composent la particule a . Cette particule a donc une masse fixe m_a et une distance inter-particulaire moyenne δ_r (voir figure 3.5). Son volume V_a et sa densité ρ_a sont liés par la relation :

$$V_a = \frac{m_a}{\rho_a} \quad (3.23)$$

Ainsi, la masse d'une particule est constante et définie par :

$$m_a = m = \rho_0 \delta_r^3 \quad (3.24)$$

Avec ρ_0 la densité au repos du fluide. δ_r correspond au diamètre des particules lorsqu'elles sont au repos. Si la masse des particules est fixe, leur volume peut varier dans l'espace et dans le temps, mais toujours autour de $V_0 = \delta_r^3$ lorsque l'on considère un écoulement faiblement compressible ou quasi-incompressible. Le volume d'une particule dépend en fait des autres particules proches.

Le fait de mettre à jour la position d'une masse de fluide avec sa vitesse a pour avantage, par rapport aux méthodes eulériennes, de ne pas requérir de calcul de termes d'advection⁷. Ceci est d'autant plus intéressant, puisque ces termes sont non-linéaires et leur calcul est complexe.

La seconde étape de discrétisation est basée sur l'approximation du continu précédente (Équation 3.8). L'intégrale continue est approchée par une somme discrète. Pour ce faire, on peut noter que toute intégrale d'une fonction arbitraire C peut être approximée par une somme de Riemann comme suit :

$$\int_{\Omega_{\vec{r}}} C(\vec{r}', t) d^n \vec{r}' \approx \sum_b C(\vec{r}'_b, t) V_b \quad (3.25)$$

Ici les points de sommations \vec{r}'_b sont les positions des particules b appartenant à l'ensemble des particules à l'intérieur du domaine $\Omega_{\vec{r}}$, ceci est représenté sur la figure 3.5. V_b est le volume respectif de chacune de ces particules dans la dimension de $d^n \vec{r}'$. En partant de l'approximation faite par le kernel d'une valeur d'un champ scalaire en un point (Équation 3.8) et en appliquant le principe de la somme de Riemann 3.25 la discrétisation SPH est obtenue :

$$\begin{aligned} A(\vec{r}, t) &\approx \int_{\Omega_{\vec{r}}} A(\vec{r}', t) w(\vec{r} - \vec{r}', h_t) d^n \vec{r}' \\ &\approx \sum_b A(\vec{r}'_b, t) W_{ab} V_b \end{aligned} \quad (3.26)$$

Avec $W_{ab} = w(\vec{r}_a - \vec{r}'_b, h_t)$ le kernel. Ceci permet d'obtenir une interpolation discrète du champ A .

Pour simuler l'interaction entre le fluide et le solide, des particules fictives, notées s , sont générées à partir de la forme de la surface du solide (Figure 3.6). Ces particules sont toujours placées à l'extérieur du volume de fluide. Soit à l'intérieur du volume solide si on parle d'un corps volumique, soit de l'autre côté de la surface si on parle uniquement d'une frontière bidimensionnelle. Un nombre suffisant de couches est créé pour que le domaine du kernel

7. L'advection est le transport de quantités (pression, masse, etc.) d'un élément donné par le mouvement (et donc la vitesse) du milieu environnant.

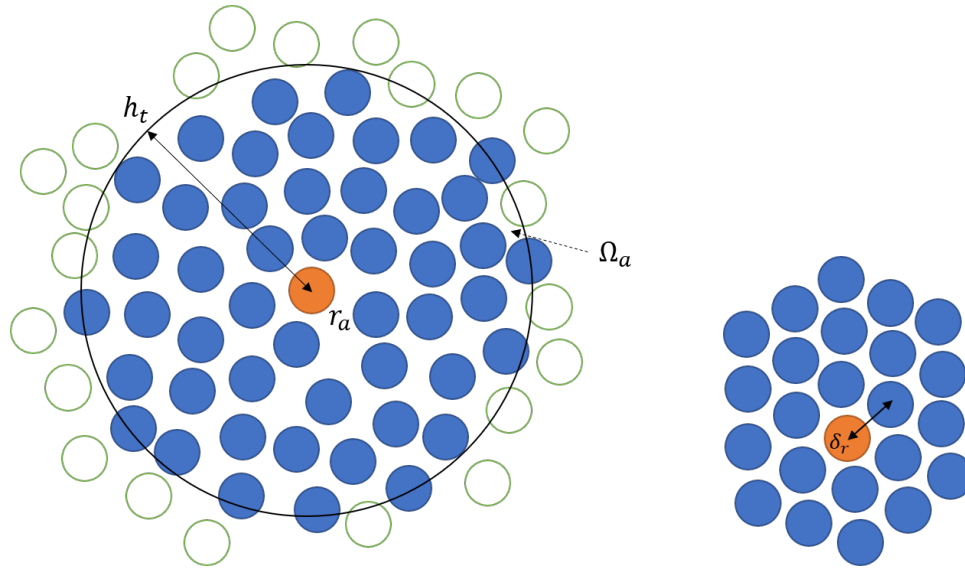


Figure 3.5 – À gauche, approximation discrète d’un champ scalaire par un groupe de particules à l’intérieur du rayon d’interaction h_t de la fonction kernel. Particule considérée en orange. Particules à l’intérieur du domaine Ω_a en bleu. Particules hors de Ω_a en blanc cerclé de vert. À droite, un champ de particules au repos qui permet d’illustrer la longueur δ_r .

d’une particule fluide puisse être rempli si cette particule se trouve proche de la surface du solide. Pour ce faire, le nombre de couches de particules limites fictives est choisi, tel que :

$$n_{layer} \geq h_t/\delta_r = \alpha_h \quad (3.27)$$

Avec α_h le rapport du rayon d’interaction. Dans le cas du B-Spline cubique, par rapport au problème d’instabilité évoqué dans la section 3.2.2.2, la valeur maximale $\alpha_h = 2$ est retenue. En passant au B-Spline quintique, α_h peut être augmenté de 2 jusqu’à au moins 2.5, qui est la valeur que nous retenons. Ainsi, le nombre moyen de particules se trouvant à l’intérieur de la portée de la fonction kernel passe de 32.5 ($\alpha_h = 2$) à 64.4 ($\alpha_h = 2.5$).

L’équation 3.28 montre l’interpolation spatiale d’une grandeur thermodynamique ou cinématique A (densité, pression, etc.) avec SPH.

$$A_a = m \left(\frac{A_a}{\rho_a} W_{aa} + \sum_{b \in \mathcal{F}_a \cup \mathcal{S}_a} \frac{A_b}{\rho_b} W_{ab} \right) \quad (3.28)$$

Avec \mathcal{F}_a et \mathcal{S}_a respectivement l’ensemble des particules liquides et solides à l’intérieur du support du noyau centré sur la particule a . Ici, le volume d’une particule est δ_r^3 , car dans le processus d’initialisation, les particules sont générées selon un maillage régulier, leur masse devant rester constante.

Calculer \mathcal{S}_a et \mathcal{F}_a peut être très coûteux. La méthode développée dans Ihmsen [IABT10] est utilisée ici. Elle fait appel au tri par index et à l’ordonnement des données de particules selon une courbe de remplissage spatial. Ceci améliore les performances en optimisant l’utilisation de la mémoire cache et en évitant des conflits d’accès concurrents aux données pendant une exécution parallèle. En particulier, la densité de la particule a peut être évaluée simplement

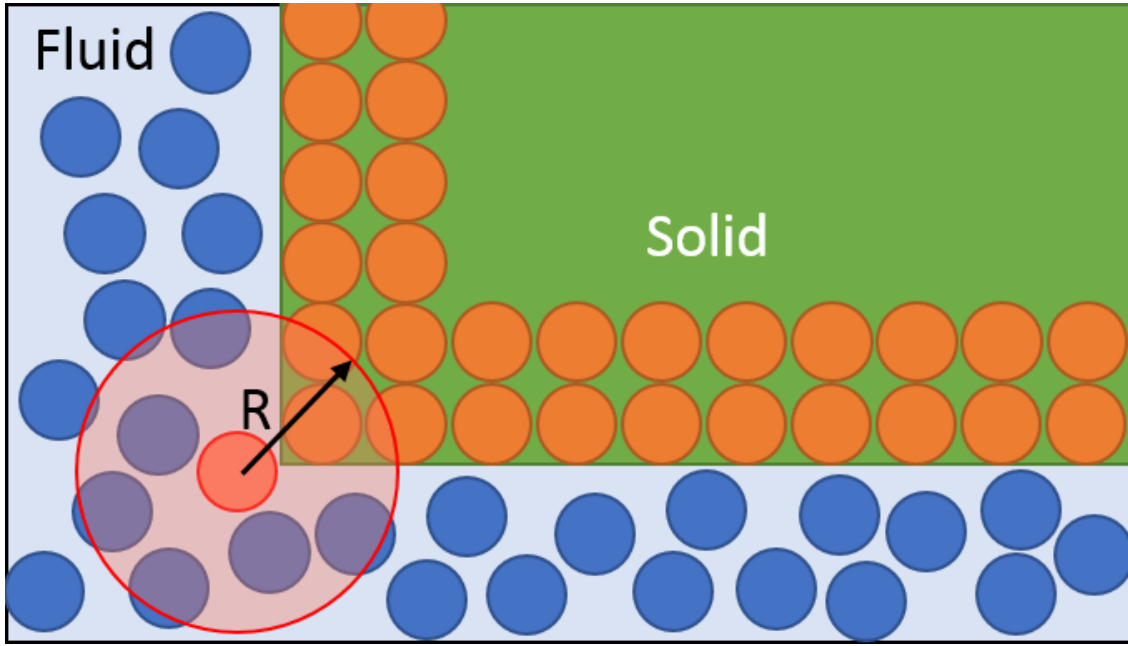


Figure 3.6 – Particules solides générées (orange) à l’intérieur de la forme solide (vert) avec des particules fluides autour (bleu) et le rayon du kernel pour une particule fluide particulière (rouge).

comme :

$$\rho_a = m \left(W_{aa} + \sum_{b \in \mathcal{F}_a \cup \mathcal{S}_a} W_{ab} \right) \quad (3.29)$$

De la même manière, l’approximation SPH est utilisée pour calculer la valeur du gradient $\vec{\nabla} A$, divergence $\vec{\nabla} \cdot \vec{A}$ et laplacien $\vec{\nabla}^2 \vec{A}$. (3.30, 3.31, 3.32), voir Violeau [Vio12] (chapitres 5.2 et 6.2).

$$\vec{\nabla} A_a = m \rho_a \sum_{b \in \mathcal{F}_a \cup \mathcal{S}_a} \left(\frac{A_a}{\rho_a^2} + \frac{A_b}{\rho_b^2} \right) \vec{\nabla} W_{ab} \quad (3.30)$$

$$\vec{\nabla} \cdot \vec{A}_a = -\frac{m}{\rho_a} \sum_{b \in \mathcal{F}_a \cup \mathcal{S}_a} (\vec{A}_a - \vec{A}_b) \cdot \vec{\nabla} W_{ab} \quad (3.31)$$

$$\vec{\nabla}^2 \vec{A}_a = 10m \sum_{b \in \mathcal{F}_a \cup \mathcal{S}_a} \frac{(\vec{A}_a - \vec{A}_b) \cdot (\vec{r}_a - \vec{r}_b)}{\rho_b |\vec{r}_a - \vec{r}_b|^2} \vec{\nabla} W_{ab} \quad (3.32)$$

3.2.4 Conclusion

Dans cette section, la méthode SPH a été présentée. Cette méthode permet de discrétiser le milieu continu et consiste en la succession d’une interpolation du continu et d’une discrétisation spatiale sous forme de particules. Les méthodes de résolution des équations du fluide sont présentées dans la prochaine section.

3.3 Modèles d'écoulement

La première partie de cette section est une analyse des modélisations existantes des écoulements avec SPH qui permettra de choisir deux modèles : un modèle d'écoulement légèrement compressible et un modèle quasi-incompressible. La seconde partie de cette section décrit deux algorithmes SPH selon chacun des modèles choisis. Le premier est un algorithme de la classe de méthodes weakly-compressible SPH (WCSPH) et le second est un algorithme de la classe incompressible SPH (ISPH). Pour ces deux algorithmes, les équations intervenant dans le calcul de la dynamique du fluide sont décrites.

3.3.1 Méthodes de résolution des équations du mouvement d'un fluide

La méthode SPH peut être divisée en trois sous-classes en fonction de la modélisation de l'écoulement, tel que présenté sur la figure 3.7 :

1. **SPH pour les écoulements compressibles** Bien que ce nom ne soit pas courant, ces méthodes pourraient s'appeler Compressible SPH. Pour ces méthodes on utilise une équation d'état et la taille du support d'interactions entre particules est variable. La méthode originale, que l'on nommera dans cette thèse SPH standard, est une méthode permettant de simuler des écoulements fortement compressibles⁸. C'est la méthode introduite par Gingold et Monaghan [GM77] qui est décrite avec plus de recul dans les articles de Monaghan [Mon92] puis [Mon05]. Au départ, l'idée était d'utiliser des fonctions statistiques pour faire des calculs approchés et la méthode ne conservait pas la quantité de mouvement, ni le moment cinétique (voir sous-section 3.1.2.1). La méthode a ensuite été développée pour les conserver. La méthode SPH standard ne comprend que l'utilisation des particules et de la fonction kernel. Elle n'englobe aucune loi physique. Cette dernière est au libre choix de l'utilisateur. Actuellement, cette méthode n'est utilisée telle quelle que pour des cas très particuliers, comme par exemple, pour simuler les lois de la mécanique du solide (voir Chen [CBJ99]).

Aujourd'hui, il existe des versions plus complexes comme la méthode Corrected SPH, qui est la plus courante pour ce type de problème. Guilcher [GCBLT13] montre l'utilisation de ce type de méthode pour simuler le comportement d'une vague qui se brise. Parmi ces méthodes on peut classer Corrected Smoothed Particle Method (CSPM) [CBC99], Discontinuous SPH (DSPH) [LL03] ou Finite Particle Method (FPM) [LXL05].

2. **SPH pour les écoulements faiblement compressibles** Les méthodes simulant un fluide faiblement compressible, dites Weakly Compressible SPH (WCSPH), sont très courantes. Le premier papier traitant de cette méthode et ayant ouvert l'utilisation de SPH à d'autres domaines d'applications est celui de Monaghan [Mon94]. Pour ces méthodes on utilise une équation d'état et la taille de support d'interactions entre particules est fixe. On peut distinguer deux types parmi les méthodes WCSPH selon la méthode de stabilisation employée. Celle appelée couramment Godunov-SPH, qui utilise un solveur de Riemann. Un exemple de l'utilisation de la méthode Godunov-SPH

8. Ces écoulements ont lieu, soit pour les gaz qui sont compressibles par nature, soit lorsque le nombre de Mach de l'écoulement est grand ou encore lorsque de grandes variations thermiques ont lieu dans le fluide. Par exemple, la rentrée en atmosphère d'une navette spatiale ou le comportement du gaz dans une étoile sont des écoulements compressibles.

se trouve dans Sigalotti [SLT09]. Et δ SPH [MAC⁺11], qui utilise un solveur pour contrôler la diffusion de la vitesse et de la densité. Cette dernière méthode a connu de récentes améliorations et a donné δ^+ SPH [SCMZ17] [SCM⁺18]. Ces méthodes pourraient être très intéressantes pour les simulations d'écoulement autour de véhicules marins ; seulement, elles ont été développées au cours de l'avancement de cette thèse et n'ont donc pas été prises en compte.

3. **SPH pour les écoulements quasiment incompressibles** La méthode simulant un fluide incompressible, dites Incompressible SPH (ISPH). Cette méthode est basée sur la projection des pressions et deux différentes approches existent pour cela : l'une se concentrant sur le critère de divergence de la vitesse [CR99] et l'autre se concentrant sur la variation de la densité [SL03]. Pour ces méthodes, la taille de support d'interactions entre particules est fixe. ISPH est actuellement préférée dans les applications d'animation 3D, avec des compromis pour obtenir un calcul en temps réel :
 - une méthode nommée Implicit Incompressible SPH (IISPH) [ICS⁺14].
 - une méthode hybride particule-maillage qui découle de IISPH, et d'une autre méthode utilisant des particules nommée Fluid-Implicit-Particle (FLIP) [BR86] [BKR88] qui s'appelle IISPH-FLIP [CIPT14].
 - une méthode qui limite la divergence du champ de vitesse, ce qui permet d'augmenter la stabilité du calcul, divergence free SPH, parfois notée DFSPH [BK15] [BK17]

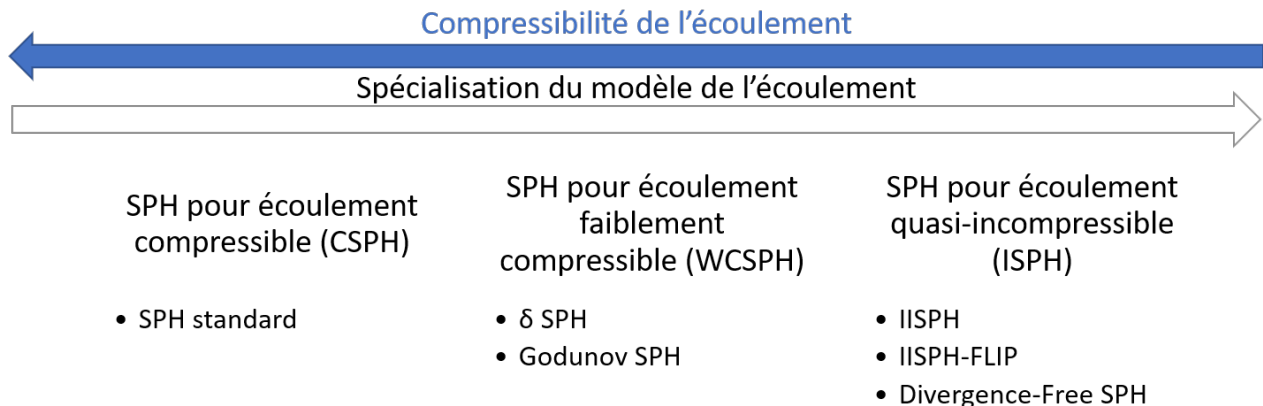


Figure 3.7 – Évolution des méthodes SPH.

Sur la figure 3.7, on voit que les méthodes d'écoulement compressible peuvent également simuler les écoulements faiblement compressibles et quasi-incompressibles. Les méthodes faiblement compressibles peuvent simuler aussi les écoulements quasi-incompressibles. Moins l'écoulement est compressible et plus il est spécialisé à certains écoulements. Ceci permet d'utiliser des techniques particulières qui peuvent mener à une amélioration du résultat ou une diminution du temps de calcul. Puisque l'objet de l'étude est un liquide, l'eau, l'écoulement à modéliser est quasi-incompressible. Il est donc préférable de se focaliser sur les méthodes WCSPH et ISPH. De plus, ces deux méthodes ont déjà démontré leurs capacités à résoudre des problèmes avec une interaction fluide-solide. Pour WCSPH, de très nombreux articles traitent du réalisme physique de l'interaction fluide solide. La plupart des algorithmes de traitement de l'interaction fluide solide étant tout d'abord développé pour ce solveur puisqu'il est le plus

simple à implémenter. On peut donc citer les articles d'Adami [AHA12], qui a développé une méthode d'extrapolation de la pression à la surface du solide, et de Band [BGPT18], qui a développé une méthode d'extrapolation utilisant les moindres carrés glissants qui permet une évolution plus précise du gradient de pression qu'avec la méthode d'Adami. Pour ISPH, des validation de simulation d'impact de vagues ont été menées par Khayyer [KGS09]. L'article de Lee [LMX⁺08] propose une comparaison des méthodes ISPH et WCSPH et démontre que la précision des champs de vitesse et pression lorsqu'il y a un écoulement autour d'un solide est meilleure avec le modèle ISPH, ce qui implique une meilleure évaluation de la force sur le solide. Ceci sera vérifié dans nos essais, notamment dans la section 3.5 et dans la section 4.1.

3.3.2 Weakly Compressible SPH (WCSPH)

L'algorithme Weakly Compressible SPH se base sur l'hypothèse d'un écoulement à faible nombre de Mach⁹. Dans ce cas, l'eau peut être considérée comme légèrement compressible. Considérons un fluide newtonien et négligeons la tension superficielle. Les équations de continuité (3.33) et du bilan de quantité de mouvement (3.34) de Navier-Stokes sont utilisées :

$$\frac{d\rho}{dt} + \rho \vec{\nabla} \cdot \vec{v} = 0 \quad (3.33)$$

$$\frac{d\vec{v}}{dt} = -\frac{1}{\rho} \vec{\nabla} p + \nu \vec{\nabla}^2 \vec{v} + \vec{g} \quad (3.34)$$

Avec p la pression, choisie relative à la pression atmosphérique¹⁰, \vec{v} le champ de vitesse, t le temps, ν la viscosité cinématique, \vec{g} le champ de gravité, ρ la densité et d/dt la dérivée lagrangienne. En considérant un écoulement légèrement compressible (nombre de Mach $Ma < 0.1$), où la pression du fluide est considérée liée à sa variation de densité, l'équation d'état de Murnaghan-Tait est applicable :

$$p = \frac{\rho_0 c_{s,num}^2}{\gamma} \left[\left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right] \quad (3.35)$$

Avec l'indice polytropique γ , qui généralement, a une valeur de $\gamma = 7$ pour l'eau. Une vitesse numérique du son $c_{s,num}$ est utilisée, car l'utilisation de la vitesse physique du son entraînerait des pas de temps excessivement courts, ce qui allongerait considérablement le temps de calcul (§ 3.3.4).

$$c_{s,num} \geq 10 \max \left(V_{max}, \sqrt{|g| H} \right) \quad (3.36)$$

Avec V_{max} et H respectivement la vitesse maximale prévue et la hauteur de fluide. Pour les deux méthodes (ISPH et WCSPH) présentées ici, la densité est interpolée en utilisant l'équation 3.29. Pour éviter l'apparition de l'instabilité d'attraction (en anglais : tensile instability) [Mon00] et appliquer implicitement la pression atmosphérique à la surface libre (ici, l'interface avec l'air), les pressions négatives sont évitées. L'annexe F explique ce problème particulier à SPH. Ce problème a été corrigé dans δ -SPH par Sun [SCM⁺18]. Néanmoins il n'a pas été possible de le résoudre pour ISPH au courant de cette thèse. L'annexe montre le point limitant actuel.

9. Le nombre de Mach est un nombre sans dimension, noté Ma , qui exprime le rapport de la vitesse locale d'un fluide à la vitesse du son dans ce même fluide. Un faible nombre de Mach signifie $Ma \ll 1$

10. C'est à dire qu'à la surface du fluide la pression est égale à 0

La première étape de l'algorithme WCSPH est appelée, dans cette thèse, étape de diffusion. A partir de l'équation 3.34, l'accélération $\vec{v}_a^{\mathcal{P}}$ due aux forces extérieures hormis celle de la pression est calculée :

$$\begin{aligned}\vec{v}_a^{\mathcal{P}} &= \nu \vec{\nabla}^2 \vec{v}_a + \vec{g} \\ &= \frac{10m\mu_{num}}{\rho_a} \sum_{b \in \mathcal{F}_a \cup \mathcal{S}_a} \frac{(\vec{v}_a - \vec{v}_b) \cdot (\vec{r}_a - \vec{r}_b)}{\rho_b |\vec{r}_a - \vec{r}_b|^2} \vec{\nabla} W_{ab} + \vec{g}\end{aligned}\quad (3.37)$$

Avec la vitesse \vec{v}_a une vitesse initiale, obtenue par intégration temporelle de l'accélération précédente, μ_{num} une viscosité dynamique numérique, supérieure à la viscosité dynamique réelle, qui aide à stabiliser la simulation. Pour les particules solides ($b \in \mathcal{S}_a$), \vec{v}_s est la vitesse de glissement, dont le calcul est expliqué dans la section 3.4.

L'ajout de cette viscosité artificielle μ_{num} est expliqué dans Monaghan [Mon05] : celle-ci est faite pour permettre la simulation de choc ou pour stabiliser l'algorithme numérique. Elle est choisie en fonction de la viscosité naturelle du fluide mais est également liée à la viscosité de volume¹¹ μ_V et, par extension, à la seconde viscosité¹² μ' . Dans l'application des équations de Navier Stokes qui a été faite ici, implicitement, l'hypothèse de Stokes a été utilisée. Cette hypothèse propose $\mu_V = 0$. Elle est vraie pour les gaz monoatomiques, et elle est une bonne approximation pour les liquides. Dans certains schéma numérique, une viscosité due au turbulence¹³ vient simuler les effets se produisant à des échelles plus fines que la résolution des particules choisies et permet de maintenir la continuité du champ de contrainte visqueux. Ici cette viscosité n'est pas calculée. L'ajout d'une viscosité artificielle permet en partie de compenser cette absence.

La seconde étape consiste à calculer l'accélération totale des particules fluides en prenant en compte les forces de pressions. Pour calculer cette accélération, il faut tout d'abord calculer le gradient de pression, pour ce faire les notations suivantes sont introduites :

$$\vec{\Sigma}_a^F = m \sum_{b \in \mathcal{F}_a} \vec{\nabla} W_{ab} \quad \text{and} \quad \vec{\Sigma}_a^S = m \sum_{s \in \mathcal{S}_a} \vec{\nabla} W_{as} \quad (3.38)$$

$$\vec{\Sigma}_a = \vec{\Sigma}_a^F + \vec{\Sigma}_a^S \quad (3.39)$$

$$\Theta_i = p_i / \rho_i^2 \quad i = a, b, s \quad (3.40)$$

$$\begin{aligned}\vec{\Gamma}_a &= m \sum_{b \in \mathcal{F}_a} \Theta_b \vec{\nabla} W_{ab} + m \sum_{s \in \mathcal{S}_a} \Theta_s \vec{\nabla} W_{as} \\ &= m \sum_{b \in \mathcal{F}_a \cup \mathcal{S}_a} \Theta_b \vec{\nabla} W_{ab}\end{aligned}\quad (3.41)$$

Enfin, l'accélération totale des particules fluides est calculée en partant de la formule du

11. La viscosité de volume μ_V d'un fluide newtonien isotrope est définie par : $\mu_V = \mu' + \frac{2}{3} \mu$ où μ est la viscosité dynamique et μ' la seconde viscosité. Son unité est le pascal · seconde (Pa · s).

12. La seconde viscosité μ' d'un fluide newtonien est le second paramètre scalaire qui caractérise complètement le caractère linéaire de la relation contrainte-vitesse de déformation pour un fluide isotrope : $\sigma = 2\mu\dot{\epsilon} + \mu' \text{tr}(\dot{\epsilon}) I$, avec ϵ la déformation. L'unité de μ' est le pascal · seconde (Pa · s).

13. en anglais : « eddy viscosity »

gradient en SPH 3.30 :

$$\begin{aligned}
\dot{\vec{v}}_a &= \dot{\vec{v}}_a^{\mathcal{P}} - \frac{1}{\rho_a} \vec{\nabla} p_a \\
&= \dot{\vec{v}}_a^{\mathcal{P}} - m \sum_{b \in \mathcal{F}_a \cup \mathcal{S}_a} \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) \vec{\nabla} W_{ab} \\
&= \dot{\vec{v}}_a^{\mathcal{P}} - m \sum_{b \in \mathcal{F}_a \cup \mathcal{S}_a} (\Theta_a + \Theta_b) \vec{\nabla} W_{ab} \\
&= \dot{\vec{v}}_a^{\mathcal{P}} - (\Theta_a \vec{\Sigma}_a + \vec{\Gamma}_a)
\end{aligned} \tag{3.42}$$

D'après les équations 3.39, 3.40, 3.41 et 3.42, on pourrait dire que $\vec{\Gamma}$ est l'accélération provenant de la somme de la contribution des particules voisines et $\Theta \vec{\Sigma}$ est la contribution de la particule elle-même à l'accélération de ses voisins. Les positions et les vitesses des particules sont finalement mises à jour avec le schéma semi-implicite d'Euler, qui est un schéma symplectique (voir annexe C).

Avec ce schéma, il est particulièrement important de noter l'utilisation de coefficients numériques pour la vitesse du son (Équation 3.35), qui est chargée de propager les ondes de pression dans le fluide, et la viscosité (Équation 3.37), qui puisqu'elle est choisie supérieure à celle de l'eau va générer des forces de frottement supplémentaires. L'algorithme WCSPH complet se trouve dans l'annexe E.

3.3.3 Incompressible SPH (ISPH)

Le modèle ISPH est basé sur l'incompressibilité de l'eau, l'objectif étant de forcer cette incompressibilité pour le calcul de l'écoulement. Pour mettre à jour les vitesses et les positions des particules et obtenir le mouvement du fluide dans le temps, les équations de Navier-Stokes incompressibles sont résolues dans leur formulation lagrangienne. Pour rappel, l'équation du bilan de quantité de mouvement 3.34 et la condition d'incompressibilité 3.33 sont :

$$\begin{aligned}
\frac{d\rho}{dt} + \rho \vec{\nabla} \cdot \vec{v} &= 0 \\
\frac{d\vec{v}}{dt} &= -\frac{1}{\rho} \vec{\nabla} p + \nu \vec{\nabla}^2 \vec{v} + \vec{g}
\end{aligned}$$

Dans le cas d'un système incompressible, l'équation de continuité devient :

$$\frac{d\rho}{dt} = \vec{\nabla} \cdot (\vec{v}) = 0 \tag{3.43}$$

Comme pour WCSPH, la première étape est une étape de diffusion. Pour mettre à jour le fluide de l'état n à l'état $n+1$, une première valeur d'accélération hormis les forces de pression est calculée :

$$\begin{aligned}
\dot{\vec{v}}_a^{\mathcal{P}} &= \nu \vec{\nabla}^2 \vec{v}_a + \vec{g} \\
&= \frac{10m\mu}{\rho_a} \sum_{b \in \mathcal{F}_a \cup \mathcal{S}_a} \frac{(\vec{v}_a - \vec{v}_b) \cdot (\vec{r}_a - \vec{r}_b)}{\rho_b |\vec{r}_a - \vec{r}_b|^2} \vec{\nabla} W_{ab} + \vec{g}
\end{aligned} \tag{3.44}$$

avec μ la viscosité dynamique réelle du fluide. Encore une fois pour les particules solides, \vec{v}_s est la vitesse de glissement, dont le calcul est expliqué dans la section 3.4. Analogiquement à WCSPH, les pressions négatives sont interdites et la densité est interpolée en utilisant l'équation 3.29. A partir des équations 3.43 et 3.31, une nouvelle valeur de la densité des particules est calculée :

$$\rho_a^P = \rho_a + m \Delta t \left(\sum_{b \in \mathcal{F}_a} (\vec{v}_a^P - \vec{v}_b^P) \cdot \vec{\nabla} W_{ab} + \sum_{s \in \mathcal{S}_a} (\vec{v}_a^P - \vec{v}_s^{fr}) \cdot \vec{\nabla} W_{as} \right) \quad (3.45)$$

Avec \vec{v}_s^{fr} la vitesse de la particule à la frontière du solide. Cette vitesse est calculée en projetant la vitesse en translation et rotation du solide en son centre de gravité au centre de la particule considérée.

$$\vec{v}_s^{fr} = \vec{v}_{Solide} + \overrightarrow{SG} \wedge \vec{\Omega} \quad (3.46)$$

Avec \vec{v}_{Solide} la vitesse du solide au centre de gravité G , \overrightarrow{SG} la distance qui sépare la particule s du centre de gravité G du solide et $\vec{\Omega}$ le vecteur instantané de rotation du solide.

On discrétise l'équation 3.34, en prenant en compte que les termes de viscosité et de gravité sont déjà représentés dans \vec{v}_a^P :

$$\vec{v}_a^{n+1} = -\frac{\Delta t}{\rho_a} \vec{\nabla} p_a + \vec{v}_a^P \quad (3.47)$$

En discrétisant l'équation 3.43 à l'état $n + 1$, en utilisant 3.47 et considérant que, à l'état $n + 1$ l'incompressibilité du fluide est atteinte, donc $\rho_a^{n+1} = \rho_0$:

$$\rho_a \vec{\nabla} \cdot \left(\frac{1}{\rho_a} \vec{\nabla} p_a \right) = \frac{\rho_0 - \rho_a^P}{\Delta t^2} \quad (3.48)$$

Ceci est une équation de Poisson avec les pressions pour inconnues que l'on résout de manière implicite via un système linéaire.

L'objectif de l'algorithme est de trouver les pressions qui font respecter l'incompressibilité. Celle-ci est modélisée comme la minimisation de $\Delta \rho_a = \rho_0 - \rho_a^*$, avec ρ_a^* la densité intermédiaire atteinte par les particules a après l'étape de diffusion puis après chaque itération. Ici, une projection de pression exacte est utilisée sur 3.48 pour une précision accrue, tout d'abord l'opérateur divergence 3.31 est utilisé :

$$\begin{aligned} & -m \sum_{b \in \mathcal{F}_a} \left(\frac{1}{\rho_a} G(p_a) - \frac{1}{\rho_b} G(p_b) \right) \vec{\nabla} W_{ab} \\ & -m \sum_{s \in \mathcal{S}_a} \left(\frac{1}{\rho_a} G(p_a) - \frac{1}{\rho_s} G(p_s) \right) \vec{\nabla} W_{as} = \frac{\rho_0 - \rho_a^*}{\Delta t^2} \end{aligned} \quad (3.49)$$

Ici $G(p_a)$ et $G(p_b)$ représentent le gradient de pression SPH (Équation 3.30) des particules a

et b qu'il reste donc à appliquer :

$$\begin{aligned}
& -m \sum_{b \in F_a} \left(m \sum_{b \in F_a} \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) \vec{\nabla} W_{ab} + m \sum_{s \in S_a} \left(\frac{p_a}{\rho_a^2} + \frac{p_s}{\rho_s^2} \right) \vec{\nabla} W_{as} \right. \\
& \left. - m \sum_{c \in F_b} \left(\frac{p_b}{\rho_b^2} + \frac{p_c}{\rho_c^2} \right) \vec{\nabla} W_{bc} - m \sum_{s' \in S_b} \left(\frac{p_b}{\rho_b^2} + \frac{p_{s'}}{\rho_{s'}^2} \right) \vec{\nabla} W_{bs'} \right) \vec{\nabla} W_{ab} \\
& - m \sum_{s \in S_a} \left(m \sum_{b \in F_a} \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) \vec{\nabla} W_{ab} + m \sum_{s \in S_a} \left(\frac{p_a}{\rho_a^2} + \frac{p_s}{\rho_s^2} \right) \vec{\nabla} W_{as} \right) \vec{\nabla} W_{as} = \frac{\rho_0 - \rho_a^*}{\Delta t^2} \quad (3.50)
\end{aligned}$$

Avec c les voisins des particules de fluide b et s' les voisins appartenant à la frontière solide des particules de la frontière solide s . On réorganise immédiatement avec les notations introduites pour WCSPH (Équation 3.39, Équation 3.38, Équation 3.40, Équation 3.41) :

$$\begin{aligned}
& -m \sum_{b \in F_a} \left(m \sum_{b \in F_a} \left(\frac{p_a}{\rho_a^2} + \Theta_b \right) \vec{\nabla} W_{ab} + m \sum_{s \in S_a} \left(\frac{p_a}{\rho_a^2} + \Theta_s \right) \vec{\nabla} W_{as} \right. \\
& \left. - m \sum_{c \in F_b} (\Theta_b + \Theta_c) \vec{\nabla} W_{bc} - m \sum_{s' \in S_b} (\Theta_b + \Theta_{s'}) \vec{\nabla} W_{bs'} \right) \vec{\nabla} W_{ab} \\
& - m \sum_{s \in S_a} \left(m \sum_{b \in F_a} \left(\frac{p_a}{\rho_a^2} + \Theta_b \right) \vec{\nabla} W_{ab} + m \sum_{s \in S_a} \left(\frac{p_a}{\rho_a^2} + \Theta_s \right) \vec{\nabla} W_{as} \right) \vec{\nabla} W_{as} = \\
& - \left(\frac{1}{\rho_a^2} (\Sigma_a^F)^2 + \frac{2}{\rho_a^2} \Sigma_a^F \Sigma_a^S + \frac{1}{\rho_a^2} (\Sigma_a^S)^2 \right) p_a - m \sum_{b \in F_a} \Theta_b (\Sigma_a - \Sigma_b) \vec{\nabla} W_{ab} \\
& \quad - m \Sigma_a \sum_{s \in S_a} \Theta_s \vec{\nabla} W_{as} + m^2 \sum_{b \in F_a} \vec{\nabla} W_{ab} \sum_{c \in F_b} \Theta_c \vec{\nabla} W_{bc} \\
& \quad + m^2 \sum_{b \in F_a} \vec{\nabla} W_{ab} \sum_{s' \in S_b} \Theta_{s'} \vec{\nabla} W_{bs'} = \frac{\rho_0 - \rho_a^*}{\Delta t^2} \quad (3.51)
\end{aligned}$$

On refactorise selon les pressions (p , Θ) pour simplifier :

$$\begin{aligned}
& - \frac{(\Sigma_a)^2}{\rho_a^2} p_a - m \sum_{b \in F_a} \Theta_b (\Sigma_a - \Sigma_b) \vec{\nabla} W_{ab} - m \Sigma_a \sum_{s \in S_a} \Theta_s \vec{\nabla} W_{as} \\
& + m^2 \sum_{b \in F_a} \vec{\nabla} W_{ab} \sum_{c \in F_b} \Theta_c \vec{\nabla} W_{bc} + m^2 \sum_{b \in F_a} \vec{\nabla} W_{ab} \sum_{s' \in S_b} \Theta_{s'} \vec{\nabla} W_{bs'} = \frac{\rho_0 - \rho_a^*}{\Delta t^2} \quad (3.52)
\end{aligned}$$

En considérant que dans $\sum_{c \in F_b}$, c peut être égal à a , on introduit $\Sigma_a^{F,2}$ pour regrouper tous les termes en p_a , qui constituent les éléments diagonaux du système linéaire, et bien les séparer des autres termes des éléments non-diagonaux (voir ci-après Équation 3.58).

$$\Sigma_a^{F,2} = m^2 \sum_{b \in F_a} \left(\vec{\nabla} W_{ab} \right)^2 \quad (3.53)$$

L'équation 3.52 devient :

$$\underbrace{- \frac{(\vec{\Sigma}_a)^2 + \Sigma_a^{F,2}}{\rho_a^2} p_a}_{D_a} + \underbrace{\Theta_a \Sigma_a^{F,2} + \vec{\Sigma}_a \cdot \vec{\Gamma}_a + m \sum_{b \in \mathcal{F}_a \cup S_a} (\Theta_b \vec{\Sigma}_b + \vec{\Gamma}_b) \cdot \vec{\nabla} W_{ab}}_{\vec{R}_a \cdot \vec{p}} = \frac{\rho_0 - \rho_a^*}{\Delta t^2} \quad (3.54)$$

On pose :

$$D_a = -\frac{(\vec{\Sigma}_a)^2 + \Sigma_a^{F,2}}{\rho_a^2} \quad (3.55)$$

$$\vec{R}_a \cdot \vec{p} = \Theta_a \Sigma_a^{F,2} + \vec{\Sigma}_a \cdot \vec{\Gamma}_a + m \sum_{b \in \mathcal{F}_a \cup \mathcal{S}_a} (\Theta_b \vec{\Sigma}_b + \vec{\Gamma}_b) \cdot \vec{\nabla} W_{ab} \quad (3.56)$$

Soit un ensemble de particules numérotées de 1 à n , et p_1 la pression de la particule 1 et p_n la pression de la particule n . On définit le vecteur \vec{p} contenant les pressions de toutes les particules de l'ensemble :

$$\vec{p} = (p_1 \quad \dots \quad p_a \quad \dots \quad p_b \quad \dots \quad p_n)^T \quad (3.57)$$

Avec ce vecteur p , on peut définir un système linéaire $M\vec{p} = \vec{b}$ à partir de l'équation 3.54 :

$$\underbrace{x_{aa}}_{D_a} p_a + \underbrace{\sum_{b \neq a} x_{ab} p_b}_{\vec{R}_a \cdot \vec{p}} = b_a \quad (3.58)$$

Avec $b_a = \frac{\rho_0 - \rho_a^p}{\Delta t^2}$ et x_{ij} les éléments de la matrice M . Cette matrice M obtenue est une matrice creuse, c'est à dire une matrice contenant beaucoup de zéros, parce que les particules n'interagissent qu'avec les particules les plus proches au travers de la fonction kernel. Conceptuellement, cela veut dire que le système est peu couplé. La méthode de Jacobi relaxée est choisie pour résoudre le système linéaire. Cette méthode présente de nombreux avantages : il n'est pas nécessaire de construire la matrice M et de la stocker en mémoire et les lignes du système peuvent être traitées efficacement en parallèle. Le faible taux de convergence est, a priori, acceptable puisque l'on vise une précision modérément élevée pour l'incompressibilité (de 1% à 0,01% d'écart de densité). Tous les coefficients diagonaux D_a étant strictement négatifs, la convergence est garantie. La pression après une itération devient :

$$p_a = (1 - \omega) p_a + \omega \frac{b_a - \vec{R}_a \cdot \vec{p}}{D_a} \quad (3.59)$$

Avec ω le facteur de relaxation. Chaque fois qu'une nouvelle pression est calculée, elle est remplacée par $p = \max(p, 0)$. La nouvelle densité est :

$$\rho_a^* = \rho_a^p + \Delta t^2 (D_a p_a + \vec{R}_a \cdot \vec{p}) \quad (3.60)$$

Enfin, le degré de forçage de l'incompressibilité est estimé en calculant la moyenne de toutes les particules liquides et en la comparant à un seuil ϵ . Une nouvelle itération est effectuée si :

$$\frac{\text{avg}_{a \in \mathcal{F}}(\max(\rho_a^*, \rho_0)) - \rho_0}{\rho_0} > \epsilon \quad (3.61)$$

Afin de s'assurer de la convergence de l'algorithme, un certain nombre d'itération minimal it_{min} est effectué et la valeur de départ avant la résolution du calcul de pression est diminuée :

$$p_a = \Omega p_a, \quad \Omega < 1 \quad (3.62)$$

Avec Ω le facteur de sous-relaxation pour la première itération.

L'algorithme ISPH complet se trouve dans l'annexe E. Notons que pour cet algorithme, contrairement à WCSPH, aucun coefficient artificiel n'est utilisé à la place d'une valeur physique pour les propriétés du fluide.

3.3.4 Pas de temps

Pour assurer une simulation stable, le pas de temps dépend de l'état de la scène simulée. La technique expliquée dans Sampath [SMA⁺16] est appliquée ici. Il est donné par la valeur minimale de ces trois conditions :

$$\Delta t_1 = \lambda_{CFL} \frac{\delta_r}{\max(c_{s,num}, \max_{a \in \mathcal{F} \cup \mathcal{S}} |\mathbf{v}_a|)} \quad (3.63)$$

$$\Delta t_2 = \lambda_{CFL} \sqrt{\frac{2\delta_r}{\max(|\mathbf{g}|, \max_{a \in \mathcal{F} \cup \mathcal{S}} |\dot{\mathbf{v}}_a|)}} \quad (3.64)$$

$$\Delta t_3 = \lambda_{diff} \frac{\delta_r^2}{\nu} \quad (3.65)$$

Avec λ_{CFL} et λ_{diff} des coefficients constants permettant d'ajuster l'importance de chacune de ces conditions par rapport aux autres. Les première et deuxième conditions (Équation 3.63, Équation 3.64) sont toutes deux liées à la condition Courant-Friedrichs-Lewy (CFL), pour empêcher les particules de bouger trop en une seule itération. Le critère basé sur la vitesse numérique du son (3.63) est spécifique au modèle WCSPH. La troisième condition (Équation 3.65) correspond à un critère limitant une trop grande diffusion pendant un pas de temps. En outre, les solveurs du fluide et des solides ont un pas de temps séparé. Typiquement, le pas de temps du solveur des solides Δt_{Rigid} est constant. Ce pas de temps Δt_{Rigid} est choisi pour qu'il soit toujours plus petit le pas de temps du solveur de fluide Δt . Pour synchroniser les deux solveurs, Δt est réduit au plus proche multiple de Δt_{Rigid} .

3.3.5 Conclusion

Cette section a présenté deux algorithmes SPH : WCSPH et ISPH. Les principales différences entre ces deux algorithmes sont :

- WCSPH est un algorithme totalement explicite, ce qui veut dire qu'entre deux états n et $n + 1$ l'algorithme consiste en une succession d'applications de formules. Contrairement à ce dernier, ISPH est un algorithme semi-implicite, ce qui veut dire qu'une partie de la résolution est résolue de manière implicite. Ici, c'est le calcul du champ de pression qui est résolu implicitement de manière à forcer l'incompressibilité.
- WCSPH nécessite, en pratique, de simuler l'écoulement comme s'il était plus compressible et visqueux qu'il ne l'est physiquement. Des coefficients numériques (vitesse du son numérique et viscosité numérique) sont ajoutés pour augmenter sa stabilité. Dans ISPH, les valeurs physiques peuvent être utilisées. Notons que le forçage de l'incompressibilité n'étant pas parfait, des effets de compression-détente du fluide peuvent être simulés malgré tout, bien que l'amplitude de ces effets soit, à priori, bien moindre qu'avec le modèle WCSPH.
- Le schéma WCSPH est bien plus simple à modifier puisqu'il est explicite. C'est pourquoi il est souvent préféré à ISPH pour le développement de nouvelles techniques à coupler avec le calcul du fluide.
- La résolution semi-implicite avec ISPH permet d'utiliser des pas de temps plus grands que ce qui est permis avec WCSPH (cf section 3.3.4)

La manière dont sont implémentés ces deux algorithmes, qui est déterminante pour le temps de calcul nécessaire, est décrite dans l'annexe J. Les algorithmes décrits dans cette

section résultent des équations appliquées au fluide, mais ne décrivent pas la manière dont les forces sont transmises entre le fluide et le solide. Ceci est détaillé dans la prochaine section.

3.4 Validation de la méthode d'interaction fluide-solide

Cette section explique la manière dont le fluide et les solides interagissent. Pour permettre une interaction, la surface des solides doit-être discrétisée avec des particules. Il existe différentes manières de calculer la pression sur les particules qui représentent la surface du solide. Ces différentes méthodes sont expliquées ici, puis comparées dans la section 3.5.

En plus des solveurs fluides, un solveur de la dynamique des solides est utilisé pour simuler la dynamique des corps rigides. Le principe du solveur rigide est de calculer et de prendre en compte toutes les forces externes appliquées sur les corps rigides. Le solveur dynamique utilisé ici calcule à chaque pas de temps une nouvelle vitesse et une nouvelle position en utilisant le schéma d'intégration de Verlet, détaillé dans l'annexe C, avec un pas de temps Δt_{Rigid} constant.

3.4.1 Force s'appliquant sur les particules frontières

Pour chaque particule fictive solide, la force totale \vec{F}_s peut être calculée ainsi :

$$\vec{F}_s = \vec{F}_s^P + \vec{F}_s^V \quad (3.66)$$

Avec \vec{F}_s^V la force visqueuse locale qui doit être appliquée sur les parois des solides pour respecter la troisième loi de Newton et compenser l'accélération appliquée sur les particules de fluides (Équation 3.37 et Équation 3.44) :

$$\vec{F}_s^V = \frac{10m^2\mu}{\rho_s} \sum_{b \in \mathcal{F}_s} \frac{(\vec{v}_b - \vec{v}_s) \cdot (\vec{r}_b - \vec{r}_s)}{\rho_b |\vec{r}_b - \vec{r}_s|^2} \vec{\nabla} W_{sb} \quad (3.67)$$

Et \vec{F}_s^P la force locale calculée à partir de la pression selon la méthode de calcul de la frontière solide utilisée (réflexion ou extrapolation). Deux méthodes différentes sont utilisées pour calculer la densité et la pression des particules de la surface du solide : une méthode d'extrapolation et une méthode de réflexion de la pression. Une condition de non-glissement du fluide à la surface du solide est appliquée pour les calculs de frottement.

3.4.2 Méthode d'extrapolation de la pression

La méthode d'extrapolation de Shepard¹⁴ a été appliquée par Adami [AHA12] à la technique WCSPH. Cette section explique comment on peut l'appliquer à ISPH également. Les équations 3.68 et 3.69 montrent comment la densité et la pression des particules de la frontière sont calculées.

$$\rho_s = \frac{\sum_{b \in \mathcal{F}_s} \rho_b W_{sb}}{\sum_{b \in \mathcal{F}_s} W_{sb}} \quad (3.68)$$

$$p_s = \frac{\sum_{b \in \mathcal{F}_s} [p_b + \rho_b (\dot{v}_b^P - \dot{v}_s) \cdot (\vec{r}_s - \vec{r}_b)] W_{sb}}{\sum_{b \in \mathcal{F}_s} W_{sb}} \quad (3.69)$$

14. Nommée en hommage à Donald Shepard [She68] [She84]

Chaque particule de la surface du solide possède donc sa propre pression et densité. La densité calculée est constante pendant un pas de temps. La pression l'est également dans le cas de WCSPH, pour ISPH elle est constante pendant une itération du solveur de pression (voir Annexe E).

Dans le cas de l'extrapolation, \vec{F}_s^P La force locale calculée à partir de la pression (Équation 3.69) :

$$\vec{F}_s^P = -m^2 \sum_{s \in S} \sum_{b \in \mathcal{F}_s} (\Theta_s + \Theta_b) \vec{\nabla} W_{sb} \quad (3.70)$$

Ensuite, la force résultante au centre de gravité du solide (CG) est calculée comme suit :

$$\vec{F}^{Fluid \rightarrow Solid} = \sum \vec{F}_s$$

et le moment résultant :

$$\vec{M}_{CM}^{Fluid \rightarrow Solid} = \sum (\vec{r}_s - \vec{r}_{CM}) \wedge \vec{F}_s$$

3.4.3 Méthode de réflexion de la pression

Dans la méthode de réflexion de la pression, qui est introduite par Akinci [Aki14], la contribution des particules à la surface du solide est directement prise en compte dans l'accélération des particules fluides et le calcul de la force appliquée aux corps rigides. Pour appliquer cette méthode, les changements suivants sont nécessaires : Le précalcul $\vec{\Gamma}_a$ de l'équation 3.41 est maintenant $\vec{\Gamma}_a^{Mir}$:

$$\vec{\Gamma}_a^{Mir} = m \left[\sum_{b \in \mathcal{F}_a} \Theta_b \vec{\nabla} W_{ab} + \frac{1}{\rho_a} \sum_{b \in \mathcal{S}_a} (\dot{\vec{v}}_b^P - \dot{\vec{v}}_s) \cdot (\vec{r}_s - \vec{r}_b) \vec{\nabla} W_{as} \right] \quad (3.71)$$

La force de pression locale appliquée sur les corps rigides (De manière similaire à l'équation 3.70) pour cette méthode d'interfaçage est calculée avec :

$$\vec{F}_s^P = -m^2 \sum_{s \in S} \sum_{b \in \mathcal{F}_s} \left(\Theta_b + \frac{(\dot{\vec{v}}_b^P - \dot{\vec{v}}_s) \cdot (\vec{r}_s - \vec{r}_b)}{\rho_b} \right) \vec{\nabla} W_{sb} \quad (3.72)$$

L'accélération totale du solveur WCSPH donnée par l'équation 3.42 devient :

$$\dot{\vec{v}}_a^{Mir} = \dot{\vec{v}}_a^P - (\Theta_a \vec{\Sigma}_a + \vec{\Gamma}_a^{Mir}) \quad (3.73)$$

et la Matrice M (Équation 3.55 & Équation 3.56) de ISPH devient les équations 3.74 et 3.75 :

$$D_a = - \frac{(\vec{\Sigma}_a)^2 + \Sigma_a^{F,2} + \vec{\Sigma}_a \cdot \vec{\Sigma}_a^S}{\rho_a^2} \quad (3.74)$$

$$\vec{R}_a \cdot \vec{p} = \Theta_a \Sigma_a^{F,2} + \vec{\Sigma}_a \cdot \vec{\Gamma}_a^{Mir} + m \sum_{b \in \mathcal{F}_a} (\Theta_b (\vec{\Sigma}_b + \vec{\Sigma}_b^S) + \vec{\Gamma}_b^{Mir}) \cdot \vec{\nabla} W_{ab} \quad (3.75)$$

3.4.4 Condition de non-glissement et de glissement libre

La manière dont la condition de non-glissement est appliquée dépend de la méthode de calcul de pression à la frontière du fluide. Dans le cas de la méthode d'extrapolation de la pression, la condition de non-glissement détermine la vitesse des particules solides à utiliser dans les calculs de forces de frottement visqueux. D'après Bouscasse [BCMA13] et Sun [SMZ15], cette vitesse de glissement est :

$$\vec{v}_s = 2\vec{v}_s^{fr} - \frac{\sum_{b \in \mathcal{F}_s} \vec{v}_b W_{sb}}{\sum_{b \in \mathcal{F}_s} W_{sb}} \quad (3.76)$$

Dans ce cas, chaque particule possède sa propre vitesse de glissement et cette vitesse est constante pendant un pas de temps.

Dans le cas de la méthode de réflexion de la pression, aucune vitesse n'est calculée pour la particule de fluide considérée. La seule vitesse prise en compte lors des calculs est la vitesse de la frontière solide \vec{v}_s^{fr} , ce qui est un autre façon, bien que non équivalente, d'imposer une condition de non-glissement.

La condition de glissement libre, tel que décrit dans Adami [AHA12] et Sun [SMZ15], consiste à omettre l'interaction visqueuse des particule fluide avec les particules fictives adjacentes. Cette condition est utilisée dans cette thèse pour limiter les effets de bords des parois des bassins de simulation. Autrement dit, dans ce cas, $\vec{F}_s^V = \vec{0}$.

3.4.5 Comparaison des méthodes de calculs de la pression

Afin de comparer la précision de l'estimation de la pression résultant des méthodes de modélisation de la frontière fluide-solide, la première série d'expériences de ce chapitre consiste à vérifier la pression hydrostatique de l'eau dans un réservoir.

3.4.5.1 Configuration expérimentale

Pour tous les tests de ce chapitre, la même scène est utilisée. La figure 3.8 est une vue de la scène d'essai, qui consiste en un réservoir rectangulaire contenant des particules de fluide générées, et dans lequel une sphère de diamètre $0.4 m$ coulera. Cette figure montre également les particules frontières générées à la surface du corps rigide.

Le fluide est de l'eau avec les propriétés physiques standard de l'eau à 293 K : la densité au repos est $\rho_0 = 998 kg.m^{-3}$ et la viscosité dynamique est $\mu = 0.000998 kg.m^{-1}.s^{-1}$. Le temps de calcul des essais est sauvegardé, puisque un algorithme rapide est recherché, et les résultats seront présentés dans la section 3.5.3. Le kernel SPH utilisé pour tous les tests présentés dans ce chapitre est le B-spline cubique (§ 3.2.2.1), ce qui permet notamment d'obtenir des résultats plus rapidement qu'avec le B-spline quintique.

3.4.5.2 Description de l'essai

Les particules d'eau se stabilisent pendant 3 secondes de temps simulé. Ainsi, elles s'auto-arrangent et on obtient un état statique ou quasi-statique du fluide. Une colonne de $1,62 m$ d'eau est formée avec environ 13 000 particules. Le paramètre utilisé pour le solveur WCSPH est donné dans le Tableau 3.1. Le choix de la vitesse du son numérique $c_{s,num}$ correspond

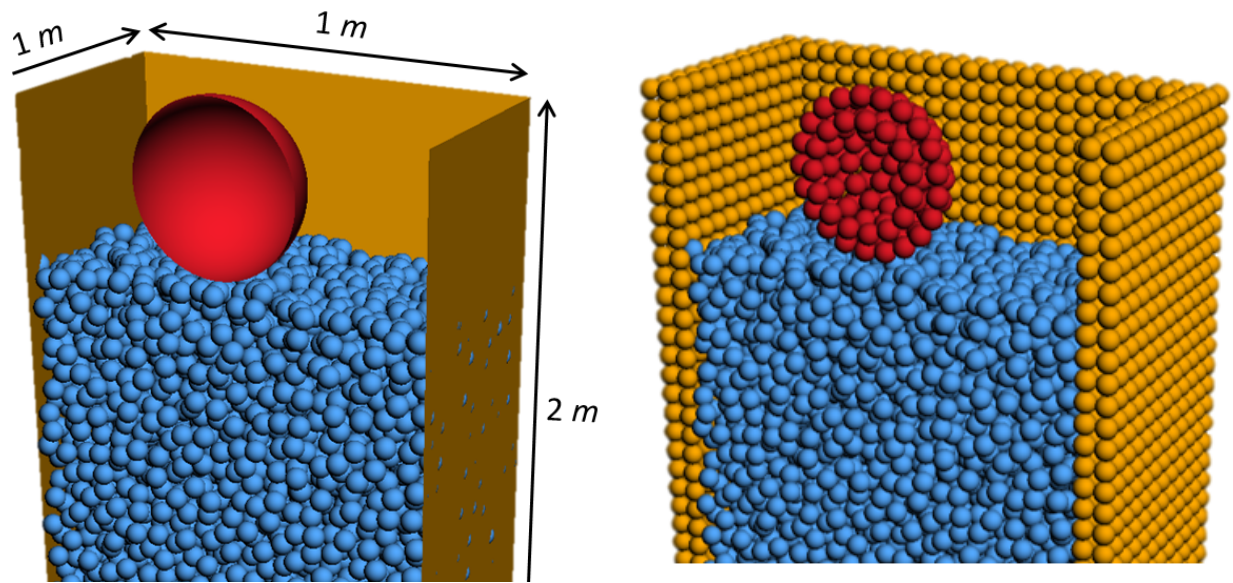


Figure 3.8 – Gauche : La scène à simuler avec les différents solides Droite : La scène à simuler avec les différentes particules frontières.

Le bassin est en jaune, les particules sont en bleus et la sphère est en rouge.

$c_{s,num}$	(Équation 3.36)	80	δ_r	(Équation 3.27)	0.05 m
λ_{CFL}	(Équation 3.63)	0.4	α_h	(Équation 3.27)	2
λ_{diff}	(Équation 3.65)	0.125	μ_{num}	(Équation 3.37)	0.01 kg.m ⁻¹ .s ⁻¹

Tableau 3.1 – Paramètres de l’algorithme WCSPH

environ au double du critère de hauteur et largement au-dessus des critères de vitesse qui se produiront dans cette scène (Équation 3.36). Cette valeur relativement faible permet d’avoir des pas de temps beaucoup plus importants que si l’on utilisait la vitesse physique du son, le critère correspondant dans l’équation 3.63 étant le plus restrictif, tout en maintenant la compressibilité à un niveau modéré. La valeur de δ_r correspond, par rapport aux dimensions du réservoir, à une résolution spatiale très grossière, pour minimiser le temps de calcul.

3.4.5.3 Résultats

La figure 3.9 montre l’ensemble du champ de pression obtenu en utilisant la technique de réflexion de la pression en comparaison avec celui obtenu par la technique d’extrapolation. La courbe noire représente la solution analytique de l’évolution de la pression statique en fonction de la profondeur.

Cette figure montre que l’erreur faite avec la technique de réflexion de pression est excessive. Elle est par endroit supérieure à 100%. Pour rappel, l’enjeu est de calculer une force de poussée d’Archimède à partir de ce champ de pression qui sera intégré sur la surface. Pour voir où se situent ces erreurs, deux couches de champ de pression à deux profondeurs différentes sont montrées sur les figures 3.10 et 3.11. Les variations de pression sont illustrées par des couleurs. Comme l’eau aurait dû se stabiliser, la valeur théorique est la pression hydrostatique :

$$p_{th}(z) = \rho g z \quad (3.77)$$

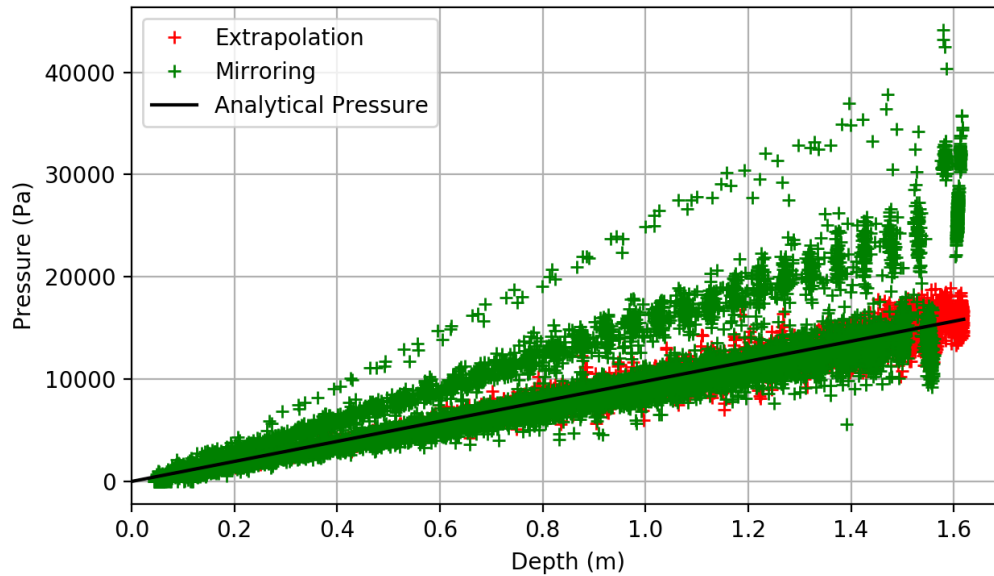


Figure 3.9 – Evolution de la pression en fonction de la profondeur en utilisant les techniques d'extrapolation et de réflexion de la pression. (WCSPH)

Pour la technique de réflexion, la pression au niveau de la frontière solide est trop élevée, en particulier dans les coins, et elle l'est encore plus au fond du bassin en raison de l'influence fictive des particules de la frontière du fond du réservoir. Cela se produit également à la surface des corps dynamiques et/ou immergés et a un effet néfaste sur l'interaction fluide-solide de la force de pression. Des résultats similaires seraient observés avec le solveur ISPH, puisque ceux-ci sont obtenus à l'état statique, où seule joue la modélisation de la frontière, qui est indépendante de la méthode de résolution de l'écoulement.

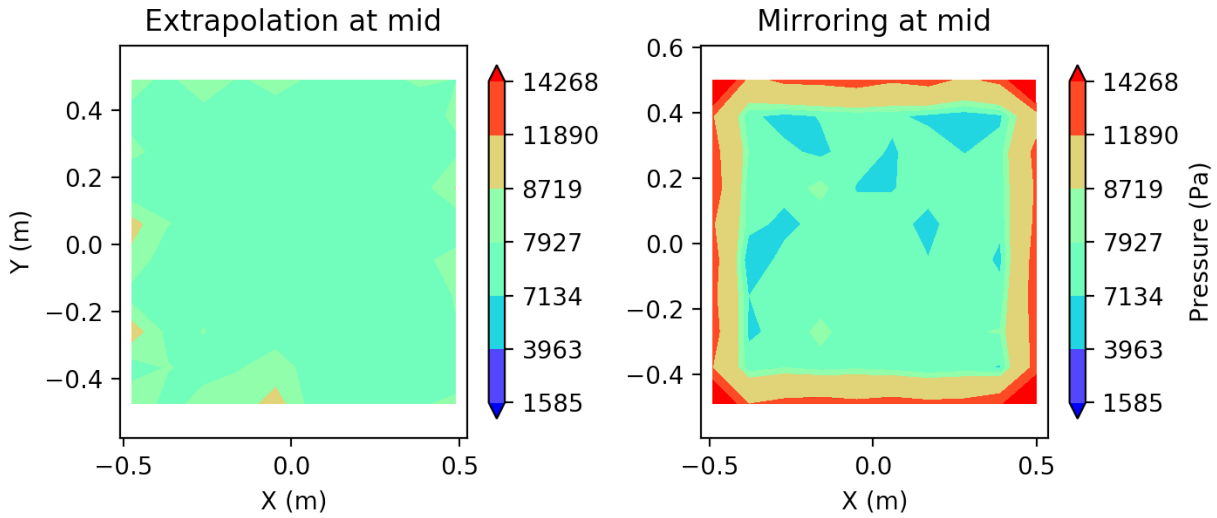


Figure 3.10 – Vue du champ de pression à la profondeur médiane du bassin avec les techniques d’extrapolation et de réflexion de pression. $z = 0.81\text{ m}$ and $p_{th} = 7931\text{ Pa}$. (WCSPH)

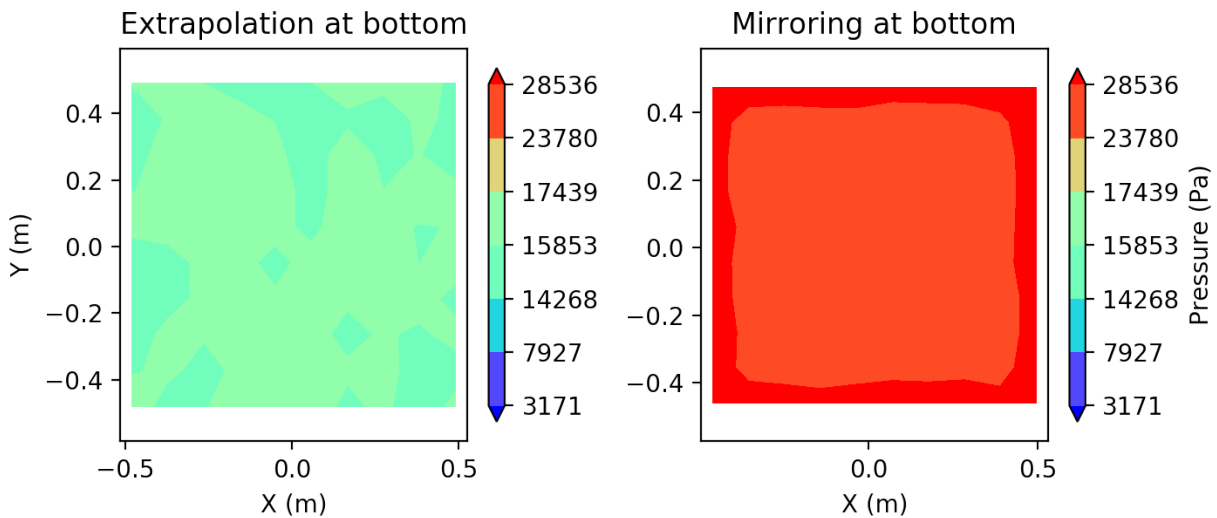


Figure 3.11 – Vue du champ de pression au fond du bassin avec les techniques d’extrapolation et de réflexion de pression. $z = 1.62\text{ m}$ and $p_{th} = 15862\text{ Pa}$. (WCSPH)

3.4.6 Conclusion

La précision de la méthode d'extrapolation est bien meilleure que la précision de la méthode de réflexion. De plus, si l'on regarde les temps de calcul à la section 3.5.3, la technique d'extrapolation requiert certes 10 à 15 % de temps supplémentaire mais ceci n'est pas significatif comparée au gain en précision. Dans la suite de cette thèse, seule la méthode d'extrapolation est utilisée pour la modélisation de l'interaction fluide-solide.

3.5 Validation de l’algorithme ISPH

Pour comparer les algorithmes WCSPH et ISPH et ainsi choisir le plus approprié pour l’évaluation des performances hydrodynamiques d’objets sous-marins, différents comportements doivent être observables (§ 1.6.1) :

1. Un champ de pression hydrostatique doit s’établir dans un bassin.
2. Le champ de pression à la surface d’un objet immergé doit correspondre à l’évolution de la pression hydrostatique lorsque l’objet est au repos. Un objet moins dense que l’eau doit être capable de flotter. Un objet plus dense que l’eau doit couler.
3. La simulation du champ de pression doit être stable/robuste au changement de direction de l’objet immergé pour permettre des simulations dynamiques précises avec les véhicules sous-marins.

Cette section compare les résultats obtenus dans différents cas tests avec les algorithmes ISPH et WCSPH. La configuration des essais est la même que celle décrite dans la section 3.4.5.

3.5.1 Comparaison des algorithmes ISPH et WCSPH dans un cas statique

3.5.1.1 Description de l’essai

Dans une deuxième série d’essais, une simulation avec un corps rigide immergé est réalisée. Les champs de pression sont comparés avec l’utilisation de la méthode d’extrapolation entre le solveur WCSPH, qui est explicite, et le solveur ISPH, qui est semi-implicite. Des particules d’eau sont générées et se stabilisent librement pendant 3 secondes de temps simulé. Ensuite, la sphère est lâchée par-dessus la surface et est arrêtée après avoir atteint une certaine profondeur. L’eau est ensuite stabilisée pendant 3 secondes supplémentaires. Les mêmes paramètres que pour la première série d’essais sont utilisés pour WCSPH (voir [Tableau 3.1](#)), et les paramètres utilisés pour ISPH se trouvent dans le [Tableau 3.2](#). La plupart d’entre eux sont communs aux deux.

λ_{CFL}	(Équation 3.63)	0.4	ϵ	(Équation 3.61)	0.0005
λ_{diff}	(Équation 3.65)	0.125	ω	(Équation 3.59)	0.5
δ_r	(Équation 3.27)	0.05m	Ω	(Équation 3.62)	0.6
α_h	(Équation 3.27)	2	it_{min}	(Annexe E)	5

Tableau 3.2 – Paramètres de l’algorithme ISPH

Le facteur d’incompressibilité ϵ est choisi assez petit pour donner un calcul de pression précis mais pas trop petit pour limiter le temps de calcul requis. Une étude plus poussée pourrait définir avec précision comment la pression p peut être calculée avec cette valeur de ϵ . Les facteurs de relaxation ω et Ω sont choisis pour avoir un algorithme résilient ¹⁵.

15. Résilient signifie en général une propriété de résistance au choc. Ici, ce terme signifie que l’algorithme est capable de donner un résultat à la simulation même lorsque l’on génère de fortes discontinuités dans la scène simulée

3.5.1.2 Résultats

La pression extrapolée sur une sphère à l'intérieur du champ de pression de l'eau avec ISPH et WCSPH est affichée sur la figure 3.12. Les points verts et rouges représentent la pression de chaque particule de la simulation. Les rouges sont les particules du fluide. Les verts sont les particules de la sphère. Le champ de pression présenté est celui à l'instant final de la simulation après stabilisation. La courbe noire représente la pression statique analytique d'après l'équation 3.77.

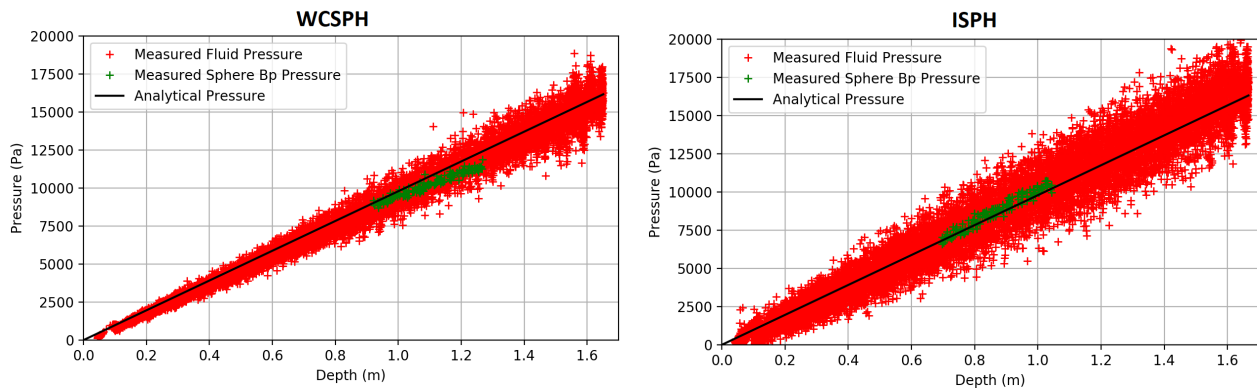


Figure 3.12 – Pression extrapolée sur la sphère avec WCSPH (*gauche*) et ISPH (*droite*).

Les points verts se trouvent à l'intérieur des points rouges et sont proches de la pression analytique à chaque profondeur. Pour ISPH, l'erreur absolue moyenne pour la pression sur les particules de l'enveloppe de la sphère est de 3,56% alors que pour la pression du fluide elle est de 14,8%. Pour WCSPH, respectivement 4,69% et 8,13%. Ainsi, les deux algorithmes ont une précision correcte pour l'évaluation de la pression sur les frontières solides (< 5% d'erreur) et WCSPH semble avoir une meilleure précision d'évaluation de la pression du fluide. La poussée d'Archimède obtenue après intégration numérique est réaliste. Toutefois, il est probable que pour des solides ayant des densités très proches de celle du fluide, le comportement du solide s'éloigne de la réalité à cause d'erreurs numériques.

Pour aller plus loin, il faut vérifier si on a les mêmes niveaux de précision dans les cas dynamiques et si cette précision n'est pas atteinte au détriment du temps de calcul.

3.5.2 Comparaison des algorithmes ISPH et WCSPH dans un cas dynamique

3.5.2.1 Description de l'essai

Une étude supplémentaire est réalisée pour vérifier la stabilité des algorithmes au comportement dynamique. Cette fois-ci, la sphère est larguée au-dessus de la surface et arrêtée à différentes profondeurs. Chaque fois que la sphère s'arrête, la pression est stabilisée. L'évolution dans le temps du coefficient de pression C_p et de l'offset Off_p , qui proviennent de l'équation 3.78, est analysée.

$$p(z) = C_p z + Off_p \quad (3.78)$$

Au regard de l'équation 3.77, C_p devrait être ρg et Off_p devrait être 0.

La figure 3.13 montre leur évolution dans le temps pour le solveur WCSPH. La ligne noire affiche la valeur théorique des coefficients.

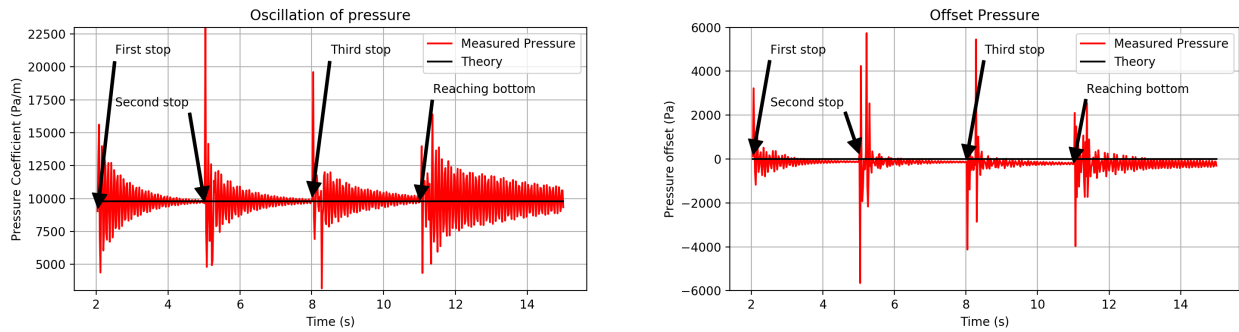


Figure 3.13 – Évolution des coefficients de l'équation 3.78 au cours du temps dans le cas de WCSPH.

Chaque fois que la sphère est arrêtée, un choc se produit dans le fluide, ce qui perturbe le champ de pression. L'offset et le coefficient de pression se stabilisent et convergent dans le temps vers la valeur théorique.

Pour comparaison, la figure 3.14 affiche l'évolution de C_p et Off_p dans le temps pour le solveur ISPH. L'évolution des coefficients avec ce solveur est très différente du cas du solveur

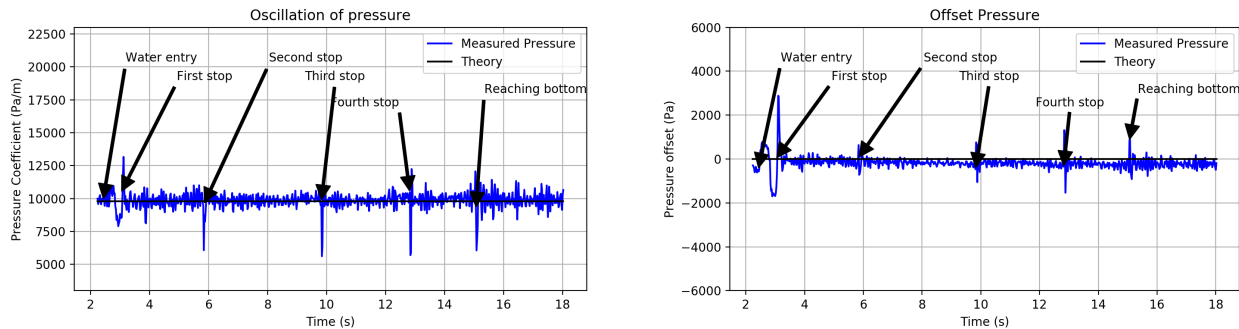


Figure 3.14 – Évolution des coefficients de l'équation 3.78 au cours du temps dans le cas de ISPH.

WCSPH. Comme dans ISPH l'incompressibilité est plus strictement appliquée et les pressions sont implicitement résolues, les variations de pression sont directement dissipées.

Ainsi, ISPH semble être plus fiable pour faire face aux chocs pouvant survenir au cours d'une simulation. Cependant, le solveur WCSPH converge vers un état statique plus précis. La précision du solveur ISPH est directement liée au facteur d'incompressibilité ϵ ; réduire ϵ , bien sûr, augmente le nombre d'itérations avant la convergence de l'algorithme. Un examen plus approfondi pourrait déterminer quel paramètre ϵ utilisé pour obtenir une précision de pression suffisante pour notre objectif.

3.5.3 Tableau récapitulatif des temps de calcul

Pour évaluer l'efficacité de l'algorithme, on compare les temps de calcul sur deux machines différentes. Les résultats d'un ordinateur de milieu de gamme et d'un ordinateur de gamme supérieure démontrent la capacité de ces algorithmes à être utilisés pour des applications en temps réel, même sur des PC standard. Notre ordinateur moyenne gamme possède un processeur Intel Core i7 6700HQ, 8 threads cadencés à 2,6 GHz, 6 Mo de cache, 16 Go de

RAM et un disque dur à 5400 tr/min. Notre ordinateur haut de gamme possède un processeur Intel Core i7 5820K, 12 threads cadencés à 3,3 GHz, 15 Mo de cache, 32 Go de RAM et un disque dur SSD. Les tableaux 3.3 et 3.4 résume le temps de calcul moyen pour 1 s de temps simulé avec les différents solveurs présentés dans cet article. En plus de l'étude avec 13 000 particules, les mêmes cas tests ont été refaits avec des particules deux fois plus petites, ce qui porte le nombre de particules dans la scène à 104 000 particules.

Ordinateur moyenne gamme					
Solveur	Frontière	Nb Part.	Temps C.	Nb PdT	Temps C. / PdT
WCSPH	Réflexion	13k	167,13	4000,00	0,042
	Extrapolation	13k	184,70	4000,00	0,046
ISPH	Réflexion	13k	17,82	199,00	0,090
	Extrapolation	13k	20,45	196,49	0,104
WCSPH	Extrapolation	104k	1890,87	8000,00	0,236
ISPH	Extrapolation	104k	314,32	402,70	0,781

Tableau 3.3 – Temps de calcul moyen (Temps C.) en secondes, nombre de pas de temps moyen (Nb PdT) et temps de calcul moyen par pas de temps pour 1 s de temps simulé avec les différents solveurs pour notre ordinateur moyenne gamme.

Ordinateur haute gamme					
Solveur	Frontière	Nb Part.	Temps C.	Nb PdT	Temps C. / PdT
WCSPH	Réflexion	13k	121,2	4000,0	0,030
	Extrapolation	13k	108,7	4000,0	0,027
ISPH	Réflexion	13k	10,9	199,0	0,055
	Extrapolation	13k	13,0	196,5	0,066
WCSPH	Extrapolation	104k	1372,1	8000,0	0,17
ISPH	Extrapolation	104k	185,7	402,70	0,46

Tableau 3.4 – Temps de calcul moyen (Temps C.) en secondes, nombre de pas de temps moyen (Nb PdT) et temps de calcul moyen par pas de temps pour 1 s de temps simulé avec les différents solveurs pour notre ordinateur haut de gamme.

Premièrement, si l'on s'intéresse au coût de l'utilisation de la méthode d'extrapolation des pressions (cf section 3.4), on remarque que l'écart de temps de calcul avec la méthode de réflexion de la pression n'est pas grand. Il est au maximum de +15% dans le cas d'ISPH, ce qui est négligeable, au regard de la précision apportée par la méthode d'extrapolation de Shepard près des frontières solides (§ 3.4). Deuxièmement, une fréquence de CPU plus rapide et plus de threads disponibles rendent les simulations encore plus rapides en réduisant le temps de calcul par seconde du temps simulé de 37 % en moyenne. Troisièmement, le solveur ISPH requiert 5 fois moins de temps de calcul que le solveur WCSPH. Bien que le nombre d'itérations et donc le temps de calcul par pas de temps soit plus élevé, il est plus que compensé par la durée des pas de temps, qui est bien plus importante. En effet, le critère

limitant (Équation 3.63), spécifique à WCSPH, rend les pas de temps de cet algorithme très courts.

3.5.4 Conclusion

Bien que, dans un cas statique, l'algorithme WCSPH permet d'obtenir un champ de pression plus précis que celui obtenu avec ISPH, nous utiliserons ce dernier pour deux raisons. Premièrement, ISPH est plus robuste face aux phénomènes dynamiques, comme cela est montré dans la section 3.5.2. Grâce au solveur de pression, ISPH rétablit bien plus rapidement le champ de pression lors d'un évènement brutal. Deuxièmement, malgré le nombre d'itérations requises par le solveur de pression, ISPH permet de diviser par 5 le temps de calcul, comme constaté dans la section 3.5.3. Nous utiliserons donc pour la suite de nos recherches le solveur ISPH couplé avec la méthode d'extrapolation qui a été choisie dans la section 3.4.

Actuellement l'algorithme ISPH ne fonctionne pas en temps réel. Les temps de calculs sont relativement courts pour une scène qui comporte un faible nombre de particules, comme c'est le cas pour nos essais, mais ce temps sera augmenté si l'on cherche à simuler un environnement marin complet. La méthode d'implémentation de notre algorithme a été décrite dans l'annexe J. Actuellement, le programme est développé pour être modulable, pour permettre des modifications et des essais. Ceci implique que le programme est codé pour que les informations soient mémorisées sous forme de tableau de structure (hormis le solveur de pression de l'algorithme ISPH, voir annexe J), ce qui n'est pas optimal en ce qui concerne l'utilisation de la mémoire cache et la latence due aux transferts de données entre les différentes couches mémoires. En figeant l'algorithme et en reprogrammant le solveur pour que l'information soit stockée sous la forme de structure de tableau, un gain significatif du temps de calcul est à espérer, ce qui permettra à terme d'utiliser les cartes graphiques (GPU).

3.6 Conclusion du chapitre

Ce chapitre a présenté les algorithmes SPH utilisés dans cette thèse pour la simulation du fluide. L'intégration de la méthode d'extrapolation pour l'interaction fluide-solide au sein de ISPH constitue la deuxième contribution de cette thèse. Celle-ci permet une meilleure précision sur le calcul de la distribution de pression à la frontière fluide-solide comparée à la méthode de calcul par réflexion. Dans la suite de cette thèse, la méthode d'extrapolation de la pression, qui est bien plus précise que la méthode de réflexion de la pression, sera utilisée. Une étude hydrodynamique heuristique¹⁶ de comparaison entre deux méthodes SPH pour la modélisation de l'écoulement représente la troisième contribution. Ces deux contributions ont été publiées dans l'article [GMR⁺19]. La comparaison des résultats des cas tests avec les méthodes WCSPH et ISPH montre que ISPH couplé à la technique d'extrapolation de la pression est bien plus stable dans un cas dynamique et son utilisation permet de réduire le temps de calcul. ISPH est retenu pour le simulateur de la dynamique des véhicules sous-marins et objets submergés. De plus, même si l'objectif de simulation temps réel n'est pas encore atteint, il reste encore beaucoup de marge pour optimiser davantage l'implémentation des algorithmes (voir annexe J). Ceux-ci pourraient d'ailleurs être portés sur une architecture massivement parallèle, comme par exemple un GPU.

Les configurations des expériences numériques réalisées dans cette section ont été imaginées en suivant la démarche de validation présentée à la section 1.6.1. Le premier point de cette démarche a été évalué. Le fluide seul possède un bon comportement et l'hydrostatique d'un objet immergé devrait être réaliste puisque la répartition de la pression à la surface de la sphère est correcte. Ceci sera évalué dans le prochain chapitre. Il reste également à valider les deux derniers points de la démarche d'évaluation. En utilisant l'algorithme ISPH et la technique d'extrapolation de la pression, le comportement du simulateur SPH dans un cas hydrodynamique sera analysé dans le chapitre 4.

16. Heuristique signifie ici que l'on a cherché à trouver une méthode de calcul qui fournit rapidement une solution réalisable, pas nécessairement optimale, pour notre problème de simulation hydrodynamique.

Chapitre 4

Résultats sur la dynamique des solides immergés avec ISPH

Le chapitre précédent a montré que notre simulateur SPH peut simuler l'écoulement d'un fluide autour d'un objet avec une précision acceptable dans un temps raisonnable sous réserve de disposer des capacités de calcul adéquates, ce qui constitue la première étape de la démarche de validation d'un simulateur marin proposée dans la section 1.6.1. Au regard de cette démarche et pour en valider la seconde étape, la valeur de la poussée d'Archimède simulée doit être analysée. Ses effets doivent aussi être vérifiés dans un cas dynamique. La simulation doit permettre qu'un objet plus dense que l'eau coule et qu'un objet moins dense que l'eau flotte, ce qui sera l'objet de la première partie de ce chapitre.

Par ailleurs, pour montrer que l'on peut simuler la dynamique d'un véhicule marin au cours d'une mission, il faut analyser les forces hydrodynamiques s'appliquant sur le solide (voir chapitre 2). Ainsi, il faut démontrer que le mouvement des objets immergés est similaire, sinon homothétique à la réalité. Pour ce faire une scène de simulation expérimentale est présentée dans la section 4.2. Celle-ci servira de base pour toutes les expérimentations de ce chapitre. Une méthode dérivée de la méthode NIPF, décrite dans la section 2.4.3, est proposée pour montrer la précision du simulateur au regard des coefficients hydrodynamiques et ainsi confronter le simulateur à la dynamique d'objets réels. Ce chapitre cherche à valider la dernière étape de la démarche de validation d'un simulateur (§ 1.6.1). En premier, la capacité du simulateur à reproduire le phénomène de masse ajoutée sera mis en évidence, notamment en utilisant l'estimation des paramètres hydrodynamiques. Puis, la capacité du simulateur à simuler des frottements est mise en lumière, en montrant qu'il permet de simuler la vitesse limite d'un objet dans le fluide simulé avec ISPH. Les forces de frottement seront analysées et des corrections seront proposées pour améliorer les simulations. Enfin, une discussion finale est proposée, permettant de conclure sur le caractère homothétique des simulations et d'analyser les temps de calculs nécessaires pour ce type d'essai.

4.1 Poussée d'Archimède

4.1.1 Poussée d'Archimède dans un cas statique

Lorsque la sphère est définie comme statique au milieu du fluide et que le fluide s'écoulant autour de la sphère s'est stabilisé, la force de pression résultante du fluide sur le solide est la poussée d'Archimède. En effet, la pression évolue avec la profondeur, tel que présenté au chapitre 3, avec l'équation 3.77 :

$$p(z) = \rho g z$$

La force qu'exerce le fluide sur le solide $\vec{F}_{fluide,stat}$ dans un cas statique est directement liée à l'intégrale de la pression s'exerçant à sa surface :

$$\vec{F}_{fluide,stat} = - \oint_S p dS \vec{n} = - \iiint_V \overrightarrow{grad}(p) dV \quad (4.1)$$

Avec dS l'élément de surface du solide, \vec{n} le vecteur normal à la surface et dV l'élément de volume du solide. L'évolution hydrostatique précédemment énoncée nous permet d'exprimer la valeur du gradient de pression :

$$\overrightarrow{grad}(p) = \rho \vec{g}$$

On peut alors remplacer cette valeur dans l'équation 4.1 et en considérant le vecteur gravité invariant dans l'espace :

$$\vec{F}_{fluide,stat} = -\rho \vec{g} \iiint_V dV$$

On retrouve alors la valeur de la poussée d'Archimède :

$$\vec{F}_{fluide,stat} = -\rho V \vec{g} \quad (4.2)$$

Avec V le volume du solide. De la même manière, on s'attend à obtenir, en sommant toutes les forces \vec{F}_s^P de l'équation 3.70 dans le cas où la sphère est statique et le fluide dans un état stable, une valeur proche de la poussée d'Archimède théorique.

$$\vec{F}_{Archi,Sim} = \sum_{s \in S} \left(-m^2 \sum_{s \in S} \sum_{b \in \mathcal{F}_s} (\Theta_s + \Theta_b) \vec{\nabla} W_{sb} \right) \quad (4.3)$$

Les valeurs obtenues avec la méthode d'extrapolation et ISPH, en se basant sur les expériences décrites dans la section 3.5, comparées à la valeur théorique sont présentées dans le tableau 4.1. De manière logique, l'erreur sur la valeur de la poussée d'Archimède simulée (4.5%) est similaire à l'erreur moyenne calculée pour le champ de pression (4.69% cf. 3.5). Cet écart est relativement faible et n'est problématique que dans le cas où la masse volumique du corps à simuler est très proche de la masse volumique du fluide.

	<i>Force(N)</i>	<i>Variation</i>
Théorique ($F_{\text{fluide,stat}}$)	328	
Simulation ($F_{\text{Arch,Sim}}$)	313	-4.5%

Tableau 4.1 – Poussée d'Archimède simulée

4.1.2 Poussée d'Archimède dans un cas dynamique

Lorsque l'on place un objet dans un fluide, on s'attend à ce que celui-ci coule si sa densité est supérieure à celle du fluide et à ce que celui-ci flotte si sa densité est inférieure à celle du fluide. Cet effet doit être vérifié dans le cas de notre simulateur. Une simulation SPH est donc mise en place avec les mêmes paramètres que ceux utilisés pour les calculs hydrostatiques de la sphère (§ 3.5 et tableau 3.2). La même sphère de diamètre $\varnothing 0,4\text{ m}$ est utilisée.

Cinq valeurs de densité pour la sphère $\rho_{\text{sphère}}$ sont testées : 500 kg.m^{-3} , 750 kg.m^{-3} , 950 kg.m^{-3} , 1050 kg.m^{-3} , 1500 kg.m^{-3} . Le principe de chaque essai est simple : la sphère est lâchée au-dessus de la surface de l'eau puis laissée totalement libre pendant 20 s . Ce temps permet la stabilisation de la simulation. Le bassin simulé représente un mètre cube d'eau pour les densités de sphère 500 kg.m^{-3} et 750 kg.m^{-3} et deux mètres cube ($1 \times 1\text{ m} \times 2\text{ m}$ de profondeur) dans les autres cas. Après ce temps, l'obtention d'un équilibre de la sphère à la surface de l'eau ou le fait que celle-ci ne remonte pas à la surface est observé. De plus, si celle-ci se trouve à la surface, le volume immergé est calculé pour valider le réalisme de la simulation de la poussée d'Archimède.

La figure 4.1 représente la situation finale après chacun des cinq essais. Les particules de fluide sont colorées en fonction de leur pression, de bleu à rouge allant des pressions basses aux hautes. Les particules de la sphère sont colorées en vert lorsqu'une pression s'applique sur elles et celles du bassin en violet. Les sphères de densité 500 kg.m^{-3} et 750 kg.m^{-3} remontent

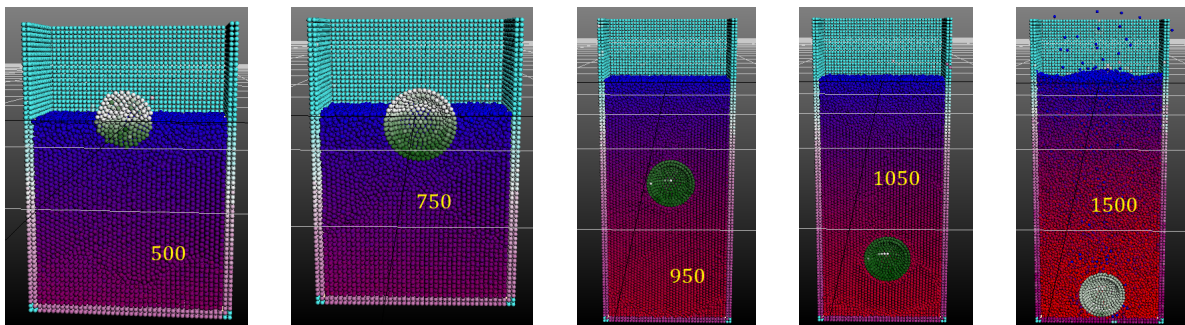


Figure 4.1 – Situation finale après les essais de coulée en cas dynamique.

bien à la surface. Elles se stabilisent avec respectivement 46,1% et 67,8% (respectivement 50% et 75% étaient attendus) de leur volume immergé, ce qui correspond à la sous-évaluation de la poussée d'Archimède mise en évidence dans le tableau 4.1. Les sphères de densité 950 kg.m^{-3} et 1050 kg.m^{-3} coulent mais se stabilisent à des profondeurs différentes. Ceci peut être dû à l'incertitude grandissante du champ de pression avec la profondeur, comme cela peut être visible sur les figures du chapitre 3. Comme annoncé avec les résultats du cas statique, le simulateur SPH a des difficultés à simuler la poussée d'Archimède quand les objets ont une densité très proche de celle du fluide. La sphère de densité 1500 kg.m^{-3} coule jusqu'au fond, ce qui était attendu. La figure montre l'état de la simulation juste après l'impact avec le fond, ce qui explique l'état du champ de pression à la surface de la sphère.

4.1.3 Conclusion

Ces résultats montrent que le simulateur SPH est bien capable de simuler la poussée d'Archimède avec une précision raisonnable. Toutefois, l'erreur faite est problématique lors de la simulation de la dynamique de corps dont la densité est très proche de celle du fluide.

4.2 Présentation du principe expérimental

Cette section présente le dispositif expérimental utilisé dans la suite de ce chapitre ainsi que la méthode d'estimation des paramètres hydrodynamiques utilisée. Elle propose également une analyse du conditionnement du problème d'estimation afin de vérifier la pertinence de nos résultats.

4.2.1 Dispositif expérimental de simulation

4.2.1.1 Description de la scène simulée

Pour déterminer les paramètres hydrodynamiques, puisque cela est possible en simulation, un essai de translation horizontale est effectué, où les mouvements dans les autres directions sont contraints. Les paramètres hydrodynamiques de l'objet sont donc obtenus pour un mouvement de cavement (§ 2.1.1). Ainsi, l'accélération de l'objet ne dépend pas de la gravité mais est définie par une force constante qui s'applique au solide. Ceci permet d'utiliser une force plus faible que le poids et qui correspond aux forces appliqués aux véhicules marins dans des conditions réelles. L'objet choisi pour ces essais est, comme dans le chapitre 3, la sphère, qui est la forme étudiée jusqu'à présent et dont, pour rappel, on a une bonne connaissance des caractéristiques hydrodynamiques. La masse volumique de cette sphère est choisie constante et égale à 2000 kg.m^{-3} . La sphère est immergée dans un bassin et est poussée par une force constante selon un axe fixe, comme illustré sur la figure 4.2. La taille du bassin est variable selon l'essai. L'algorithme utilisé pour cette simulation est ISPH (§ 3.3.3). La technique de calcul de pression à la frontière fluide-solide est celle de l'extrapolation (§ 3.4). La vitesse de la sphère est mesurée au cours du temps. Une condition de glissement libre est appliquée

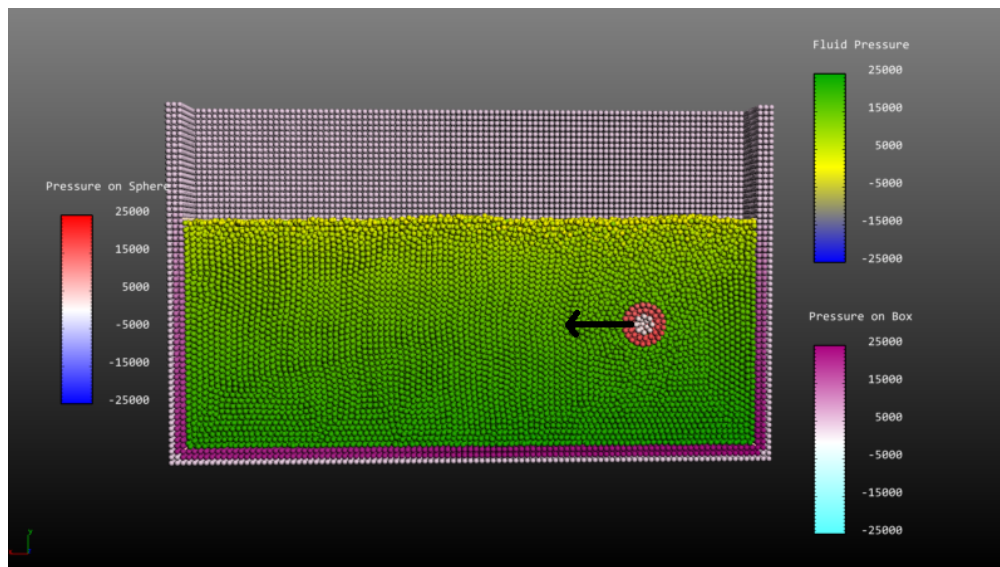


Figure 4.2 – Expérience numérique d'accélération d'une sphère dans un bassin.

sur la surface du bassin, interdisant toute force de frottement visqueux sur cette dernière. Dans l'équation 3.67, \vec{F}_s^V est tout simplement égale à 0. Il en va de même pour le calcul de l'accélération (Équation 3.44) due aux effets visqueux, par principe de réciprocité. Ainsi, aucune accélération n'est calculée à partir des valeurs de vitesse des particules solides hormis

la gravité. On a tout simplement (les changements apportés sont marqués en rouge) :

$$\dot{\vec{v}}_a^P = \frac{10m\mu}{\rho_a} \sum_{b \in \mathcal{F}_a} \frac{(\vec{v}_a - \vec{v}_b) \cdot (\vec{r}_a - \vec{r}_b)}{\rho_b |\vec{r}_a - \vec{r}_b|^2} \vec{\nabla} W_{ab} + \vec{g}$$

Différentes tailles de sphères sont utilisées pour ces expérimentations. La conservation du caractère homothétique des simulations est étudiée avec trois tailles de sphère : une sphère de diamètre $\varnothing 0.3$, une autre de diamètre $\varnothing 0.4$ et une dernière de diamètre $\varnothing 0.5$. Ainsi, les masses de ces sphères sont respectivement, 28.3 kg , 67.0 kg et 130.9 kg . La valeur des paramètres hydrodynamiques est estimée avec ces sphères et plusieurs niveaux de discrétisation pour la simulation, tel que représenté sur la figure 4.3.

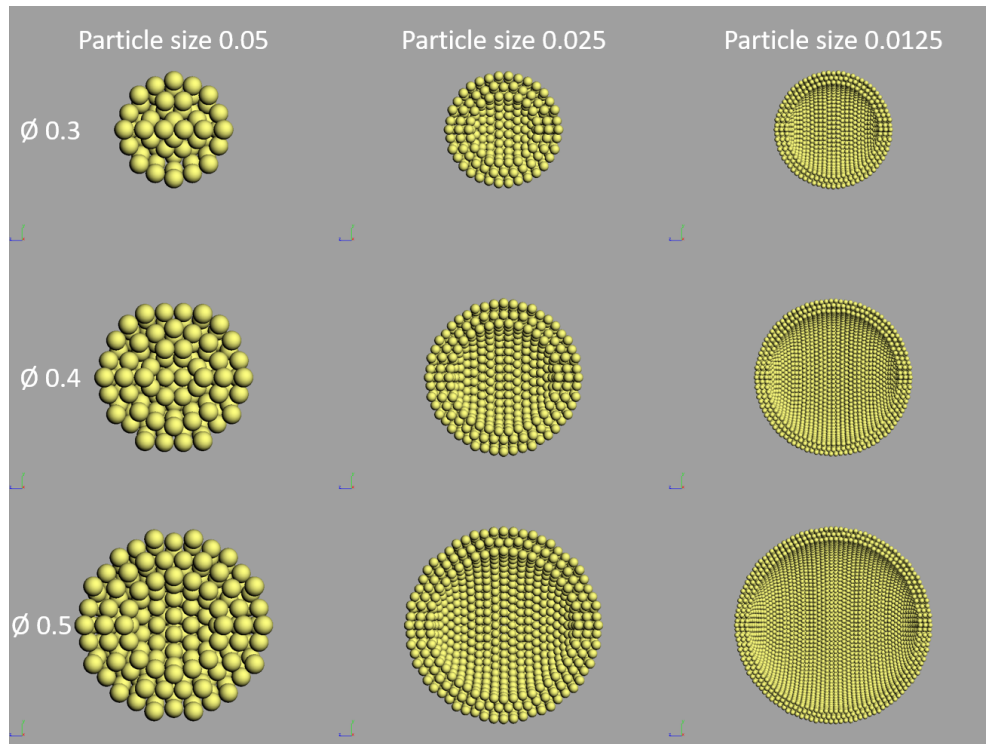


Figure 4.3 – Différentes tailles de sphères et de particules utilisées pour l'identification de paramètres hydrodynamiques.

Le kernel utilisé pour cette expérience est le B-Spline quintique (§ 3.2.2.2) qui est plus précis et permet d'approcher plus finement les champs physiques, ce qui améliore la stabilité de l'algorithme ISPH et probablement la précision des forces hydrostatiques et hydrodynamiques. En passant au B-Spline quintique, α_h , le rapport entre le rayon d'interaction et la taille des particules, peut être augmenté de 2 jusqu'à au moins 2.5, qui est la valeur retenue. Ainsi, le nombre moyen de particules se trouvant à l'intérieur de la portée de la fonction kernel passe de 32.5 à 64.4. Les paramètres de l'algorithme ISPH utilisés sont décrits dans le tableau 4.2.

Si on compare les tableaux 4.2 et 3.2, on constate qu'en dehors des paramètres dont les changements ont été expliqués plus haut (δ_r et α_h), seul le paramètre ϵ a changé. ϵ a été divisé par 2 pour aboutir à une meilleure convergence de l'algorithme ISPH et ainsi limiter les oscillations du champ de pression observées sur la figure 3.14.

λ_{CFL}	0.4	ϵ	0.00025
λ_{diff}	0.125	ω	0.5
δ_r	0.05, 0.025 et 0.0125 m	Ω	0.6
α_h	2.5	it_{min}	5

Tableau 4.2 – Paramètres de l’algorithme ISPH

4.2.1.2 Modèle dynamique de la sphère

La somme des forces projetée sur l’axe \vec{x} de translation de la sphère est :

$$F_{Input} - K_l v_x - K_q v_x |v_x| = (m + m_a) a_x \quad (4.4)$$

Avec $v_x = \vec{v}_{sphere} \cdot \vec{x}$ la vitesse de la sphère projetée sur l’axe \vec{x} , $a_x = \vec{a}_{sphere} \cdot \vec{x}$ son accélération. K_l , K_q et m_a les paramètres hydrodynamiques de la sphère et m sa masse, F_{Input} la force appliquée à la sphère pour sa mise en mouvement. Les termes de Coriolis sont absents car les vitesses de rotation sont nulles (§ 2.1.2).

Cette équation est utilisée pour calibrer les scènes à simuler d’une part, et pour l’estimation des paramètres hydrodynamiques d’autre part. Pour rappel, pour la sphère, le coefficient quadratique adimensionnel C_d et K_q sont liés par l’équation (§ 2.2.2.5) :

$$K_q = \frac{1}{2} C_d \rho A \quad (4.5)$$

Dans la suite de ce chapitre, le coefficient C_d sera employé, car c’est le coefficient que l’on retrouve dans la plupart des ouvrages de référence traitant des forces de frottement s’appliquant sur une sphère.

4.2.1.3 Description du ordinateur utilisé dans ce chapitre

Le ordinateur utilisé est un ordinateur muni de deux processeurs Intel Xeon E5 2680 v2, cadencé à 2,80 GHz et pouvant aller jusqu’à 3.60 GHz et possédant chacun 10 coeurs, 20 Threads et 25 Mo de mémoire cache, soit au total 40 coeurs et 50 Mo de mémoire cache. Le calcul est fait en parallèle sur 25 Threads. Le ordinateur possède 64 Go de mémoire RAM. Le disque dur est un disque SSD. L’affichage est assuré au moyen d’une carte Nvidia Quadro K2000. Les temps de calculs résultant des expériences menées dans ce chapitre seront analysés dans la section 4.5.4.

4.2.2 Numerical Integration Fitting

Pour déterminer les paramètres hydrodynamiques de la sphère, une catégorie de méthodes similaires à la méthode NIPF, développée dans la section 2.4.3, est introduite ici. Cette méthode, pour rappel, consiste à intégrer numériquement l’équation de la dynamique du véhicule pour obtenir une trajectoire, puis à utiliser un algorithme d’optimisation point-intérieur pour minimiser l’écart entre la vitesse obtenue analytiquement par intégration et l’évolution temporelle de la vitesse mesurée de l’objet dont on veut connaître les paramètres hydrodynamiques. L’algorithme point intérieur utilisé est basé sur une fonction de coût qui a

été définie dans le chapitre 2 comme étant (équation 2.59) :

$$J = \frac{(f_s^P - f_m^P)^2}{(f_m^P)^2} G_f + \frac{(\varphi_s^P - \varphi_m^P)^2}{4\pi^2} G_\varphi + \frac{\overrightarrow{\Delta\theta} \cdot \overrightarrow{\Delta\theta}^T}{\overrightarrow{\theta}^m \cdot \overrightarrow{\theta}^m{}^T} G_\Delta$$

Ici, il est cependant impossible d'utiliser les informations de fréquence et de phase, puisque nous ne sommes pas dans le cas d'un mouvement de balancier. La position angulaire est remplacée par la position cartésienne suivant x . La fonction de coût de l'algorithme NIPF devient alors :

$$J = \frac{\overrightarrow{\delta r}_x \cdot \overrightarrow{\delta r}_x^T}{\overrightarrow{r}_x^m \cdot \overrightarrow{r}_x^m{}^T} G_\Delta, \quad \overrightarrow{\delta r}_x = \overrightarrow{r}_x^A - \overrightarrow{r}_x^m \quad (4.6)$$

Les paramètres de l'équation 4.6 sont :

- \overrightarrow{r}_x^m le vecteur contenant toutes les positions mesurées sur une plage temporelle donnée,
- \overrightarrow{r}_x^A le vecteur contenant toutes les positions obtenues analytiquement à l'aide de paramètres hydrodynamiques sur la même plage temporelle,
- $G_\Delta = 100$ le même gain permettant d'ajuster la vitesse de convergence de l'algorithme.

On notera que l'utilisation du dénominateur dans cette fonction ne sert pas à la convergence de l'algorithme, puisque ce terme est constant lors de l'optimisation. Il sert néanmoins a posteriori dans l'analyse du résultat de l'optimisation, puisqu'il permet de s'assurer de l'écart maximal obtenu entre la courbe mesurée et la courbe analytique optimisée.

Lors de la simulation, il est possible d'enregistrer la vitesse du solide en parallèle de la position. Ainsi, deux autres méthodes peuvent être imaginées. La première est une méthode qui se base uniquement sur l'écart de vitesse et qui ne nécessite donc qu'une intégration simple. La seconde est une méthode qui compare les écarts de position et de vitesse simultanément. Ces méthodes pourraient alors devenir une classe de méthodes nommée Numerical Integration Fitting (NIF). Elles se retrouvent dans d'autres articles traitant de l'identification de paramètres hydrodynamiques, comme par exemple Chen [CCCT07].

Le set initial de paramètres utilisé pour tous les algorithmes de cette section est le même que dans le chapitre 2, c'est à dire (voir Équation 2.60) :

$$\overrightarrow{\xi}_{init} = [m_a^{th}, 0, C_d^*]$$

4.2.2.1 Numerical Integration Velocity Fitting (NIVF)

Dans le cas de la simulation SPH, il est possible d'utiliser directement la vitesse mesurée au cours de l'essai, ce qui évite d'utiliser la double intégration numérique lors de l'intégration du modèle dynamique de la sphère. Cette intégration est alors réalisée avec Runge Kutta 4 (Voir annexe C). Pour la méthode NIVF, la fonction de coût devient simplement :

$$J = \frac{\overrightarrow{\Delta v}_x \cdot \overrightarrow{\Delta v}_x^T}{\overrightarrow{v}_x^m \cdot \overrightarrow{v}_x^m{}^T} G_\Delta \quad (4.7)$$

Avec \overrightarrow{v}_x^m le vecteur contenant toutes les vitesses mesurées sur une plage temporelle donnée, \overrightarrow{v}_x^A le vecteur contenant toutes les vitesses obtenues analytiquement à l'aide de paramètres hydrodynamiques sur la même plage temporelle, $\overrightarrow{\Delta v}_x = \overrightarrow{v}_x^A - \overrightarrow{v}_x^m$ la différence entre ces deux vecteurs et $G_\Delta = 100$ le même gain que NIPF permettant d'ajuster la vitesse de convergence de l'algorithme.

4.2.2.2 Numerical Integration Velocity and Position Fitting (NIVPF)

Pour conclure sur l'utilité d'utiliser la position combinée à la vitesse, vérifions s'il existe une évolution entre les paramètres obtenus avec une combinaison des deux algorithmes, en définissant une méthode que l'on nomme Numerical Integration Velocity and Position Fitting (NIVPF) qui nécessite une double intégration pour que l'on puisse se baser sur la position et dont la fonction de coût est la suivante :

$$J = \frac{\overrightarrow{\delta r_x} \cdot \overrightarrow{\delta r_x^T}}{\overrightarrow{r_x^m} \cdot \overrightarrow{r_x^m^T}} G_\Delta + \frac{\overrightarrow{\Delta v_x} \cdot \overrightarrow{\Delta v_x^T}}{\overrightarrow{v_x^m} \cdot \overrightarrow{v_x^m^T}} G_\Delta \quad (4.8)$$

4.2.3 Conditionnement du problème d'estimation de paramètres hydrodynamiques

Cette partie présente une analyse de conditionnement du problème de l'estimation de paramètres hydrodynamiques selon le modèle de frottement utilisé. Le conditionnement mesure la dépendance de la solution d'un problème numérique par rapport aux données du problème. Ici, il exprime la difficulté d'estimation des paramètres m_a , C_d et K_l et permet de contrôler la validité des résultats de l'estimation. Si le conditionnement est élevé, le problème est dit mal conditionné et la confiance dans les résultats de l'estimation des paramètres hydrodynamiques est faible. Dans le cas d'un problème mal conditionné, un faible changement dans les paramètres d'entrées a un fort effet sur le résultat de l'estimation. Si le conditionnement est idéal alors le conditionnement vaut 1. Dans notre cas le problème d'estimation revient à un problème linéaire $A \vec{x} = \vec{b}$, avec A la matrice contenant les données mesurées, \vec{x} le vecteur de paramètres à estimer et \vec{b} le second membre de l'équation dynamique. Le conditionnement de la matrice A peut être calculé tel que décrit dans Gallouët [Gal19]. Soit \mathbb{R}^n muni de la norme euclidienne et $\mathcal{M}_n(\mathbb{R})$ muni de la norme induite. Soit $A \in \mathcal{M}_n(\mathbb{R})$ une matrice inversible. On note $cond(A)$ le conditionnement associé à la norme induite par la norme euclidienne sur \mathbb{R}^n et il vaut :

$$cond(A) = \sqrt{\frac{\sigma_n}{\sigma_1}} \quad (4.9)$$

Avec σ_1 et σ_n respectivement la plus petite et la plus grande des valeurs propres de $A^T A$ (qui est une matrice symétrique définie positive).

Il permet d'établir le rapport maximal entre les variations dans la matrice de conditionnement et toute variation des paramètres estimés. En effet, soit $A \in \mathcal{M}_n(\mathbb{R})$ une matrice inversible, et $\vec{b} \in \mathbb{R}^n$, $b \neq 0$. On munit \mathbb{R}^n d'une norme $|\cdot|$, et $\mathcal{M}_n(\mathbb{R})$ de la norme induite. Soit $\delta A \in \mathbb{R}^n$; on suppose que $A + \delta A$ est une matrice inversible. Si \vec{x} est solution de $A \vec{x} = \vec{b}$ et $\vec{x} + \delta \vec{x}$ est solution de :

$$(A + \delta A) (\vec{x} + \delta \vec{x}) = \vec{b}$$

Alors :

$$\frac{|\delta \vec{x}|}{|\vec{x} + \delta \vec{x}|} \leq cond(A) \frac{|\delta A|}{|A|} \quad (4.10)$$

Ainsi, on majore l'erreur relative de l'estimation par rapport à une erreur sur la matrice A .

Pour définir la matrice de conditionnement du problème, on se réfère au modèle dynamique de la sphère (Équation 4.4)

$$F_{Input} - K_l v_x - K_q v_x |v_x| = (m + m_a) a_x$$

Les variables à déterminer sont m_a , C_d et K_l dans le cas du modèle LQ. Les coefficients liés à l'estimation de m_a sont les accélérations, ceux liés à l'estimation de C_d sont les vitesses au carré et ceux liés à l'estimation de K_l sont les vitesses. La matrice de conditionnement du problème avec les deux forces de frottement et la masse ajoutée est la matrice K_{LQ} :

$$K_{LQ} = \begin{pmatrix} v_0 & v_0^2 & a_0 \\ v_1 & v_1^2 & a_1 \\ \vdots & \vdots & \vdots \\ v_n & v_n^2 & a_n \end{pmatrix} \quad (4.11)$$

Avec $v_0 \dots v_n$ les vitesses à chaque pas de temps et $a_0 \dots a_n$ les accélérations à chaque pas de temps et n le pas de temps final. Les valeurs utilisées pour le calcul du conditionnement sont les données recalculées analytiquement avec les paramètres estimés.

On peut également analyser le problème dans le cas où l'on utilise un seul coefficient de frottement. Dans le cas du modèle linéaire L ($C_d = 0$), la matrice de conditionnement K_L est :

$$K_L = \begin{pmatrix} v_0 & a_0 \\ v_1 & a_1 \\ \vdots & \vdots \\ v_n & a_n \end{pmatrix} \quad (4.12)$$

Dans le cas du modèle quadratique Q ($K_l = 0$), la matrice de conditionnement K_Q est :

$$K_Q = \begin{pmatrix} v_0^2 & a_0 \\ v_1^2 & a_1 \\ \vdots & \vdots \\ v_n^2 & a_n \end{pmatrix} \quad (4.13)$$

4.3 Estimation de la masse ajoutée d'une sphère

Cette section présente les résultats d'estimation de masse ajoutée obtenus lors des expérimentations avec les sphères. L'influence des coefficients de frottement est étudiée dans la section suivante (§ 4.4). Ces expériences doivent, premièrement, montrer que l'on peut étendre les applications des méthodes Numerical Integration Fitting (NIF) (§ 4.2.2) à d'autres dispositifs que le balancier hydrodynamique, et deuxièmement, analyser les paramètres hydrodynamiques simulés avec l'algorithme SPH et différents niveaux de discrétisation. La première étape consiste donc à choisir un algorithme parmi la classe NIF qui servira à estimer les paramètres dans la suite du chapitre. Puis, les résultats de masses ajoutées obtenus et ses variations, notamment en fonction de la discrétisation et de la vitesse seront analysés.

4.3.1 Numerical Integration Velocity Fitting

Pour chaque essai présenté dans cette section, une sphère est placée dans un bassin dont la taille est de $6 \times 2,5 \times 1.5 \text{ m}$, comme illustré sur la figure 4.2. Vu la taille du bassin, pour obtenir une plus grande plage de vitesses, la force appliquée au solide est choisie à $201,6 \text{ N}$. Cette force correspond à la force requise pour accélérer la sphère de 0 à 1 m.s^{-1} en 1 s en utilisant les paramètres de référence pour la sphère. Elle est donc appliquée à la sphère pendant 1 s puis est coupée subitement, et la sphère est laissée libre pendant 2 s . Ainsi, la décélération de la sphère est observée. Le choix d'une trajectoire où la vitesse est croissante puis décroissante est crucial pour la détermination des paramètres hydrodynamiques et permet de dissocier masse ajoutée et frottement (cf. § 4.5.2). Ici, seule la masse ajoutée sera étudiée et seule la sphère de diamètre $\varnothing 0,4 \text{ m}$ (masse 67.0 kg) sera utilisée pour la simulation. Cette masse est assez proche de la masse d'un véhicule marin autonome, comme par exemple, le Sparus II de masse 50 kg . Dans un premier temps, contrairement au chapitre 2, un seul modèle de frottement est utilisé pour analyser la masse ajoutée estimée. Le modèle LQ est choisi car il combine les deux frottements et doit permettre de laisser l'algorithme converger au mieux.

Les résultats issus de la méthode d'identification NIVF et les différentes tailles de particule δ_r sont donnés dans le tableau 4.3. Dans ce tableau, pour chaque δ_r , la première ligne donne la

δ_r	Type	m_a	K_l	C_d	Coût
0.05	Référence	16.72	0	0.41	47.75
	Simulation	22.66	86.17	0	0.05
0.025	Référence	16.72	0	0.41	5.87
	Simulation	21.31	53.09	0	0.016
0.0125	Référence	16.72	0	0.41	1.53
	Simulation	19.35	27.02	0.0	0.003

Tableau 4.3 – Résultats de l'estimation des paramètres hydrodynamiques avec une sphère de diamètre $\varnothing 0.4 \text{ m}$ en utilisant le modèle de frottement LQ et NIVF

valeur des paramètres hydrodynamiques utilisés pour la simulation de référence qui représente également le set initial de paramètres avant optimisation avec NIVF. La seconde ligne donne les valeurs estimées à partir des données de simulation. La dernière colonne donne la valeur de la fonction de coût en utilisant les paramètres hydrodynamiques de chaque ligne, c'est à

dire en faisant la différence entre la vitesse obtenue avec l'équation 4.4 et ces paramètres et l'évolution temporelle de la vitesse provenant de la simulation avec l'algorithme SPH. Pour les paramètres estimés à partir de la simulation, plus cette valeur est faible et meilleure est la précision de l'estimation de paramètres. En ce qui concerne les paramètres de référence, plus la trajectoire obtenue avec l'algorithme SPH est proche de la courbe analytique de référence et plus la valeur du coût est faible.

Comme on peut le constater, l'algorithme d'estimation de paramètres hydrodynamiques converge dans tous les cas. Les écarts entre les valeurs estimées de masse ajoutée et la valeur théorique sont assez faibles, l'écart maximal pour toutes les tailles des particules simulées est de 35% et l'écart minimal est de 19%. Logiquement, plus la discrétisation est fine et plus l'écart entre la valeur simulée et la valeur de référence/théorique est réduit. Les résultats des valeurs de forces de frottement sont discutés dans la prochaine section (4.4). La valeur élevée de la fonction de coût sur la ligne de référence pour l'essai avec $\delta_r = 0.05$ signifie que la courbe de l'évolution temporelle de la vitesse de la sphère est assez éloignée de la courbe de référence obtenue analytiquement.

Les résultats issus de la méthode d'identification NIPF et les différentes tailles de particule δ_r sont donnés dans le tableau 4.4. Ce dernier est construit de la même manière que le tableau 4.3. On constate tout d'abord que les valeurs estimés avec cet algorithme sont très proches

δ_r	Type	m_a	K_l	C_d	Coût
0.05	Référence	16.72	0	0.41	95.34
	Simulation	23.58	86.03	0	0.006
0.025	Référence	16.72	0	0.41	12.73
	Simulation	22.08	52.67	0	0.002
0.0125	Référence	16.72	0	0.41	1.58
	Simulation	19.71	26.76	0.0	0.0003

Tableau 4.4 – Estimation des paramètres hydrodynamiques avec une sphère de diamètre $0.4m$ en utilisant le modèle de frottement LQ et NIPF

des paramètres estimés avec l'algorithme NIVF. La variation maximale est observée pour la masse ajoutée dans le cas de la simulation avec $\delta_r = 0.05$ et est de l'ordre de 4%, ce qui est tout à fait négligeable. Les valeurs des fonctions de coût changent mais gardent le même ordre de grandeur et la même évolution en comparant la simulation avec la courbe issue des paramètres analytiques.

Les résultats obtenus avec la méthode d'identification NIVPF et les différentes tailles de particule δ_r sont donnés dans le tableau 4.5. Ce dernier contient uniquement les valeurs de fonction de coût dans le cas de la discrétisation $\delta_r = 0.025$ et d'une sphère de diamètre $\varnothing 0,4m$ et met en évidence les résultats obtenus avec les trois différentes méthodes pour comparaison. On constate que la valeur du coût avec la méthode NIVPF est la somme des coûts obtenus avec les méthodes NIPF et NIVF lorsque l'on analyse le coût obtenu à partir des valeurs de référence. Par ailleurs, les variations obtenues dans les valeurs estimées avec ces trois méthodes sont négligeables, ainsi, on peut donc au choix utiliser, la méthode NIPF, la méthode NIVF ou une combinaison des deux. Par souci de simplicité et pour éviter la double intégration numérique qui pourrait entraîner des erreurs, la méthode NIVF sera adoptée dans la suite de ce chapitre.

Type	Méthode	m_a	K_l	C_d	Coût
Référence	NIVF	16.72	0	0.41	5.87
	NIPF	16.72	0	0.41	12.73
	NIVPF	16.72	0	0.41	18.59
Simulation	NIVF	21.31	53.09	0	0.016
	NIPF	22.08	52.67	0	0.002
	NIVPF	21.61	53.02	0	0.019

Tableau 4.5 – Comparaison des résultats obtenus avec les techniques NIPF, NIVF et NIVPF avec le modèle de frottement LQ et la simulation de la sphère de diamètre $\varnothing 0,4 m$

Afin d'illustrer les résultats de l'estimation de paramètres, les courbes de l'évolution temporelle des vitesses obtenues avec les simulations SPH sont représentées sur la figure 4.4. Les vitesses simulées avec SPH sont en couleurs, et la vitesse obtenue analytiquement avec les paramètres de référence est en noir. On remarque que plus la discrétisation est fine et

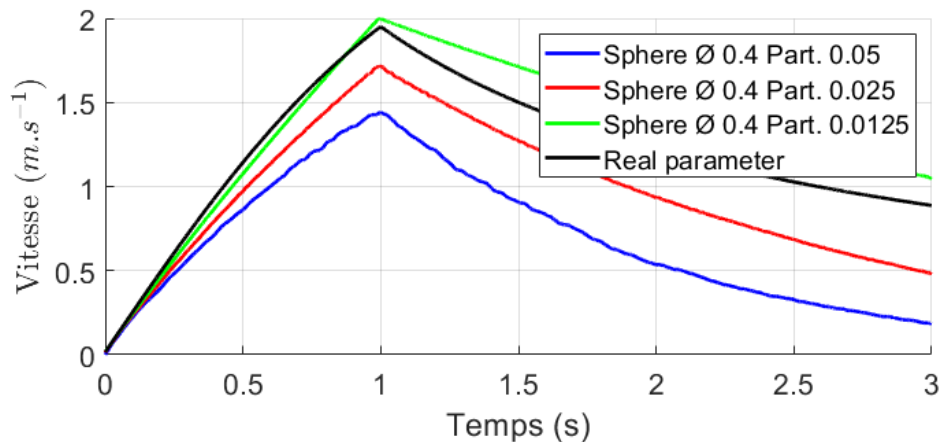


Figure 4.4 – Évolution temporelle de la vitesse de la sphère de $\varnothing 0,4 m$ avec différentes tailles de particules δ_r

plus on se rapproche du comportement obtenu analytiquement à partir des coefficients de référence. Avec la discrétisation la plus fine (courbe verte), la vitesse semble par contre aller au-delà de la courbe de référence. Puisque la masse ajoutée s'approche de la masse ajoutée théorique, le coefficient de frottement est peut-être sous-estimé dans ce cas. Ceci sera analysé dans la section (4.4). Le résultat obtenu est très satisfaisant en ce qui concerne le caractère homothétique de la simulation : une masse ajoutée et un coefficient de frottement sont bien simulés.

Il s'agit maintenant de vérifier, premièrement, que l'évolution constatée de la masse ajoutée avec la sphère de diamètre $\varnothing 0,4 m$ est similaire avec les autres sphères de diamètres $\varnothing 0,3 m$ et $\varnothing 0,5 m$ présentées dans la section 4.2. Après avoir analysé les résultats obtenus avec les frottements (§ 4.4), dans la dernière section de ce chapitre (§ 4.5), la convergence de l'algorithme NIVF basé sur une optimisation de type point intérieur sera discutée.

4.3.2 Influence de la discrétisation des sphères

Sur la figure 4.3, on peut observer que la taille des particules a une grande influence sur la rugosité de la surface, ce qui peut expliquer que la force de frottement soit surévaluée au cours de la simulation, comme précédemment évoqué. En outre, plus la discrétisation est grossière et moins il y a de particules à la surface, donc plus une erreur de calcul de pression sur une unique particule a d'impact sur le calcul de la force de pression résultante. Pour ces essais, la même force est appliquée aux sphères de diamètre 0,3 et 0,5 que celle pour la sphère de diamètre 0,4, même si cette force ne correspond plus à une accélération empirique de 0 à 1 $m.s^{-2}$ pour pouvoir comparer ces évolutions dans des conditions similaires. Ceci permet de démontrer qu'il y a bien une évolution de la vitesse atteinte par la sphère selon son diamètre avec une même taille de particule δ_r , ce qui est représenté sur la figure 4.5.

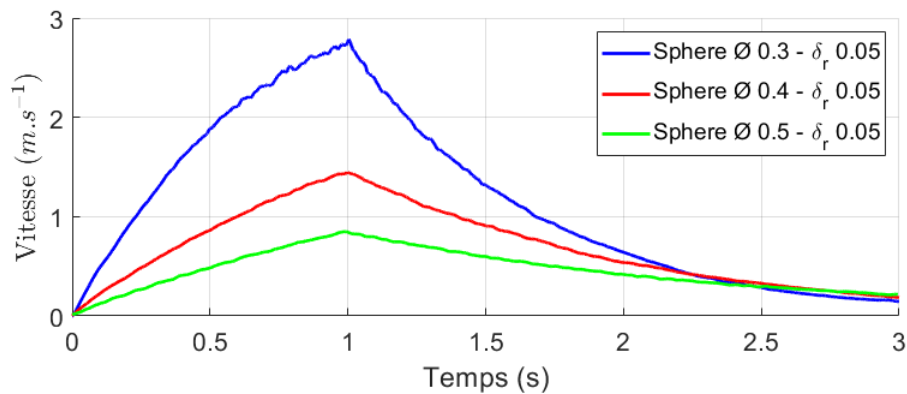


Figure 4.5 – Évolution temporelle de la vitesse des sphères pour une taille de particules δ_r fixe (0.05)

Cette figure montre qu'avec une même force, la manière dont les sphères se comportent dans l'eau évolue, une sphère plus petite (donc plus légère) se déplace beaucoup plus qu'une sphère plus grosse (donc plus lourde). Cette évolution est conforme à la réalité.

Le tableau 4.6 permet de comparer la valeur de la masse ajoutée estimée par l'algorithme NIVF pour toutes les tailles de sphères et toutes les discrétisations.

Diamètre	$\delta_r = 0.05$		$\delta_r = 0.025$		$\delta_r = 0.0125$		Référence
	m_a (kg)	Var	m_a (kg)	Var	m_a (kg)	Var	m_a (kg)
0.3	9.67	37%	8.56	21%	8.4	19%	7.05
0.4	22.66	35%	21.31	27%	19.35	16%	16.72
0.5	45.0	38%	43.13	32.5%	41.71	28%	32.66

Tableau 4.6 – Masse ajoutée simulée avec SPH avec toutes les tailles de particules et de sphère. Comparaison avec la valeur théorique.

Pour toutes les tailles de sphère, les valeurs de masse ajoutée estimées se rapprochent de la valeur analytique lorsque δ_r diminue. La valeur maximale de l'erreur sur la masse ajoutée est inférieure à 40% même en utilisant un niveau de discrétisation faible, ce qui est très satisfaisant car cela implique que cette méthode pourrait permettre d'obtenir dans un temps raisonnable une estimation de la valeur de masse ajoutée. Le fait que les variations de masse ajoutée ne

soient pas similaires avec les trois tailles de sphère s'explique peut-être par l'estimation des forces de frottement qui sera étudiée dans la prochaine section. Une autre possibilité pourrait être liée à la simulation de la dynamique du fluide qui dépendrait de la résolution (δ_r) utilisée.

4.3.3 Influence de la vitesse de la sphère

Pour vérifier si la vitesse de la sphère influe sur la masse ajoutée simulée de la sphère, comme cela a pu être supposé au chapitre 2, la masse ajoutée de la sphère de diamètre $\varnothing 0,4 m$ est calculée selon une méthode de plage glissante avec le modèle LQ. Pour cette étude, l'algorithme NIVF est utilisé sur une plage temporelle mobile d'une durée de $0.2 s$ et qui avance de $0.1 s$ le long des données de simulation. Le pas de temps pour les données recueillies après la simulation est un pas de temps régulier de longueur $\delta t = 0.0005 s$, ce qui veut dire que l'algorithme utilise 400 points de la trajectoire pour converger sur chaque plage.

L'évolution de la masse ajoutée en fonction de la vitesse est représentée sur la figure 4.6. Sur cette figure, les points de couleurs représentent les valeurs de masse ajoutée estimée pour une vitesse. La vitesse utilisée est la vitesse moyenne sur la plage de données sélectionnée pour l'estimation des paramètres. La valeur de la masse ajoutée théorique pour cette sphère est représentée en noir.

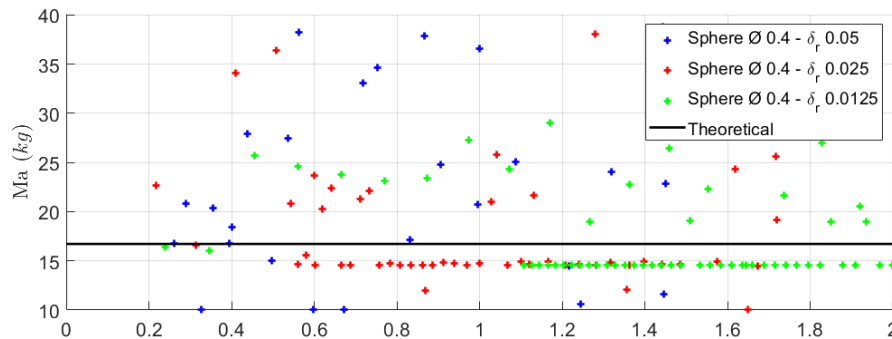


Figure 4.6 – Evolution de la masse ajoutée en fonction de la vitesse pour une sphère de diamètre $\varnothing 0,4 m$ et le modèle de frottement LQ

Les valeurs de masse ajoutée estimée sont assez disparates pour les particules les plus grossières ($\delta_r = 0.05$). Dans les autres cas et hormis quelques points, une valeur qui semble constante est estimée. L'algorithme point intérieur est peut-être mis en difficulté par l'utilisation simultanée de deux forces de frottement, impliquant l'estimation de trois paramètres. L'utilisation d'un seul paramètre de frottement est probablement plus approprié pour la détermination de la masse ajoutée avec une plage de données glissante.

La figure 4.7 montre l'évolution de la masse ajoutée en fonction de la vitesse, mais avec cette fois-ci l'estimation des paramètres se basant sur le modèle L (qui est le modèle où $C_d = 0$).

L'évolution est beaucoup plus resserrée autour de la valeur théorique avec ce modèle de frottement. Ces résultats sont très satisfaisants car ils démontrent, d'une part, que le simulateur SPH permet bien de simuler la masse ajoutée, et d'autre part, que notre algorithme d'estimation converge la plupart du temps vers la bonne valeur. Dans certains cas, néanmoins, l'algorithme semble ne pas converger correctement. Ceci est peut-être dû au conditionnement lors de l'estimation des paramètres hydrodynamiques et doit être analysé.

4.3.4 Analyse des conditionnements avec les modèles L, Q et LQ

Le conditionnement des matrices décrites dans la section 4.2.3 pour les essais avec les trois tailles de sphère ($\varnothing 0,3\text{ m}$, $\varnothing 0,4\text{ m}$ et $\varnothing 0,5\text{ m}$) et différentes valeurs de δ_r est présenté dans le tableau 4.7. La première colonne montre la taille de la sphère, la seconde, la taille des particules, la troisième, la valeur du conditionnement avec le modèle LQ, la quatrième, celui du modèle L et la cinquième, celui du modèle Q.

Sphère	δ_r	$cond(K_{LQ})$	$cond(K_L)$	$cond(K_Q)$
$\varnothing 0,3\text{ m}$	0.05	7.79	1.16	1.22
	0.025	15.57	1.27	2.36
	0.0125	30.96	1.34	3.33
$\varnothing 0,4\text{ m}$	0.05	7.09	1.29	1.24
	0.025	8.12	1.08	1.19
	0.0125	12.63	1.20	1.84
$\varnothing 0,5\text{ m}$	0.05	6.54	1.34	1.25
	0.025	9.27	1.06	1.42
	0.0125	13.85	1.29	1.10

Tableau 4.7 – Conditionnement des matrices d’estimation pour les simulations de translation horizontale

Le conditionnement de la matrice K_{LQ} est très élevé : il vaut en moyenne 12.35 pour nos essais. Ceci est peut-être dû à l’utilisation de deux coefficients de frottement. Pour les valeurs les plus élevées et très éloignées de 1, l’estimation avec le modèle LQ n’est pas fiable. C’est probablement ce qui se retrouve sur les figure 4.18 et 4.21. L’algorithme point intérieur est mis en difficulté par le nombre de paramètres à estimer. Dans le cas des modèles avec un seul coefficient de frottement, la valeur du conditionnement est alors grandement diminuée et vaut en moyenne 1.23 avec le modèle linéaire (L) et 1.66 avec le modèle linéaire (Q). Ces valeurs étant proches de 1, l’estimation à partir de ces modèles peut être considérée comme relativement fiable.

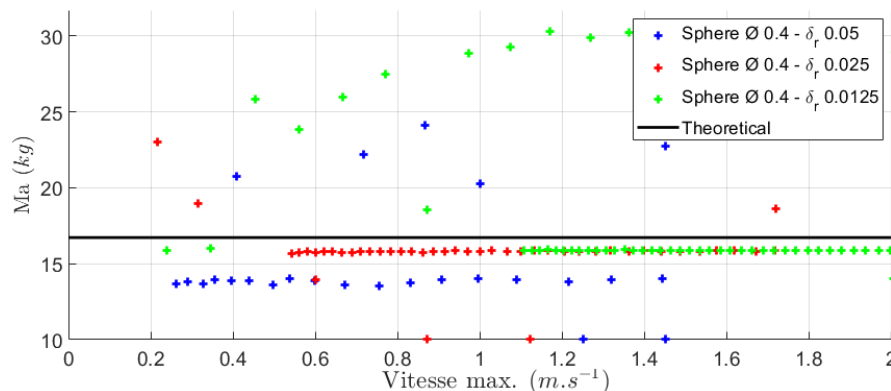


Figure 4.7 – Evolution de la masse ajoutée en fonction de la vitesse pour une sphère de diamètre $\varnothing 0,4\text{ m}$ et le modèle de frottement L

Ceci met en évidence que l'utilisation du modèle de frottement LQ met en difficulté l'algorithme d'estimation des paramètres hydrodynamiques, comme cela a été constaté dans la sous-section précédente. L'utilisation d'un unique modèle de frottement semble une meilleur approche pour l'estimation des paramètres. Néanmoins, la convergence de l'algorithme avec le modèle LQ sera vérifiée dans la suite de ce chapitre.

4.3.5 Mouvement de va-et-vient et masse ajoutée

Pour confirmer que la masse ajoutée diminue lors d'un va-et-vient, comme cela a été observé lors de l'expérience du balancier hydrodynamique au chapitre 2, l'effort périodique de la gravité sur la sphère va être reproduit en simulation. Cet effort hydrostatique est obtenu en combinant le modèle dynamique (Équation 2.50) et l'équation de la trajectoire avec le modèle de frottement linéaire (Équation 2.61). Celle-ci est ici considérée comme une bonne approximation pour le mouvement décrit par le balancier, l'important étant d'observer le comportement du simulateur face à une oscillation. Pour rappel, pour le balancier, dans le cas du modèle L :

$$\begin{aligned}\ddot{\theta}(t) &= \frac{P_{eq}}{Mr} \sin \theta(t) - \frac{K_l}{M} \dot{\theta}(t), \quad M = m + m_a \\ \theta(t) &= \left(\theta_0 \cos \beta t + \frac{-\alpha \theta_0 + \dot{\theta}_0}{\beta} \sin \beta t \right) e^{\alpha t} \\ \alpha &= \frac{-K_l}{2M} \\ \beta &= \frac{\sqrt{-4rP_{eq}M - K_l^2 r^2}}{2rM}\end{aligned}$$

Ainsi, la force $F_{input}(t)$ que l'on choisit d'appliquer à la sphère en simulation est :

$$F_{input}(t) = P_{eq} \frac{m_{sim}}{m} \sin \left(\left(\theta_0 \cos \beta t + \frac{-\alpha \theta_0 + \dot{\theta}_0}{\beta} \sin \beta t \right) e^{\alpha t} \right) \quad (4.14)$$

Avec m_{sim} la masse de la sphère simulée, qui sert à s'assurer que l'amplitude du mouvement de la sphère en simulation soit du même ordre de grandeur que celui de la sphère réelle. Les paramètres issus des mesures expérimentales et de l'exploitation des résultats avec NIPF provenant du chapitre 2 sont utilisés. Ces paramètres sont décrits dans le tableau 4.8.

P_{eq}	10,62 N	θ_0	18,38°
m	6,43 kg	$\dot{\theta}_0$	0 rad.s ⁻¹
r	1,1 m	K_l	1,35 N.s.m ⁻¹
m_a	1,31 kg		

Tableau 4.8 – Paramètres d'entrée pour la force à appliquer à la sphère dans le simulateur SPH

La simulation dure 25 s ce qui permet d'observer 3 périodes d'oscillations complètes. L'évolution temporelle de la position de la sphère simulée obtenue lors de deux essais similaires, dont l'origine du mouvement a été légèrement décalé, est illustrée sur la figure 4.8. La courbe

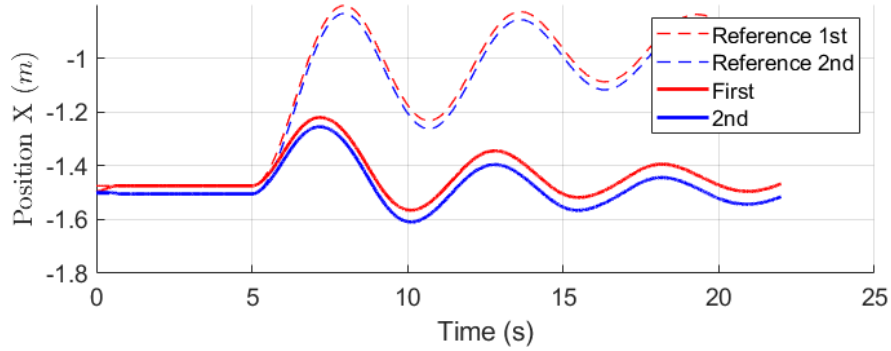


Figure 4.8 – Trajectoire décrite par la sphère au cours de l’expérience avec la force d’entrée du balancier hydrodynamique.

rouge représente la trajectoire obtenue lors du premier essai et la bleue celle lors du second. Les courbes en pointillés représentent les trajectoires avec les paramètres de référence.

On constate que les trajectoires obtenues avec le simulateur SPH ne correspondent pas à la trajectoire avec les paramètres hydrodynamiques de référence. Néanmoins, la simulation SPH possède une bonne répétabilité puisque les trajectoires sont tout à fait identiques, comme en atteste la courbe des vitesses. La méthode NIPF est réutilisée telle que dans le chapitre 2 pour estimer les paramètres hydrodynamiques de la sphère pour cet essai. Ainsi, la fonction de coût utilisée est :

$$J = \frac{(f_s^P - f_m^P)^2}{(f_m^P)^2} G_f + \frac{(\varphi_s^P - \varphi_m^P)^2}{4\pi^2} G_\varphi + \frac{\vec{\delta r}_x \cdot \vec{\delta r}_x^T}{\vec{r}_x^m \cdot \vec{r}_x^m T} G_\Delta \quad (4.15)$$

Le modèle de frottement linéaire est utilisé. La figure 4.9 montre la trajectoire estimée si l’on applique la méthode NIPF sur toute la durée de la simulation. Sur cette figure les courbes rouge et bleue pleines sont les mêmes que celles présentées sur la figure 4.8 mais les courbes pointillées représentent ici les courbes estimées par notre algorithme NIPF.

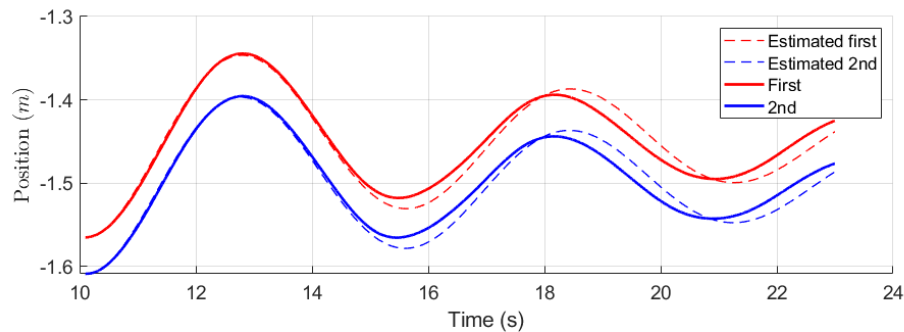


Figure 4.9 – Trajectoire estimée avec NIPF pour l’expérience avec la force d’entrée du balancier hydrodynamique.

Sur cette figure on constate que l’algorithme NIPF converge, et est en phase notamment pour la première oscillation, mais il y a un déphasage qui apparaît après la première oscillation. Ceci s’explique par l’utilisation de la fonction de coût (équation 4.15) qui permet d’aligner les signaux sur la même phase. Les paramètres estimés avec l’algorithme NIPF et le modèle L sont présentés dans le tableau 4.9. Ce tableau est organisé de la même manière que les

tableaux précédents (tableaux 4.3, 4.4 et 4.5). Une dernière colonne, notée Cond, est ajoutée pour exprimer le conditionnement du problème d'estimation. Les résultats obtenus avec le

Type	m_a	K_l	C_d	Coût	Cond
Référence	16.72	0	0.41	32.47	
L	28.17	125.9	0	0.0052	4.4
Q	34.6	0	20.98	0.019	9.8

Tableau 4.9 – Résultats de l'estimation des paramètres hydrodynamiques lors de l'essai du mouvement de balancier avec une sphère de diamètre $\varnothing 0.4 m$ en utilisant le modèle de frottement L et NIPF

modèle Q sont moins bons que les résultats obtenus avec le modèle L, ce qui sera analysé et expliqué dans la prochaine section (§ 4.4). Les masses ajoutées estimées sont étonnamment bien supérieures (+30%) à celles obtenues lors des essais en ligne droite, qui ont été présentées dans les tableaux 4.3, 4.4 et 4.5. Ces résultats sont inverses de ceux obtenus avec le balancier réel dans le chapitre 2, où la masse ajoutée estimée était inférieure à la masse ajoutée théorique. L'obtention d'une telle valeur de masse ajoutée s'explique peut-être par l'absence de véritable phénomène de turbulence dans la simulation, comme cela sera expliqué dans la section 4.4. Le conditionnement vaut 4.4 dans le cas du modèle L et 9.8 dans le cas du modèle Q. Le conditionnement plutôt élevé dans le cas du modèle Q expliquerait en partie ce résultat.

On peut alors se poser la question de l'évolution des paramètres hydrodynamiques au cours de l'essai et notamment l'évolution de la masse ajoutée. Une analyse du conditionnement est effectuée avec une plage de données glissante au cours du temps. Une plage temporelle d'une durée de 2,2 s est sélectionnée et décalée au fur et à mesure d'un quart de plage. Sur cette plage temporelle, cette fois, l'algorithme NIPF décrit dans la section 4.2.2 est utilisé, puisqu'il n'est pas possible d'utiliser les informations de fréquences et de phases sur une plage temporelle si courte (voir figure 4.9). La figure 4.10 présente l'évolution temporelle de la masse ajoutée au cours de l'essai du balancier avec le modèle de frottement L comparée à la valeur moyenne estimée, qui est ici la valeur de référence, ainsi que l'évolution du conditionnement du problème d'estimation. On constate sur cette figure que le conditionnement du problème

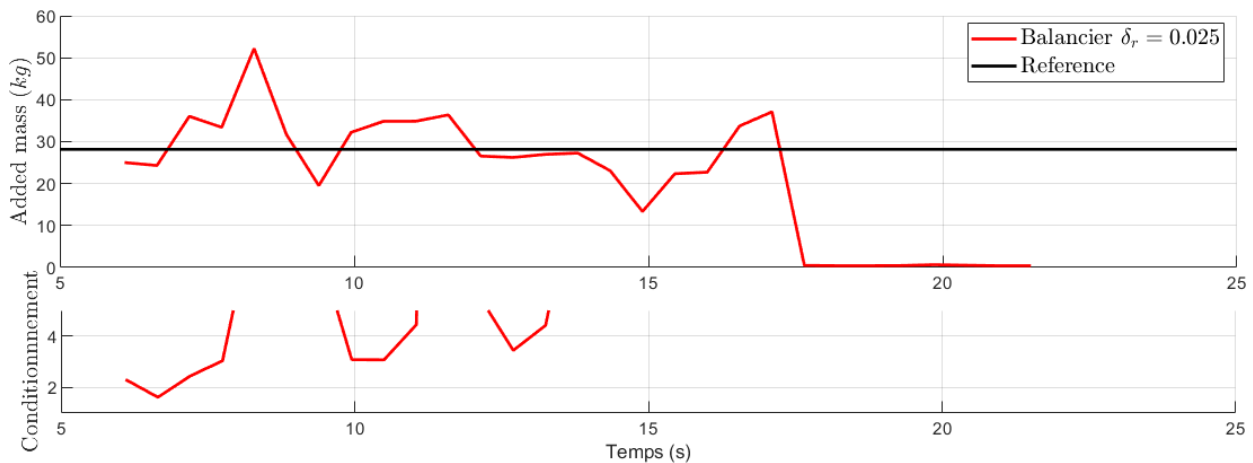


Figure 4.10 – Évolution de la masse ajoutée lors de l'expérience du balancier hydrodynamique et conditionnement de la matrice décrivant le problème d'estimation.

d'estimation est assez mauvais selon les périodes. Un grand nombre de points doit donc être écarté pour l'analyse des résultats. Si l'on choisit de ne conserver que les points où le conditionnement est inférieur à 5¹, il est difficile de conclure sur une éventuelle évolution du paramètre de masse ajoutée, même s'il est possible de l'observer sur la figure. En effet, il apparaît que la masse ajoutée est successivement inférieure puis supérieure à la valeur obtenue avec une estimation basée sur toutes les données de l'essai. Au vu des résultats, cette expérience ne sera pas analysée plus amplement.

1. Un conditionnement de 5 signifie que l'incertitude sur la matrice de conditionnement donc sur les vitesses et accélérations est multipliée par 5 sur les paramètres estimés. Cf équation [4.10](#)

4.3.6 Conclusion

Dans cette section, une estimation des paramètres hydrodynamiques à partir d'une simulation avec l'algorithme SPH a été conduite grâce aux algorithmes de la classe NIF, dérivés de la méthode NIPF précédemment introduite au chapitre 2. L'utilisation de l'algorithme NIVF a été préférée pour l'estimation des paramètres hydrodynamiques simulés avec SPH, car celui-ci permet de n'effectuer qu'une intégration simple et conduit au même résultat. Avec cette méthode, une valeur de masse ajoutée correcte a été obtenue. Après une étude avec différents niveaux de discrétisation, il apparaît que la masse ajoutée trouvée est relativement constante et dépend principalement de la taille de la sphère, ce qui correspond à un comportement réaliste de la simulation. Dans le pire des cas, la masse ajoutée estimée diffère de 40% de la masse ajoutée théorique. Ces résultats sont donc bien plus proches de la valeur théorique, que ceux obtenus avec l'expérience du balancier au chapitre 2. Dans certains cas, la précision de la simulation de la masse ajoutée est même de l'ordre de 20%. Le conditionnement des matrices du problème d'estimation remet en question l'utilisation du modèle de frottement LQ et, dans la suite, les modèles de frottement simple (L et Q) seront étudiés. Il faut maintenant étudier les valeurs des coefficients de frottement simulés qui pourraient expliquer certaines variations de la masse ajoutée estimée.

4.4 Estimation des forces de frottement

Cette section étudie les valeurs des forces de frottement s'appliquant sur la sphère. Pour démontrer la capacité du simulateur à simuler des frottements, il faut démontrer que l'on peut atteindre une vitesse limite en translation avec un objet immergé, comme décrit au point 3.(a) dans la section 1.6.1, ce qui fait l'objet de la première partie de cette section. Les 3 modèles de frottement analysés dans le chapitre 2 sont réutilisés ici pour l'estimation des forces de frottement : un modèle avec un frottement linéaire seul (L), un modèle avec un frottement quadratique (Q) et un modèle avec une combinaison de ces deux frottements (LQ). Ensuite, l'évolution des paramètres hydrodynamiques en fonction de la vitesse avec différentes résolutions de particules sera observée. Une discussion sera proposée sur l'analyse des forces provenant du fluide et s'exerçant sur le solide. Une méthode de correction de ces forces sera également introduite.

4.4.1 Vérification de la vitesse limite

Pour observer l'apparition d'une vitesse limite, la sphère est accélérée avec une force de 40 N pendant 12 s puis laissée en décélération libre. Ainsi, la distance parcourue par l'objet avant d'atteindre sa vitesse limite est plus faible qu'en utilisant la force définie dans la section précédente. Pour dimensionner le bassin à simuler, l'équation 4.4 est utilisée pour calculer analytiquement une trajectoire à partir de paramètres de référence. La figure 4.11 montre l'évolution de la position et de la vitesse de la sphère avec les paramètres hydrodynamiques de référence, c'est à dire en utilisant un coefficient de frottement C_d de 0.41 et une masse ajoutée calculée à partir de l'équation 2.37.

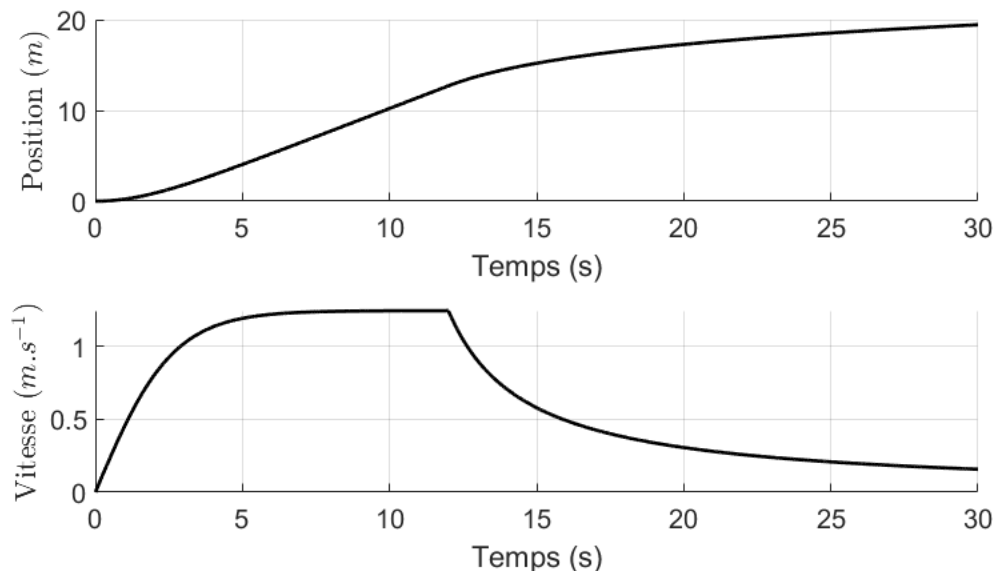


Figure 4.11 – Évolution attendue de la position et de la vitesse de la sphère au cours de l'essai de vitesse maximale

La simulation est effectuée dans un grand bassin ($21 \times 2,5 \times 3\text{ m}$) (Figure 4.11). Les temps de calculs colossaux requis pour faire des simulations avec un bassin de 21 m de long poussent, en général, à faire des essais dans des bassins plus petits, voir section 4.5.4. La vitesse limite

devrait être atteinte et maintenue pendant au moins une seconde. Puis, à la fin des 12 s, la vitesse limite n'est plus maintenue puisqu'il n'y a plus de force s'appliquant sur la sphère.

Deux tailles de particules sont utilisées pour la simulation : 0.05 m et 0.025 m (voir figure 4.3). L'état de la scène à la fin des 26 s de simulation avec les particules de taille $\delta_r = 0.025\text{ m}$ est représenté sur la figure 4.12. Sur cette figure, la sphère plongée dans le champ de particules

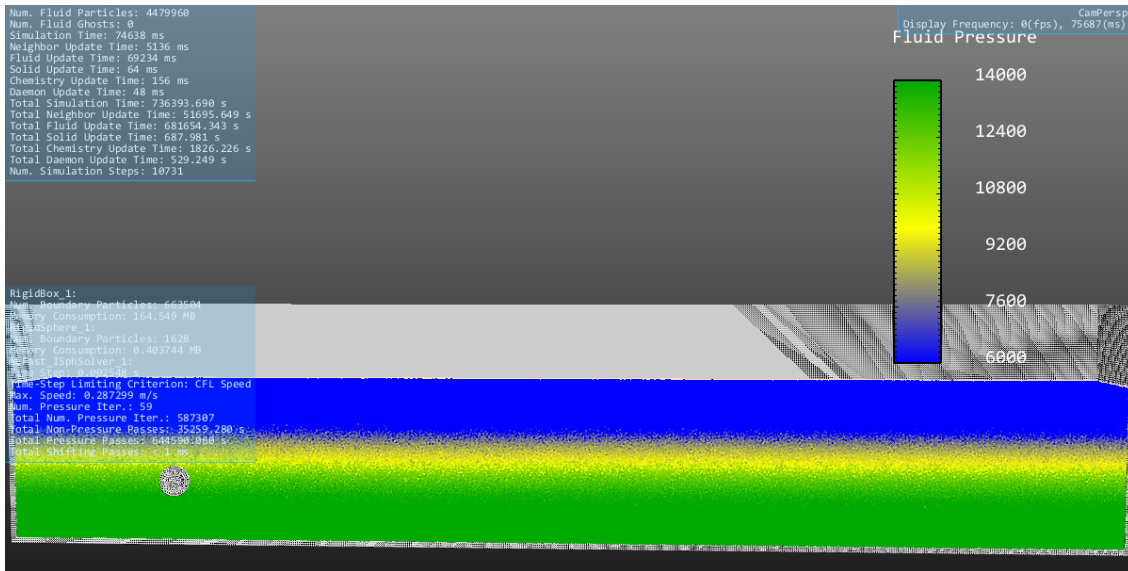


Figure 4.12 – Etat final de la simulation pour l'essai de vitesse maximale avec des particules de taille $\delta_r = 0.025\text{ m}$

de fluide peut être observée. Les particules de fluide sont coloriées en bleu, vert et jaune selon leur pression. Les parois solides sont représentées en blanc. La sphère a commencé sa trajectoire à droite du bassin et l'a traversé jusqu'à arriver sur la gauche.

La figure 4.13 montre la courbe de vitesse obtenue au cours de ces simulations, en bleu avec $\delta_r = 0.025$ et en rouge avec $\delta_r = 0.05\text{ m}$. La courbe noire est la courbe d'évolution attendue déjà présentée sur la figure 4.11.

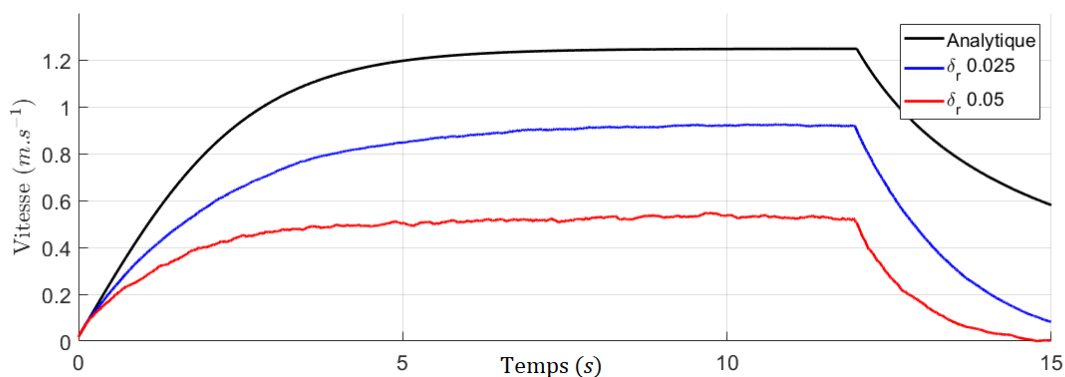


Figure 4.13 – Courbe de vitesse limite obtenue au cours de la simulation avec deux différentes tailles de discrétisation

Comme on peut le constater, une vitesse limite est bien atteinte au cours de chaque simulation. Ceci est très satisfaisant car cela montre que le simulateur est homothétique avec

la réalité physique. De plus, l'influence de la discrétisation a été vérifiée. Dans le cas de la courbe rouge, la taille des particules est particulièrement grossière ($0,05\text{ m}$ de diamètre) si on la compare à la taille de la sphère ($0,4\text{ m}$ de diamètre), voir la figure 4.3. On constate que plus la discrétisation est fine, plus la vitesse limite atteinte est proche de la vitesse réelle. Ceci peut s'expliquer de différentes manières :

- La taille des particules influe sur la précision des calculs numériques. En effet, on constate qu'il y a des oscillations sur la courbe de vitesse rouge, qui n'apparaissent pas ou sont moins prononcées sur la courbe bleue.
- La taille des particules joue sur la rugosité de la surface de la sphère. Il a déjà été démontré que la rugosité des surfaces des objets influe sur la valeur du coefficient de frottement estimé, notamment pour les sphères. En particulier, le coefficient de frottement C_d augmente avec la rugosité de la surface [Ach74], ce qui correspond au comportement observé ici.
- Le champ de particules autour de la sphère est mieux discrétisé et les mouvements dans le fluide ainsi que l'énergie dissipée dans le calcul des forces visqueuses sont plus réalistes.

4.4.2 Étude du type de frottement simulé avec SPH

Les modèles de frottement L, Q et LQ sont utilisés pour l'estimation des paramètres hydrodynamiques. Le choix du modèle de frottement à appliquer dépend de l'utilisation finale du véhicule marin. Si la vitesse du véhicule donc l'écoulement autour du véhicule, caractérisé par le nombre de Reynolds, est faible alors le modèle L est préférable. Au contraire, si le véhicule est rapide alors le modèle Q est à préférer. Le modèle LQ est une combinaison permettant de lier ces deux modèles et d'obtenir des paramètres pour toutes les situations. Il est néanmoins moins fiable, comme nous l'avons montré dans la section précédente en exprimant la valeur de conditionnement du problème. Puisqu'il y a normalement un seul coefficient de frottement, la répartition des frottements entre linéaire et quadratique dépendra fortement de l'expérience réalisée pour les déterminer. En pratique, vu les dimensions des véhicules marins (qui est généralement de l'ordre de 0.1 m^3) et les vitesses de ceux-ci (généralement de l'ordre de $(0.01 \text{ à } 2\text{ m.s}^{-1})$), le nombre de Reynolds de l'écoulement autour du véhicule est de l'ordre de 10^4 à 10^6 et seul un coefficient quadratique devrait être utilisé. Le tableau 4.10 reprend le formalisme du tableau 4.3 mais décrit les résultats obtenus en utilisant uniquement un frottement linéaire avec le modèle de frottement linéaire (L) ou quadratique (Q) pour une sphère de diamètre $\varnothing 0,4$. Pour chaque δ_r , la première ligne donne la valeur des paramètres hydrodynamiques utilisés pour la simulation de référence².

La valeur du coefficient de frottement estimée varie fortement, ce qui était prévisible au regard des essais de vitesse limite précédents (§ 4.4.1). L'algorithme converge moins bien vers le modèle quadratique comme nous l'avons montré au début de cette section. Les résultats de l'estimation sont donc moins fiables avec ce modèle. Peut-être que cela s'explique par un conditionnement du problème d'estimation moins bon avec le modèle Q (voir section 4.3.4). Il est intéressant de noter que les valeurs estimées avec le modèle L sont égales à celles obtenues avec le modèle LQ (Voir tableau 4.3). Si on compare les valeurs de coefficient de frottement obtenues on constate que l'on a le bon ordre de grandeur et que l'on simule même dans le cas où δ_r est petit un frottement inférieur au frottement réel. Le fait que le frottement

2. Ces lignes sont les mêmes que dans le tableau précédent

δ_r	Type	m_a	K_l	C_d	Coût
0.05	Référence	16.72	0	0.41	47.75
	Linéaire	22.66	86.17	0	0.051
	Quadratique	21.58	0	1.42	1.49
0.025	Référence	16.72	0	0.41	5.87
	Linéaire	21.31	53.09	0	0.016
	Quadratique	22.09	0	0.69	0.44
0.0125	Référence	16.72	0	0.41	1.53
	Linéaire	19.35	27.02	0	0.003
	Quadratique	21.04	0	0.27	0.053

Tableau 4.10 – Résultats de l’estimation des paramètres hydrodynamiques avec une sphère de diamètre $0.4m$ en utilisant les modèles de frottement L et Q

obtenu soit linéaire est peut-être dû à l’absence des phénomènes de turbulences qui génère ce frottement dans le cas des simulations SPH où δ_r est grand. Il n’est pas étonnant que la valeur du coefficient de frottement augmente selon la taille des particules, puisque l’état de la surface des sphères change entre les différents essais. Dans le chapitre 2, l’influence de l’état de surface³ de la sphère sur la valeur du coefficient C_d a été évoquée, notamment en faisant référence à Achenbach [Ach74].

3. Ici on parle de la rugosité de la surface de la sphère, qui dépend de la taille des particules

4.4.3 Influence de la vitesse

Les coefficients de frottement sont influencés par la vitesse, comme cela a été constaté au chapitre 2. Une évolution similaire est attendue avec la simulation SPH. Comme pour la masse ajoutée, l'algorithme NIVF est utilisé sur une plage glissante mobile, qui sélectionne une plage temporelle d'une durée de 0.2 s et qui avance de 0.1 s le long des données de simulation. Le pas de temps pour les données recueillies après la simulation est un pas de temps régulier de longueur $\delta t = 0.0005\text{ s}$, ce qui veut dire que l'algorithme utilise 400 points de la trajectoire pour converger sur chaque plage. Les résultats obtenus avec le modèle de frottement linéaire, c'est à dire avec $C_d = 0$ sont illustrés sur la figure 4.14. Sur cette figure, la valeur du coefficient K_l est présentée par rapport à la vitesse maximale atteinte par la sphère dans la plage temporelle sélectionnée.

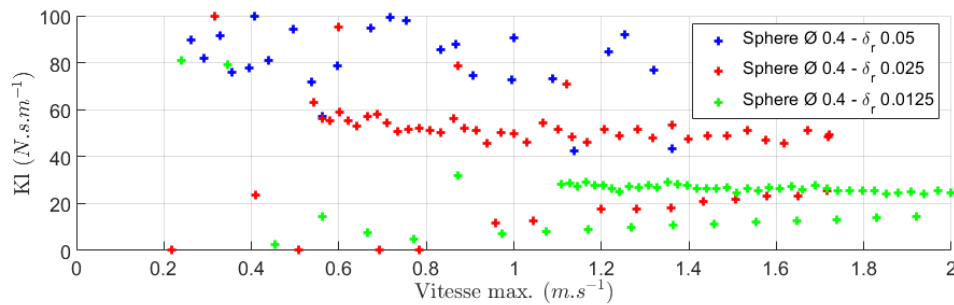


Figure 4.14 – Évolution du frottement linéaire K_l en fonction de la vitesse maximale dans la plage sélectionnée avec le modèle linéaire L pour trois taille de particules δ_r .

On observe que le coefficient de frottement linéaire diminue légèrement en fonction de la vitesse pour chaque δ_r . La variation du nuage de points dans le cas où $\delta_r = 0.05$ montre qu'il est difficile d'avoir une simulation précise des frottements dans ce cas. On constate un changement d'ordre de grandeur entre les différents niveaux de discrétisation δ_r , ce qui correspond aux résultats d'estimation faits sur l'ensemble des points, qui ont été présentés dans le tableau 4.3. Ces résultats sont assez satisfaisants, si on les compare aux résultats obtenus dans le chapitre 2.

S'il est impossible de comparer les forces de frottement linéaires avec des valeurs de référence, il est possible de faire cette comparaison en utilisant le modèle quadratique. Les résultats obtenus avec l'algorithme NIVF et ce modèle (Q) sont présentés sur la figure 4.15. Cette figure est basée sur le même principe que la figure précédente (Figure 4.14), le seul changement est le paramètre de frottement estimé qui n'est plus K_l mais C_d .

Sur cette figure, le même constat peut être fait avec l'évolution du coefficient de frottement : il diminue lorsque la vitesse augmente et il est variable selon la finesse de la discrétisation. Néanmoins, vu le nombre de Reynolds de l'écoulement autour de la sphère, une valeur relativement constante du coefficient de frottement était attendue (voir figure 2.6). Pour analyser plus finement le phénomène de frottement, une analyse des frottements simulés dans le cas où $\delta_r = 0.0125$ est réalisé, car c'est avec ce niveau de discrétisation que nous avons obtenu les données les plus proches des valeurs de référence. Les mêmes simulations que précédemment sont réalisées, avec une force constante s'appliquant sur la sphère pour l'accélérer pendant 1 s et l'observation de la décélération pendant 2 s . Le coefficient de frottement C_d étant un paramètre adimensionnel, l'évolution de ce coefficient avec différentes tailles de sphère est comparée sur la figure 4.16. En appliquant une force égale avec 3 tailles de sphère, une

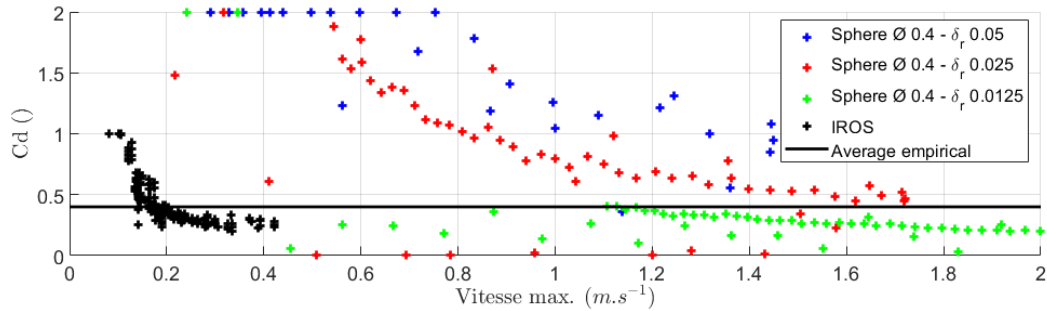


Figure 4.15 – Évolution du frottement quadratique C_d en fonction de la vitesse maximale dans la plage sélectionnée avec le modèle quadratique Q pour trois tailles de particules δ_r .

variation de la vitesse maximale atteinte est observée, ainsi la plage de vitesse sur laquelle l'estimation des paramètres hydrodynamiques peut être faite est différente pour les trois sphères.

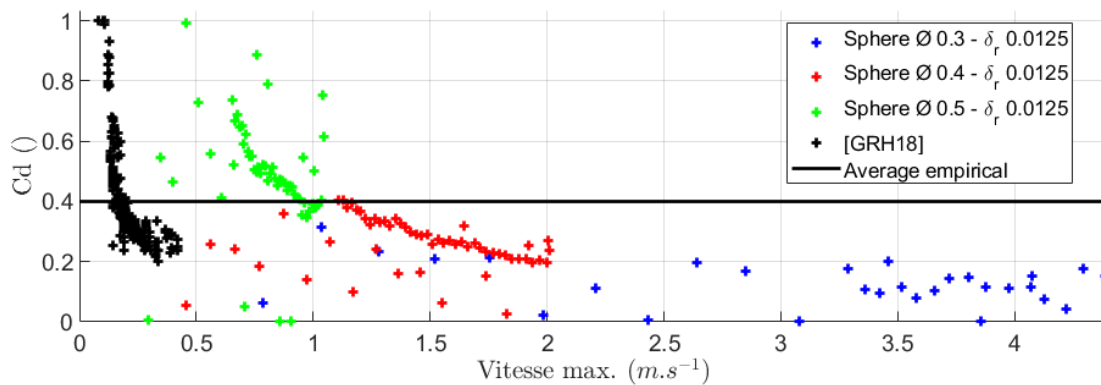


Figure 4.16 – Évolution du coefficient de frottement C_d en fonction de la vitesse maximale dans la plage sélectionnée avec différentes tailles de sphère pour $\delta_r = 0.0125$.

Sur cette figure, on constate que le paramètre C_d simulé est bien indépendant de la taille de la sphère, puisqu'il y a une continuité entre les nuages de points présents sur la figure. Cette continuité est notamment visible entre les courbes rouge et verte pour les vitesses entre 0.8 et 1.2 m.s^{-1} , mais moins entre les courbes rouges et bleues aux alentours des vitesses de 2 m.s^{-1} . A titre de comparaison sur cette figure, le nuage de points provenant de la publication [GRH18] et de la figure 2.20 est représenté. Bien que dans les deux cas, une décroissance est observée, il est plus probable qu'elle ait lieu pour des vitesses plus faibles où le nombre de Reynolds est inférieur à 10^3 , comme cela était le cas avec le balancier. En observant les variations entre les courbes de la sphère de diamètre $\varnothing 0,4 \text{ m}$ sur les figures 4.15 et 4.16, on constate que la pente s'aplatit et que le coefficient varie moins fortement (entre 1,5 et 0,5 pour $\delta_r = 0,025$ et entre 0,4 et 0,2 pour $\delta_r = 0,0125$). Ceci est encourageant et il est probable qu'en ajustant la résolution de la simulation du fluide autour de la sphère on obtienne de meilleurs résultats avec la simulation SPH. Ceci sera rendu possible avec un solveur multi-résolution.

En conclusion, si l'estimation de la force de masse ajoutée tendait à montrer que le simulateur la simule de manière réaliste avec une précision relativement bonne, la simulation des forces de frottement est plus problématique. Le coefficient de frottement estimé semble

être linéaire à la vitesse alors qu'en réalité pour l'écoulement qui a lieu autour de la sphère, cette force devrait être quadratique à la vitesse et varier dans une limite de 30% par rapport au nombre de Reynolds de l'écoulement. En analysant les valeurs de ce coefficient quadratique avec le niveau de discrétisation le plus grossier, la force appliquée sur la sphère est excessive et avec le niveau de discrétisation le plus faible, la force de frottement est inférieure à la réalité. Néanmoins l'évolution du coefficient en fonction de la vitesse semble être plus réaliste lorsque la discrétisation s'affine, ce qui est encourageant. Une analyse de l'ordre de grandeur des forces de frottement visqueux décrites dans le chapitre 3 est nécessaire pour expliquer ces résultats.

4.4.4 Analyse des forces issues du fluide s'appliquant sur le solide

Dans le chapitre 3, deux forces d'interactions fluide-solide sont proposées avec les algorithmes SPH. D'une part, les forces de pression \vec{F}_s^P (Équation 3.70) et, d'autre part, les forces de frottement visqueux \vec{F}_s^V (Équation 3.67). Les forces de pression \vec{F}_s^P sont le résultat du calcul de la dynamique des fluides fait par le solveur ISPH de Neutrino. Les forces de frottement visqueux \vec{F}_s^V sont dues au frottement surfacique au niveau de la surface de la sphère. Dans le chapitre 2, dans la section 2.2.2, les phénomènes à l'origine des forces de frottement ont été expliqués. Vu que le nombre de Reynolds de l'écoulement autour de la sphère est de l'ordre de $10^4 - 10^5$, il est probable que les forces de pression l'emportent sur les forces visqueuses. La figure 4.17 montre l'évolution des forces de pression $\vec{F}^P = \sum \vec{F}_s^P$ au cours du temps comparé à l'évolution des forces de frottement visqueux $\vec{F}^V = \sum \vec{F}_s^V$ s'appliquant sur la sphère.

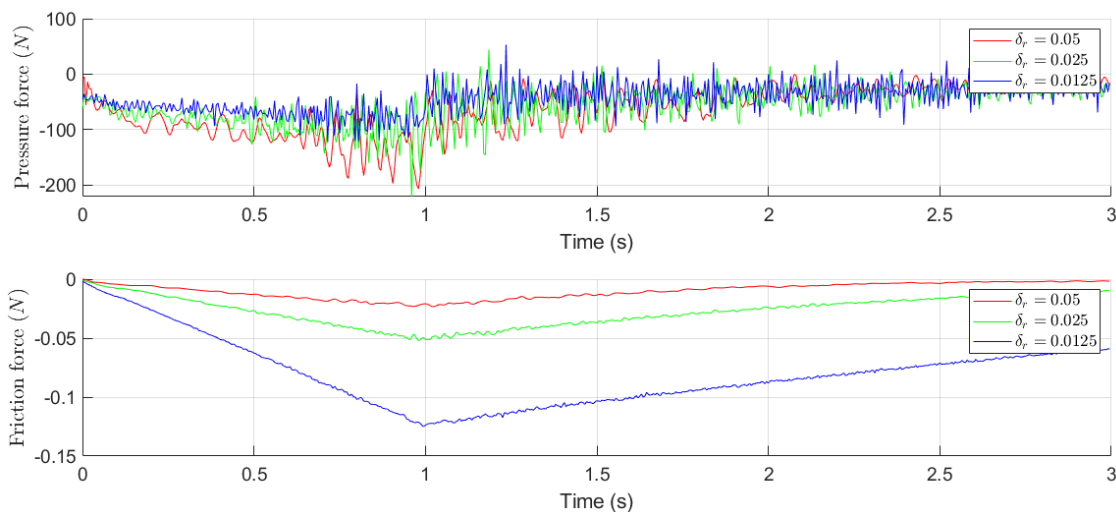


Figure 4.17 – Évolution des forces de pression et de frottements visqueux s'appliquant sur la sphère au cours de l'essai avec la sphère de $\varnothing.4$ et différentes valeurs de δ_r

Sur ce graphique, on remarque, premièrement, que l'ordre de grandeur des forces de frottement visqueux et des forces de pression est très différent : il y a un rapport 1000 entre ces deux forces ; et, deuxièmement, que les deux forces varient logiquement avec la vitesse de la sphère. Au vu des résultats et de la trajectoire, on peut estimer que les forces de pression sont du bon ordre de grandeur. Plus la discrétisation est fine et plus la force de frottement visqueux calculée est grande ; cette force reste néanmoins très faible comparée aux forces de

pression, comme c'était attendu au vu du nombre de Reynolds. Avec la discrétisation utilisée, toutes les turbulences censées apparaître dans le fluide n'ont pas lieu, ce qui peut expliquer que les frottements observés ne soient pas quadratiques. Toutefois, la force à l'origine d'un frottement linéaire devrait être la force de frottement visqueux \vec{F}^V qui dans notre cas est quasi nulle, ce qui est problématique. Cette force est en revanche très stable, la force due à la pression en revanche oscille assez fortement.

Les oscillations observées dans la force de pression \vec{F}^P sont dues à la résolution du champ de pression ISPH et ces résultats ressemblent aux résultats obtenus dans la section 3.5.2. Lors de la résolution du solveur de pression de l'algorithme ISPH (E.2), un certain niveau d'incompressibilité est atteint. Augmenter ce critère d'incompressibilité ϵ , qui est pour rappel :

$$\epsilon < \frac{\text{avg}_{a \in \mathcal{F}}(\max(\rho_a^*, \rho_0)) - \rho_0}{\rho_0} \quad (4.16)$$

permet de réduire les oscillations dans le champ de pression, mais requiert de bien plus nombreuses itérations pour la convergence du solveur et conduit à une augmentation significative du temps de calcul, voire empêche le solveur de converger dans certains cas. La trajectoire obtenue avec la valeur ϵ choisie est satisfaisante mais il est possible que ce facteur d'incompressibilité ait un effet sur la trajectoire de la sphère et sur la nature de la force de frottement.

Pour compenser les turbulences trop faibles à l'arrière de la sphère, un coefficient de mise à l'échelle pour les forces de frottement visqueux pourrait permettre de les compenser. Ainsi, il pourrait contribuer à rapprocher l'estimation de la force de frottement d'un modèle quadratique. Il serait variable selon la finesse de la discrétisation de la scène simulée.

4.4.5 Correction des forces de frottement visqueux

Pour approcher les forces de frottement réelles, un coefficient de frottement proportionnel au carré de la vitesse est ajouté. Ainsi, la force de frottement visqueux \vec{F}_s^V (Équation 3.67) devient :

$$\vec{F}_{s,modif}^V = \frac{10m^2\mu}{\rho_s} \sum_{b \in \mathcal{F}_s} (A(\vec{v}_b - \vec{v}_s) + 1) \circ \frac{(\vec{v}_b - \vec{v}_s) \cdot (\vec{r}_b - \vec{r}_s)}{\rho_b |\vec{r}_b - \vec{r}_s|^2} \vec{\nabla} W_{sb} \quad (4.17)$$

Avec A un facteur multiplicateur qui permet de rendre cette force proportionnelle à la vitesse au carré et \circ qui indique l'utilisation du produit d'Hadamard. Ce produit est utilisé pour dissocier l'application du coefficient selon l'axe du mouvement. Pour respecter la troisième loi de Newton, l'accélération visqueuse (Équation 3.37) est également modifiée selon le même principe. Ce facteur multiplicateur A s'explique par le fait que notre modèle ne prend pas en compte les phénomènes pouvant survenir à de plus petites échelles que la taille des particules simulées. La force de frottement due aux turbulences, voir section 2.2.2 n'est donc pas correctement simulée dans notre cas et l'ajout du facteur multiplicateur permet de la compenser. L'objet de cette partie est de trouver une valeur appropriée pour A . Pour cela, les essais sont réalisés avec une taille de sphère donnée. La taille choisie est celle de la sphère de diamètre $0.4 m$ avec la discrétisation $\delta_r = 0.025 m$. Le tableau 4.11 référence les valeurs de fonction de coût pour la trajectoire simulée avec le simulateur ISPH et différentes valeurs de coefficient A comparée à une trajectoire simulée avec les paramètres de référence. Pour chaque essai deux calculs sont effectués : le premier correspond à la détermination de la valeur de la

fonction de coût dans le cas où l'on utilise les paramètres de référence. Le second correspond à une détermination des paramètres hydrodynamiques simulés avec l'algorithme SPH en utilisant l'algorithme NIF. Les cellules vertes mettent en évidence les résultats présentés dans chaque ligne du tableau.

	A	m_a	K_l	C_d	Coût	A	m_a	K_l	C_d	Coût
Référence	0	16,72	0	0,41	6,08	120	16,72	0	0,41	5,2
Simulation	0	21,82	0	0,70	0,41	120	21,2	0	0,68	0,34
Référence	60	16,72	0	0,41	4,92	200	16,72	0	0,41	14,96
Simulation	60	21,30	0	0,67	0,35	200	19,93	0	0,91	0,37

Tableau 4.11 – Valeurs de fonction de coût pour la trajectoire simulée avec le simulateur ISPH et différentes valeurs de coefficient A comparée à une trajectoire simulée avec les paramètres de référence.

En vert les valeurs calculées à chaque ligne.

Comme on peut le constater sur le tableau 4.11, l'ajout du coefficient A a permis de diminuer la valeur de la fonction de coût, tant que A reste faible. Ceci est visible sur les lignes où l'on calcule la fonction de coût en comparant la courbe de référence et la courbe simulée. Ce coefficient permet bien de compenser l'absence de turbulence dans le sillage de la sphère et semble aider à ce que l'algorithme converge vers une valeur de masse ajoutée plus proche de la valeur théorique. La sphère semble toutefois être trop ralentie par l'eau qui l'entoure donc par les forces de pression \vec{F}_s^P (Équation 3.70), voir l'équation 3.66. L'essai avec $A = 200$ étant plus éloignée de la trajectoire réelle que celui avec $A = 120$.

La diminution de cette force de pression pourrait donc permettre de s'approcher plus encore de la trajectoire réelle. Dans ce cas, la diminution se ferait uniquement sur la force appliquée au solide dans les directions différentes de celle de la gravité pour ne pas perturber le calcul de poussée d'Archimède. La force \vec{F}_s^P (Équation 3.70) devient alors :

$$\vec{F}_{s,modif}^P = -A_{Press} m^2 \sum_{s \in S} \sum_{b \in \mathcal{F}_s} (\Theta_s + \Theta_b) \vec{\nabla} W_{sb} \quad (4.18)$$

Avec A_{Press} un coefficient choisi pour faire varier l'intensité de la force de pression. Le tableau 4.12 référence les valeurs de fonction de coût obtenues et de l'algorithme NIF de la même manière que dans le tableau 4.11. Également, les résultats présentés à chaque ligne sont mis en valeur en vert pour faciliter la lecture du tableau. L'évolution des coefficients A et A_{Press} a été choisie par dichotomie en fonction des résultats précédents.

L'ajout du coefficient sur la force de pression A_{Press} permet de diminuer la valeur de la fonction de coût lorsque l'on compare la courbe simulée et la courbe expérimentale. Sur les deux dernières lignes du tableau, une trajectoire simulée très proche de celle avec les paramètres de référence est observée. Ceci démontre que notre algorithme SPH est capable, après modification, de simuler de manière très réaliste un mouvement ou une trajectoire d'un objet immergé. Toutefois, il est difficile de donner une explication physique à l'ajout du coefficient A_{Press} , si ce n'est un facteur dû à une erreur numérique au vue de l'échelle des particules. De plus, ce coefficient dépend probablement du mouvement de l'objet ou de sa nature et l'on ne peut envisager son utilisation dans le cas d'un mouvement vertical.

	A_{press}	A	m_a	K_l	C_d	Coût
Référence	0,85	120	16,72	0	0,41	4,49
Simulation	0,85	120	17,72	0	0,66	0,29
Référence	0,75	60	16,72	0	0,41	0,70
Simulation	0,75	60	16,61	0	0,49	0,21
Référence	0,725	60	16,72	0	0,41	0,5
Simulation	0,725	60	15,75	0	0,48	0,19

Tableau 4.12 – Valeurs de fonction de coût pour la trajectoire simulée avec le simulateur ISPH et différentes valeurs de coefficient A et A_{press} comparée à une trajectoire simulée avec les paramètres de référence.

Une approche radicalement différente de celle employée dans cette section serait de considérer que les forces générées par les turbulences des particules de fluide (donc par les frottements visqueux entre particules du fluide) à l'arrière de l'objet sont trop faibles. L'ajout d'un modèle mathématique pour étudier et contrôler les effets de ces turbulences, à la manière des calculs Reynolds Averaged Navier Stokes (voir section 2.3.1), pourrait être une piste de recherche.

4.4.6 Conclusion

Cette partie a montré que le coefficient de frottement avec la simulation SPH n'était pas tout à fait réaliste. Certes, un coefficient de frottement est bien simulé, et une vitesse limite due aux phénomènes de frottements peut être atteinte, mais cette force de frottement semble plutôt correspondre à un frottement linéaire que quadratique. La valeur du coefficient de frottement semble être assez proche des frottements réels dans le cas où l'on simule une scène avec une discrétisation fine du fluide et de la surface du solide. L'utilisation d'un solveur SPH multi-résolution, comme celui de Hu [HPR⁺17], qu'il faudrait adapter à notre algorithme SPH, permettrait probablement d'améliorer les résultats obtenus tout en maintenant le temps de calcul à un niveau acceptable. L'évolution du frottement quadratique semble bien être homothétique à la réalité, puisque le caractère adimensionnel du coefficient C_d a notamment été vérifié, en montrant la continuité obtenue avec trois différentes tailles de sphère pour un même niveau de discrétisation. En comparant avec l'évolution réelle, cette évolution est décalée. Au regard des plages de vitesses et en considérant le nombre de Reynolds de l'écoulement simulé qu'impliquent ces vitesses, le coefficient de frottement simulé ne devrait pas autant varier.

Les forces s'appliquant sur le solide ont été analysées en se basant sur les algorithmes présentés dans le chapitre 3. Il semble que l'ajout d'une correction à la force de frottement visqueux et à la force de pression pour compenser l'absence de turbulence lors de la simulation du fluide permettrait de s'approcher encore plus précisément de la réalité. Néanmoins, l'ajout de tels coefficients signifie qu'il faut caractériser chaque objet à simuler, ce qui va à l'encontre de notre objectif initial, et une autre solution doit être trouvée dans le futur pour obtenir une simulation plus réaliste, comme une méthode de calcul prenant en compte les effets des turbulences ou l'utilisation d'un solveur SPH multi-résolution.

4.5 Discussions

4.5.1 Étude de la convergence de l'algorithme NIVF

Pour analyser le résultat de l'algorithme NIVF, une étude de l'évolution de la valeur de la fonction de coût, comme celle de l'étude précédente du balancier hydrodynamique (voir section 2.5) est réalisée. Dans ce but, une trajectoire est simulée (Équation 4.4) avec un certain set de paramètres $\xi = [m_a, K_l, C_d]$. La fonction de coût est ensuite calculée à partir de cette trajectoire simulée et de l'essai réalisé avec l'algorithme SPH avec la sphère de diamètre $\varnothing 0,4$ et avec la discrétisation $\delta_r = 0.025$. Dans un premier temps, l'estimation de la masse ajoutée sera comparée avec un seul type de frottement, soit linéaire, soit quadratique.

Sur la figure 4.18, l'évolution de la fonction de coût est observée avec le modèle de frottement L, c'est à dire en choisissant $C_d = 0$. Sur la figure de droite, toute la surface correspondant à la valeur du coût en fonction des paramètres m_a et K_l est représentée. Sur la figure du gauche, la zone où la fonction de coût est inférieure à 0.75 est représentée. Pour rappel, la fonction de coût est définie dans l'équation 4.7. La moyenne du carré des écarts $\overline{\Delta v_x^2}$ peut être calculée ainsi :

$$\overline{\Delta v_x^2} = \frac{\overrightarrow{\Delta v_x} \cdot \overrightarrow{\Delta v_x}^T}{n_{mes}} \quad (4.19)$$

Avec n_{mes} le nombre de points de mesure. En remplaçant $\overrightarrow{\Delta v_x} \cdot \overrightarrow{\Delta v_x}^T$ avec l'expression de la fonction J (équation 4.7) on obtient :

$$\overline{\Delta v_x^2} = \frac{J_{max} \left(\overrightarrow{v_x^m} \cdot \left(\overrightarrow{v_x^m} \right)^T \right)}{G_{\Delta} n_{mes}} \quad (4.20)$$

Avec J_{max} le coût pour lequel on souhaite calculer l'imprécision. Ce coût est noté J_{max} pour mettre en évidence qu'il servira de valeur maximale admise et ainsi sélectionner un ensemble de valeurs de paramètres estimés. On prendra ensuite la valeur absolue de l'erreur en faisant l'approximation suivante :

$$\overline{\Delta v_x} \approx \sqrt{\overline{\Delta v_x^2}} \quad (4.21)$$

L'application numérique donne : $\overline{\Delta v_x} = 0.008 \text{ m.s}^{-1}$. Au regard de la vitesse maximale atteinte, cela veut dire qu'en moyenne il y a entre 0.4 et 1% d'erreur sur la trajectoire.

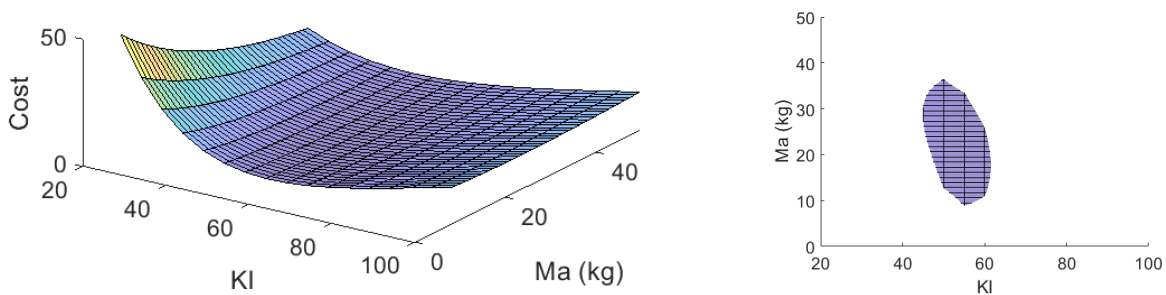


Figure 4.18 – À gauche : Évolution de la fonction de coût de l'algorithme NIVF avec une sphère de $\varnothing 0,4$, avec la discrétisation $\delta_r = 0.025$ et le modèle de frottement L À droite : Zone où cette fonction de coût est inférieure à 0.75.

Sur la figure 4.18, on constate que l'algorithme peut converger assez facilement vers la valeur du coefficient de frottement car la zone où la fonction de coût est inférieure à 0.75 est assez resserrée et la valeur varie entre 45 et 60, soit un écart de 20% autour de la valeur moyenne pour K_f . En revanche, la valeur de la masse ajoutée est difficile à estimer avec cette fonction de coût, car de très nombreuses valeurs de masse ajoutée correspondent à un coût très faible et l'écart autour de la valeur moyenne de m_a est supérieur à 50%. Un minimum local existe bel et bien, comme le montre la figure 4.19, ce qui permet à l'algorithme de converger. La valeur estimée reste néanmoins peu fiable et ceci rappelle l'intérêt de l'expérience du balancier pour la détermination de la masse ajoutée d'un objet. Dans le cas du balancier, la force de masse ajoutée était dissociable des forces de frottement grâce à l'utilisation de la période d'oscillation et de la phase qui ne dépendent que de l'inertie. La réalisation d'une telle expérience est toutefois beaucoup plus complexe avec une simulation SPH car elle nécessite de modéliser la liaison pivot de l'axe du balancier, et de minimiser tous les effets de dissipation autour de cette liaison virtuelle.

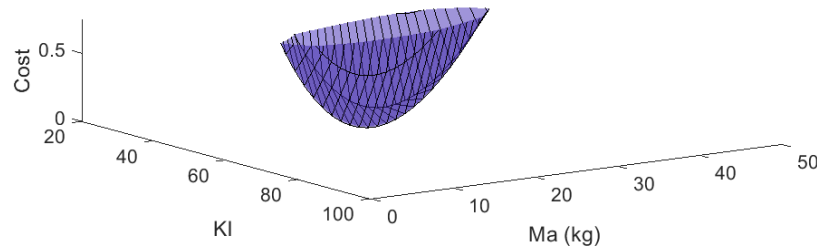


Figure 4.19 – Evolution de la fonction de coût de l'algorithme NIVF avec une sphère de $\varnothing 0,4$, avec la discrétisation $\delta_r = 0.025$ et le modèle de frottement L dans la zone où elle est inférieure à 0.75

Les forces mises en jeu lors de la simulation sont du même ordre de grandeur, comme en atteste la figure 4.20. Sur cette figure, l'évolution de la force générée par la masse ajoutée au cours du temps est en rouge et l'évolution de la force de frottement est en bleu. D'après

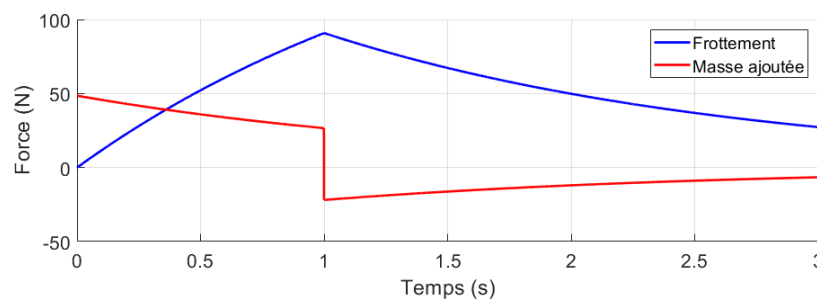


Figure 4.20 – Forces mises en jeu pour un essai avec une sphère de $\varnothing 0,4$ et avec la discrétisation $\delta_r = 0.025$ en considérant un modèle de frottement linéaire.

l'évolution de ces forces, il n'est pas possible de considérer que la différence d'ordre de grandeur de celle-ci est problématique pour l'estimation dans le cas où l'on se sert de l'essai complet. Par contre, lors de l'étude de l'influence de la vitesse, il est possible que la plage de données où les vitesses sont les plus faibles, c'est à dire au début de l'essai, ait conduit à des résultats approximatifs, la vitesse et donc la force de frottement étant nulle au départ.

La même étude est réalisée avec le modèle de frottement Q, et dans ce cas, le coefficient K_l est nul. L'essai utilisé est toujours celui avec la discrétisation $\delta_r = 0.025$. La figure 4.21 montre l'évolution de la fonction de coût avec le modèle Q. Sur cette figure, à gauche, le même focus sur les valeurs pour lesquelles le coût est inférieur à 0.75 est présenté.

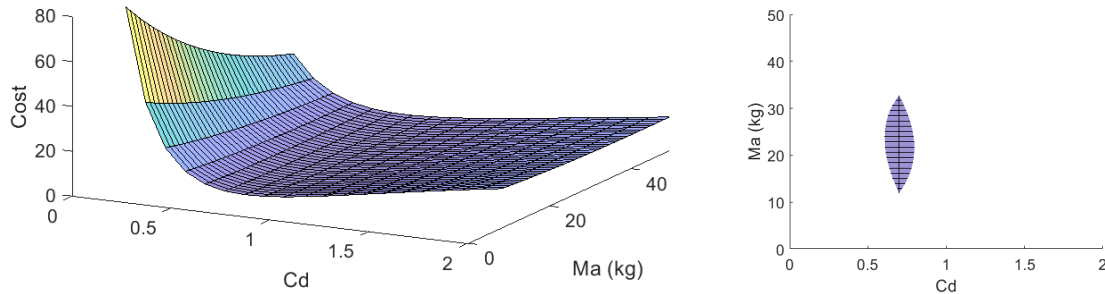


Figure 4.21 – À gauche : Évolution de la fonction de coût de l’algorithme NIVF avec une sphère de $\varnothing 0,4$, la discrétisation $\delta_r = 0.025$ et le modèle Q
À droite : Zone où cette fonction de coût est inférieure à 0.75.

Le même constat que celui fait avec le modèle L peut être tiré de cette figure : la valeur du coefficient C_d estimée est assez précise et varie entre 0.6 et 0.8, mais la valeur de la masse ajoutée affecte moins la trajectoire de la sphère, ce qui rend son estimation moins précise.

Pour vérifier la pertinence de l'utilisation du modèle de frottement LQ et pour choisir le modèle de frottement correspondant le plus précisément à ce qui est observé au cours des simulations, la valeur de la fonction de coût est calculée en choisissant une masse ajoutée constante et en faisant varier les paramètres K_l et C_d . Le cas de la sphère de diamètre $\varnothing 0,4$ en utilisant un δ_r de 0.025 est étudié. Sur la figure 4.22, on peut voir l'évolution de ces deux paramètres, ainsi qu'un focus sur la zone où la fonction de coût est inférieure à 0.75. Ces résultats ont été obtenus pour une masse ajoutée égale à 21, ce qui correspond approximativement à la masse ajoutée estimée d'après le tableau 4.3.

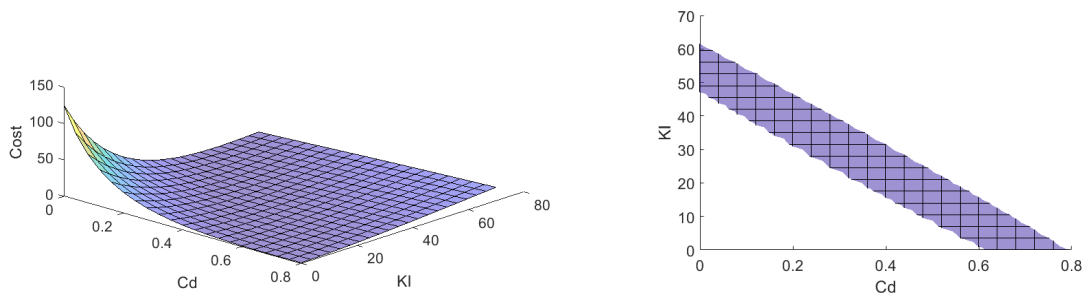


Figure 4.22 – À gauche : Évolution de la fonction de coût de l’algorithme NIVF avec une sphère de $\varnothing 0,4$, la discrétisation $\delta_r = 0.025$ et le modèle LQ
À droite : Zone où cette fonction de coût est inférieure à 0.75.

On constate sur cette figure que la valeur de la fonction de coût reste très faible pour une grande plage de valeur de K_l et de C_d et qu'il est difficile de choisir entre un modèle linéaire et un modèle quadratique. La valeur minimale du coût est atteinte pour $C_d = 0$, ce qui veut dire que le modèle linéaire représente mieux la dynamique de la sphère simulée. Les frottements linéaires semblent donc mieux correspondre à ce qui est simulée avec l'algorithme

ISPH. Les résultats obtenus avec chacun des deux modèles de frottements ont été présentés dans la section [4.4](#).

4.5.2 Convergence de l'algorithme d'estimation liée aux procédures expérimentales

Ce paragraphe permet de montrer l'importance de la procédure expérimentale choisie dans ce chapitre, c'est à dire l'importance de la présence d'une force extérieure connue pendant tout ou partie de l'expérience conduite pour l'estimation des paramètres hydrodynamiques a posteriori. Deux cas sont considérés :

- **Cas d'un mouvement actionné.** Une force extérieure connue agit en plus des forces hydrodynamiques (masse ajoutée et frottement) sur le solide et peut être prise en compte dans l'estimation des paramètres hydrodynamiques. Ce cas correspond à celui étudié jusqu'à présent dans ce chapitre. Pour rappel, dans ce cas le modèle dynamique de la sphère est (équation 4.4) :

$$F_{Input} - K_l v_x - K_q v_x |v_x| = (m + m_a) a_x$$

- **Cas d'un mouvement libre.** Aucune autre force que les forces hydrodynamiques n'agit sur le solide. Pour étudier ce cas, les données de trajectoire simulées utilisées jusqu'à présent sont réutilisées. Mais seules les données après 1 s de simulation sont prises en compte, c'est à dire lorsque la décélération libre de la sphère est observée. Le modèle dynamique de la sphère dans ce cas est :

$$-K_l v_x - K_q v_x |v_x| = (m + m_a) a_x \quad (4.22)$$

Avec ce modèle, il est possible de lier directement les paramètres en fonction du ratio vitesse et accélération, et il n'y a plus de valeurs de force de référence pour l'estimation de paramètre. Dans le cas du modèle linéaire, on a :

$$-\frac{K_l}{m + m_a} = \frac{a_x}{v_x} \quad (4.23)$$

La figure 4.23 montre l'évolution de la fonction de coût dans le cas d'un mouvement libre, de la même manière que sur la figure 4.18, c'est à dire en considérant le modèle de frottement L et l'essai avec une sphère de $\varnothing 0,4$ et la discrétisation $\delta_r = 0.025$. L'algorithme NIVF

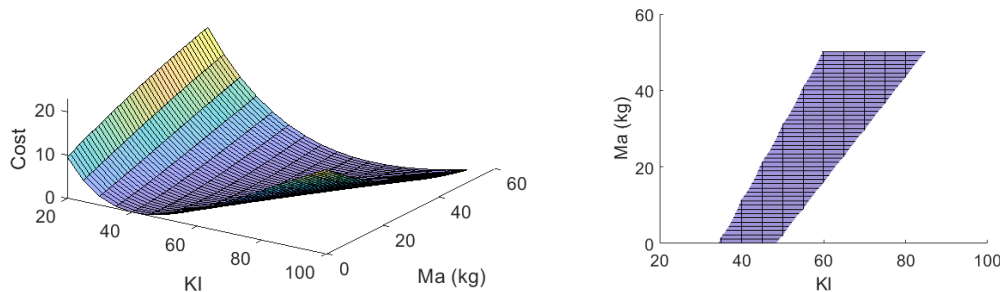


Figure 4.23 – À gauche : Évolution de la fonction de coût de l'algorithme NIVF avec une sphère de $\varnothing 0,4$ pour un essai de mouvement libre, avec la discrétisation $\delta_r = 0.025$ et le modèle de frottement L

À droite : Zone où cette fonction de coût est inférieure à 0.75.

converge pour toutes les valeurs de masse ajoutée testées et, comme le suggère l'équation 4.23,

on peut imaginer qu'il existe une infinité de points de convergence (donc de couples $[m_a, K_l]$) pour l'algorithme d'estimation.

Ceci met en évidence que l'utilisation d'essais où le mouvement du véhicule est libre dans l'eau ne permet pas d'estimer correctement le couple masse ajoutée et coefficient de frottement. Avec des essais où le mouvement est libre, il est nécessaire de connaître a priori la masse ajoutée ou le coefficient de frottement du véhicule pour déterminer l'autre paramètre. Pour les véhicules sous-marins, cela implique que l'on peut obtenir les paramètres hydrodynamiques du véhicule en le laissant évoluer librement dans la direction de pilonnement (heave), puisque l'on peut connaître la poussée d'Archimède, mais pas dans les autres directions. Pour les autres directions, il est possible d'utiliser le balancier hydrodynamique décrit au chapitre 2 ou un autre mécanisme permettant d'actionner l'objet avec une force connue.

4.5.3 Caractère homothétique des paramètres hydrodynamiques simulés

En prévision de l'utilisation future du simulateur, la question de l'évolution des paramètres hydrodynamiques lors du changement de la dimension d'un objet se pose. Le ratio entre la masse ajoutée et les coefficients de frottement obtenus entre deux sphères de diamètre différent est donc analysé. Les résultats obtenus avec les sphères de diamètre $\varnothing 0,3 m$, $\varnothing 0,4 m$ et $\varnothing 0,5 m$ présentés dans le tableau 4.6 sont comparés. Le tableau 4.13 présente les ratios entre les masses ajoutées obtenues avec les différentes tailles de sphère pour chaque niveau de discrétisation. Par exemple, pour $\delta_r = 0.05$, le ratio affiché sur la ligne 0.3/0.4 du tableau est le suivant :

$$\text{Ratio} = \frac{m_{a,\varnothing 0,3}}{m_{a,\varnothing 0,4}} \quad (4.24)$$

Ratio \varnothing	$\delta_r = 0.05$	$\delta_r = 0.025$	$\delta_r = 0.0125$	Référence
0.3/0.4	0.43	0.40	0.43	0.42
0.3/0.5	0.21	0.20	0.20	0.22
0.4/0.5	0.50	0.49	0.46	0.51

Tableau 4.13 – Ratio entre les masses ajoutées obtenues

L'évolution de la masse ajoutée avec la taille des sphères est bien homothétique à l'évolution théorique de la masse ajoutée. Ce résultat est important car il veut dire que même si la masse ajoutée simulée ne correspond pas à la masse ajoutée réelle, celle-ci évolue correctement avec la géométrie du solide. Dans le cas des forces de frottement, le coefficient étant adimensionnel, cette analyse n'a pas lieu d'être.

4.5.4 Temps de calcul

Pour les simulations où la sphère est accélérée avec une force constante pendant 1 s, pendant huit secondes de simulation, qui comprennent une première étape où la sphère est stabilisée dans le fluide, le temps de calcul avec les particules les plus grossières ($\delta_r = 0.05$) se situe entre 4300 et 4500 s (1,5 h), avec 190 000 particules. Pour la discrétisation moyenne ($\delta_r = 0.025$), le temps de calcul se situe entre 70 000 et 80 000 s (20 h), avec environ 1 500

000 particules. Pour la discrétisation fine ($\delta_r = 0.0125$), le temps de calcul se situe entre 480 000 et 500 000 s (presque 6 jours), avec environ 8 500 000 particules. Aussi satisfaisant soit le résultat d'une simulation du comportement de la sphère avec des particules très fines, actuellement, pour l'application souhaitée du simulateur SPH et pour ce niveau de finesse, les calculs sont trop longs. Pour ces essais, le nombre d'itération moyen du solveur ISPH par pas de temps était de 52 pour $\delta_r = 0.05$, 49 pour $\delta_r = 0.025$ et 37 pour $\delta_r = 0.0125$. Cette évolution est probablement due à la précision attendue de la densité dans la convergence du solveur qui est plus simple à atteindre avec des particules plus petites (voir Annexe E). Réduire le nombre d'itération requise avec un solveur de pression plus efficace permettrait également d'accélérer le temps de calcul.

Dans le cas de la simulation où l'obtention d'une vitesse limite a été recherchée, en ce qui concerne la simulation utilisant les particules de fluide de taille $\delta_r = 0.05 m$, le nombre de particules était de 559 997. Pour 26 s simulées, le temps de calcul fut de 48 241 s, ce qui représente 13,4 h. En comparaison, pour la simulation utilisant des particules de fluide de taille $\delta_r = 0.025 m$, le nombre de particules était de 4 479 960. Pour 26 s simulées, le temps de calcul fut de 735 193 s, ce qui représente 8 jours et demi de calculs.

Les temps de calculs requis sont considérables dans le cas où l'on choisit la discrétisation la plus fine ou dans le cas où la zone à simuler est grande, même si les résultats avec ce niveau de discrétisation sont très bons. Il n'est pas envisageable de concevoir un simulateur qui utiliserait ce niveau de discrétisation pour des simulations de robotique plus complexe que celles menées ici. Le temps de calcul est relié au nombre de particules à simuler. Ainsi, simuler le comportement d'un grand volume de fluide est assez difficile et l'utilisation de solutions multi-échelle ou multi-résolution semble la meilleure option possible puisque ces solutions permettent de diminuer drastiquement le nombre de particules.

4.6 Conclusion du chapitre

Ce chapitre a montré, grâce à différents résultats d'essais, que l'algorithme de simulation ISPH développé au cours de cette thèse est capable de simuler des phénomènes dynamiques. La poussée d'Archimède est correctement simulée par le solveur ISPH. Toutefois, cette force est simulée avec une marge d'erreur et dans le cas où la flottabilité de l'objet est neutre, cela devient problématique. La démonstration de la capacité du solveur ISPH à obtenir une vitesse limite avec un objet traversant un champ de particules fluide constitue la quatrième contribution de cette thèse. La classe de méthodes NIF, dont la méthode NIPF avait été introduite au chapitre 2, a été présentée ici. La convergence des algorithmes NIF a été validée. L'algorithme NIVF basé uniquement sur la vitesse a été utilisé dans ce chapitre puisqu'il évitait l'utilisation d'une double intégration numérique. Les paramètres estimés sont de plus en plus précis lorsque l'on diminue la taille des particules. La masse ajoutée simulée est très réaliste et sa valeur est obtenue avec une erreur maximale de 40% pour les simulations avec une discrétisation grossière et 28% pour la discrétisation la plus fine. De plus, cette masse ajoutée évolue de manière homothétique avec la masse ajoutée théorique lorsque l'on modifie la taille de l'objet étudié. Par contre, les frottements simulés sont plus éloignés des frottements réels. Ils sont plus proches d'un modèle de frottement linéaire que d'un modèle quadratique, ce que l'on peut expliquer par l'absence de modèle de turbulence, qui est notamment problématique lors de la simulation avec une discrétisation grossière du fluide. De nombreuses pistes d'améliorations existent pour nos algorithmes, comme par exemple, l'utilisation d'une taille de particule variable qui permettrait de modéliser plus fidèlement la frontière fluide-solide tout en minimisant le temps de calcul nécessaire pour simuler le comportement du fluide loin du solide. Il serait également possible d'ajouter un modèle de turbulence au calcul de fluide pour rendre celui-ci plus réaliste. L'utilisation de la méthode NIF sur une simulation SPH et l'analyse des résultats menant aux futures améliorations à apporter à ce solveur constituent la cinquième contributions de cette thèse. Une possibilité de correction des forces provenant de la résolution du fluide SPH afin d'obtenir une trajectoire plus proche d'une trajectoire de référence a été présentée. Mais cette correction va à l'encontre de l'objectif de notre recherche qui est de simuler la dynamique d'un véhicule marin et du fluide qui l'entoure pour éviter une pré-détermination des paramètres hydrodynamiques du véhicule pour la simulation. Dans ce chapitre, la plupart des essais ont été menés avec une force unique et constante pour mettre en mouvement l'objet, l'utilisation d'une force variable ou de plusieurs valeurs de force et l'observation des variations induites pourrait être une perspective de recherche intéressante. Le simulateur SPH conçu remet en question l'utilisation de coefficients de frottement constants pour la simulation ou pour le contrôle dynamique lors de missions avec un véhicule sous-marin qui, au regard de nos résultats, n'est fiable que pour des trajectoires uniformes, sans changement de vitesses. L'utilisation de cet algorithme pourrait également permettre de mieux comprendre les phénomènes se déroulant lors de missions en milieu très perturbé, comme la zone de surf. Il permet notamment de simuler les effets des vagues et de la houle sur l'écoulement autour du véhicule, comme expliqué au chapitre 3, et permettra donc de simuler ces effets sur le véhicule.

Chapitre 5

Conclusion générale

Cette thèse s'est déroulée au sein du laboratoire COSMER de l'université de Toulon. Elle faisait partie du projet SURFBOT qui avait pour objectif de développer un véhicule sous-marin à faible coût capable de naviguer dans la zone de surf. Pour ce projet, cette thèse devait permettre le développement d'une solution permettant d'évaluer le comportement dynamique de véhicules sous-marins dans un environnement aquatique susceptible d'être perturbé. Les développements réalisés dans le cadre de cette thèse apportent une première contribution à cet objectif en proposant un outil de simulation basé sur une modélisation particulière du fluide. La nouvelle approche pour la simulation du véhicule proposée dans cette thèse repose sur l'utilisation de la méthode Smoothed Particles Hydrodynamics afin de simuler en même temps la dynamique du fluide et celle du solide, contrairement aux simulateurs classiques qui ne simulent que le solide. L'objectif de cette thèse a été de montrer qu'il est possible d'utiliser cette méthode pour simuler la dynamique d'objets immergés, donc de véhicules sous-marins, et notamment de permettre de déterminer leurs paramètres hydrodynamiques. Les résultats obtenus au cours de cette thèse sont résumés dans la prochaine section.

5.1 Bilan des contributions apportées

La navigation des véhicules sous-marins dans la zone de surf représente un objectif de recherche à long terme. Elle sera rendue possible si l'on parvient à doter le véhicule de capacités à affronter dans une certaine mesure les perturbations pouvant survenir dans cette zone. Les perturbations (vagues et courants) ainsi que le design des véhicules actuels ont été caractérisés dans cette thèse. Pour concevoir un véhicule qui sera capable de naviguer dans cette zone, il faut concevoir un outil qui permette d'évaluer correctement le comportement dynamique des véhicules dans cette zone. Cet outil sera utile, par exemple, pour tester des algorithmes de commandes ou pour aider à la conception de véhicules reconfigurables. À partir d'une étude des différents environnements de simulation existants et au regard des variations possibles des perturbations que peut subir un véhicule dans la zone de surf, il a été établi que la méthode utilisée dans les simulateurs de robotique (Gazebo, UWSim, Vortex, etc.) classiques ne permettait pas de mettre en évidence tous les phénomènes pouvant se produire. En effet, pour cette zone particulière, il est intéressant de modéliser le comportement du fluide en même temps que celui du solide. Pour cela, deux possibilités existent pour modéliser un volume de fluide : représenter ce volume avec un maillage ou avec un ensemble de particules. Cette deuxième possibilité est très intéressante, car elle est particulièrement adaptée à la simulation d'interface mobile, comme par exemple la surface de l'eau ou la surface du véhicule immergé.

SPH est une méthode particulièrement populaire de simulation de fluide avec des particules. Cette méthode a déjà été utilisée pour différentes applications, et notamment, dans le domaine de l'ingénierie marine pour analyser certains phénomènes d'écoulement hors ligne, et dans le domaine de l'animation et des jeux vidéo, où elle est utilisée en temps réel mais au prix d'approximations sur la réalité physique. Pour utiliser cette méthode, il a été tout d'abord nécessaire de comprendre et de caractériser les effets de l'eau sur le véhicule. Dans cette optique, une expérience avec un balancier hydrodynamique a été réalisée. La sphère est l'objet d'étude utilisé tout au long de cette thèse, car c'est la forme la plus étudiée jusqu'à aujourd'hui. Une nouvelle méthode d'estimation des paramètres hydrodynamiques baptisée Numerical Integration Position Fitting, a été développée pour l'estimation des paramètres avec l'expérience du balancier. Ensuite, SPH a été présentée avec différentes méthodes utilisables pour la modélisation de l'écoulement. Cette méthode a ensuite été étendue à une classe de méthode nommée Numerical Integration Fitting. La méthode Incompressible SPH couplée avec la méthode d'extrapolation de la pression à la frontière des objets solides a montré qu'il était possible de réaliser une simulation dynamique d'un objet immergé. Le caractère homothétique des simulations a été observé, en particulier, à travers une expérience de vitesse maximale due aux frottements. Les résultats obtenus avec le balancier hydrodynamique et les résultats de simulation SPH ont été comparés, cependant ils ne correspondent pas. Cette différence pourrait provenir du fait que l'écoulement simulé avec SPH ne rend pas bien compte des turbulences pouvant apparaître dans le fluide, ce qui nécessiterait de futures études.

5.1.1 Nouvelle méthode d'estimation des paramètres hydrodynamiques

Les expériences hydrodynamiques sont complexes à analyser car de nombreux phénomènes simultanés se produisent dans le fluide. Les forces hydrodynamiques sont décomposées en

masse ajoutée et force de frottements qu'il faut estimer à partir d'une seule trajectoire. La masse ajoutée jouant sur l'inertie du pendule, il est possible de la dissocier des forces de frottements en utilisant la méthode NIPF qui a été introduite au chapitre 2 et étendue à une classe de méthode, nommée NIF, au chapitre 4. Les résultats obtenus montrent en revanche qu'il est difficile de différencier les forces de frottements linéaires des forces de frottements quadratiques. Ceci peut provenir, soit des données expérimentales, soit du fait que ces coefficients évoluent en fonction de la vitesse ou de l'accélération au cours d'un essai. La procédure expérimentale est déterminante pour l'obtention de résultats corrects, ce qui a été montré dans la discussion (§ 4.5.2), et celle-ci doit donc être choisie avec soin pour obtenir des paramètres réalistes. Les paramètres de frottements estimés dépendent fortement de la vitesse mesurée. Ainsi, si l'incertitude de mesure est trop grande et si le signal doit être post-traité et notamment filtré, les paramètres de frottement estimés peuvent être fortement impactés. Dans le cas d'une simulation, des résultats d'estimation non cohérent pourrait être obtenu si celle-ci n'est pas réaliste. La pertinence de l'utilisation simultanée des frottements linéaires et quadratiques a été discutée dans cette thèse. Les forces de frottements, qui peuvent être quadratiques, c'est à dire proportionnelles à la vitesse au carré, ou linéaires, c'est à dire proportionnelles à la vitesse, sont liées au nombre de Reynolds qui caractérise l'écoulement. Avec des objets de la taille des véhicules sous-marins compacts actuels, dont l'ordre de grandeur est d'environ $0.15 m^3$ ($0.04 m^3$ pour le BlueROV 2 et $0.26 m^3$ pour le Sparus 2), ce nombre de Reynolds est suffisamment grand pour que les frottements soient quadratiques, même pour des vitesses de l'ordre de $0.05 m.s^{-1}$. On peut alors se poser la question de l'utilisation courante d'une combinaison entre frottements linéaire et quadratique pour modéliser la dynamique des véhicules sous-marins. De plus, ces coefficients sont variables, car ils dépendent de la vitesse et de l'orientation de l'écoulement (voir chapitre 2). Ces arguments renforcent la pertinence de l'utilisation d'une méthode capable de simuler le fluide pour la navigation en environnement perturbé.

5.1.2 Solution adaptée pour la simulation de véhicules marins avec la méthode Smoothed Particles Hydrodynamics

La méthode SPH est une méthode de modélisation de la matière du fluide, mais cette méthode seule ne caractérise par la résolution de l'écoulement. Les algorithmes SPH peuvent être séparés en trois différentes classes : Compressible SPH, Weakly Compressible SPH et Incompressible SPH. Compressible SPH ne concerne pas la mécanique des liquides sur Terre. Weakly Compressible SPH et Incompressible SPH sont de bons candidats pour la modélisation des écoulements autour des solides. Ces méthodes ne concernent que le calcul du fluide et non la modélisation de la frontière solide. La frontière des solides est modélisée avec des particules fictives placées à l'extérieur du volume censé représenter le fluide. Pour les calculs hydrodynamiques, deux méthodes ont été comparées dans cette thèse : la méthode d'extrapolation et la méthode de réflexion. La première méthode consiste à extrapoler, à partir des valeurs du fluide, une valeur pour chaque particule fictive représentant la surface du solide. La seconde méthode consiste à ne pas calculer de quantité pour les particules de la frontière fluide-solide et à simuler une réflexion de la pression par la surface du solide. Cette seconde méthode, bien que légèrement moins coûteuse en temps de calcul, se révèle bien trop imprécise.

Les travaux de cette thèse ont permis de montrer que le solveur Weakly Compressible SPH explicite utilisé n'est pas adapté à la simulation temps réel de phénomènes dynamiques.

Premièrement, l'utilisation de ce solveur nécessite d'utiliser des pas de temps extrêmement courts, ce qui augmente le temps de calcul. Deuxièmement, malgré la précision de la simulation dans un cas statique, la simulation de phénomènes dynamiques n'est pas très stable car le déplacement des ondes de choc n'est pas résolu à travers un solveur. Toute discontinuité entraîne de grandes imprécisions avec le solveur WCSPH. Le solveur ISPH, par contre, permet des calculs relativement rapides, notamment avec des scènes de petite dimension ou faiblement discrétisées. Malgré le nombre d'itérations requises par le solveur au cours d'un pas de temps, leur durée permet finalement un calcul plus rapide.

5.1.3 Viabilité du solveur ISPH

Premièrement, la capacité du solveur ISPH à simuler la poussée d'Archimède a été analysée. Les résultats montrent que l'on peut la simuler de manière fiable pour des solides ayant une densité éloignée de celle du fluide dans lequel ils évoluent. Pour les objets dont la flottabilité est parfaitement neutre, le solveur ISPH ne sera pas capable de restituer la valeur de la poussée d'Archimède avec une précision suffisante pour simuler leur dynamique de manière réaliste. Cette flottaison neutre parfaite peut toutefois être simulée si l'on supprime toutes les forces s'appliquant selon l'axe vertical sur le solide. Le solveur ISPH couplé à la méthode d'extrapolation offre la possibilité de simuler une vitesse limite due aux forces de frottements. Peu importe le niveau de discrétisation, une vitesse limite est atteinte pour la sphère. Celle-ci se rapproche de la vitesse théorique lorsque la discrétisation s'affine. L'estimation des paramètres hydrodynamiques a été obtenue grâce à la classe d'algorithme Numerical Integration Fitting et notamment l'algorithme Numerical Integration Velocity Fitting. La masse ajoutée simulée reste assez proche de la masse ajoutée réelle, et, les variations de masse ajoutée observées sont homothétiques avec les variations réelles de masse ajoutée (§ 4.5.3). Les simulations SPH pourraient donc être utilisées pour calibrer la masse ajoutée d'un véhicule sous-marin. Les frottements simulés sont plus proches de frottements linéaires que de frottements quadratiques. Une correction pour simuler les coefficients hydrodynamiques réels a été proposée, néanmoins, cette solution n'est pas satisfaisante si l'on souhaite se servir de l'algorithme SPH pour simuler le comportement d'un véhicule sous-marin sans avoir à entrer d'autres paramètres que sa forme géométrique et sa densité.

5.2 Perspectives

L'utilisation d'un simulateur qui couple la dynamique du fluide avec celle du solide est prometteuse pour l'étude de la dynamique des véhicules marins. Cette section présente plusieurs pistes d'amélioration du solveur SPH, un essai avec une forme différente de celle de la sphère et des pistes pour l'utilisation future de cette technique de simulation dans le domaine de la robotique marine et sous-marine. La fin de cette section souligne l'importance des paramètres hydrodynamiques pour la recherche en robotique sous-marine ainsi que les perspectives de recherche dans le domaine de l'estimation de paramètres hydrodynamiques.

5.2.1 Optimisation des méthodes pour le solveur ISPH

Le solveur ISPH présenté dans cette thèse pourrait être grandement amélioré grâce à certaines techniques récentes. Par exemple, l'utilisation d'un système multi-résolution pour la taille des particules qui s'adapte à la simulation en cours pourrait permettre de simuler des zones bien plus grandes, tout en conservant, voir en améliorant la précision des forces hydrodynamiques s'appliquant sur l'objet immergé. Des algorithmes multi-résolution existent, comme par exemple celui de Gonnet [GSTC], celui de Hu [HPR⁺17], ou encore celui de Sun [SCM⁺18]. Tous les algorithmes existants à ce jour sont adaptés à un algorithme faiblement compressible, comme l'algorithme WCSPH que nous avons présenté au chapitre 3, mais pas à un algorithme ISPH. Le principe est illustré sur la figure 5.1 Cet algorithme multi-résolution

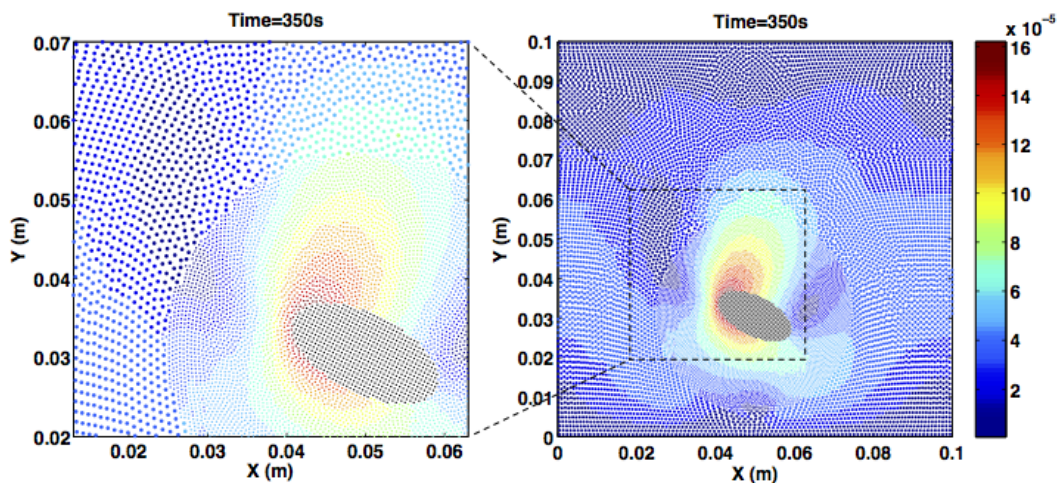


Figure 5.1 – Simulation avec un algorithme multi-résolution. Les particules sont coloriées en fonction de leur vitesse. (Source : Hu [HPR⁺17])

permettrait d'adapter la taille des particules en fonction de la vorticit  du fluide, en fonction de la proximit  de la particule avec une fronti re solide. Mais pour cela, il faut r soudre plusieurs probl mes :

- Ces algorithmes impliquent une masse variable des particules ; il faut pouvoir en tenir compte dans les calculs de la dynamique des fluides et donc revoir toute la construction de l'algorithme ISPH.
- Une m thode qui permette de d terminer la vorticit  locale du fluide au niveau de chaque particule est   choisir.

- Des limites de variations de la taille des particules dans un espace donné sont à définir pour éviter que des particules énormes ne se retrouvent à côté de particules minuscules.
- Dans le cas particulier de solides en mouvement ou de solides déformables, le mouvement des particules du solide sera à prendre en compte lors de la nouvelle discrétisation de l'espace.

En plus de cet algorithme multi-résolution, un algorithme capable de prendre en compte la vorticit  du fluide pourrait permettre de prendre en compte les effets des turbulences dans la r solution de la dynamique de l' coulement   la mani re des solveurs RANS (voir chapitre 2). Bien qu' voqu e dans la litt rature, comme par exemple, dans l'article traitant d'animation 3D de Peer [PT17], o  la vorticit  est utilis e pour simuler des fluides tr s visqueux, ou dans Jang [JiRBN11], o  elle est prise en compte via une m thode hybride avec un algorithme SPH coupl    un mod le eul rien, tel qu'illustr  sur la figure 5.2, elle ne semble pas avoir  t 

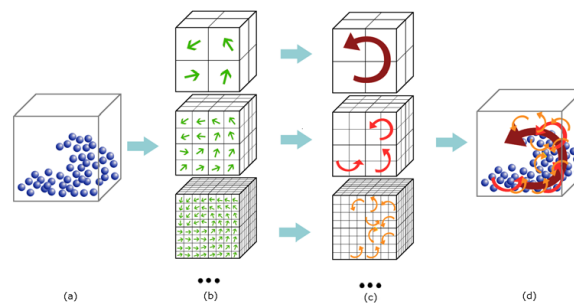


Figure 5.2 – Principe de la m thode de calcul de vorticit  de Jang (Source : [JiRBN11])

prise en compte dans un cas o  l'on utilise uniquement un algorithme SPH en cherchant   obtenir une interaction entre un fluide et le solide.

Pour diminuer le temps de calcul, il serait  galement possible de modifier le code source de l'algorithme ISPH qui est actuellement cod  sous forme de tableaux de structures, mais qui serait plus rapidement ex cut  s'il  tait cod  sous forme de structures de tableaux. Ce changement se r v lerait  galement n cessaire pour pouvoir utiliser les cartes graphiques (GPU).

5.2.2 Param tres hydrodynamiques d'une forme de capsule

Dans cette th se, la forme  tudi e a  t  celle d'une sph re, notamment parce que les param tres hydrodynamiques de r f rence pour la sph re ont  t  bien  tudi s et sont connus. Cependant, seuls quelques v hicules sous-marins poss dent cette forme et ceux-ci sont notamment des v hicules du domaine de la recherche comme par exemple SUR-II et SUR-III de l'universit  de Kagawa ([YGS13] & [GG17]) ou le v hicule du projet UNEXMIN ([FRM⁺18]). Ainsi, il faut v rifier que des param tres hydrodynamiques peuvent  tre obtenus avec d'autres formes, plus proches de v hicules sous-marins r els. Comme d crit dans le chapitre d'introduction (1), la forme la plus courante pour les AUV est une forme de capsule.

Un essai suppl mentaire de mouvement de cavement (Section 2.1.1) a donc  t  r alis  avec cette forme pour d montrer que le simulateur SPH d velopp  ici permet de simuler d'autres formes que la sph re. Malgr  une recherche approfondie dans la litt rature, il n'a pas  t  possible de trouver des coefficients de frottement de r f rence (voir chapitre 2). Les dimensions d'un AUV existant, le SPARUS II, ont  t  utilis es pour mod liser la capsule   simuler. La capsule poss de un diam tre de $\varnothing 0,23\text{ m}$ et une longueur de $1,6\text{ m}$. La densit 

simulée pour le véhicule était de 1150 kg.m^{-3} ce qui devrait permettre d'obtenir une poussée d'Archimède acceptable lors de la coulée initiale du véhicule et qui confère au SPARUS simulé une masse de $72,8 \text{ kg}$ contre 52 kg pour le véhicule réel. L'estimation des paramètres hydrodynamiques est néanmoins indépendante de la masse et dépend plutôt de la forme du véhicule. Une masse simulée proche de celle réelle permet de s'assurer qu'il n'y a pas trop d'incertitude lors de l'estimation de la masse ajoutée. Dans la direction de cavalement, la masse ajoutée est supposée égale à une sphère du diamètre des hémisphères de la capsule, en considérant que l'écoulement autour de la capsule serait similaire à celui de la sphère à l'avant et à l'arrière de la capsule et que l'écoulement potentiel autour de la partie cylindrique n'apporte aucun effort de pression sur la capsule (bien qu'il puisse y avoir des frottements qui eux génèrent une force à relier à la vitesse). Les paramètres utilisés pour le solveur SPH sont les mêmes que ceux décrits dans le tableau 4.2. La taille des particules est $\delta_r = 0.025$ ce qui représente un bon compromis entre temps de calcul et précision du résultat. La figure 5.3 montre la scène de simulation. La procédure de simulation est similaire à la procédure de simulation de la sphère. La capsule est tout d'abord immergée (jusqu'à atteindre une profondeur de 0.5 m), puis la simulation du fluide est stabilisée pendant 3 s , puis la capsule est accélérée selon la direction de cavalement. Les particules bleues, vertes et jaunes montrent

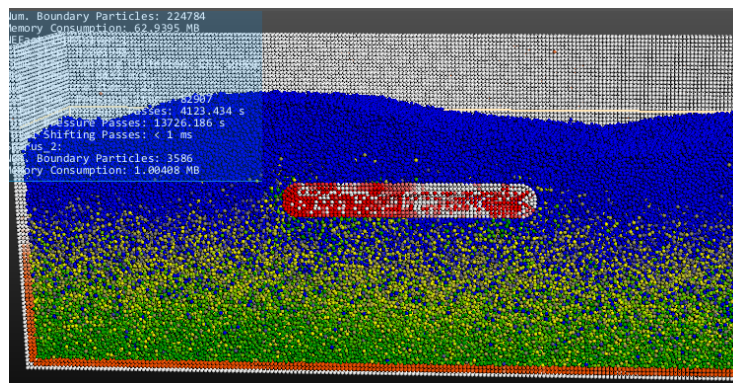


Figure 5.3 – Scène simulée avec la capsule

les particules de fluide selon leur pression. La pression sur la surface du bassin est représentée en orange et la pression à la surface de la capsule est représentée en rouge. Lors de cette simulation, les particules autour de la capsule ont réagi plus fortement au mouvement du solide que pour le cas de la sphère et des vagues ont été observées à la surface. Ceci pourrait être dû à un problème de répartition des particules sur la surface de la capsule simulée. Cette répartition doit être parfaitement symétrique pour garantir les propriétés de continuité du schéma SPH.

Le principe de l'essai avec la capsule est le même que ceux des sphères : la capsule est immergée dans un premier temps et le fluide est stabilisé, puis la capsule est accélérée dans la direction de cavalement avec une force égale à 3 fois sa masse. Les paramètres de référence utilisés pour la comparaison sont la masse ajoutée théorique et la valeur de coefficient $C_d = 1.0$, en se basant sur les résultats d'Ouchene [OKT14] évoqués dans le chapitre 2 (tableau 2.2). Les résultats d'Ouchene sont difficiles à comparer avec nos résultats puisque le ratio entre la longueur et le diamètre ($L/D=5$) utilisé par celui-ci est différent du nôtre ($L/D=7$). L'aire utilisée dans le calcul du K_q est l'aire du disque de diamètre $\varnothing 0.23$. L'évolution temporelle de la vitesse obtenue est représentée sur la figure 5.4.

La vitesse simulée de la capsule est très proche de celle de référence, ce qui peut supposer que

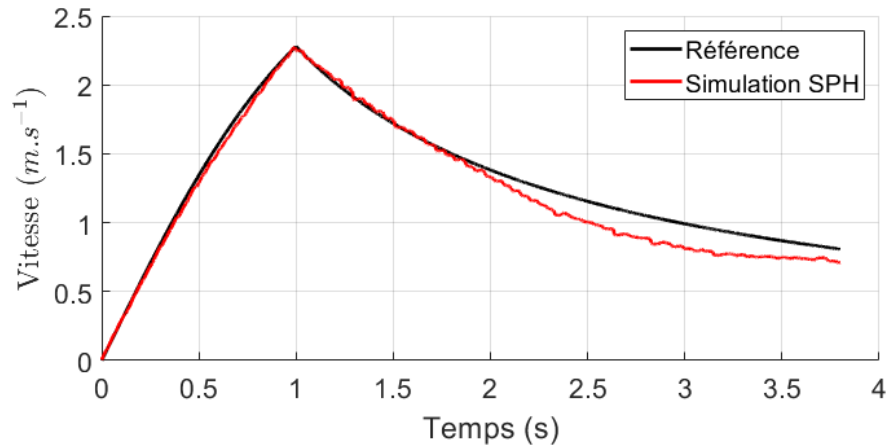


Figure 5.4 – Évolution temporelle de la vitesse de la capsule au cours du temps comparée à une trajectoire de référence

les paramètres hydrodynamiques estimés soient proches de ceux de référence. Les paramètres hydrodynamiques avec les trois modèles de frottements L, Q et LQ obtenus avec l'algorithme NIVF sont présentés dans le tableau 5.1

δ_r	Type	m_a	K_l	C_d	Coût
0.025	Référence	3.18	0	1.0	0.59
	L	3.51	36.56	0	0.17
	Q	1.91	0	1.15	0.23
	LQ	2.07	20.54	0.5	0.079

Tableau 5.1 – Résultats de l'estimation des paramètres hydrodynamiques avec une capsule de diamètre $\varnothing 0.23\text{ m}$ et de longueur $\varnothing 1.6\text{ m}$ en utilisant le modèle de frottement LQ et NIVF

On constate que les paramètres hydrodynamiques estimés sont du même ordre de grandeur que les paramètres hydrodynamiques de référence. Il n'est pas possible de véritablement confirmer la valeur du coefficient de frottement puisque celui-ci n'a pas été déterminé par une expérience réelle, ce qui constitue une piste de recherche future. La masse ajoutée est dans le pire des cas estimée avec une erreur de 40%, ce qui correspond aux erreurs observées avec la sphère. Dans cette simulation le coefficient de frottement estimé avec le modèle LQ n'est pas un frottement uniquement linéaire mais c'est un frottement qui est également quadratique ce qui est intéressant et montre peut-être qu'un plus grand nombre de turbulences ont été simulées dans le fluide comme suggéré dans la description du déroulement de la simulation (5.3).

5.2.3 Utilisation du solveur SPH en robotique

Le simulateur SPH pourrait être utilisé de diverses manières en robotique. La première utilité de ce solveur pourrait être l'étude de manœuvres réalisées par des véhicules sous-marins. La simulation de ces manœuvres pourrait, par exemple, permettre d'améliorer les algorithmes de commande du véhicule marin ou encore de comprendre quels effets ce véhicule a sur son environnement et sur les potentiels autres véhicules proches. Ceci pourrait être une piste de recherche pour la commande d'une formation de plusieurs robots ou d'un essaim de robots. Toutefois, le réalisme des effets que le véhicule génère dans le fluide autour de lui n'a pas été vérifié dans cette thèse. Dans l'optique d'une application militaire, mieux pouvoir appréhender les manœuvres du véhicule peut constituer un avantage tactique. Dans l'optique d'une application civile, cela pourrait permettre une meilleure fluidité de mouvement dans le déplacement du véhicule.

Une seconde utilité serait l'étude de véhicules déformables. Que ce soit des véhicules reconfigurables, qui par exemple seraient capables de réorienter des ailerons ou déplacer/orienter des propulseurs, ou bien des véhicules déformables, dans la lignée de la robotique souple, ce simulateur pourrait permettre d'étudier les variations de paramètres hydrodynamiques induites par les déformations des véhicules. En suivant la même idée, il serait également possible d'utiliser ce simulateur en amont de conception pour pouvoir, sans avoir à réaliser de maquette préalable, tester et améliorer le design des véhicules sous-marins.

Enfin, ce simulateur pourrait servir au développement d'un algorithme de stabilisation dans un environnement fortement perturbé. Actuellement, il n'est pas possible de maintenir un véhicule en place dans la zone de surf. En utilisant le simulateur, il est peut-être possible de concevoir un véhicule déformable capable de s'adapter aux vagues et aux courants, et de maintenir sa position dans cette zone.

5.2.4 Paramètres hydrodynamiques et perspectives en robotique sous-marine

La connaissance des paramètres hydrodynamiques d'un véhicule marin sert actuellement principalement à la simulation de la dynamique du véhicule et à la préparation de nouvelles missions. L'idée de la simulation SPH est de s'affranchir de l'estimation de ces paramètres. Néanmoins, pour l'instant, le simulateur développé dans cette thèse reste incomplet et une validation expérimentale physique est toujours requise pour l'obtention de ces paramètres. En ce qui concerne les expériences qui permettent d'obtenir des paramètres hydrodynamiques, l'intérêt d'utiliser un mécanisme pour mettre en mouvement le véhicule dont on veut obtenir les paramètres a été démontré. Les futures expérimentations à mener pour valider le simulateur devront donc viser à déplacer le véhicule grâce à un mécanisme qui génère une force connue, comme par exemple, un système de traction (tel que celui du bassin de traction de l'école Centrale Nantes [Cou92]) ou un système de propulsion à bord du véhicule dont on connaît précisément la force d'action.

Les paramètres hydrodynamiques peuvent être utilisés pour la commande des véhicules marins et sous-marins. Actuellement, le modèle dynamique des véhicules marins est principalement employé pour des applications où les vitesses restent faibles. Cependant, notamment dans des milieux perturbés, l'utilisation du modèle dynamique pourrait être un atout pour la stabilisation de la trajectoire du véhicule sous-marin soumis aux effets de la mer. La précision dans l'obtention de ces paramètres sera alors un facteur important, et une piste de recherche

consisterait à mettre au point une méthode expérimentale garantissant une certaine précision.

Annexe A

Modélisation cinématique des véhicules sous-marins

A.1 Référentiels

Pour expliquer certains phénomènes pouvant apparaître pour le robot sous-marin nous avons besoin de placer le robot par rapport au référentiel terrestre. Deux référentiels terrestres seront considérés, pour enfin définir référentiel géographique par rapport aux référentiels terrestres. Un premier référentiel terrestre dit inertiel (ECI : Earth Centered Inertial), dont le centre est fixé au centre de la terre et dont l'orientation est fixe dans l'espace : l'axe \vec{Z} est défini par l'alignement du centre et du pôle nord, l'axe \vec{X} pointe vers le point vernal. Le point vernal est défini par le Service International de la Rotation Terrestre et des Systèmes de Référence (IERS) [PL10] comme étant le point d'intersection de l'équateur et de l'écliptique à midi du temps terrestre (TT) à la date du 1^{er} janvier 2000, noté $\gamma_{J2000.0}$. Il fut nécessaire de fixer ce point à une date précise à cause des phénomènes de nutations et de précession de la Terre.

A partir de l'ECI, on peut définir un second référentiel dont l'orientation est fixe par rapport à la terre (ECEF : Earth centered Earth fixed). Il tourne à la vitesse de rotation $\omega_E = 7.2921 \times 10^{-5} rad.s^{-1}$ par rapport au référentiel ECI. Pour le mouvement des véhicules marins les effets de la rotation de la Terre, particulièrement lente, peuvent être négligée et ce référentiel peut être considéré inertiel, donc tel que les lois de Newton s'appliquent.

Nous utilisons le référentiel ECEF comme référence pour le référentiel Est-Nord-Haut (ENU : East North Up) qui est le référentiel utilisé par ROS [ROS]. Le centre de ce référentiel se déplace avec le navire sur la surface de la terre. Dans ce référentiel, l'axe x est orienté vers l'est géographique, l'axe y est vers le nord géographique et l'axe z orienté vers le ciel. Ce repère ENU diffère du repère VEH par son orientation. La figure A.1 montre ces trois référentiels.

Un dernier repère est choisi tel que l'axe x soit parallèle aux lignes de courant et nommé COUR. Ce référentiel est particulièrement utile pour appliquer les efforts de trainée et de portance qui s'appliquent selon la ligne de courant ou perpendiculairement à cette ligne de courant.

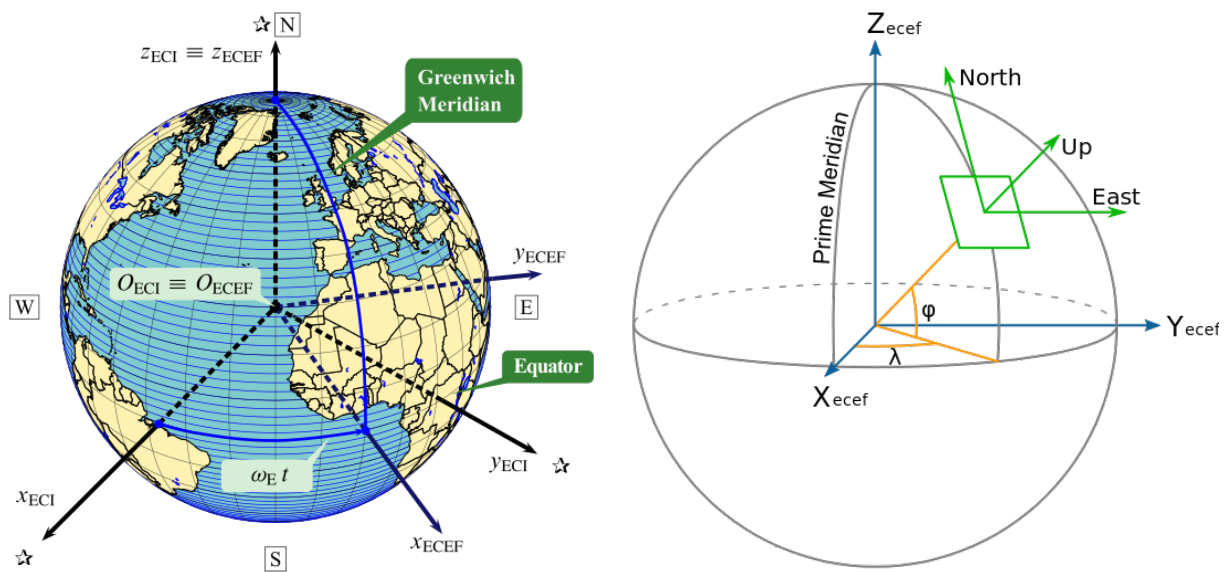


Figure A.1 – Présentation des repères liés à la géographie de la terre. Le repère fixe par rapport à la terre (ECEF) en rotation autour de l'axe z du repère (ECI) et le repère Est-Nord-Haut (ENU) lié au navire à la surface de la terre. (Source image de gauche : [JSB])

A.2 Transformation entre VEH et ENU

La transformation entre le repère du véhicule VEH et Est-Nord-Haut ENU est nécessaire pour permettre de placer le véhicule dans le référentiel terrestre. La transformation entre le repère VEH et le repère ENU, si les deux repères sont centrés au même point, est le résultat de trois rotations :

- Une première rotation d'angle ψ (lacet) autour de l'axe haut du repère ENU U pour aligner l'axe N à l'axe y du navire.
- Une seconde rotation d'angle ϕ (roulis) autour de l'axe x obtenu à la première rotation qui permet d'aligner l'axe z du navire.
- Une dernière rotation d'angle θ (tangage) autour du nouvel axe y qui permet d'aligner l'axe x du navire à celui créé par la seconde rotation.

Ces rotations successives nous donnent une matrice de transformation entre le repère ENU et le repère VEH :

$$\begin{pmatrix} E \\ N \\ U \end{pmatrix} = R_z(\psi)R_x(\phi)R_y(\theta) \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (\text{A.1})$$

Avec les matrices de rotations :

$$R_z(\psi)R_x(\phi)R_y(\theta) = \begin{pmatrix} \sin \psi \cos \theta & \cos \psi \cos \phi + \sin \phi \sin \theta \sin \psi & \sin \theta \sin \psi \cos \phi - \cos \psi \sin \phi \\ \cos \psi \cos \theta & \sin \phi \sin \theta \cos \psi - \sin \psi \cos \phi & \sin \psi \sin \phi + \sin \theta \cos \psi \cos \phi \\ \sin \theta & -\cos \theta \sin \phi & -\cos \theta \cos \phi \end{pmatrix} \quad (\text{A.2})$$

Le vecteur $\vec{\eta}_{ENU} = (E, N, U)$ et le vecteur $\vec{\Theta}_{ENU} = (\phi_{ENU}, \theta_{ENU}, \psi_{ENU})$ sont les vecteurs décrivant, respectivement, la position et l'orientation du véhicule dans le repère ENU. Pour passer du vecteur vitesse $\vec{v}_{VEH} = (u, v, w)$ au vecteur vitesse du référentiel ENU $\vec{\eta}_{ENU}$ il faut donc effectuer la transformation suivante :

$$\vec{\eta}_{ENU} = R_z(\psi)R_x(\phi)R_y(\theta)\vec{v}_{VEH} \quad (\text{A.3})$$

Ainsi on obtient en détaillant :

$$\begin{aligned} \dot{E} &= u \sin \psi \cos \theta + v [\cos \psi \cos \phi + \sin \phi \sin \theta \sin \psi] \\ &\quad + w [\sin \theta \sin \psi \cos \phi - \cos \psi \sin \phi] \end{aligned} \quad (\text{A.4})$$

$$\begin{aligned} \dot{N} &= u \cos \psi \cos \theta + v [\sin \phi \sin \theta \cos \psi - \sin \psi \cos \phi] \\ &\quad + w [\sin \psi \sin \phi + \sin \theta \cos \psi \cos \phi] \end{aligned} \quad (\text{A.5})$$

$$\dot{U} = u \sin \theta - v \cos \theta \sin \phi - w \cos \theta \cos \phi \quad (\text{A.6})$$

$$(\text{A.7})$$

Pour passer du vecteur vitesse angulaire $\vec{\omega}_{VEH} = (p, q, r)$ au vecteur vitesse angulaire du référentiel ENU $\vec{\Theta}_{ENU}$ la transformation est différente car les rotations ne sont pas toutes affectées de la même manière par les transformations de repère successives :

$$\vec{\omega}_{VEH} = \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + R_y(\theta) \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + R_x(\phi)R_y(\theta) \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} = T_{\Theta}^{-1} \vec{\Theta}_{ENU} \quad (\text{A.8})$$

avec T_{Θ}^{-1} l'inverse de la transformation attendue pour passer du repère VEH au repère ENU.

$$T_{\Theta}^{-1} = \begin{pmatrix} \cos \theta & 0 & \sin \theta \cos \phi \\ 0 & 1 & -\sin \phi \\ -\sin \theta & 0 & \cos \theta \cos \phi \end{pmatrix} \implies T_{\Theta} = \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ \tan \phi \sin \theta & 1 & \tan \phi \cos \theta \\ \frac{\sin \theta}{\cos \phi} & 0 & \frac{\cos \theta}{\cos \phi} \end{pmatrix} \quad (\text{A.9})$$

Ce qui nous donne, dans le détail, la transformation suivante :

$$\dot{\phi}_{ENU} = p \cos \theta - r \sin \theta \quad (\text{A.10})$$

$$\dot{\theta}_{ENU} = p \tan \phi \sin \theta + q + r \tan \phi \cos \theta \quad (\text{A.11})$$

$$\dot{\psi}_{ENU} = p \frac{\sin \theta}{\cos \phi} + \frac{\cos \theta}{\cos \phi}, \phi \neq \pm 90^\circ \quad (\text{A.12})$$

$$(\text{A.13})$$

A.3 Transformation entre ENU et ECEF

La transformation pour passer du repère Est-Nord-Haut ENU au repère ECEF permet de placer le véhicule dans le référentiel de la terre, ce référentiel est utile pour des déplacements sur de très longues distance. Pour passer du repère ENU au repère ECEF deux rotations successives sont nécessaires en utilisant les données de longitude l et de latitude λ :

- La première est une rotation d'angle $\pi/2 - l$ permettant d'aligner l'axe haut avec l'axe nord de la terre.
- La seconde est une rotation d'angle $-\pi/2 - \lambda$ permettant d'aligner l'axe est du repère ENU et l'axe \vec{X} de la terre.

La transformation s'obtient donc avec :

$$\begin{pmatrix} x_{ECEF} \\ y_{ECEF} \\ z_{ECEF} \end{pmatrix} = R_z(\pi/2 - l)R_x(-\pi/2 - \lambda) \begin{pmatrix} E \\ N \\ U \end{pmatrix} \quad (\text{A.14})$$

Avec les matrices de rotations :

$$R_z(\pi/2 - l)R_x(-\pi/2 - \lambda) = \begin{pmatrix} -\sin \lambda & -\cos \lambda \sin l & \cos \lambda \cos l \\ \cos \lambda & -\sin \lambda \sin l & \sin \lambda \cos l \\ 0 & \cos l & \sin l \end{pmatrix} \quad (\text{A.15})$$

A.4 Transformation entre VEH et COUR

Le repère COUR est très souvent utilisé pour les efforts hydrodynamiques. L'axe x du repère COUR est l'axe parallèle aux lignes de courant, l'axe z lui reste dans la plan de référence du navire. Par définition, la portance est définie perpendiculaire au lignes de courant et la trainée est définie parallèle à ces lignes. Le grand intérêt de l'utilisation de ce repère est donc que la force de trainée est selon son axe x et celle de portance selon son axe z . L'objectif est donc d'obtenir un alignement entre la vitesse U du courant autour du véhicule et l'axe x . La transformation pour passer du repère du véhicule VEH au repère COUR est composé de deux rotations :

- Une première rotation en fonction de l'angle de glissement latérale β autour de l'axe z
- Une seconde rotation en fonction de l'angle d'attaque α autour de l'axe y

La transformation du repère COUR au repère VEH peut s'écrire ainsi :

$$\begin{pmatrix} U \\ 0 \\ 0 \end{pmatrix} = R_y(\alpha)R_z(-\beta) \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (\text{A.16})$$

Avec les matrices de rotations :

$$R_y(\alpha)R_z(-\beta) = \begin{pmatrix} \cos \beta \cos \alpha & \sin \beta & \cos \beta \sin \alpha \\ -\sin \beta \cos \alpha & \cos \beta & -\sin \beta \sin \alpha \\ \sin \alpha & 0 & \cos \alpha \end{pmatrix} \quad (\text{A.17})$$

La transformation inverse est obtenue en utilisant la transposée des matrices de rotations. A partir de la transposée on obtient les relations suivantes :

$$u = U \cos \alpha \cos \beta \quad (\text{A.18})$$

$$v = U \sin \beta \quad (\text{A.19})$$

$$w = U \sin \alpha \cos \beta \quad (\text{A.20})$$

Ainsi on obtient les valeurs des angles α et β pour un véhicule marin à vitesse constante $U > 0$:

$$\alpha = \tan^{-1} \left(\frac{w}{u} \right) \quad (\text{A.21})$$

$$\beta = \sin^{-1} \left(\frac{v}{U} \right) \quad (\text{A.22})$$

Dans le cas d'un véhicule marin soumis à un courant océanique, la différence entre la vitesse du véhicule et la vitesse du courant doit être prise en compte ainsi on a la relation avec les vitesses relatives, notées avec un indice r , les vitesses d'avance et les vitesses de courant, notées avec un indice c :

$$\begin{pmatrix} u_r \\ v_r \\ w_r \end{pmatrix} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} - \begin{pmatrix} u_c \\ v_c \\ w_c \end{pmatrix} \quad (\text{A.23})$$

avec une vitesse relative U_r :

$$U_r = \sqrt{u_r^2 + v_r^2 + w_r^2} \quad (\text{A.24})$$

A partir de la même méthode, on retrouve une équation similaire à l'équation [A.20](#) :

$$u_r = U_r \cos \alpha \cos \beta \quad (\text{A.25})$$

$$v_r = U_r \sin \beta \quad (\text{A.26})$$

$$w_r = U_r \sin \alpha \cos \beta \quad (\text{A.27})$$

Et les angles d'attaque α_r et β_r :

$$\alpha_r = \tan^{-1} \left(\frac{w_r}{u_r} \right) \quad (\text{A.28})$$

$$\beta_r = \sin^{-1} \left(\frac{v_r}{U_r} \right) \quad (\text{A.29})$$

Annexe B

Environnements de simulations et leurs composants principaux

Dans la première section de cette annexe, des informations sur les éléments principaux qui composent les logiciels nommés environnement de simulation seront présentés. Pour rappel, un environnement de simulation est un logiciel qui permet de d'accueillir et/ou de fabriquer un modèle puis de simuler son fonctionnement, voir section [1.5.1](#). Dans la seconde section de cette annexe, différents logiciels seront présentés.

B.1 Composants des environnements de simulations

Dans cette section, sera expliqué ce que sont les middlewares, les bibliothèques graphiques et les moteurs physiques. Une attention particulière sera portée sur les moteurs physiques, qui sont les bibliothèques qui gèrent la partie dynamique¹ de la plupart des environnements de simulation que nous présenterons dans la prochaine section.

B.1.1 Les middlewares

Un middleware est une application qui permet de créer un canal de requêtes et de réponses entre différents composants logiciels. Chacun des composants peut choisir de transmettre ou non des informations sur le canal et peut réaliser des actions lorsqu'un nouveau composant se connecte au canal. Les composants logiciels du middleware sont reliés comme sur un véritable réseau informatique et peuvent se trouver sur différents appareils (robot, serveur, camera, second ordinateur, etc.). Ces middlewares sont utilisés en robotique, mais ont aussi d'autres applications, comme l'Internet of Things (IoT), c'est à dire les objets connectés. Grâce à ces middlewares, il est possible de relier différents environnements de simulation entre eux, comme dans Kermorgant [Ker14], où UWSIM est relié à Gazebo. Des comparatifs de middleware pour la robotique existent. Par exemple, Magyar [MSK15] compare 4 middlewares selon leurs fonctionnalités, Elkady [ES12] en compare 21 différents avec plus de détails et Fortino [FGRS14] en compare 13 pour la mise en réseau d'objets connectés. Aucun middleware ne sera utilisé dans cette thèse, alors nous avons choisi de ne pas présenter leurs capacités en détails. Parmi les middlewares les plus couramment utilisés en robotique on peut citer :

- ROS de l'IRSF
- YARP
- Orocos
- RT-Middleware

B.1.2 Les bibliothèques graphiques

Les bibliothèques graphiques sont des bibliothèques qui permettent d'ajouter des fonctionnalités d'affichage à un programme. Elle permettent de dessiner des éléments en plan (2D) ou en volume (3D), ce qui permet d'afficher des icônes, des images, ou des scènes en trois dimensions. Le principe est de déclarer la géométrie des objets sous forme de points, de vecteurs, de polygones, de bitmaps et de textures. La bibliothèque graphique effectue ensuite des calculs de projection en vue de déterminer l'image à l'écran, en tenant compte de la distance, de l'orientation, des ombres, de la transparence et du cadrage. Les bibliothèques graphiques actuelles tirent parti, quand elle le peuvent, des processeurs parallèles et des technologies accélératrices, c'est à dire des circuits intégrés dédiés qui exécuteront certaines fonctions plus efficacement, des cartes graphiques actuelles. La première à permettre un affichage en volume et la plus célèbre des bibliothèques graphiques est OpenGL [Mar06], qui veut dire « Open Graphic Library ». Cette bibliothèque est sous licence libre. Parmi ses concurrentes, il y a la bibliothèque de Microsoft : DirectX, qui est une bibliothèque plus haut niveau car elle intègre également des fonctions pour la gestion de son et de périphérique d'entrée. L'affichage 3D est réalisé avec une sous-bibliothèque de DirectX : Direct3D. Plus récemment, depuis 2016, une nouvelle bibliothèque

1. Ce qui est le cœur de cette thèse

nommée Vulkan est apparue. Cette API, d'abord annoncée sous l'appellation OpenGL Next, est développée par Khronos Group, un consortium industriel regroupant les plus grandes entreprises de l'industrie numérique, et a pour but de remplacer à terme OpenGL. Dans la suite nous allons évoquer d'autres bibliothèques graphiques comme :

- Blender, qui est un logiciel basé sur OpenGL est un logiciel gratuit qui permet d'utiliser de nombreuses fonctionnalités avancées de rendu 3D, un modèleur, une interface graphique, des fonctions d'animation et capable d'être déployé sur différents systèmes d'exploitation. Ce logiciel peut être inclus en tant que logiciel tiers à un environnement de simulation ou utilisé directement. Un simulateur de ROV, reliant Blender et ROS, a été créé par Guth [GSA⁺13] pour la simulation d'un « Simultaneous Localization And Mapping (SLAM) », en français cartographie et localisation simultanée. Ce simulateur ne nécessitait aucun calcul de la dynamique du robot.
- OGRE, qui est une bibliothèque open source basée sur OpenGL qui doit être ajoutée comme logiciel tiers.
- OpenSceneGraph, moteur 3D open source qui est concurrent de OGRE, encore basé sur OpenGL.

Si certains logiciels utilisent ces bibliothèques graphiques, précédemment citées qui permettent un rendu en temps réel, d'autres, notamment ceux dédiés à l'animation hors ligne, utilisent des techniques de rendus basées sur le lancer de rayon², qui est une technique plus lente, et se passent des bibliothèques de rendu temps réel.

Cet aspect de rendu ne nous intéresse pas directement dans le cas de l'étude de la dynamique des véhicules sous-marins. Néanmoins certaines fonctionnalités de commande du véhicule peuvent être basées sur un retour de la caméra embarquée et sur une utilisation des données images. Il faudrait dans ce cas s'assurer du rendu graphique de la simulation.

B.1.3 Les moteurs physiques

Les bibliothèques logicielles indépendantes permettant de résoudre des problèmes de la mécanique classique sont couramment appelées moteurs physiques ou en anglais « Physic Engine ». Typiquement, ces bibliothèques permettent de résoudre les actions des forces de gravité, des collisions, de liaison mécanique et des effets inertiels (notamment pour les rotations de solide). Les principales difficultés à surmonter pour ces moteurs physiques sont la gestion de collisions multiples, la prise en compte de liaisons mécaniques, la synchronisation avec la scène, la réalisation des calculs en temps réel et la conservation de l'énergie dans l'intégration temporelle. Ces moteurs physiques sont dédiés à l'animation en temps réel et de ce fait, de nombreux sacrifices sont faits sur le réalisme physique pour obtenir une simulation plus stable et plus rapide à calculer.

Il existe différents articles comparatifs de ces moteurs physiques, comme par exemple Boeing [BB07], qui date de 2007 et qui est donc aujourd'hui légèrement désuet, mais qui néanmoins compare en détails les fonctionnalités de 6 moteurs physiques de cette époque. Erez [ETT15] propose une évaluation du même type, mais propose des cas tests pour la comparaison des performances des moteurs physiques : ODE, PhysX, Havok, Bullet et MuJoCo. Erez mesure notamment la vitesse d'exécution des calculs avec différentes scènes complexes à résoudre, comme par exemple une main robotisée à 35 degrés de libertés attrape un objet, une forme humanoïde à 25 degrés de libertés tombe sur le sol, 27 capsules qui tombent sur le

2. En anglais « ray tracing »

sol. Selon cette étude aucun moteur physique ne dépasse les autres sur tous les plans. Pour certaines applications spécifiques, comme par exemple celles liées à la robotique articulée³, certains logiciels arrivent à se distinguer. Pour cet exemple, il s’agit de MuJoCo. Pour d’autres applications, comme par exemple lors de la collision d’un grand nombre de solides, MuJoCo, est moins performant que d’autres. Notre simulation de véhicule sous-marin va comporter un véhicule sans articulations pouvant naviguer au milieu de différents autres solides, nous écartons donc l’utilisation de MuJoCo.

Parmi les moteurs physiques restant, nous allons présenter les moteurs qui sont les plus présents dans les environnements de simulations que nous allons présenter dans la prochaine sous-section : PhysX et Bullet, ainsi que deux bibliothèques qui ont une approche particulièrement intéressante pour la modélisation du fluide et du solide : FleX et Bifrost. Bifrost est une bibliothèque qui diffère des moteurs précédemment présentés puisqu’elle est plutôt dédiée à l’animation hors ligne.

B.1.3.1 Nvidia PhysX

Historiquement, Nvidia PhysX a été développée initialement par l’équipe de Mathias Müller⁴ au sein de la société Novodex AG [MSJT10]. Le programme commercialisé se nommait initialement Novodex et ne comportait alors qu’un système de solides rigides [AG03]. Novodex a été ensuite racheté par AGEIA en 2004, pour développer le premier Physics Processing Unit⁵ (PPU) une solution hardware dédiée aux calculs physiques en utilisant le multithreading [Phy05]. Enfin en 2008, Nvidia rachète AGEIA Technologies et adapte PhysX [Phy08] pour l’utiliser avec le langage Compute Unified Device Architecture (CUDA) [Phy17] et incorporer la technologie PPU directement dans les cartes graphiques de la série GeForce 8 [Sma08].

Nvidia PhysX intégrait jusqu’en 2017⁶ une bibliothèque qui permettait de calculer le comportement d’un fluide en considérant celui-ci comme un ensemble de particules qui se déplacent dans l’espace [MCG03]. Cette extension utilisait la méthode appelée Smoothed Particle Hydrodynamics (SPH), voir chapitre 3.

B.1.3.2 Bullet

Bullet est un moteur physique open source développé par Erwin Coumans⁷. La dernière version de Bullet est Bullet 3. Une bibliothèque spécifique, PyBullet a été développée pour qu’il soit facilement intégrable sur les projets en langage Python. Bullet est de plus en plus utilisé en robotique (mais notamment par les équipes de Google), comme dans Zeng où il est utilisé pour l’apprentissage d’un robot lanceur de balle [ZSL⁺], ou dans Choromanski pour des simulations avec un robot marcheur [CPPH⁺]. Néanmoins il ne prend pas en compte les phénomènes hydrodynamiques [E.19].

3. système robotisé présentant des articulations

4. Mathias Müller est diplômé d’un doctorat de l’Eidgenössische Technische Hochschule (ETH) Zürich et a cofondé Novodex

5. Les cartes PPU sont tombés en désuétude et ne sont plus aujourd’hui fabriquées ni même supportées [Phy11].

6. Jusqu’à la version 3.4

7. Erwin Coumans, originaire des Pays Bas, travaille sur ce projet depuis plus de 15 ans et fait aujourd’hui partie des équipes de recherche de Google

B.1.3.3 Nvidia FleX

Nvidia FleX est une bibliothèque, développée par la société Nvidia, qui souhaite rendre possible le calcul multi-physique en temps réel en s'appuyant sur le langage CUDA. Ce moteur physique est un peu à part car il souhaite permettre de modéliser à la fois des solides, des liquides et différentes sortes d'objets déformables interagissant entre eux et ceci uniquement avec un système de particules [MMCK14]. La dynamique des fluides avec Nvidia FleX est présentée dans l'article de Macklin [MM13]. Le modèle utilisé est PCISPH [SP09] Ce logiciel est dérivé du logiciel PhysX de Nvidia et semble assez prometteur.

B.1.3.4 Bifrost

Bifrost est une bibliothèque dédiée à l'animation hors ligne, directement intégrée à Autodesk Maya, qui permet de créer des effets mettant en jeu des fluides, qui sont simulés à l'aide d'un solveur « FLuid Implicit Particle » (FLIP), voir chapitre 3. Ainsi, il permet de générer des liquides à partir d'un émetteur, de lui faire subir la gravité, de simuler des éclaboussures en interagissant avec le « collider⁸ » du logiciel dans lequel il est implémenté, de gérer des effets avec des fumées, ou encore de simuler des jets d'eau. Cette bibliothèque possède également une sous-partie qui permet de simuler la surface de l'eau : Bifrost Ocean Simulation System (BOSS). BOSS permet de simuler les vagues et la houle, en prenant en compte l'échelle de Douglas. Il permet également de prendre en compte les vagues qui peuvent être générées à la surface de l'eau par un véhicule marin. Néanmoins, ce moteur physique ne permet pas de gérer, à lui seul, les solides et ne renvoie pas d'effort du fluide vers les solides, l'objectif étant d'obtenir un rendu visuel réaliste.

8. Le collider est la partie du programme permettant le calcul des collisions

B.2 Présentation de plusieurs environnements de simulations existants

Chaque logiciel est développé pour un type d'utilisateur particulier. Les logiciels répondant aux critères d'environnements de simulation que nous avons évoqués dans la section précédente sont variés, et il a été choisi de les présenter ici selon le public cible de ces logiciels. Les logiciels sont regroupés selon ceux dédiés aux roboticiens, ceux pour l'animation et ceux dédiés à la simulation d'écoulement de fluide.

B.2.1 Logiciels dédiés à la robotique

La simulation est un devenu un outil essentiel pour les roboticiens, car elle permet de tester en amont les algorithmes, le design des robots et plus récemment l'entraînement d'intelligence artificielle à implanter dans le futur robot, comme dans Rusu [RVR⁺]. Différents articles comparatifs parlent des simulateurs dédiés à la robotique, comme celui de Mouret [MC17] ou de Ivaldi [IPN14], où les simulateurs sont comparés en fonction des retours d'expérience des utilisateurs. Le logiciel de robotique le plus utilisé est donc Gazebo, mais les raisons principales de cet intérêt sont sa licence libre et l'activité de la communauté scientifique autour de Gazebo. L'article montre que le logiciel le plus abouti à cette époque (2014), qui néanmoins n'a pas de licence libre est V-REP, car il a une bonne documentation, de bons tutoriels et qu'il propose de nombreuses fonctionnalités avancées. Cette étude néanmoins mélange deux types de logiciels que nous avons distingués dans cette thèse : les environnements de simulations et les moteurs physiques, qui eux ne permettent pas directement l'affichage. Plus récemment, le même auteur a écrit le chapitre 35 du livre [Hum18], où elle présente et compare les différentes solutions de simulations pour la robotique humanoïde. L'article de Cook [CVL14], passe en revue les différents simulateurs existants en 2014 pour la simulation de véhicules sous-marins et propose également l'utilisation de logiciels dédiés aux jeux vidéo et à l'animation pour la simulation de véhicules sous-marins. Nous avons choisi de présenter cinq simulateurs qui correspondent à ceux qui semblent être les plus performants pour la simulation de véhicules sous-marins. Ils ont été choisis pour leur capacité à simuler des capteurs, à être connectés avec d'autres programmes comme Matlab ou avec le système réel, à travers un middleware ou non, et/ou parce qu'ils intègrent déjà une modélisation de l'environnement marin.

B.2.1.1 Vortex de CM Labs

Vortex est un simulateur multi-physique dédié à la modélisation de l'environnement marin et sous-marin. Il permet de simuler l'évolution d'un robot sous-marin dans cet environnement ainsi que certains effets visuels avancés du sous-marin sur son environnement. En plus d'une très jolie visualisation de l'environnement marin : prise en compte de la réflexion de la lumière dans l'eau, de la création de vague en surface et de leur prise en compte dans le calcul hydrodynamique (pour les véhicules à la surface uniquement), et de la visibilité sous-marine, Vortex possède un moteur physique hydrodynamique assez complet. Il prend en compte la poussée d'Archimède et les forces hydrodynamiques à condition de connaître les coefficients à appliquer sur le véhicule. Il permet également dans une certaine mesure de prendre en compte la transmission de la puissance du moteur à l'eau en fonction de la vitesse de rotation. Enfin, grâce à son modèle multi-physique il permet de réaliser des missions amphibies complètes

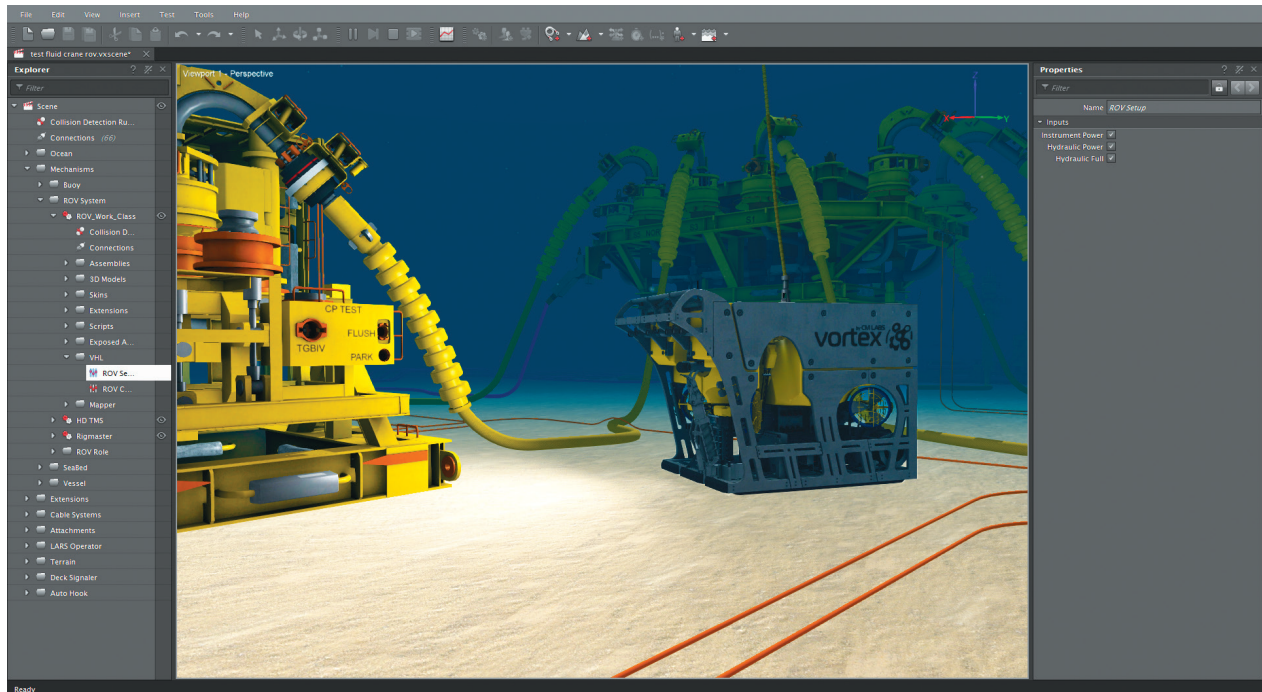


Figure B.1 – Vue d’une simulation de mission d’un ROV sous Vortex

(terre, surface et sous l’eau) et permet de simuler la météo maritime (échelle de Beaufort et échelle de Douglas). Ce logiciel a été utilisé lors de recherches en robotique sous-marine par Sverdrup-Thygeson [STKPG16], qui a effectivement utilisé le modèle dynamique présent dans Vortex, qui est celui de Fossen [Fos11], pour simuler les mouvements de son robot, Zhang [ZDM⁺16] où est décrit la manière dont a été créé un simulateur d’installation de complexe sous-marin, ou encore par Laroche [LDC⁺13] qui se concentre sur la phase de développement d’un ROV pour répondre au besoin de l’utilisateur. La figure B.1 montre l’interface de Vortex au cours d’une simulation de mission d’un ROV.

Caractéristiques :

- Licence : Commerciale
- Rendu graphique : Pas d’information
- Librairie pour le calcul de la dynamique :
 - Dynamique des solides : Module interne
 - Dynamique des fluides : Module interne permettant la simulation de vagues (haute mer) et d’effets à la surface.
- Middleware : non, mais lien possible avec Matlab Simulink.
- Script : Python
- Dernière version : 2019

B.2.1.2 MORSE et UW MORSE du LAAS-CNRS et NTNU

MORSE est un environnement de simulation académique développé par le Laboratoire d’Analyse et d’Architecture Système (LAAS) qui permet de générer des scènes 3D à partir de simples scripts python [ELDL11]. Morse peut être couplé avec de nombreux middleware dont ROS, et est basé sur le moteur graphique Blender et utilise le moteur physique Bullet.

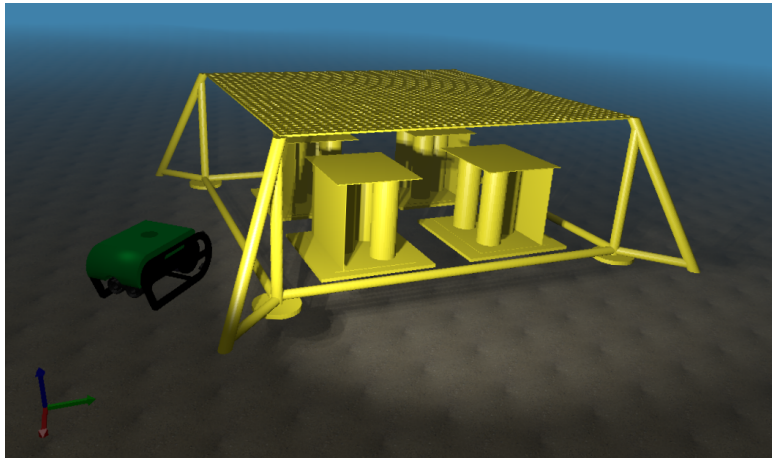


Figure B.2 – Image issue de l’environnement de simulation UWMORSE (Source : [HSG16])

D’après l’étude [BB07] ce moteur ne prend pas en compte la dynamique des fluides. Cet environnement est donc peu compatible avec la simulation de la dynamique de véhicules marins et sous-marins. MORSE permet de simuler un grand nombre de capteurs et il serait néanmoins possible de partager leurs données à travers un middleware et d’utiliser un autre logiciel pour la simulation de la dynamique des fluides. Ce logiciel, a récemment été adapté pour la simulation de véhicules sous-marins et une nouvelle branche UW MORSE de ce logiciel a été créée [HSG16] à l’université de Trondheim (NTNU) et permet de simuler la dynamique des véhicules sous-marins ainsi que leur environnement. Dans cette nouvelle branche, une couche supplémentaire est ajoutée et permet de calculer les efforts hydrodynamiques à appliquer sur le véhicule à partir de paramètres. La figure B.2 montre une capture d’écran de l’environnement au cours d’une simulation avec UWMORSE.

Caractéristiques :

- Licence : Open Source
- Rendu graphique : Blender
- Librairie pour le calcul de la dynamique :
 - Dynamique des solides : Bullet + un solveur interne pour la dynamique des solides dans l’eau avec UW MORSE.
 - Dynamique des fluides : Non
- Middleware : ROS, MAVlink protocol
- Script :
- Dernière version : MORSE 1.4 (2016) / UW MORSE (2017)

B.2.1.3 UWSIM de l’université Jaume I

UWSim est un projet de l’IRS Lab de l’université de Castellon. C’est un environnement de simulation open source spécifiquement conçu pour les véhicules sous-marins. Il utilise le moteur physique Bullet comme MORSE et le moteur graphique OpenSceneGraph.[PPFS12]. Un moteur de rendu 3D extérieur, comme Blender ou 3D Studio Max, peut facilement être ajouté pour produire des images de hautes qualités. UWSim est compatible avec ROS. Il contient un certain nombre de modélisation de capteurs et permet de transmettre les données de ceux-ci via ROS de la même manière que pour le programme MORSE. Ce logiciel permet

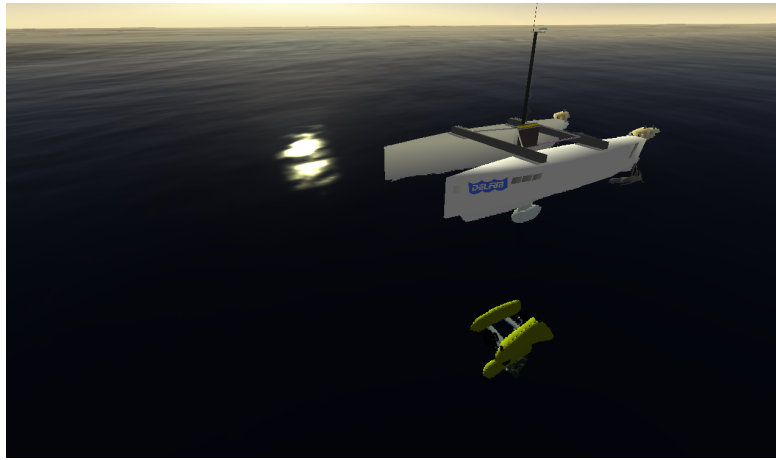


Figure B.3 – Capture d’écran d’UWSim

de faire des simulations en surface et sous l’eau comme Vortex. Il est néanmoins difficile de trouver une documentation pour ce logiciel qui semble peu maintenu. La figure B.3 montre les capacités du moteur graphique d’UWSim.

Caractéristiques :

- Licence : Open Source
- Rendu graphique : OpenSceneGraph (OSG)
- Librairie pour le calcul de la dynamique :
 - Dynamique des solides : Bullet + un solveur interne pour la dynamique des solides dans l’eau
 - Dynamique des fluides : aucun
- Middleware : ROS
- Script :
- Dernière version : 1.4.1 (2017)

B.2.1.4 Gazebo de l’Open Source Robotics Foundation (OSRF)

Gazebo est un environnement de simulation, initialement a été développé par l’University of Southern California, mais plus récemment continué par l’ « Open Source Robotics Foundation » (OSRF), dont fait également partie le projet ROS et qui est soutenu par la DARPA (équivalent de la DGA pour les USA). Par défaut, le moteur physique de Gazebo est ODE, qui est un moteur physique dont le réalisme est fiable. Gazebo est intégré dans ROS et de nombreux services ont été développés autour de Gazebo comme le format .sdf ou des programmes Cloud. [Fou16] [KH04]. De nombreuses recherches se sont appuyées sur Gazebo pour la simulation de robot : aussi bien pour la simulation de véhicules terrestres avec, par exemple, les travaux de Fernandes[FSP⁺14], ou de véhicules volants avec, par exemple, les travaux de Meyer [MSK⁺12], ou encore pour la simulation de robot humanoïde avec, par exemple, les travaux d’Ajoudani [ALR⁺14]. Certains travaux existent pour la simulation de véhicules sous-marins avec les travaux de Nadj [NVS⁺14], Kozhemyakin [KRR⁺16] et VanderWardt [VdW11]. D’après la publication d’Ivaldi [IPN14], on peut toutefois supposer que Gazebo est principalement utilisé pour la recherche en robotique mobile terrestre et aérienne. Gazebo n’intègre pas directement de librairie hydrodynamique mais des projets spécifiques ont créé

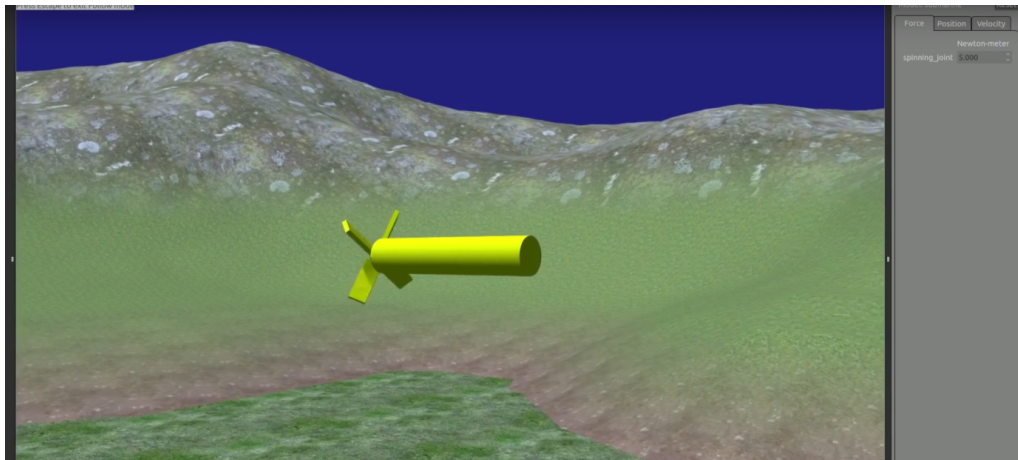


Figure B.4 – Capture d'écran d'un robot sous-marins dans Gazebo

ces bibliothèques, comme UUV Simulator [MSV⁺16]. La figure B.4 montre un AUV sous l'eau avec l'interface graphique de Gazebo.

Caractéristiques :

- Licence : Open Source
- Rendu graphique : OGRE
- Bibliothèque pour le calcul de la dynamique :
 - Dynamique des solides : ODE, Bullet, Simbody, DART
 - Dynamique des fluides : Non
- Middleware : ROS
- Script :
- Dernière version : 10.0 (2019)

B.2.1.5 V-REP

V-REP est un environnement de simulation développé par Coppelia Robotics, une entreprise basée en suisse. V-REP propose 4 moteurs physiques : Bullet, ODE, Vortex (celui de CM Labs), et Newton. [Rob16]. De tous les environnements de simulation présentés précédemment V-REP est le plus récent. [FSOM10] [RSF13] Le moteur physique Vortex dans V-REP n'est malheureusement disponible que dans la version payante de V-REP. Dans la version éducation on n'a accès qu'aux 3 autres moteurs physiques. La figure B.5 montre l'interface graphique de V-REP.

Caractéristiques :

- Licence : Commerciale
- Rendu graphique : OpenGL
- Bibliothèque pour le calcul de la dynamique :
 - Dynamique des solides : Bullet, ODE, Vortex (celui de CM Labs), Newton
 - Dynamique des fluides : non
- Middleware : ROS, BlueZero
- Script :
- Dernière version : 3.6.2 2019

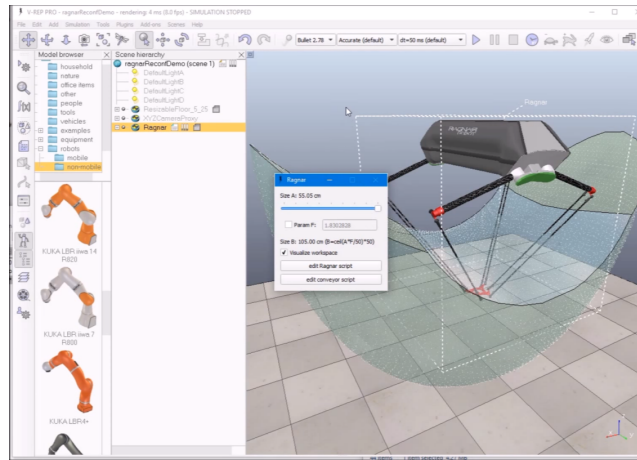


Figure B.5 – Capture d'écran de V-REP

B.2.2 Logiciels dédiés aux jeux vidéo ou à l'animation 3D

À côté des logiciels dédiés à la robotique, de nombreux roboticiens se tournent vers les logiciels de développement pour l'animation ou les jeux vidéo, comme par exemple dans l'article d'Andaluz [ACG⁺16], où Unity est couplé avec Matlab pour la simulation de robot mobile avec bras articulé ou Mattingly [MjCP⁺12] où les capacités de Unity pour les simulations pour la robotique sont discutées. L'intérêt de l'utilisation de ces logiciels est qu'ils sont très ergonomiques. Parmi les logiciels existants, les plus populaires ont été choisis ici.

B.2.2.1 Unity

Unity est un moteur de jeux multiplateforme développé par Unity Technologies, une société basée en Californie. De nombreuses bibliothèques existantes ont été développées pour simuler le comportement de l'eau et des navires pour les jeux vidéo ou par des roboticiens/hydrodynamiciens. Ce logiciel est capable de déployer les jeux ou animations réalisés sur un très grand nombre de plateformes : Windows, Ubuntu, iOS, les consoles de jeux, et bien d'autres. Ce logiciel est l'un des plus présents sur le marché, notamment pour la réalité virtuelle où il représente 2/3 du marché. Il permet également de créer des jeux connectés sur internet. Une simulation créée avec cet environnement peut donc être incluse dans un réseau, avec ou sans middleware. La figure B.6 montre une capture d'écran d'une simulation avec un BlueROV 2 sous Unity :

Caractéristiques :

- Licence : Gratuite (Utilisation) / Commerciale (Support technique/Distribution)
- Rendu graphique : API native
- Bibliothèque pour le calcul de la dynamique :
 - Dynamique des solides : NVIDIA PhysX
 - Dynamique des fluides : non
- Middleware : Non mais mise en réseau du simulateur possible.
- Script : C#
- Dernière version : Unity 2019

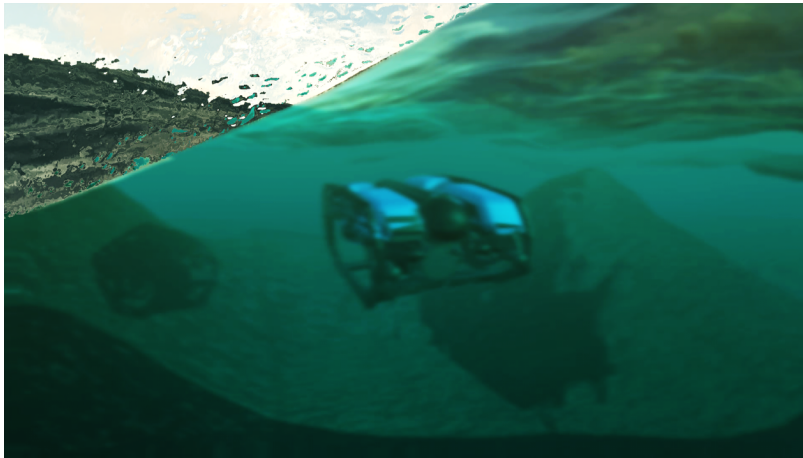


Figure B.6 – Simulation avec un BlueROV 2 sous Unity

B.2.2.2 Unreal Engine

Unreal Engine est un moteur de jeu vidéo développé par Epic Games. Il utilise PhysX comme solveur dynamique et a également intégré Nvidia FleX pour les fluides. Aujourd’hui, Unreal Engine est à la version 4. Unreal Engine 3 avait déjà été utilisé par Carpin [CLW⁺07] avant que les premiers logiciels environnements de simulation dédiés à la robotique ne voient le jour, pour démontrer l’intérêt de simuler des robots évoluant dans leur environnement pour des robots de sauvetage.

Caractéristiques :

- Licence : Gratuite (Utilisation) / Commerciale (Support technique/Distribution)
- Rendu graphique : API native
- Librairie pour le calcul de la dynamique :
 - Dynamique des solides : Nvidia PhysX, FleX, APEX
 - Dynamique des fluides : non
- Middleware :
- Script : Python, C++
- Dernière version : Unreal Engine 4 (2019)

B.2.2.3 Autodesk Maya

Autodesk Maya est un logiciel d’animation, qui permet la modélisation, le rendu et la simulation de scène 3D. Il permet notamment de modéliser la surface de l’eau avec le système Bifröst Ocean Simulation System, ou de simuler des cascades avec un système de particules avec Bifröst que nous allons présenter dans la section des moteurs physiques.

Caractéristiques :

- Licence : Commerciale
- Rendu graphique : API native
- Librairie pour le calcul de la dynamique :
 - Dynamique des solides : Bullet, Bifröst Ocean Simulation System (BOSS)
 - Dynamique des fluides : Bifröst (surface)
- Middleware : non
- Script : Python, Maya Embedded Language

- Dernière version : Maya 2019.2

B.2.3 Logiciels dédiés à la simulation du fluide

Dans l'architecture navale, pour calculer et optimiser la forme d'un navire, on utilise le plus souvent des logiciels qui simulent l'écoulement du fluide autour du véhicule. Cette simulation se fait à travers la résolution des équations de Navier-Stokes ou un système d'équations équivalent. Ces logiciels permettent aujourd'hui d'analyser le comportement aérodynamique des véhicules marins et sont même utilisés comme outils de validation pour des expériences réelles, comme dans Eng [ELLS08], où ils sont utilisés pour valider une expérience de détermination de paramètres hydrodynamiques avec un pendule. Ou encore dans [ES16] où la simulation est utilisée pour valider une expérience avec un rail instrumenté. Il y a plusieurs façons de modéliser l'espace pour la résolution des équations de Navier-Stokes. On peut, d'une part, découper un certain volume de l'espace en petits éléments de volume et obtenir ainsi un maillage de l'espace. Ou d'autre part, on peut considérer que l'espace qui contient de la matière peut être découpé en un ensemble de particules de matière ayant chacune une certaine masse. Ceci nous donne deux types de logiciels distincts pour la simulation de fluide : les logiciels utilisant des particules et ceux utilisant un maillage. La technique de résolution avec des particules est assez récente, c'est pourquoi la majorité des logiciels existants utilise la technique du maillage. Deux logiciels parmi les plus courants dans le domaine de la simulation avec maillage seront présentés ici, ainsi que deux logiciels utilisant une méthode avec des particules.

B.2.3.1 ANSYS Fluent

Fluent est un logiciel de la société ANSYS, une société qui développe des solutions pour la simulation de tous types de problèmes (mécanique des solides déformable, machine thermique, électronique, champ magnétique, ...). Ce logiciel permet de modéliser et simuler les écoulements, les phénomènes de turbulences, les transferts de chaleur ou encore les réactions chimiques dans un système. Il utilise pour cela des modèles maillés. Il prend en charge un grand nombre de modèles (ou systèmes d'équations) pour la dynamique des fluides et permet d'obtenir un résultat particulièrement précis. ANSYS est un logiciel utilisé notamment dans les études hydro- et aérodynamiques, pour le calcul de machine hydraulique ou thermique. ANSYS Fluent peut être couplé au logiciel de calcul Matlab, notamment pour la résolution de problèmes d'optimisation.

Caractéristiques :

- Licence : Commerciale
- Rendu graphique : API native
- Librairie pour le calcul de la dynamique :
 - Dynamique des solides : non
 - Dynamique des fluides : Partie intégrante du logiciel
- Middleware : non
- Script : Script ANSYS .UDF
- Dernière version : ANSYS 2019 R2

B.2.3.2 STAR-CCM+

STAR-CCM+ est un logiciel qui a été développé à l'origine par la société CD-Adapco qui a été rachetée par Siemens en Avril 2016. Tout comme Fluent, c'est un logiciel qui permet de faire des simulations multi-physiques et notamment des simulations hydrodynamiques. Il n'y a pas de véritable différence entre les résultats que peuvent donner les logiciels Fluent et STAR-CCM+, puisqu'ils contiennent, à peu près, les mêmes modèles mathématiques et les mêmes fonctionnalités de modelage et maillage. L'un des deux ayant parfois l'avantage sur l'autre pendant une courte période (quelques mois, années) d'une fonctionnalité supplémentaire issue d'une innovation récente.

Caractéristiques :

- Licence : Commerciale
- Rendu graphique : API native
- Librairie pour le calcul de la dynamique :
 - Dynamique des solides : non
 - Dynamique des fluides : Partie intégrante du logiciel
- Middleware : non
- Script : Java
- Dernière version : Simcenter STAR-CCM+ (2019)

B.2.3.3 GPUSPH

GPUSPH est un logiciel de simulation de la dynamique des fluides qui utilise une modélisation du fluide avec des particules. GPUSPH a été la première implémentation de la méthode WCSPH (Weakly Compressible Smoothed Particle Hydrodynamics, voir chapitre 3) à fonctionner entièrement sur Graphics Processing Unit (GPU) avec Compute Unified Device Architecture (CUDA) [HBD10], qui est une technologie permettant de réaliser des calculs en utilisant un processeur graphique (GPU) pour exécuter des calculs généraux à la place du processeur (CPU). Il est basé sur des plateformes externes, comme Salome et Paraview pour, respectivement, la préparation de la scène à simuler (modèle, maillage) et l'affichage du résultat final. GPUSPH représente donc uniquement le code de calcul permettant la résolution des équations du fluide sur GPU.

Caractéristiques :

- Licence : OpenSource
- Rendu graphique : Salome, Paraview
- Librairie pour le calcul de la dynamique :
 - Dynamique des solides : Chrono
 - Dynamique des fluides : Partie intégrante du logiciel
- Middleware : non
- Script : C++
- Dernière version : 5.0 (2019)

B.2.3.4 Neutrino

Neutrino est un logiciel de simulation de la dynamique des fluides qui, tout comme GPUSPH, utilise une modélisation du fluide avec des particules et qui est édité par la société Centroid Lab. Neutrino est une suite complète permettant de modéliser une scène en 3D, d'ajouter des objets solides avec une géométrie simple et de réaliser des simulations réalistes

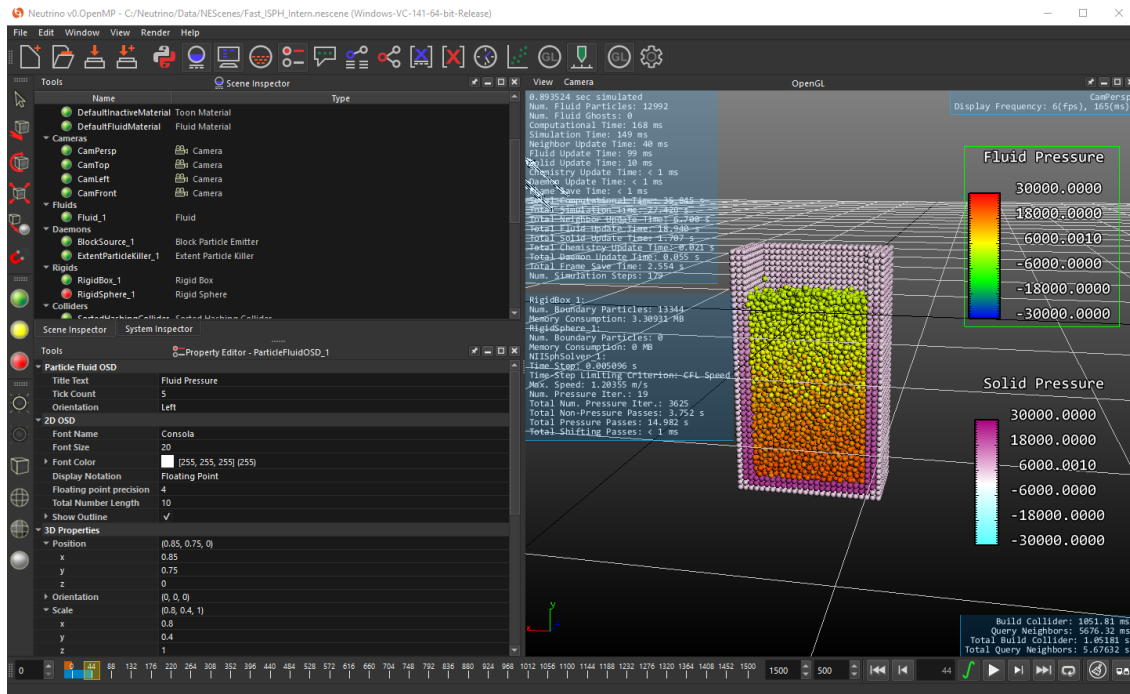


Figure B.7 – Environnement de simulation Neutrino

du mouvement du fluide. Neutrino possède donc une interface graphique qui lui est propre et qui est représenté sur la figure B.7.

Caractéristiques :

- Licence : Pas disponible sur le marché (Commerciale)
- Rendu graphique : OpenGL
- Librairie pour le calcul de la dynamique :
 - Dynamique des solides : Bullet, interne
 - Dynamique des fluides : Partie intégrante du logiciel
- Middleware : non
- Script : Python
- Dernière version : 1.4.2 (2019)

Annexe C

Méthodes d'intégration numérique

En mécanique, bien souvent on s'intéresse à l'évolution des objets dans l'espace au cours du temps. Très facilement, on est capable de connaître l'accélération instantanée d'un objet en faisant le bilan des forces s'appliquant sur un solide, grâce au principe fondamental de la dynamique ;

$$\begin{aligned}\sum \vec{F} &= m \vec{a} \\ \sum \vec{M}_A &= J_A \vec{\alpha}\end{aligned}\tag{C.1}$$

Avec \vec{F} les différentes forces s'appliquant sur le solide, m la masse du solide, \vec{a} l'accélération du solide, \vec{M}_A l'ensemble des moments projetés au point A , J_A la matrice d'inertie du solide au point A et $\vec{\alpha}$ l'accélération angulaire autour du point A . Pour connaître les effets de ces accélérations instantanées sur la position du solide il faut alors résoudre le problème suivant ¹ :

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0\tag{C.2}$$

Avec $y'(t)$ la dérivée temporelle de $y(t)$ étant égale à une fonction continue $f(t, y)$ qui dépend du temps et de $y(t)$ qui en t_0 vaut y_0 . L'objectif est de déterminer $y(t+h)$, h le pas de temps. Il faut faire cette intégration une première fois pour obtenir la vitesse de l'objet à partir de l'accélération puis une seconde fois pour obtenir sa position à partir de sa vitesse. Les méthodes décrites dans cette annexe permettent de résoudre ce problème, certaines sont plus simples à mettre en place et moins coûteuses en calcul et d'autres sont plus précises. Il existe deux types de méthode d'intégration :

- Les méthodes explicites, qui en partant du point de départ y_0 et de la dérivée première déterminent le prochain point en faisant une erreur connue que l'on va négliger.
- Les méthodes implicites, qui vont essayer de prédire ce qu'il se passera au point suivant et ainsi gagner en stabilité comparé au méthode explicite.

1. Ce problème fait partie des problèmes de Cauchy qui regroupent les problèmes constitués d'une équation différentielle dont on recherche une solution vérifiant une certaine condition initiale

C.1 Méthode d'intégration explicite/progressive

Le principe des méthodes d'intégration explicite est d'approcher la valeur de l'intégrale de la dérivée temporelle à l'aide des valeurs connues. Les méthodes les plus répandues et les plus générales sont les méthodes de Runge-Kutta². Ces méthodes permettent d'obtenir une intégration numérique pour un ordre N défini. On dit qu'une méthode d'intégration numérique est d'ordre N si elle est exacte pour les polynômes de degré inférieur ou égal à N , et si elle est fautive pour au moins un polynôme de degré $N + 1$.

C.1.1 Méthode d'Euler ou de Runge-Kutta d'ordre 1

La méthode d'Euler est une méthode d'intégration du premier ordre. Cette méthode utilise le principe de dérivation numérique, aussi appelé de différence finie. Soit une fonction $y(t)$ de classe \mathcal{C}^1 sur $[a, b]$. L'intervalle $[a, b]$ est découpée en petits instants $[a, b] = [a, t_1, t_2, \dots, b]$, tous distants d'un pas constant h ³ de tel sorte que $t_{i+1} = t_i + h$ avec le point arbitraire $t_i \in [a, b]$. Dans ce cas trois méthodes permettent d'obtenir la dérivée $y'(t_i)$:

$$y'(t_i) = \frac{y(t_i + h) - y(t_i)}{h} \quad \text{Différence finie progressive} \quad (\text{C.3})$$

$$y'(t_i) = \frac{y(t_i) - y(t_i - h)}{h} \quad \text{Différence finie rétrograde} \quad (\text{C.4})$$

$$y'(t_i) = \frac{y(t_i + h) - y(t_i - h)}{2h} \quad \text{Différence finie centrée} \quad (\text{C.5})$$

Ces 3 méthodes de dérivation sont illustrées sur la figure C.1. En noir on a la fonction $y(t)$. Trois droites représentent les différentes différences finies en t_i . Comme on peut le constater,

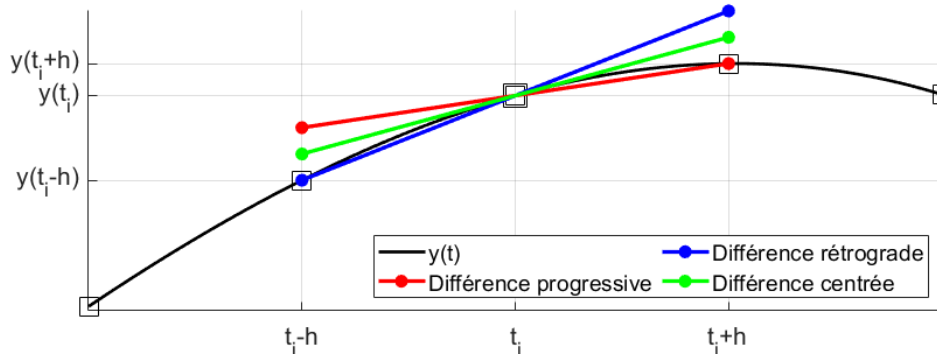


Figure C.1 – Dérivée de $y(t_i)$ avec trois méthodes de différenciation finie.

pour une fonction $y(t)$ qui n'est pas d'ordre 1, ces trois dérivées diffèrent. La dérivée progressive passe par le point $y(t_i)$ et le point $y(t_i + h)$ mais ne prend pas en compte le point précédent ($y(t_i - h)$). La dérivée rétrograde passe par ce point précédent mais pas par le point $y(t_i + h)$ et la dérivée centrée est la pente entre $y(t_i + h)$ et $y(t_i - h)$.

2. Ces méthodes sont ainsi nommées en l'honneur des mathématiciens Carl Runge et Martin Wilhelm Kutta. Lesquels élaborèrent ces méthodes en 1901.

3. Ceci représente le pas d'intégration. Souvent, ce pas correspondra au pas de temps noté Δt . Il est noté dans cet annexe h car cette méthode est applicable pour toute fonction numérique

L'objectif de nos méthodes d'intégration est de résoudre le problème C.2. Connaissant une valeur initiale $y(t_i)$ et en cherchant la valeur $y(t_i + h)$ nous utilisons la formule de différence finie progressive (Équation C.3) et obtenons :

$$y_{n+1}(t_i + h) = y_n(t_i) + h y'_n(t_i, y_n) \quad (\text{C.6})$$

Ici nous rappelons que y'_n peut dépendre de la valeur de y_n . Cette méthode d'intégration est illustrée sur la figure C.2. La valeur de y_{n+1} est obtenue par intégration numérique. On

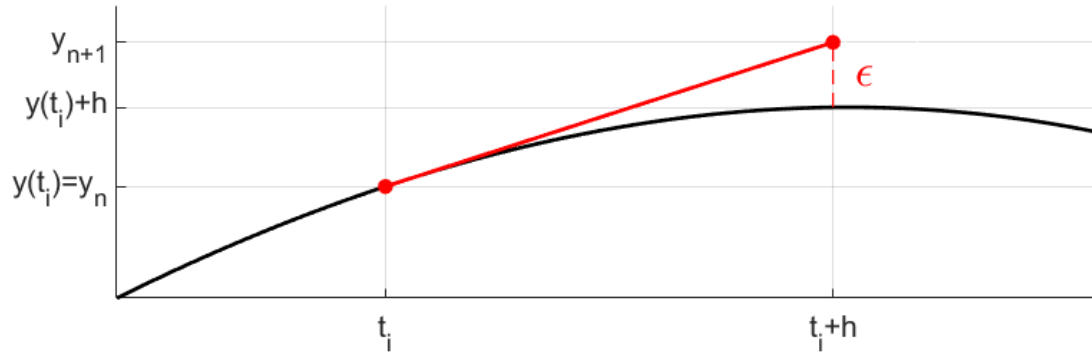


Figure C.2 – Illustration de la méthode d'intégration d'Euler

constate que cette valeur $y_{n+1}(t_i + h)$ diffère de la valeur de $y(t_i + h)$ et qu'il existe une erreur ϵ faite lors de l'intégration. Cette erreur ϵ faite sur un pas d'intégration peut être obtenue par les développements limités :

$$\begin{aligned} y(t_i + h) &= \underbrace{y(t_i) + h y'(t_i)}_{y_{n+1}} + \frac{1}{2} h^2 y''(t_i) + \mathcal{O}(h^3) \\ \epsilon &= y(t_i + h) - y_{n+1} = \frac{1}{2} h^2 y''(t_i) + \mathcal{O}(h^3) \end{aligned} \quad (\text{C.7})$$

Cette erreur ϵ dépend donc fortement du pas de temps h mais également de la dérivée seconde de la fonction $y(t_i)$.

C.1.2 Méthode général de Runge-Kutta

Les méthodes de Runge-Kutta sont des méthodes d'analyse numérique d'approximation de solutions d'équations différentielles. On rencontre couramment deux méthodes de Runge Kutta : La méthode d'ordre 1 qui est en fait la méthode d'Euler et la méthode d'ordre 4. Ces méthodes permettent de résoudre le problème décrit par l'équation C.2. La solution exacte de ce problème est :

$$y_{n+1} = y(t_i + h) = \underbrace{y(t_i)}_{=y_n} + \int_{t_i}^{t_i+h} \underbrace{f(t, y(t))}_{=y'(t)} dt \quad (\text{C.8})$$

Le principe des méthodes de Runge-Kutta est d'approcher la valeur de l'intégrale avec une méthode de quadrature :

$$\int_{t_i}^{t_i+h} f(t, y(t)) dt = \sum_{k=1}^m b_k \kappa_k \quad (\text{C.9})$$

Avec b_k des valeurs constantes qui peuvent être obtenues en suivant un tableau de Butcher et dépendent de m et κ_k des valeurs intermédiaires de la fonction $f(t)$:

$$\begin{aligned}\kappa_1 &= f(t_n, y_n) \\ \kappa_2 &= f(t_n + c_2 h, y_n + h a_{21} k_1) \\ \kappa_3 &= f(t_n + c_3 h, y_n + h(a_{31} k_1 + a_{32} k_2)) \\ &\vdots\end{aligned}\tag{C.10}$$

$$\kappa_l = f\left(t_n + c_l h, y_n + h \sum_{j=1}^{l-1} a_{lj} k_j\right)\tag{C.11}$$

$$\vdots\tag{C.12}$$

$$\kappa_m = f\left(t_n + c_m h, y_n + h \sum_{j=1}^{m-1} a_{mj} k_j\right)\tag{C.13}$$

Le tableau de Butcher⁴ est un moyen mnémotechnique qui relie les coefficients a_{ij} , c_l et b_k ainsi⁵ :

0					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
\vdots	\vdots		\ddots		
c_m	a_{m1}	a_{m2}	\cdots	$a_{m\ m-1}$	
	b_1	b_2	\cdots	b_{m-1}	b_m

Pour que la méthode de Runge-Kutta soit consistante il faut choisir :

$$c_l = \sum_{j=1}^{l-1} a_{lj}\tag{C.14}$$

Pour que la méthode de Runge Kutta soit consistante avec une intégration d'ordre 1, il faut :

$$\sum_{j=1}^m b_j = 1\tag{C.15}$$

D'autres conditions s'ajoutent pour le choix des constantes a_{ij} , c_l et b_k selon l'ordre de l'intégrateur souhaité. Pour plus de détails sur ces méthodes voir Butcher [But16].

Pour un ordre $p < 5$, m correspond à l'ordre et au nombre d'étapes d'intégration de Runge Kutta. Pour un ordre $p \geq 5$, cette relation n'est plus vrai et le nombre d'étapes d'intégration $s \geq p + 1$ [But16].

4. Nommé en l'honneur de John C. Butcher, un mathématicien Néo-Zélandais né en 1933, qui s'est particulièrement investi dans la résolution des problèmes de Runge-Kutta

5. Le tableau présenté ici est un cas particulier pour les méthodes explicites, où la partie triangulaire supérieure du tableau est nulle et donc ici laissée vide. Dans le cas de méthode implicite ou semi-implicite, cette partie doit être remplie.

C.1.2.1 Méthode de Runge-Kutta d'ordre 2

Ce schéma d'intégration est couramment appelé schéma de prédiction-correction explicite ou méthode du point milieu. Le principe est d'estimer la dérivée au milieu du pas d'intégration :

$$y_{n+\frac{1}{2}} = y_n + \frac{h}{2} f(t_i, y_n) = y_n + \frac{h}{2} y'_n \quad (\text{C.16})$$

$$y'_{n+\frac{1}{2}} = f\left(t_i + \frac{h}{2}, y_{n+\frac{1}{2}}\right) \quad (\text{C.17})$$

Enfin l'intégration est faite à partir de cette estimation de dérivée estimée :

$$y_{n+1} = y_n + h y'_{n+\frac{1}{2}} \quad (\text{C.18})$$

Cette méthode d'intégration est représentée sur la figure C.3, où nous réalisons l'intégration de la même courbe $y(t)$ au même point t_i qu'avec la méthode d'Euler. Comme on peut le voir

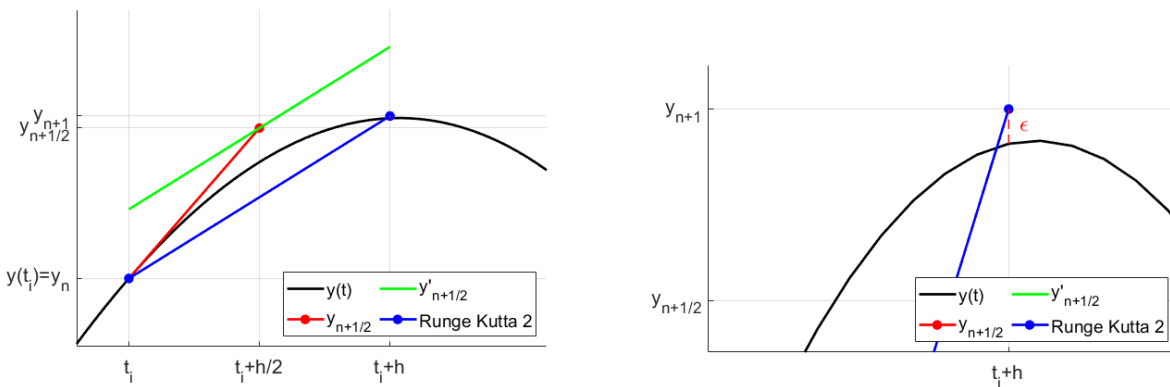


Figure C.3 – Méthode d'intégration de Runge-Kutta d'ordre 2. A droite : zoom sur le point $t_i + h$ pour montrer l'erreur

sur ces figures la méthode de Runge-Kutta 2 permet de diminuer significativement l'erreur faite lors de l'intégration. Ces erreurs seront comparées dans la section C.1.2.3. Ici l'erreur locale de troncature⁶ est de l'ordre de $\mathcal{O}(h^3)$. L'erreur de troncation globale⁷ est de l'ordre de $\mathcal{O}(h^2)$.

Pour ce cas le tableau de Butcher est simplement :

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ \hline & 0 & 1 \end{array}$$

La méthode présentée ici n'est pas la seule méthode de Runge-Kutta d'ordre 2 existante. Elle fait partie de la famille des méthodes d'intégration de Runge-Kutta d'ordre 2 existantes, qui peuvent être résumées avec le tableau de Butcher suivant :

$$\begin{array}{c|cc} 0 & & \\ \alpha & \alpha & \\ \hline & 1 - \frac{1}{2\alpha} & 1 - \frac{1}{2\alpha} \end{array}$$

6. L'erreur faite sur une itération.

7. L'erreur cumulée causée par de nombreuses itérations.

Une formule générale pour cette famille de méthode est donc :

$$y_{n+1} = y_n + h \left(\left(1 - \frac{1}{2\alpha}\right) y'_n + \frac{1}{2\alpha} f(t_n + \alpha h, y_n + \alpha h y'_n) \right) \quad (\text{C.19})$$

Dans cette famille $\alpha = 1/2$ est la méthode présentée jusqu'ici et $\alpha = 1$ est une autre méthode assez courante appelée la méthode de Heun⁸.

C.1.2.2 Méthode de Runge-Kutta d'ordre 4

La méthode de Runge-Kutta 4 (RK4) la plus courante, qui permet de résoudre ce problème, est décrite par l'équation C.20.

$$y_{n+1} = y_n + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4) \quad (\text{C.20})$$

où

$$\begin{aligned} k_1 &= f(t_n, y_n) \\ k_2 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) \\ k_3 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right) \\ k_4 &= f(t_n + h, y_n + hk_3) \end{aligned}$$

Ici la valeur y_{n+1} est calculée à partir de la valeur actuelle qui est actualisée après un pas h . La pente utilisée pour le calcul d'intégration est la moyenne pondérée de 4 pentes :

- k_1 : la pente du début de l'intervalle.
- k_2 : la pente du milieu de l'intervalle que l'on calcule en utilisant une valeur de y au milieu de l'intervalle déduite de k_1 .
- k_3 : la pente du milieu de l'intervalle que l'on calcule en utilisant une valeur de y au milieu de l'intervalle déduite cette fois de k_2 .
- k_4 : la pente de la fin de l'intervalle que l'on calcule en utilisant une valeur de y calculée à la fin de l'intervalle avec k_3 .

Ici l'erreur locale de troncature est de l'ordre de $\mathcal{O}(h^5)$. L'erreur de troncature globale est de l'ordre de $\mathcal{O}(h^4)$.

Pour ce cas le tableau de Butcher est :

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

Une autre méthode courante parmi les méthodes de Runge-Kutta d'ordre 4, appelée

8. Karl Heun était un mathématicien allemand au début du xx^e siècle.

méthode des $3/8$ de Simpson, est :

0				
$\frac{1}{3}$	$\frac{1}{3}$			
$\frac{2}{3}$	$-\frac{1}{3}$	1		
1	1	-1	1	
	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$

Cette méthode est plus précise que la première méthode décrite mais puisqu'il y a plus de valeurs non-nulles dans le tableau de Butcher, il y a plus d'opérations flottantes à réaliser pour exploiter cette méthode à chaque pas de temps.

C.1.2.3 Comparaison de l'erreur faite avec les méthodes de Runge-Kutta explicite

Nous allons comparer ces méthodes avec une fonction $y(t)$ ne dépendant pas des valeurs précédentes. Cette fonction est choisie volontairement sinusoïdale pour ne pas être approchée par des fonctions polynomiales :

$$y(t) = \sin(2t^2) + 0.2t \quad (\text{C.21})$$

Cette fonction a une dérivée connue qui est utilisée pour calculer les dérivées temporelles :

$$y'(t) = 4t \cos(2t^2) + 0.2 \quad (\text{C.22})$$

En partant du point initial $t = 0.75$ et en utilisant un pas $h = 0.025$, nous observons l'évolution des courbes avec différentes méthodes d'intégration sur la figure C.4. On peut constater que

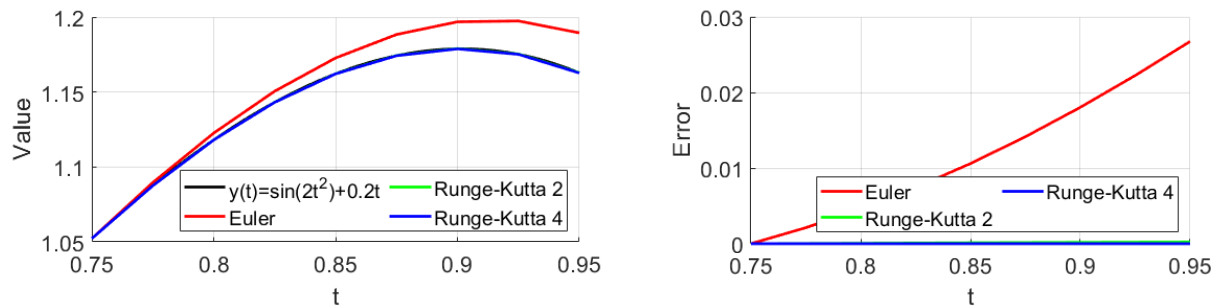


Figure C.4 – Comparaison des méthodes d'intégration de Runge-Kutta. A droite : Différence entre la courbe obtenue avec la méthode et la courbe $y(t)$

l'erreur est très grande avec la méthode d'Euler en comparaison des erreurs avec les méthodes d'ordre supérieur. L'erreur finale avec la méthode d'Euler est $\epsilon_{Euler,fin} = 0.027$, celle avec la méthode Runge-Kutta 2 est $\epsilon_{RK2,fin} = 2.24 \times 10^{-4}$ et celle avec la méthode Runge-Kutta 4 est $\epsilon_{RK4,fin} = 4.07 \times 10^{-8}$.

C.1.3 Double intégration avec Runge Kutta 4

Dans le cas d'une double intégration, on voudrait démontrer que l'on peut résoudre le problème de l'Équation C.23 avec les équations C.24.

$$y'' = f(t, y, y'), \quad y(t_0) = y_0, \quad y'(t_0) = y'_0 \quad (\text{C.23})$$

$$\begin{aligned}
y'_{n+1} &= y'_n + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4) \\
y_{n+1} &= y_n + hy'_n + \frac{h^2}{6} (k_1 + k_2 + k_3)
\end{aligned} \tag{C.24}$$

où

$$\begin{aligned}
k_1 &= f(t_n, y_n, y'_n) \\
k_2 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}y'_n, y'_n + \frac{h}{2}k_1\right) \\
k_3 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}y'_n + \frac{h^2}{4}k_1, y'_n + \frac{h}{2}k_2\right) \\
k_4 &= f\left(t_n + h, y_n + hy'_n + \frac{h^2}{2}k_2, y'_n + hk_3\right)
\end{aligned}$$

Démonstration Pour cela, nous allons poser le système C.25 et nous allons utiliser la méthode simple d'intégration RK4 sur le système $u' = g(t, u)$ et déterminer u_{n+1} .

$$u = \begin{pmatrix} y' \\ y \end{pmatrix}, \quad u' = \begin{pmatrix} y'' \\ y' \end{pmatrix} = g(t, u) = \begin{pmatrix} f(t, y, y') \\ y' \end{pmatrix} \tag{C.25}$$

En appliquant la méthode RK4 Simple on obtient l'Équation C.26.

$$u_{n+1} = u_n + \frac{h}{6} (k_{1,u} + 2k_{2,u} + 2k_{3,u} + k_{4,u}) \tag{C.26}$$

où

$$\begin{aligned}
k_{1,u} &= g(t_n, u_n) = \begin{pmatrix} f(t_n, y_n, y'_n) \\ y'_n \end{pmatrix} \\
k_{2,u} &= g\left(t_n + \frac{h}{2}, u_n + \frac{h}{2}k_{1,u}\right) \\
k_{3,u} &= g\left(t_n + \frac{h}{2}, u_n + \frac{h}{2}k_{2,u}\right) \\
k_{4,u} &= g\left(t_n + h, u_n + hk_{3,u}\right)
\end{aligned}$$

Il faut maintenant remplacer u_n et la fonction g . Développons $k_{2,u}$:

$$\begin{aligned}
u_n + \frac{h}{2}k_{1,u} &= \begin{pmatrix} y'_n \\ y_n \end{pmatrix} + \frac{h}{2} \begin{pmatrix} f(t_n, y_n, y'_n) \\ y'_n \end{pmatrix} \\
k_{2,u} &= g\left(t_n + \frac{h}{2}, u_n + \frac{h}{2}k_{1,u}\right) = \begin{pmatrix} f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}y'_n, y'_n + \frac{h}{2}f(t_n, y_n, y'_n)\right) \\ y'_n + \frac{h}{2}f(t_n, y_n, y'_n) \end{pmatrix}
\end{aligned}$$

On pose :

$$\begin{aligned}k_{1,y} &= f(t_n, y_n, y'_n) \\k_{2,y} &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}y'_n, y'_n + \frac{h}{2}k_{1,y}\right)\end{aligned}$$

On réécrit alors $k_{1,u}$ et $k_{2,u}$:

$$\begin{aligned}k_{1,u} &= g(t_n, u_n) = \begin{pmatrix} f(t_n, y_n, y'_n) \\ y'_n \end{pmatrix} = \begin{pmatrix} k_{1,y} \\ y'_n \end{pmatrix} \\k_{2,u} &= g\left(t_n + \frac{h}{2}, u_n + \frac{h}{2}k_{1,u}\right) = \begin{pmatrix} f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}y'_n, y'_n + \frac{h}{2}k_{1,y}\right) \\ y'_n + \frac{h}{2}k_{1,y} \end{pmatrix} = \begin{pmatrix} k_{2,y} \\ y'_n + \frac{h}{2}k_{1,y} \end{pmatrix}\end{aligned}$$

Donc :

$$\begin{aligned}u_n + \frac{h}{2}k_{2,u} &= \begin{pmatrix} y'_n \\ y_n \end{pmatrix} + \frac{h}{2} \begin{pmatrix} k_{2,y} \\ y'_n + \frac{h}{2}k_{1,y} \end{pmatrix} \\k_{3,u} &= g\left(t_n + \frac{h}{2}, u_n + \frac{h}{2}k_{2,u}\right) = \begin{pmatrix} f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}y'_n + \frac{h^2}{4}k_{1,y}, y'_n + \frac{h}{2}k_{2,y}\right) \\ y'_n + \frac{h}{2}k_{2,y} \end{pmatrix} = \begin{pmatrix} k_{3,y} \\ y'_n + \frac{h}{2}k_{2,y} \end{pmatrix}\end{aligned}$$

Avec :

$$k_{3,y} = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}y'_n + \frac{h^2}{4}k_{1,y}, y'_n + \frac{h}{2}k_{2,y}\right)$$

Enfin on calcule $k_{4,u}$:

$$\begin{aligned}u_n + hk_{3,u} &= \begin{pmatrix} y'_n \\ y_n \end{pmatrix} + h \begin{pmatrix} k_{3,y} \\ y'_n + \frac{h}{2}k_{2,y} \end{pmatrix} \\k_{4,u} &= g(t_n + h, u_n + hk_{3,u}) = \begin{pmatrix} f\left(t_n + h, y_n + hy'_n + \frac{h^2}{2}k_{2,y}, y'_n + hk_{3,y}\right) \\ y'_n + hk_{3,y} \end{pmatrix} = \begin{pmatrix} k_{4,y} \\ y'_n + hk_{3,y} \end{pmatrix}\end{aligned}$$

Avec :

$$k_{4,y} = f\left(t_n + h, y_n + hy'_n + \frac{h^2}{2}k_{2,y}, y'_n + hk_{3,y}\right)$$

On obtient alors, en remplaçant les valeurs de $k_{1,u}, k_{2,u}, k_{3,u}$ et $k_{4,u}$ dans l'Équation C.26, le système C.27.

$$\begin{pmatrix} y'_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} y'_n \\ y_n \end{pmatrix} + \frac{h}{6} \left(\begin{pmatrix} k_{1,y} \\ y'_n \end{pmatrix} + 2 \begin{pmatrix} k_{2,y} \\ y'_n + \frac{h}{2}k_{1,y} \end{pmatrix} + 2 \begin{pmatrix} k_{3,y} \\ y'_n + \frac{h}{2}k_{2,y} \end{pmatrix} + \begin{pmatrix} k_{4,y} \\ y'_n + hk_{3,y} \end{pmatrix} \right) \quad (\text{C.27})$$

En développant on obtient bien le système présenté à l'Équation C.24.

C.2 Méthode d'intégration implicite/rétrograde

Le principe des méthodes d'intégration implicite est d'approcher la valeur de l'intégrale de la dérivée temporelle en utilisant des informations sur les dérivées temporelles (éventuellement estimées) au pas de temps suivant. L'obtention de ces informations est assez complexe et nécessite souvent des opérations d'inversion de matrice très coûteuses en opérations flottantes. Les méthodes implicites permettent souvent de choisir de plus grand pas que pour la méthode explicite : ceci peut permettre de diminuer le temps de calcul global si le temps requis pour l'inversion du problème est inférieur au temps pour calculer les nombreux pas de temps de l'algorithme explicite. De plus, ces méthodes permettent, en général, d'obtenir des simulations plus stables que les méthodes explicites, car elles permettent de prédire et prendre en compte les changements brusques.

C.2.1 Méthode d'Euler implicite

Dans la section C.1.1, trois différentes méthodes de dérivation numérique ont été montrées. La différence finie progressive (Équation C.3) est utilisée pour la méthode d'Euler explicite. Dans le cas de la méthode d'Euler implicite, c'est la différence finie rétrograde (Équation C.4) que l'on utilise. A partir de cette équation on obtient :

$$y_{n+1}(t_i + h) = y_n(t_i) + hy'_{n+1}(t_i + h, y_{n+1}) \quad (\text{C.28})$$

Comme on peut le voir, l'utilisation de ce schéma nécessite d'estimer au préalable la dérivée $y'_{n+1}(t_i + h, y_{n+1})$ suivante. Ce schéma est inclus dans les méthodes de Runge-Kutta d'ordre 1. Le comportement de ce schéma d'intégration comparé au schéma d'Euler explicite est montré sur la figure C.5. Nous réutilisons la fonction C.21 pour comparer les méthodes d'Euler mais sur une plage de valeur plus longue. Sur cette figure, on se rend compte que l'erreur

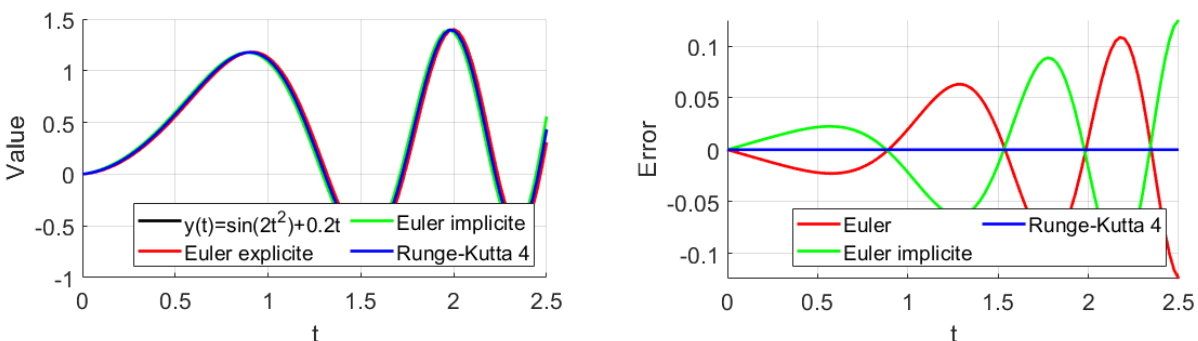


Figure C.5 – Comparaison des résultats avec une fonction temporelle et Euler explicite et implicite. A droite : Différence entre la courbe obtenue avec la méthode et la courbe $y(t)$

faite avec la méthode d'Euler implicite est du même ordre de grandeur que celle avec Euler explicite puisque l'on utilise une fonction ayant une dérivée connue. La différence entre ces deux méthodes est qu'avec la méthode d'Euler implicite on est en avance de phase sur le signal $y(t)$. En pratique, la fonction y' dépend souvent de $y(t)$. Ainsi, faire une erreur sur la prochaine valeur de $y(t)$ peut être fatal pour toute la suite de l'intégration.

La méthode d'Euler explicite fait une erreur qui se propage dans le temps. Si on additionne les deux signaux d'erreur on obtient une valeur non nulle qui pour cette fonction sinusoïdale oscille et dont l'amplitude augmente au fil du temps, comme représenté sur la figure C.6.

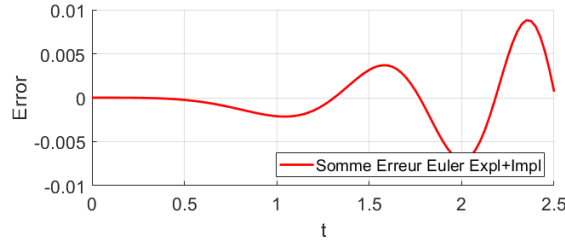


Figure C.6 – Propagation de l'erreur avec la méthode explicite

Le schéma d'Euler implicite amène à de grandes erreurs sur la valeur de y_n (ici nous l'avons comparé à Runge-Kutta 4 explicite), nous allons voir dans la suite une méthode implicite d'ordre 2.

C.2.2 Méthode d'Euler semi-implicite

C.2.3 Méthode de Runge-Kutta implicite d'ordre supérieur

Nous allons utiliser les tableaux de Butcher pour présenter deux méthodes de Runge-Kutta implicite. Ces méthodes sont intéressantes puisqu'elles permettent une meilleure précision comparées à la méthode d'Euler implicite⁹. La première méthode est relativement simple :

$$\begin{array}{c|cc} \frac{1}{3} & \frac{1}{3} & 0 \\ 1 & 1 & 0 \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array}$$

Cette méthode est une méthode de Runge-Kutta d'ordre 3.

La seconde méthode est dérivée de la méthode de quadrature de Gauss-Legendre et permet avec seulement 2 étapes d'intégration d'obtenir un intégrateur de Runge-Kutta d'ordre 4.

$$\begin{array}{c|cc} \frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

D'autres méthodes, qui sont plus simples à implémenter et qui ne font pas partie des méthodes de Runge-Kutta, existent.

C.2.4 Méthode de Verlet et « Velocity Verlet »

La méthode de Verlet¹⁰ est une méthode d'ordre 2 qui ne fait pas partie des méthodes de Runge-Kutta, parce qu'elle utilise la dérivée seconde de la fonction y que l'on veut intégrer. Cette méthode est particulièrement intéressante dans le cas d'une simulation mécanique, puisque l'on est capable de déterminer l'accélération de l'objet avec le principe fondamental de la dynamique C.1. Cette méthode se découpe en deux sous-méthodes : la méthode dite élémentaire et la méthode « Velocity Verlet ».

9. Sur un pas de temps l'erreur est du même ordre de grandeur qu'avec les méthodes de Runge-Kutta 2 explicite, comme nous l'avons montré pour les méthodes d'Euler implicite et explicite.

10. Cette méthode a été développée par le physicien français Loup Verlet en 1967 [Ver67]

C.2.4.1 Méthode d'intégration élémentaire de Verlet

La méthode d'intégration élémentaire de Verlet réduit le taux d'erreurs introduites par l'intégration en calculant la valeur de $y(t)$ au pas de temps suivant à partir des valeurs courante et précédente, sans faire appel à la dérivée $y'(t)$. En utilisant deux développements de Taylor de $y(t)$ aux instants $t + h$ et $t - h$:

$$\begin{aligned} y(t + h) &= y(t) + hy'(t) + \frac{h^2}{2}y''(t) + \frac{h^3}{6}y^{(3)}(t) + \mathcal{O}(h^4) \\ y(t - h) &= y(t) - hy'(t) + \frac{h^2}{2}y''(t) - \frac{h^3}{6}y^{(3)}(t) + \mathcal{O}(h^4) \end{aligned}$$

Avec $y^{(3)}(t)$ la dérivée 3^e de y . En faisant l'addition de ces deux lignes on obtient :

$$y(t + h) = 2y(t) - y(t - h) + h^2y''(t) + \mathcal{O}(h^4) \quad (\text{C.29})$$

On observe que l'un des grands intérêts de cette méthode est qu'elle est localement plus précise d'un ordre. Le point faible de cette méthode est qu'elle nécessite le stockage de la valeur de $y(t - h)$. En pratique, avec de nombreux éléments, le stockage de cette information est coûteux.

C.2.4.2 Velocity Verlet

L'un des points faibles de la méthode précédente est le stockage de la valeur à l'instant précédent. Le schéma Velocity Verlet se base sur la valeur actuelle de la vitesse et de l'accélération. Il est introduit en 1967 dans l'annexe de l'article de Swope [SABW82], et il est considéré comme équivalent à la méthode élémentaire. Il se décrit ainsi :

$$y(t + h) = y(t) + h y'(t) + \frac{h^2}{2} y''(t) \quad (\text{C.30})$$

$$y'(t + h) = y'(t) + \frac{h}{2} (y''(t) + y''(t + h)) \quad (\text{C.31})$$

Avec ce schéma, l'erreur est du même ordre que celle du schéma d'intégration élémentaire de Verlet. Pour implémenter ce schéma il faut donc réaliser les étapes suivantes :

1. Calculer $y(t + h)$ avec l'équation C.30
2. Déterminer une valeur de $y''(t + h)$ à partir de $y(t + h)$
3. Calculer la nouvelle vitesse $y'(t + h)$ avec l'équation C.31

Cette méthode est particulièrement adaptée à l'intégration temporelle du mouvement d'un système mécanique puisque l'évaluation de la nouvelle accélération revient à faire un bilan des forces au nouvel emplacement.

C.2.5 Intégrateur symplectique

La notion d'intégrateur symplectique est fortement liée à la mécanique hamiltonienne (voir section 3.1.2.1). Un intégrateur symplectique est une transformation canonique¹¹ qui

11. En mécanique hamiltonienne, une transformation canonique est un changement des coordonnées canoniques $(q, p, t) \leftarrow (Q, P, t)$ qui conserve la forme des équations de Hamilton, sans pour autant nécessairement conserver le Hamiltonien en lui-même.

permet de s'assurer que l'on conserve la structure $dp \wedge dq$ au cours du temps, avec ici \wedge qui ne représente pas le produit vectoriel mais bien le produit extérieur. Pour les problèmes mécaniques, cela permet de s'assurer que l'on reste dans l'espace de phase et que l'on conserve l'énergie mécanique lors de l'intégration ¹².

Les intégrateurs implicites que nous avons présentés jusqu'ici sont tous des intégrateurs symplectiques à condition de choisir un pas de temps fixe. Symplectique ne signifie pas forcément réversible temporellement. La méthode d'Euler implicite n'est pas réversible temporellement. Les méthodes d'ordre supérieur sont par contre réversibles dans le temps et si l'on applique un pas de temps négatif on récupère la position précédente.

12. Une partie de cette énergie peut toujours être dissipée avec d'autres forces comme des forces de frottements

C.3 Conclusion : Méthode explicite à pas variable ou méthode implicite à pas fixe ?

Finalement, les méthodes explicite et implicite ont des avantages et des inconvénients. Nous allons les énoncer pour pouvoir déterminer la meilleure méthode d'intégration pour nos simulations.

Pour les méthodes explicites, il est possible de connaître l'erreur faite au cours d'un pas de temps. En pratique, il est possible de choisir la taille du pas de temps à chaque itération, par exemple en fonction des collisions ou de la courbure de trajectoire, ce qui permet de diminuer fortement le temps de calcul. Néanmoins, une dérive est faite à chaque itération de la simulation. Sur un temps assez long, si on calcule un grand nombre de pas de temps, cette erreur devient significative. La méthode explicite est également plus simple à mettre en place car elle ne nécessite aucune prédiction.

Les méthodes implicites ont l'avantage, si l'on considère un pas de temps fixe, d'être symplectiques, c'est à dire de conserver les énergies des systèmes mécaniques considérés. Ceci implique qu'elles sont bien plus précises si l'on souhaite réaliser une simulation avec de nombreuses itérations. Vu que ces méthodes se basent sur des prédictions, ces méthodes sont beaucoup plus stables que les méthodes explicites. Ces méthodes permettent ainsi de choisir des pas de temps beaucoup plus grands que les méthodes explicites.

Dans notre cas, celui de la simulation de mouvement de solide dans un fluide dynamique, nous allons faire beaucoup d'itérations. Notamment, parce que l'on réalisera beaucoup de pas de temps pour rendre compte de tous les effets dans le fluide. Les méthodes implicites sont à préconiser dans ce cas. La dynamique des solides dans Neutrino utilise l'algorithme « Velocity Verlet ». La dynamique des particules de fluide utilise un algorithme d'Euler semi-implicite.

C.4 Adaptation dans les moteurs physiques des jeux vidéo

Pour les jeux vidéo, l'important est de permettre une simulation de la dynamique d'un grand ensemble de solides en temps réel pour assurer une certaine fluidité de l'animation affichée au joueur. Les moteurs physiques dédiés à cet effet utilisent donc des méthodes ne requérant que peu d'opérations flottantes. La précision des autres calculs numériques (e.g. collisions, précision numérique en fonction de la taille de mémoire allouée, etc.) fait qu'on a rarement besoin de dépasser une précision d'intégration supérieure à l'ordre 4. Les méthodes les plus employées sont les méthodes de Runge-Kutta d'ordre 1 : Euler explicite et Euler semi-implicite en particulier, et la méthode explicite de Runge-Kutta d'ordre 4. On retrouve cela, par exemple, dans MuJoCo [TET12], où l'utilisateur peut choisir entre Euler semi-implicite et Runge-Kutta 4 en fonction de son application comme évoqué dans la section C.3. Nous allons détailler une partie du fonctionnement de trois moteurs physiques : Bullet, PhysX et FleX car ils ont été utilisés dans cette thèse.

Dans les deux cas, le principe est d'utiliser un algorithme de calcul de force prenant en compte les collisions puis de mettre à jour cette position. Ce principe est bien décrit dans l'algorithme 1 de l'article de Macklin [MMCK14]. Comme on peut le voir sur cet article FleX utilise pour l'intégration numérique : Euler semi-implicite. Dans PhysX, c'est l'intégrateur Velocity Verlet qui est utilisé [MHL⁺09]. Dans Bullet,

Pour les moteurs physiques actuels, permettre une détection rapide et efficace des collisions se produisant au cours d'un pas de temps est le point essentiel pour assurer la simulation temps réel, car c'est probablement le plus coûteux en temps de calcul. L'autre point important pour ces moteurs physiques, est d'assurer une simulation stable. Ainsi des techniques permettant d'optimiser le calcul des contraintes à appliquer aux différents éléments de la simulation au cours d'un pas de temps sont développées.

Les méthodes d'intégration utilisées devraient être très précises, néanmoins les algorithmes de détection de collisions et de stabilisation de la simulation peuvent parfois rajouter des erreurs. Par exemple, dans plusieurs algorithmes des fonctions de sous-relaxation sont appliquées. Une fonction de sous-relaxation est une fonction qui applique un facteur de relaxation α_{relax} au résultat final et prend en compte le résultat précédent pour la valeur $n + 1$:

$$y_{n+1} = \alpha_{relax} y_{n+1}^* + (1 - \alpha_{relax}) y_n \quad (\text{C.32})$$

Avec $0 < \alpha_{relax} < 1$ et y_{n+1}^* la valeur après prise en compte des forces et collisions s'appliquant sur le solide.

Essai de chute de cubes avec Bullet et comparaison avec des méthodes d'intégration classique Quatre essais ont été menés avec quatre intégrateurs différents : Bullet, l'intégrateur symplectique d'Euler (ordre 1), l'intégrateur symplectique de Verlet (ordre 2) et enfin la méthode d'intégration de Runge Kutta d'ordre 4 (non symplectique). Notre référence sera l'intégrateur symplectique de Verlet.

Le cube choisi a une masse de 12,1 kg, donc l'accélération que l'on doit obtenir est de l'ordre de 0.81 m.s^{-2} . Les courbes montrant la différence entre l'accélération obtenue et l'accélération obtenue avec l'intégrateur symplectique de Verlet sont présentées sur la figure C.7. Sur cette courbe, on peut observer en rouge la différence avec Bullet, en vert la différence avec Euler symplectique et en bleu la différence avec Runge-Kutta 4 et l'intégration du cube avec les

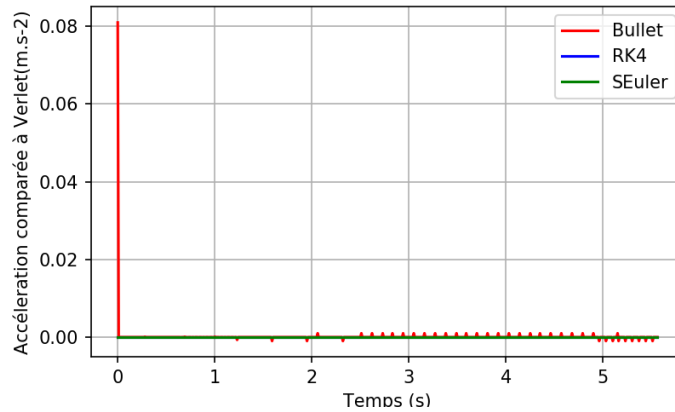


Figure C.7 – Différence d'accélération entre les méthodes d'intégration classique et le moteur physique Bullet

autres intégrateurs. Comme on peut le constater il y a un léger décalage au départ avec la trajectoire de Bullet. Ce décalage est dû à une fonction de Bullet permettant de stabiliser les simulations et limitant les accélérations brusques. Bullet prend en compte le bilan des forces mais aussi l'accélération instantanée du pas de temps précédent et fait une sous-relaxation (Équation C.32). L'erreur induite en accélération est de l'ordre de 10% sur le premier pas de temps.

Annexe D

Conception d'un balancier hydrodynamique

La conception du balancier est inspirée de balanciers déjà existants comme celui décrit dans Eng [ELLS08]. Contrairement à celui décrit dans celui-ci, le balancier conçu est prévu pour permettre des essais tailles réelles avec les ROV possédés par le laboratoire COSMER. Ces ROV sont parmi les plus courants : OpenROV, BlueROV et BlueROV 2. Le BlueROV 2 étant le ROV le plus lourd et le plus volumineux, le balancier est conçu pour supporter son poids une fois dans l'eau. Les données du BlueROV 2 utiles à la conception du balancier résumées dans le tableau D.1. Le mécanisme doit s'adapter aux bassins prévus pour les essais.


	Hauteur : 254 mm
	Largeur : 338 mm
	Longueur : 457 mm
	Poids max. dans l'air (avec ballasts) : 11 kg
	Flottabilité (sans Ballast) : 1.4 kg
	Flottabilité (avec Ballast) : 0.2 kg

Tableau D.1 – Données du BlueROV 2 utiles à la conception du balancier

La profondeur d'immersion des ROV doit l'être également. Ainsi, sa hauteur et largeur doivent être réglables. L'un des points sensibles de la conception de ce balancier est qu'il doit induire le moins de perturbations possibles sur la mesure des coefficients de trainée du ROV. Le coût de fabrication du dispositif est également un critère important pour le choix des éléments qui le composent. Une première série de mesures sera faite avec une sphère pour s'assurer de la fiabilité du dispositif expérimental.

D.1 Dispositif mécanique

Les éléments retenus pour réaliser la liaison pivot permettant la rotation de l'axe principal sont des roulements à billes. Ces roulements à billes permettent de limiter le frottement et d'assurer le guidage de la rotation. Un frottement existe néanmoins dans le roulement et une mesure de ce frottement sera nécessaire pour l'écartier lors de la détermination des forces de frottements et masses ajoutées dans l'eau. Le diamètre de l'axe principal est un élément important pour le choix des roulements. Cet axe doit être correctement dimensionné car il devra supporter le poids de l'objet à étudier. Il faudrait notamment limiter la flexion de la barre et faire en sorte que l'on reste dans la limite élastique. Nous allons étudier le cas d'une barre de 2 m de longueur encastrée de part en part avec une force appliquée au centre correspondant à la valeur du poids maximal que peut exercer le BlueROV dans l'air, voir figure D.1. Nous considérons donc ici que les roulements contraignent localement la pente de

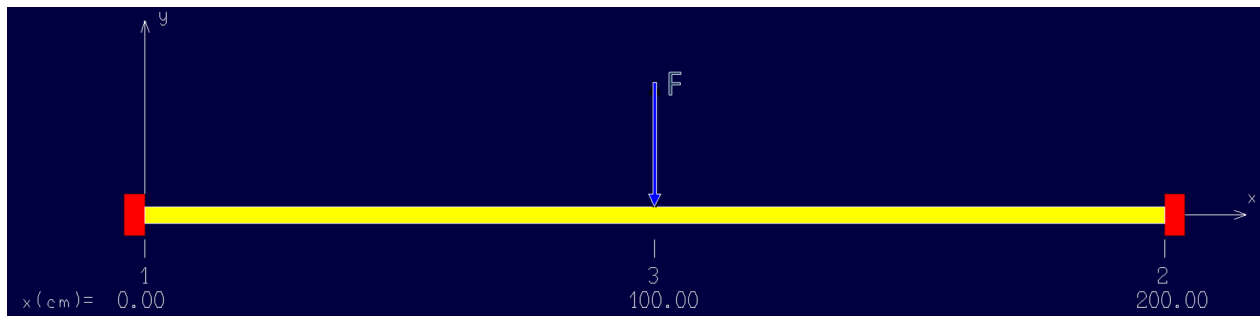


Figure D.1 – Calcul de la flèche maximale de la barre de flexion

la déformation de l'axe principal. La valeur de la flèche maximale peut être calculée ainsi :

$$f = -\frac{FL^3}{192EI_G} \quad (\text{D.1})$$

Avec f la flèche, F la force, L la longueur de la poutre, E le module d'Young du matériau, $I_G = \pi D^4/32$ le moment quadratique de la poutre et D le diamètre de la poutre. On choisit de limiter la flèche $f < 4\text{ cm}$ en considérant ce critère suffisant pour que l'axe reste dans sa limite élastique. Dans ce cas, on peut calculer le diamètre minimal de l'axe pour respecter cette contrainte dans le cas où le BlueROV est placé sur le système dans l'air :

$$D = \sqrt[4]{\frac{FL^3}{6\pi Ef}} \quad (\text{D.2})$$

En considérant une poutre en aluminium standard $E = 67500\text{ MPa}$ et un poids de $P = mg = 11 \times 9.81 = 107.91\text{ N}$, on obtient un diamètre minimal de $D > 10.8\text{ mm}$. Une barre de diamètre 16 mm est choisie. Dans la pratique, les roulements à bille permettent un léger rotulage au niveau de la liaison et donc la déformée et la flèche de cet axe sont différentes. Nous allons vérifier à l'aide du logiciel RDM Le Mans [DEB19] que les contraintes dans cette barre restent dans le domaine élastique et que la flexion maximale est bien inférieure à 4 mm . Nous considérons alors une poutre supportée par deux appuis et dont la pente de part et d'autre est contrainte à 1° . La figure D.2 montre la déformation de l'axe principal en fonction de cette nouvelle contrainte. La flèche maximale est de $3.071 < 4\text{ cm}$ ce qui est bien inférieur

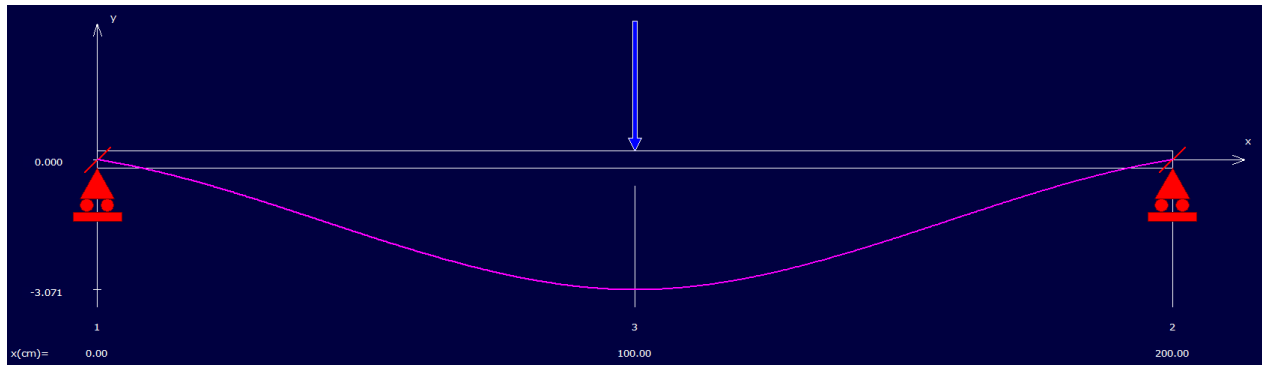


Figure D.2 – Déformation de l'axe principal

au critère défini précédemment. La figure D.3 montre la répartition des contraintes dans la poutre. La résistance élastique minimale pour les alliages d'aluminium série 1000 à 7000 est

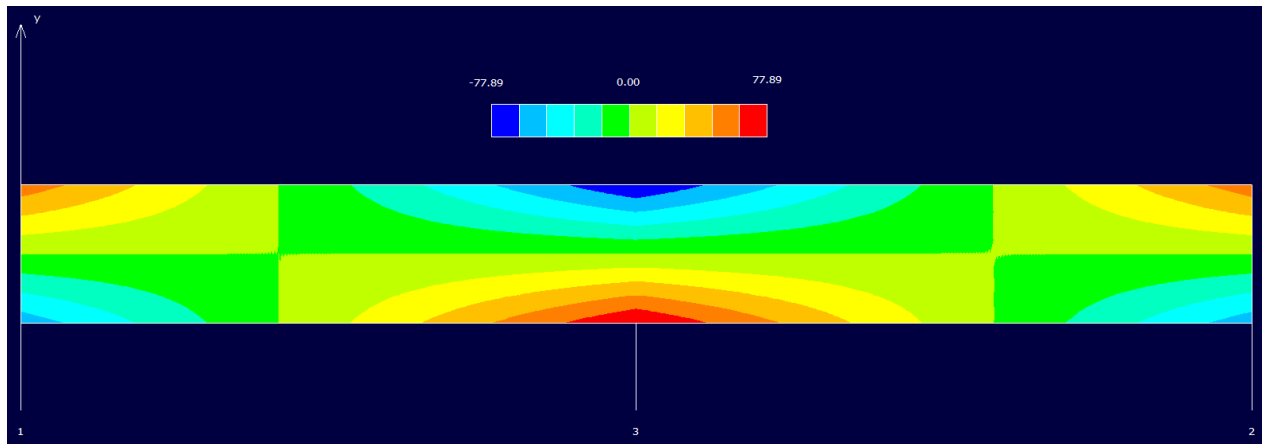


Figure D.3 – Répartition des contraintes locales dans l'axe principal (Contrainte en MPa)

de $R_e = 90 \text{ MPa}$. La valeur maximale de la contrainte dans l'axe n'atteint que 78 MPa . Il n'y aura donc pas de déformation plastique de l'axe principale s'il est soumis au poids du ROV dans l'air. Le roulement choisi est un roulement à bille linéaire, NSK-RHP LPB16.

La liaison entre l'axe principale et la tige d'attache est faite avec une noix de serrage. Les contraintes en traction sur la tige d'attache ne devrait pas poser de problème. On choisit d'utiliser une barre en aluminium de diamètre 8 mm . La contrainte en traction sur cette tige est d'à peine $\sigma = 4F/\pi D^2 = 21.5 \text{ kPa}$. Nous devons par contre vérifier que l'inertie de rotation de cette barre d'aluminium ramenée à l'axe de rotation est négligeable par rapport à la masse de l'objet que nous allons placer sur le balancier.

La figure 2.13 montre le dispositif expérimental en place dans le bassin d'essai avec l'objet que l'on souhaite étudier, une sphère, réalisée ici avec un médecine ball de 3 kg que l'on a rempli d'eau. Le médecine ball rempli d'air étant d'une densité inférieure à celle de l'eau : 560 kg.m^{-3}

D.2 Acquisition des données

Le système d'acquisition des données doit être suffisamment précis et rapide pour permettre une analyse du signal. Il faut choisir un couple capteur et carte électronique qui permette cela. Ce capteur doit être suffisamment étanche pour résister à l'humidité et une projection d'eau, il faudra donc une certaine résistance à l'eau (A minima respecter la norme IP 65). La carte électronique sera placée loin du bassin. La mesure de l'angle doit permettre de localiser avec une précision minimale de l'ordre du *mm* l'objet au bout du bras du balancier. Par exemple, pour une variation avec l'horizontale Δy , une rayon de balancier r et une variation d'angle $\Delta\theta$, comme représenté sur la figure D.4 :

$$\Delta y = r \sin \Delta\theta$$

$$\Delta\theta = \sin \frac{\Delta y}{r} \tag{D.3}$$

$$\tag{D.4}$$

Pour obtenir une précision $\Delta y < 1 \text{ mm}$ avec $r = 1 \text{ m}$ il faut choisir un capteur d'une précision

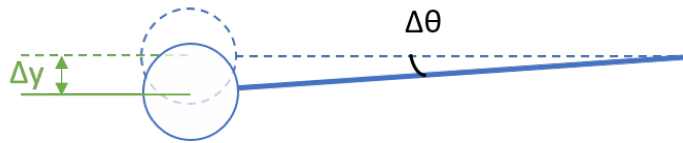


Figure D.4 – Schéma expliquant le lien entre précision angulaire et précision de la position de l'objet à étudier

d'au moins 0.057° . Le capteur choisi est le codeur rotatif WDGI 58B de Wachendorff, qui permet une résolution de 0.018° . La fréquence d'acquisition maximale est donnée par la carte d'acquisition. La carte choisie est PhidgetEncoder HighSpeed. L'ensemble permet une fréquence d'acquisition maximale de 100 Hz .

Annexe E

Algorithme SPH

E.1 Algorithme WCSPH explicite

Algorithm 1 WCSPH Solver

- 1: **for each** particle $a \in \mathcal{F}$ **do**
 - 2: Compute domains \mathcal{F}_a and \mathcal{S}_a
 - 3: Compute \mathcal{S}^* , $\forall a \in \mathcal{F}$ ($\mathcal{S}_a \subset \mathcal{S}^*$)
 - 4: Compute δt with CFL conditions
 - 5: **for each** particle $a \in \mathcal{F}$ **do**
 - 6: Compute $\forall b \in \mathcal{F}_a \cup \mathcal{S}_a$, $\vec{\nabla} W_{ab} = W'_{ab} \vec{e}_{ab}$
 - 7: Compute ρ_a (Équation 3.29)
 - 8: Compute p_a (Équation 3.35)
 - 9: Compute $\Theta_a = p_a/\rho_a^2$
 - 10: **if** using Shepard extrapolation **then**
 - 11: **for each** particle $s \in \mathcal{S}^*$ **do**
 - 12: Extrapolate ρ_s and \vec{v}_s (Équation 3.68 & Équation 3.76)
 - 13: **for each** particle $a \in \mathcal{F}$ **do**
 - 14: Compute non-pressure acceleration $\dot{\vec{v}}_a^P$ (Équation 3.37)
 - 15: **if** using Shepard extrapolation **then**
 - 16: **for each** particle $s \in \mathcal{S}^*$ **do**
 - 17: Compute p_s (Équation 3.69) and ρ_s using reverse (Équation 3.35)
 - 18: Compute $\Theta_s = p_s/\rho_s^2$
 - 19: **for each** solid S **do**
 - 20: Compute \vec{F}_s^P and \vec{F}_s^V depending on Shepard extrapolation or Mirroring
 - 21: **for each** particle $a \in \mathcal{F}$ **do**
 - 22: Compute $\dot{\vec{v}}_a$ depending on Shepard extrapolation (Équation 3.42) or Mirroring (Équation 3.73)
 - 23: Compute $\vec{v}_a = \vec{v}_a + \delta t \dot{\vec{v}}_a$
 - 24: Compute $\vec{r}_a = \vec{r}_a + \delta t \vec{v}_a$
-

E.2 Algorithm ISPH semi-implicite

Algorithm 2 ISPH Algorithm

- 1: **for each** particle $a \in \mathcal{F}$ **do**
 - 2: Compute domains \mathcal{F}_a and \mathcal{S}_a
 - 3: Compute \mathcal{S}^* , $\forall a \in \mathcal{F}$ ($\mathcal{S}_a \subset \mathcal{S}^*$)
 - 4: Compute δt with CFL conditions
 - 5: **for each** particle $a \in \mathcal{F}$ **do**
 - 6: Compute $\forall b \in \mathcal{F}_a \cup \mathcal{S}_a$, $\vec{\nabla} W_{ab} = W'_{ab} \vec{e}_{ab}$
 - 7: Compute ρ_a (Équation 3.29)
 - 8: **if** using Shepard extrapolation **then**
 - 9: **for each** particle $s \in \mathcal{S}^*$ **do**
 - 10: Extrapolate ρ_s and \vec{v}_s (Équation 3.68 & Équation 3.76)
 - 11: **for each** particle $a \in \mathcal{F}$ **do**
 - 12: Compute non-pressure acceleration $\dot{\vec{v}}_a^P$ (Équation 3.37)
 - 13: **for each** particle $a \in \mathcal{F}$ **do**
 - 14: $\vec{v}_a^P = \vec{v}_a + \Delta t \dot{\vec{v}}_a^P$
 - 15: **if** using Shepard extrapolation **then**
 - 16: **for each** particle $s \in \mathcal{S}^*$ **do**
 - 17: Extrapolate pressure p_s (Équation 3.69)
 - 18: Compute $(p_a, \Theta_a, \vec{\Sigma}_a)$, $\forall a \in \mathcal{F}$ and (p_s, Θ_s) , $\forall s \in \mathcal{S}^*$ with alg. 3
 - 19: **for each** solid S **do**
 - 20: Compute \vec{F}_s^P and \vec{F}_s^V depending on Shepard extrapolation or Mirroring
 - 21: **for each** particle $a \in \mathcal{F}$ **do**
 - 22: Compute $\vec{\Gamma}_a$ or $\vec{\Gamma}_a^{Mir}$
 - 23: Compute $\dot{\vec{v}}_a$ depending on Shepard extrapolation (Équation 3.42) or Mirroring (Équation 3.73)
 - 24: Compute $\vec{v}_a = \vec{v}_a + \delta t \dot{\vec{v}}_a$
 - 25: Compute $\vec{r}_a = \vec{r}_a + \delta t \dot{\vec{v}}_a$
-

Algorithm 3 ISPH Pressure Solver Algorithm

- 1: **for each** particle $a \in \mathcal{F}$ **do**
- 2: Compute $\rho_a^* = \rho_a^P$ (Équation 3.45)
- 3: $b_a = \frac{\rho_0 - \rho_a^P}{\delta t^2}$
- 4: Compute $\vec{\Sigma}_a^F \Sigma_a^{F,2}$, $\vec{\Sigma}_a^S$, $\vec{\Sigma}_a$
- 5: Compute D_a depending on Shepard extrapolation (Équation 3.55) or Mirroring (Équation 3.74)
- 6: $p_a = \Omega p_a$, $\Omega < 1$ (Under relaxed first guess)
- 7: Compute Θ_a
- 8: $it = 0$
- 9: **while** $\frac{\text{avg}_{a \in \mathcal{F}}(\max(\rho_a^*, \rho_0)) - \rho_0}{\rho_0} > \epsilon$ **and** $it_{min} < it$ **do**
- 10: **for each** particle $a \in \mathcal{F}$ **do**
- 11: Compute $\vec{\Gamma}_a$ or $\vec{\Gamma}_a^{Mir}$

```

12:   for each particle  $a \in \mathcal{F}$  do
13:     Compute  $\vec{R}_a \cdot \vec{p}$  depending on Shepard extrapolation (Équation 3.56) or Mirroring
      (Équation 3.75)
14:   for each particle  $a \in \mathcal{F}$  do
15:     Compute  $p_a$  (Équation 3.59)
16:     Compute  $\rho_a^*$  (Équation 3.60)
17:      $b_a = \frac{\rho_0 - \rho_a^*}{\Delta t^2}$ 
18:   if using Shepard extrapolation then
19:     for each particle  $s \in \mathcal{S}^*$  do
20:       Extrapolate pressure  $p_s$  (Équation 3.69)
21:    $it = it + 1$ 

```

Annexe F

Le problème « Tensile Instability » et ISPH

F.1 Le problème d’instabilité d’attraction

Comme lorsqu’un objet se déplace dans l’eau, la pression derrière lui peut devenir légèrement négative (en dessous de la pression atmosphérique), il est important pour nous de permettre l’apparition de pression négative. L’instabilité d’attraction, en anglais « Tensile Instability », est un problème qui se produit lorsque l’on autorise les pressions négatives avec un algorithme SPH. Ce problème a été décrit par Monaghan [Mon00]. Celui-ci propose d’essayer de donner un sens physique à ce problème. En considérant un solide qui serait constitué de particules SPH qui obéirait aux lois physiques de déformations des solide, les forces entre ces particules sont dérivées des équations du mouvement et dépendent de la pression. Lorsque la pression est positives, les particules SPH se repoussent tout comme les atomes. Lorsque le solide est étiré, les particules SPH s’attirent mutuellement. Cependant, contrairement au comportement des atomes dans un solide, l’attraction peut entraîner une instabilité qui se manifeste par des particules formant de petits amas. Ces amas rendent l’ensemble du groupe de particules instable. Pour être plus précis : physiquement, le gradient des pressions correspond en fait à une différence de pression $p_a - p_b$ entre la pression de la particule p_a et celles des particules p_b voisines. Or, l’opérateur SPH du gradient qui garantit la conservation de la quantité de matière est de la forme $p_a + p_b$, voir l’équation 3.30. En décomposant $p_a + p_b = (p_a - p_b) + 2p_a$, il est possible d’interpréter l’opérateur SPH du gradient des pressions comme étant le gradient des pressions physique additionné d’un terme « artificiel ». Considérons l’équation 3.30 indépendamment du type de particules, qu’elles soient solides ou fluides et appliquées à la pression, cette équation peut-être réécrite ainsi :

$$m \sum_{b \in F_a} \left(\frac{p_b}{\rho_b^2} + \frac{p_a}{\rho_a^2} \right) \vec{\nabla} W_{ab} = m \sum_{b \in F_a} \left(\frac{p_b}{\rho_b^2} - \frac{p_a}{\rho_a^2} \right) \vec{\nabla} W_{ab} + 2m \frac{p_a}{\rho_a^2} \sum_{b \in F_a} \vec{\nabla} W_{ab} \quad (\text{F.1})$$

Sun [SCM⁺18] a noté que le dernier des termes du côté droit se comporte comme une force de régularisation, qui repousse les particules et qui conduit à la réorganisation des particules vers des configurations spatiales uniformes dans le cas où p_a est positif. Au contraire, dans le cas $p_a < 0$, ce terme a un effet d’attraction. Ainsi, il se comporte de manière inverse, génère éventuellement des amas et augmente le désordre des particules. Ce terme pousse les particules à se déplacer de la région clairsemée vers la région dense, voir Sun [SMZW18]. Le choix de la

forme de l'équation [F.1](#), permet aussi d'expliquer qu'une approximation est faite dans le cas d'une fonction continue. Puisqu'ici le gradient du fonction continue $p_a = p_b$, vaut en fait $2p_a$. Ce problème est étudié par Price [Pri12], section 4.

F.2 Tensile Instability Control et ISPH

Sun [SCM⁺18] a proposé une méthode pour contrôler cette instabilité d'attraction en introduisant un changement dans le gradient de pression. Une façon robuste d'éviter cette instabilité d'attraction, ce qu'on appelle le contrôle de l'instabilité d'attraction, en anglais Tensile Instability Control (TIC), est d'utiliser la valeur absolue de p_a :

$$\vec{\nabla} p_{a \text{ TIC}} = m \sum_{b \in F_a} \left(\frac{p_b}{\rho_b^2} + \frac{|p_a|}{\rho_a^2} \right) \vec{\nabla} W_{ab} \quad (\text{F.2})$$

Ce changement est très simple à mettre en place lors de l'utilisation de WCSPH, mais ce n'est pas si simple pour les solveurs incompressibles, car cela change le système linéaire à résoudre (Voir section 3.3.3 Équation 3.58). Réécrivons l'équation 3.50 selon le principe TIC :

$$\begin{aligned} & -m \sum_{b \in F_a} \left(m \sum_{b \in F_a} \left(\frac{|p_a|}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) \vec{\nabla} W_{ab} + m \sum_{s \in S_a} \left(\frac{|p_a|}{\rho_a^2} + \frac{p_s}{\rho_s^2} \right) \vec{\nabla} W_{as} \right. \\ & \left. -m \sum_{c \in F_b} \left(\frac{|p_b|}{\rho_b^2} + \frac{p_c}{\rho_c^2} \right) \vec{\nabla} W_{bc} - m \sum_{s' \in S_b} \left(\frac{|p_b|}{\rho_b^2} + \frac{p_{s'}}{\rho_{s'}^2} \right) \vec{\nabla} W_{bs'} \right) \vec{\nabla} W_{ab} \\ & -m \sum_{s \in S_a} \left(m \sum_{b \in F_a} \left(\frac{|p_a|}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) \vec{\nabla} W_{ab} + m \sum_{s \in S_a} \left(\frac{|p_a|}{\rho_a^2} + \frac{p_s}{\rho_s^2} \right) \vec{\nabla} W_{as} \right) \vec{\nabla} W_{as} = \frac{\rho_0 - \rho_a^*}{\Delta t^2} \quad (\text{F.3}) \end{aligned}$$

Après développement l'équation F.3 devient :

$$\begin{aligned} & - \left(\frac{1}{\rho_a^2} (\Sigma_a^F)^2 + \frac{2}{\rho_a^2} \Sigma_a^F \Sigma_a^S + \frac{1}{\rho_a^2} (\Sigma_a^S)^2 \right) |p_a| - m \sum_{b \in F_a} \frac{1}{\rho_b^2} (\Sigma_a^F + \Sigma_a^S) \vec{\nabla} W_{ab} p_b \\ & + m \sum_{b \in F_a} \frac{1}{\rho_b^2} (\Sigma_b^S + \Sigma_b^{S'}) \vec{\nabla} W_{ab} |p_b| - (\Sigma_a^F + \Sigma_a^S) m \sum_{s \in S_a} \frac{1}{\rho_s^2} \vec{\nabla} W_{as} p_s \\ & + m^2 \sum_{b \in F_a} \vec{\nabla} W_{ab} \sum_{c \in F_b} \frac{1}{\rho_c^2} \vec{\nabla} W_{bc} p_c + m^2 \sum_{b \in F_a} \vec{\nabla} W_{ab} \sum_{s' \in S_b} \frac{1}{\rho_{s'}^2} \vec{\nabla} W_{bs'} p_{s'} = \frac{\rho_0 - \rho_a^*}{\Delta t^2} \quad (\text{F.4}) \end{aligned}$$

Ce qui peut être simplifié grâce au notation (Équation 3.39, Équation 3.40, Équation 3.41) en :

$$\begin{aligned} & - \left(\text{sign}(p_a) (\Sigma_a)^2 + \Sigma_a^{F;2} \right) \theta_a - m \sum_{b \in F_a} (\Sigma_a - \text{sign}(p_b) \Sigma_b) \vec{\nabla} W_{ab} \theta_b \\ & - m \Sigma_a \sum_{s \in S_a} \vec{\nabla} W_{as} \theta_s + m^2 \sum_{b \in F_a} \vec{\nabla} W_{ab} \sum_{\substack{c \in F_b \\ c \neq a}} \vec{\nabla} W_{bc} \theta_c \\ & + m^2 \sum_{b \in F_a} \vec{\nabla} W_{ab} \sum_{s' \in S_b} \vec{\nabla} W_{bs'} \theta_{s'} = \frac{\rho_0 - \rho_a^*}{\Delta t^2} \quad (\text{F.5}) \end{aligned}$$

On note que les valeurs absolues de p_a ont été transformées en un facteur de signe ($\text{sign}(p_a)$) pour montrer ce qui se passe dans les termes diagonaux et non diagonaux de la matrice qui

doit être inversée en utilisant Jacobi relaxée. Les nouveaux termes $D_{a\ TIC}$ et $R_{a\ TIC}$ sont :

$$D_{a\ TIC} = - \frac{\text{sign}(p_a) (\Sigma_a)^2 + \Sigma_a^{F,2}}{\rho_a^2} \quad (\text{F.6})$$

$$R_{a\ TIC} = - m \sum_{b \in F_a} \frac{\Sigma_a - \text{sign}(p_b) \Sigma_b}{\rho_b^2} \vec{\nabla} W_{ab} + m^2 \sum_{b \in F_a} \vec{\nabla} W_{ab} \sum_{\substack{c \in F_b \\ c \neq a}} \frac{1}{\rho_c^2} \vec{\nabla} W_{bc} + \\ m \Sigma_a \sum_{s \in S_a} \frac{1}{\rho_s^2} \vec{\nabla} W_{as} + m^2 \sum_{b \in F_a} \vec{\nabla} W_{ab} \sum_{s' \in S_b} \frac{1}{\rho_{s'}^2} \vec{\nabla} W_{bs'} \quad (\text{F.7})$$

Ces changements rendent le problème non linéaire. Il est important d'analyser la valeur de $(\Sigma_a)^2$ et $\Sigma_a^{F,2}$ pour vérifier si la diagonale ne change pas de signe pendant la résolution du PPE, ce qui entraîne une divergence du solveur. Malheureusement, en pratique ce cas survient et il faudrait trouver une stratégie pour y remédier.

Annexe G

Équation du mouvement d'un véhicule marin à un point d'origine arbitraire

Pour un véhicule sous-marin, il est parfois intéressant d'avoir un modèle cinétique du véhicule à un point d'origine arbitraire (CO) pour utiliser certaines propriétés géométriques du véhicule. Vu que les forces et moments hydrodynamiques vont être exprimés à ce point arbitraire, les lois de Newton vont être formulées pour ce point également.

Pour cela, nous allons déplacer le modèle cinétique du point CG au point CO en utilisant l'équation 2.8 :

$$\begin{aligned}\vec{v}_{g/i} &= \vec{v}_{S/i} + \vec{\omega}_{S/i} \wedge \vec{r}_g \\ &= \vec{v}_{S/i} - \vec{r}_g \wedge \vec{\omega}_{S/i} \\ &= \vec{v}_{S/i} + S^T(\vec{r}_g) \vec{\omega}_{S/i}\end{aligned}\quad (\text{G.1})$$

Ce qui nous permet d'introduire la matrice de transformation $H(\vec{r}_g)$ entre les deux repères S et g :

$$\begin{pmatrix} \vec{v}_{g/i} \\ \vec{\omega}_{S/i} \end{pmatrix} = H(\vec{r}_g) \begin{pmatrix} \vec{v}_{S/i} \\ \vec{\omega}_{S/i} \end{pmatrix}\quad (\text{G.2})$$

Avec \vec{r}_g le vecteur allant de CO à CG, représenté sur la figure 2.2 et $H(\vec{r}_g)$ la matrice de transformation 3×3 :

$$H(\vec{r}_g) = \begin{pmatrix} I_{3 \times 3} & S^T(\vec{r}_g) \\ 0_{3 \times 3} & I_{3 \times 3} \end{pmatrix}, \quad H^T(\vec{r}_g) = \begin{pmatrix} I_{3 \times 3} & 0_{3 \times 3} \\ S(\vec{r}_g) & I_{3 \times 3} \end{pmatrix}\quad (\text{G.3})$$

La vitesse de rotation angulaire n'est pas modifiée au cours de cette transformation. Ainsi, à l'aide de l'équation G.2 on peut transformer l'équation du mouvement au centre de gravité (Équation 2.13) :

$$H^T(\vec{r}_g) M_{RB}^{CG} H(\vec{r}_g) \begin{pmatrix} \dot{\vec{v}}_{S/i} \\ \dot{\vec{\omega}}_{S/i} \end{pmatrix} + H^T(\vec{r}_g) C_{RB}^{CG} H(\vec{r}_g) \begin{pmatrix} \vec{v}_{S/i} \\ \vec{\omega}_{S/i} \end{pmatrix} = H^T(\vec{r}_g) \begin{pmatrix} \vec{F}_g \\ \vec{M}_g \end{pmatrix}\quad (\text{G.4})$$

Ainsi on peut définir les matrices de masse et matrices des effets centripètes et de Coriolis au point CO :

$$M_{RB}^{CO} = H^T(\vec{r}_g) M_{RB}^{CG} H(\vec{r}_g)\quad (\text{G.5})$$

$$C_{RB}^{CO} = H^T(\vec{r}_g) C_{RB}^{CG} H(\vec{r}_g)\quad (\text{G.6})$$

Ce qui en développant nous donne :

$$M_{RB}^{CO} = \begin{pmatrix} mI_{3 \times 3} & -mS(\vec{r}_g) \\ mS(\vec{r}_g) & I_G - mS^2(\vec{r}_g) \end{pmatrix} \quad (\text{G.7})$$

$$C_{RB}^{CO} = \begin{pmatrix} mS(\vec{\omega}_{S/i}) & -mS(\vec{\omega}_{S/i})S(\vec{r}_g) \\ mS(\vec{r}_g)S(\vec{\omega}_{S/i}) & -S(I_G - mS^2(\vec{r}_g))\vec{\omega}_{S/i} \end{pmatrix} \quad (\text{G.8})$$

On remarque que l'on retrouve le théorème de Huygens pour le déplacement de l'inertie autour d'un axe parallèle : $I_G - mS^2(\vec{r}_g)$.

Annexe H

Calcul de masse ajoutée pour une sphère

Prenons le cas d'une sphère de rayon R immergée dans un volume infini de fluide incompressible, qui a une accélération constante \dot{u} dans la direction x . Pour calculer la masse ajoutée de l'eau, nous cherchons la pression s'appliquant sur la surface de la sphère. En considérant que l'écoulement autour de la sphère est irrotationnel ($\overrightarrow{rot} \vec{u} = \vec{0}$), la vitesse du fluide \vec{u} découle du gradient du potentiel d'écoulement ϕ :

$$\vec{u} = \overrightarrow{\text{grad}}\phi = \vec{\nabla}\phi \quad (\text{H.1})$$

Ce faisant, il faut souligner que les effets des forces visqueuses dans le fluide sont négligés.

On applique la formule de Bernoulli en transitoire [Bat10] :

$$\frac{\partial\phi}{\partial t} + \frac{1}{2}u^2 + \frac{p}{\rho} + gz = 0 \quad (\text{H.2})$$

Avec p le champ de pression, z l'altitude des points considérés.

Enfin, nous négligeons les forces de pesanteur, vu que le mouvement de la sphère se fera dans un plan horizontal et que la taille de notre sphère n'est pas significative. La force hydrostatique pourra être prise en compte séparément pour le calcul de la dynamique du solide¹. En tenant compte de l'équation H.1, l'équation H.2 devient :

$$p = \rho \left(\frac{\partial\phi}{\partial t} + \frac{1}{2} |\vec{\nabla}\phi|^2 \right) \quad (\text{H.3})$$

On cherche maintenant à déterminer le potentiel d'écoulement circulant autour de cette sphère. Dans un fluide incompressible, la divergence de la vitesse est nulle :

$$\overrightarrow{\text{div}} \cdot \vec{u} = \vec{\nabla} \cdot \vec{u} = 0 \quad (\text{H.4})$$

Ceci implique que le potentiel d'écoulement obéit à l'équation de Laplace :

$$\text{Par définition : } \vec{u} = \vec{\nabla}\phi \implies (\vec{\nabla} \cdot \vec{\nabla})\phi = \Delta\phi = 0 \quad (\text{H.5})$$

La figure H.1 montre la sphère ainsi que différents petits éléments de surface et différentes longueurs qui serviront au calcul qui suit. La géométrie de la sphère fait qu'il est plus

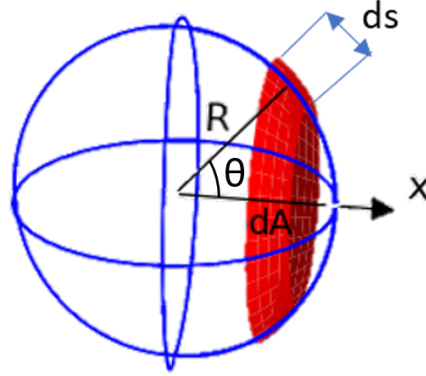


Figure H.1 – Calcul de la masse ajoutée d’une sphère : intégration de la pression s’appliquant à sa surface

intéressant d’utiliser les coordonnées polaires pour les calculs. Ces éléments de volumes et longueurs sont :

$$d\vec{A}_x = -\cos\theta dA \vec{e}_x \quad (\text{H.6})$$

$$dA = 2\pi R ds \quad (\text{H.7})$$

$$ds = R d\theta \quad (\text{H.8})$$

Avec \vec{e}_x le vecteur unitaire dans la direction de l’axe x . En considérant que le potentiel d’écoulement est axisymétrique $\phi \equiv \phi(r, \theta)$ autour de l’axe x , l’équation de Laplace peut s’écrire (Équation H.5) en coordonnées polaires (Josza [Joz]) :

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial u}{\partial r} \right) + \frac{1}{r^2 \sin\theta} \frac{\partial}{\partial \theta} \left(\sin\theta \frac{\partial u}{\partial \theta} \right) = 0 \quad (\text{H.9})$$

Avec r le rayon sphérique : partant du centre géométrique de la sphère traversant l’écoulement et allant jusqu’au point considéré.

Une solution générale de l’équation de Laplace peut être obtenue grâce à une méthode de séparation de variable d’après le cours de Josza [Joz]. Cette solution générale de l’équation de Laplace (Équation H.9) est donnée par :

$$\phi(r, \theta) = \sum_{n=0}^{\infty} (A_n r^n + B_n r^{-(n+1)}) P_n(\cos\theta) \quad (\text{H.10})$$

Avec A_n et B_n des constantes arbitraires et P_n le polynôme de Legendre de degrés n . Ce polynôme peut s’écrire sous forme de somme de la manière suivante :

$$P_n(x) = \frac{1}{2^n} \sum_{k=0}^n \binom{n}{k}^2 (x-1)^{n-k} (x+1)^k \quad (\text{H.11})$$

Avec $\binom{n}{k}$ le coefficient binomial. Ainsi, $P_0(x) = 1$, $P_1(x) = x$, $P_2(x) = \frac{3}{2}x^2 - \frac{1}{2}$, etc.. D’après le livre de Guyon, Hulin et Petit [GHP01], il existe un unique champ de vitesse, potentiel et

1. Dans la force de poussée d’Archimède

incompressible, qui correspond à des composantes de vitesse normale données sur les surfaces solides et à une vitesse donnée à l'infini. La démarche adoptée dans la suite de ce calcul est de trouver une solution au problème, puisque celle-ci est unique. Puisque le domaine de résolution du problème est infini et que A_0 est une constante à appliquer en tout point du domaine ($r \rightarrow \infty$), et qu'à la surface de la sphère le flux est nul, A_0 est nécessairement nul. Regardons maintenant les cas particuliers avec chacun des trois premiers termes B_0 , A_1 et B_1 .

1. Tous les A et B sont nuls sauf B_0 . Dans ce cas,

$$\phi(r, \theta) = \frac{B_0}{r} \implies \vec{u} = \vec{\nabla} \phi(r, \theta) = -\frac{B_0 \vec{e}_r}{r^2}$$

Avec \vec{e}_r le vecteur unitaire radial. Ceci représente une source ou un puits de flux en fonction du signe de B_0 . Un flux radial sortant d'un point source dans le cas où B_0 est négatif et un flux entrant dans un point puits dans le cas où B_0 est positif. Ceci est représenté sur la figure H.2



Figure H.2 – Puits et source de potentiel

2. Tous les A et B sont nuls sauf A_1 . Dans ce cas, en notant que $x = r \cos \theta$

$$\phi(r, \theta) = A_1 r \cos \theta = A_1 x \implies \vec{u} = \vec{\nabla} \phi(r, \theta) = A_1 \vec{e}_x \quad (\text{H.12})$$

Avec \vec{e}_x le vecteur unitaire selon l'axe x . Ceci représente simplement un flux uniforme selon l'axe x , comme représenté sur la figure H.3.

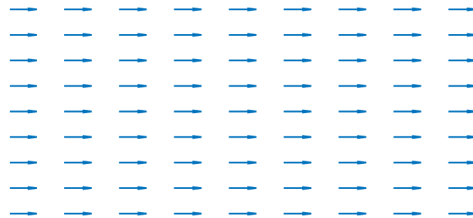


Figure H.3 – Circulation rectiligne uniforme

3. Tous les A et B sont nuls sauf B_1 . Dans ce cas,

$$\phi = \frac{B_1 \cos \theta}{r^2} = \frac{B_1 x}{r^3} \implies \vec{u} = \vec{\nabla} \phi = B_1 \left(\frac{\vec{e}_x}{r^3} - \frac{3x \vec{e}_r}{r^4} \right) \quad (\text{H.13})$$

On peut noter que cette équation est l'opposée de la dérivée selon x ($-\partial/\partial x$) de l'équation H.12. Cette solution (Équation H.13) peut être vue comme un écoulement symétrique autour de l'axe x avec une recirculation qui part du centre de la sphère.

Considérons maintenant le cas d'un écoulement stationnaire autour de la sphère. A l'infini, lorsque $r \rightarrow \infty$, l'écoulement doit être colinéaire au déplacement de la sphère², donc :

$$\begin{aligned} u_{r \rightarrow \infty} &= U \\ \phi &\rightarrow Ux \rightarrow Ur \cos \theta \end{aligned}$$

Ceci ressemble donc au deuxième cas présenté ci-dessus. Il n'y a pas d'écoulement normal à la sphère, donc la variation du flux par rapport à r à la surface de la surface est nulle :

$$\left(\frac{\partial \phi}{\partial r} \right)_{r=R} = 0 \quad (\text{H.14})$$

Ainsi, l'unique solution du polynôme de Legendre proportionnelle à $\cos \theta$ est $P_1(\cos \theta)$. Puisque l'écoulement y est proportionnel à l'infini nous allons vérifier si la solution de l'équation H.5 est de la forme :

$$\phi = \left(A_1 r + \frac{B_1}{r^2} \right) \cos \theta \quad (\text{H.15})$$

Pour que cela corresponde à la condition à $r \rightarrow \infty$, on choisit $A_1 = U$. Pour que cela corresponde à la seconde condition (Équation H.14), qui veut qu'il n'y ait pas d'écoulement normal on a :

$$\frac{\partial \phi}{\partial r} = \left(A_1 - \frac{2B_1}{r^3} \right) \cos \theta = 0$$

Pour $r = R$ on a :

$$A_1 = \frac{2B_1}{R^3}$$

donc :

$$B_1 = U \frac{R^3}{2}$$

On obtient finalement le potentiel d'écoulement :

$$\phi = U \left(r + \frac{R^3}{2r^2} \right) \cos \theta \quad (\text{H.16})$$

Ce potentiel d'écoulement correspond à une vitesse d'écoulement de :

$$\vec{u} = \vec{\nabla} \phi = \left(\frac{\partial \phi}{\partial r}, \frac{1}{r} \frac{\partial \phi}{\partial \theta}, 0 \right) = \left(U \cos \theta \left(1 - \frac{R^3}{r^3} \right), -U \sin \theta \left(1 + \frac{R^3}{2r^3} \right), 0 \right) \quad (\text{H.17})$$

Cet écoulement autour de la sphère est représenté sur la figure H.4.

Ce potentiel d'écoulement décrit le cas où le fluide est en mouvement par rapport à la sphère. Dans le cas, où la sphère se déplace et le fluide est au repos il faut soustraire le potentiel d'écoulement à l'infini ($\phi = r \cos \theta$) :

$$\phi_A = \phi - r \cos \theta = U \frac{R^3}{2r^2} \cos \theta \quad (\text{H.18})$$

Dans ce cas le champ de vitesse du fluide autour de la sphère devient :

$$\vec{u}_A = \vec{\nabla} \phi_A = \left(\frac{\partial \phi}{\partial r}, \frac{1}{r} \frac{\partial \phi}{\partial \theta}, 0 \right) = \left(-U \cos \theta \frac{R^3}{r^3}, -U \sin \theta \frac{R^3}{2r^3}, 0 \right) \quad (\text{H.19})$$

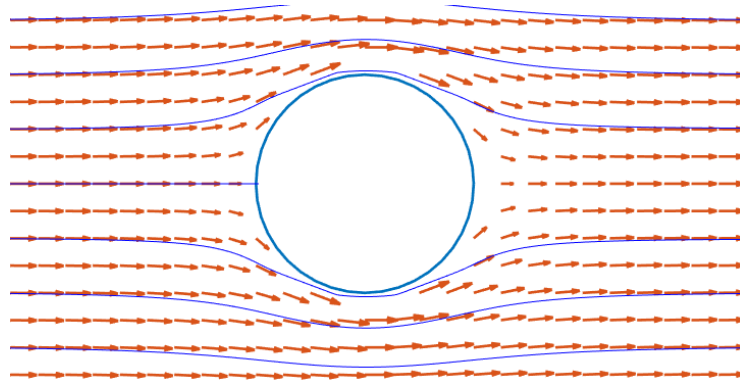


Figure H.4 – Circulation du fluide lors d’un écoulement stationnaire autour d’une sphère. En bleu les lignes de courants et en orange le champ de vitesse.

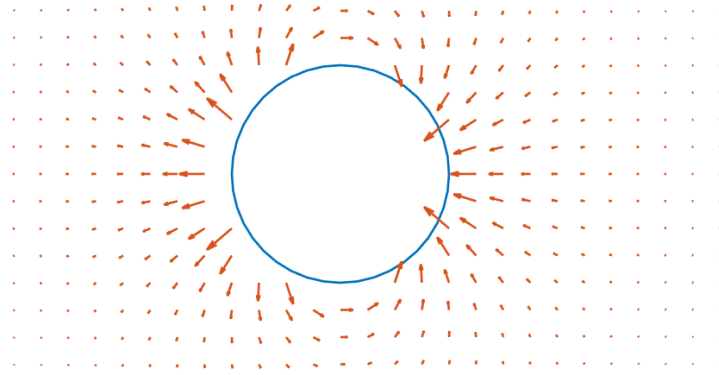


Figure H.5 – Circulation du fluide autour d’une sphère qui accélère dans un fluide statique. En orange le champ de vitesse.

Cette circulation est représentée sur la figure H.5.

À partir de ce potentiel d’écoulement ϕ_A , on calcule maintenant les valeurs qui nous intéressent pour l’équation H.3, c’est à dire, celle de l’écoulement à la surface de la sphère :

$$\left. \frac{\partial \phi_A}{\partial t} \right|_{r=R} = \frac{R}{2} \dot{U} \cos \theta \quad (\text{H.20})$$

$$\begin{aligned} \frac{1}{2} \left| \vec{\nabla} \phi_A \right|_{r=R}^2 &= \frac{1}{2} \left| \left(\frac{\partial \phi_A}{\partial r}, \frac{1}{r} \frac{\partial \phi}{\partial \theta}, 0 \right) \right|_{r=R}^2 = \frac{1}{2} \left| \left(-U \cos \theta \frac{R^3}{r^3}, -U \sin \theta \frac{R^3}{2r^3}, 0 \right) \right|_{r=R}^2 \\ \frac{1}{2} \left| \vec{\nabla} \phi_A \right|_{r=R}^2 &= \frac{U^2}{2} \left(\cos^2 \theta + \frac{\sin^2 \theta}{4} \right) \quad (\text{H.21}) \end{aligned}$$

À partir des équations H.20, H.3 et H.21 on obtient la pression due au déplacement de l’eau s’exerçant sur la surface de la sphère. On peut donc calculer la force de masse ajoutée $\vec{F}_{A,x}$

2. Rappelons que, dans notre cas, c’est la sphère qui se déplace. Il n’y a donc pas d’écoulement à l’infini mais nous utilisons ce biais pour obtenir les constantes du polynôme de Legendre

en intégrant cette pression sur la surface de la sphère :

$$\vec{F}_{A,x} = \int p d\vec{A}_x \quad (\text{H.22})$$

$$\vec{F}_{A,x} = - \int_0^\pi \rho \left(\frac{\partial \phi_A}{\partial t} + \frac{1}{2} |\vec{\nabla} \phi|^2 \right) 2\pi R^2 \cos \theta \sin \theta d\theta \vec{e}_x$$

$$\vec{F}_{A,x} = -2\rho\pi R^2 \int_0^\pi \left(\frac{R}{2} \dot{U} \cos \theta + \frac{U^2}{2} \left(\cos^2 \theta + \frac{\sin^2 \theta}{4} \right) \right) \cos \theta \sin \theta d\theta \vec{e}_x$$

$$\vec{F}_{A,x} = -2\rho\pi R^2 \left(\frac{R}{2} \dot{U} \int_0^\pi \cos^2 \theta \sin \theta d\theta + \frac{U^2}{2} \int_0^\pi \left(\cos^3 \theta \sin \theta + \frac{\cos \theta \sin^3 \theta}{4} \right) d\theta \right) \vec{e}_x$$

$$\vec{F}_{A,x} = -2\rho\pi R^2 \left(\frac{R}{2} \dot{U} \underbrace{\left[-\frac{\cos^3 \theta}{3} \right]_0^\pi}_{=2/3} + \frac{U^2}{2} \underbrace{\left[\frac{\sin^4 \theta}{16} - \frac{\cos^4 \theta}{4} \right]_0^\pi}_{=0} \right) \vec{e}_x$$

$$\vec{F}_{A,x} = -\frac{2}{3} \rho\pi R^3 \dot{U} \vec{e}_x \quad (\text{H.23})$$

Annexe I

Calcul des B-Spline pour le schéma SPH

I.1 B-Spline cubique

Pour obtenir la fonction B-Spline cubique $f_{W,3}$ en partant de l'équation 3.20 on remplace $M = 3$:

$$\begin{aligned} f_3(q) &= \frac{1}{3!} \sum_{k=0}^4 (-1)^k \binom{4}{k} \max(0; q+2-k)^3 \\ &= \frac{1}{6} \left(\max(0; q+2)^3 - 4 \max(0; q+1)^3 + 6 \max(0; q)^3 - 4 \max(0; q-1)^3 + \right. \\ &\quad \left. \max(0; q-2)^3 \right) \end{aligned}$$

En considérant q positif, en séparant les différents cas où la fonction max s'annule et en développant on obtient :

$$\begin{aligned} f_3(q) &= \begin{cases} \frac{1}{6} ((q-2)^3 - 4(q-1)^3) & 0 \leq q \leq 1 \\ \frac{1}{6}(q-2)^3 & 1 \leq q \leq 2 \\ 0 & q \geq 2 \end{cases} \\ &= \begin{cases} q^2 \left(\frac{-q}{2} + 1 \right) - \frac{2}{3} & 0 \leq q \leq 1 \\ \frac{1}{6}(q-2)^3 & 1 \leq q \leq 2 \\ 0 & q \geq 2 \end{cases} \end{aligned}$$

Pour simplifier nous allons multiplier par $^{-3/4}$ ces fonctions :

$$f_3(q) = \begin{cases} \frac{1}{2} - \frac{3q^2}{4} \left(1 - \frac{q}{2} \right) & 0 \leq q \leq 1 \\ \frac{1}{8}(2-q)^3 & 1 \leq q \leq 2 \\ 0 & q \geq 2 \end{cases}$$

Pour obtenir un rayon ι_t limité à 1 on choisit de faire le changement de variable suivant $q = 2q^*$ Ainsi on obtient la fonction finale du kernel cubique $f_{W,3}$:

$$f_{W,3}(q) = \begin{cases} \frac{1}{2} - 3(q^*)^2 (1 - q^*) & 0 \leq q^* \leq \frac{1}{2} \\ (1 - q^*)^3 & \frac{1}{2} \leq q^* \leq 1 \\ 0 & q^* \geq 1 \end{cases}$$

Il reste à calculer le coefficient de normalisation α_3 (Équation 3.19) :

$$\begin{aligned}
\alpha_3 &= \left[4\pi \int_0^1 f_{W,3}(q) q^2 dq \right]^{-1} \\
&= \frac{1}{4\pi} \left[\int_0^{\frac{1}{2}} \left(\frac{1}{2} - 3q^2(1-q) \right) q^2 dq + \int_{\frac{1}{2}}^1 (1-q)^3 q^2 dq \right]^{-1} \\
&= \frac{1}{4\pi} \left[\int_0^{\frac{1}{2}} \frac{q^2}{2} dq - \int_0^{\frac{1}{2}} 3q^4(1-q) dq + \int_{\frac{1}{2}}^1 q^2 - 3q^3 + 3q^4 - q^5 dq \right]^{-1} \\
&= \frac{1}{4\pi} \left[\frac{1}{48} - 3 \left(\frac{1}{160} - \frac{1}{384} \right) + \left(\frac{3}{5} - \frac{3}{4} + \frac{1}{3} - \frac{1}{6} - \frac{3}{160} + \frac{3}{64} - \frac{1}{24} + \frac{1}{384} \right) \right]^{-1} \\
&= \frac{1}{4\pi} \left[\frac{1}{64} \right]^{-1} \\
&= \frac{16}{\pi}
\end{aligned} \tag{I.1}$$

On obtient finalement, en remplaçant la notation q^* par q , et en prenant en compte le coefficient ι^n (Équation 3.17) dans le coefficient $\alpha_{W,3}$:

$$\begin{aligned}
\alpha_{W,3} &= \frac{16}{\pi \iota_t^3} \\
f_{W,3}(q) &= \begin{cases} \frac{1}{2} - 3q^2(1-q) & 0 \leq q \leq \frac{1}{2} \\ (1-q)^3 & \frac{1}{2} \leq q \leq 1 \\ 0 & q \geq 1 \end{cases}
\end{aligned} \tag{I.2}$$

I.2 B-Spline quintique

Pour la fonction B-Spline quintique en partant de l'équation 3.20 on remplace $M = 5$:

$$\begin{aligned} f_5(q) &= \frac{1}{5!} \sum_{k=0}^6 (-1)^k \binom{6}{k} \max(0; q + 3 - k)^5 \\ &= \frac{1}{120} \left(\max(0; q + 3)^5 - 6 \max(0; q + 2)^5 + 15 \max(0; q + 1)^5 - 20 \max(0; q)^5 + \right. \\ &\quad \left. 15 \max(0; q - 1)^5 - 6 \max(0; q - 2)^5 + \max(0; q - 3)^5 \right) \end{aligned}$$

Comme pour le B-Spline cubique, considérons q positif et séparons les différents cas où la fonction max s'annule :

$$\begin{aligned} f_5(q) &= \begin{cases} \frac{1}{120} ((q - 3)^5 - 6(q - 2)^5 + 15(q - 1)^5) & 0 \leq q \leq 1 \\ \frac{1}{120} ((q - 3)^5 - 6(q - 2)^5) & 1 \leq q \leq 2 \\ \frac{1}{120} (q - 3)^5 & 2 \leq q \leq 3 \\ 0 & q \geq 3 \end{cases} \\ &= \begin{cases} \frac{1}{120} (10q^5 - 30q^4 + 60q^2 - 66) & 0 \leq q \leq 1 \\ \frac{1}{120} ((q - 3)^5 - 6(q - 2)^5) & 1 \leq q \leq 2 \\ \frac{1}{120} (q - 3)^5 & 2 \leq q \leq 3 \\ 0 & q \geq 3 \end{cases} \end{aligned}$$

On effectue alors le changement de variable $q = 3q^*$ pour normaliser la fonction $f_5(q)$:

$$f_5(q) = \begin{cases} \frac{243}{120} \left(10q^5 - 10q^4 + \frac{20}{9}q^2 - \frac{22}{81} \right) & 0 \leq q \leq \frac{1}{3} \\ \frac{243}{120} \left((q - 1)^5 - 6 \left(q - \frac{2}{3} \right)^5 \right) & \frac{1}{3} \leq q \leq \frac{2}{3} \\ \frac{243}{120} (q - 1)^5 & \frac{2}{3} \leq q \leq 1 \\ 0 & q \geq 1 \end{cases}$$

Pour simplifier on multiplie par $-\frac{243}{120}$ et on obtient ainsi, en factorisant la première ligne :

$$f_5(q) = \begin{cases} 10 \left(\frac{11}{405} - q^2 \left(\frac{2}{9} - q^2(1 - q) \right) \right) & 0 \leq q \leq \frac{1}{3} \\ (1 - q)^5 - 6 \left(\frac{2}{3} - q \right)^5 & \frac{1}{3} \leq q \leq \frac{2}{3} \\ (1 - q)^5 & \frac{2}{3} \leq q \leq 1 \\ 0 & q \geq 1 \end{cases}$$

Il reste à calculer le coefficient de normalisation α_5 (Équation 3.19) :

$$\begin{aligned} \alpha_5 &= \left[4\pi \int_0^1 f_{W,5}(q) q^2 dq \right]^{-1} \\ &= \frac{1}{4\pi} \left[\int_0^{\frac{1}{3}} 10 \left(\frac{11}{405} - q^2 \left(\frac{2}{9} - q^2(1 - q) \right) \right) q^2 dq + \int_{\frac{1}{3}}^{\frac{2}{3}} \left((1 - q)^5 - 6 \left(\frac{2}{3} - q \right)^5 \right) q^2 dq + \right. \\ &\quad \left. \int_{\frac{2}{3}}^1 (1 - q)^5 q^2 dq \right]^{-1} \\ &= \frac{2187}{40\pi} \end{aligned} \tag{I.3}$$

Enfin on obtient la fonction kernel B-Spline quintique $f_{W,5}$:

$$\alpha_{W,5} = \frac{2187}{40\pi\iota_t^5}$$

$$f_{W,5}(q) = \begin{cases} 10 \left(\frac{11}{405} - q^2 \left(\frac{2}{9} - q^2(1-q) \right) \right) & 0 \leq q \leq \frac{1}{3} \\ (1-q)^5 - 6 \left(\frac{2}{3} - q \right)^5 & \frac{1}{3} \leq q \leq \frac{2}{3} \\ (1-q)^5 & \frac{2}{3} \leq q \leq 1 \\ 0 & q \geq 1 \end{cases} \quad (\text{I.4})$$

Annexe J

Méthodes d'implémentations des algorithmes

J.1 Organisation des classes

Les algorithmes ISPH et WCSPH sont déployés sur un programme C++ qui se nomme Neutrino et qui est présenté dans l'annexe B. Neutrino est un programme qui possède de nombreuses fonctionnalités, notamment pour l'affichage de la scène, le calcul et la sauvegarde de nombreux types de données. Cette section n'explique pas de manière exhaustive le fonctionnement complet de ce programme mais traite uniquement de la partie simulation dynamique.

Pour décrire l'organisation des classes, une méthode standard dérivée du langage de modélisation unifié, en anglais « Unified Modeling Language » (UML), pourrait être utilisée mais le résultat serait très lourd et peu visuel. La figure J.1 représente un diagramme simplifié des interactions entre les classes. Ces classes sont :

- Une classe « Particule », qui est celle qui permet de stocker les informations sur chaque particule de la simulation. Cette classe est décomposée en deux sous-classes, les particules fluides et les particules solides. Le nombre de variables de ces classes de particules à une importance particulière, car selon leur nombre et leur précision (int, float, double, etc.) la taille mémoire d'une particule est modifiée. Notre simulation étant vouée à en contenir un très grand nombre, une attention particulière doit lui être portée.
- Une classe « Solveur fluide global » qui permet de donner un cadre aux classes de chaque solveur de fluide et qui contient les fonctions qui sont similaires pour tous les solveurs comme par exemple celles qui décrivent l'interaction fluide-solide.
- Une classe par solveur de fluide (ISPH et WCSPH) qui contient les méthodes de calcul à appliquer aux particules de fluide.
- Une classe « Moteur physique » ou plus rigoureusement « Solveur de corps rigides », qui sert à calculer la dynamique des objets solides et prendre en compte les interactions avec le fluide.
- Une classe par type d'objet solide, qui permet de décrire et calculer les propriétés mécaniques de chaque objet, notamment leur inertie.
- Une classe « Collider », qui permet de déterminer, par exemple, les particules qui interagissent. Cette classe n'est pas représentée sur le diagramme.

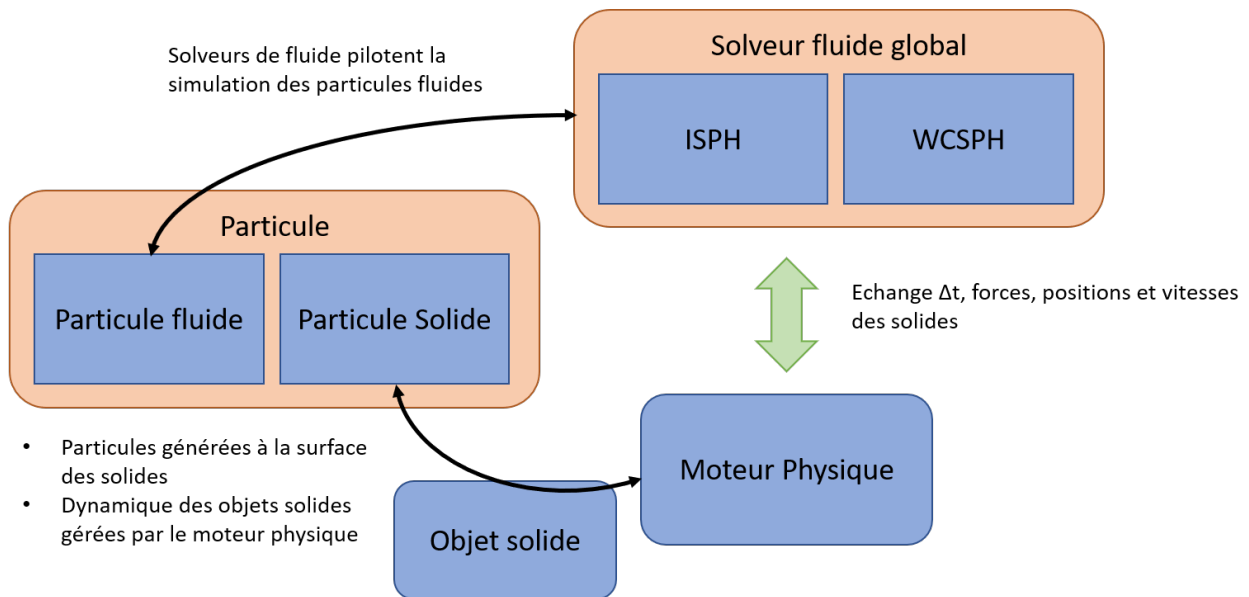


Figure J.1 – Schéma des interactions entre les classes dans Neutrino

Les particules qui existent en tant que classe sont stockées en mémoire sous forme de tableau de structure ("array of structures"), par opposition à une structure de tableaux ("structure of arrays"). Dans notre cas, il y a un tableau contenant l'ensemble des particules du fluide, et chacune de ces particules possèdent des informations sur ces propriétés (pression, vitesse, etc.). Si l'on souhaitait stocker nos informations sous la forme de structure de tableau, il faudrait que chaque propriété soit un tableau qui contient les données de toutes les particules et que toutes ces données soient dans une unique structure. Néanmoins, lors de la résolution du calcul des pressions dans le solveur ISPH, les données de la matrice à inverser sont stockées sous cette forme, ce qui contribue à la diminution du temps de calcul avec ce solveur. Ce concept est particulièrement important pour l'organisation des données dans la mémoire et le taux d'utilisation de la mémoire cache du processeur. Les processeurs modernes (x86) sont tous équipés de « Single Instruction on Multiple Data » (SIMD), en français, « instruction unique, données multiples ». Ce mode de fonctionnement permet d'exécuter la même instruction simultanément sur plusieurs données pour produire plusieurs résultats. En utilisant un tableau de structure, les données ne sont pas organisées de manière suffisante dans la mémoire et l'accès mémoire est limitant pour la vitesse de calcul. La génération d'un code en structure de tableaux, en revanche, est bien plus complexe à mettre en place et surtout à modifier.

Les particules sont enregistrées dans une structure mémoire ayant la forme d'une grille compacte et une table de hachage est utilisée pour accélérer la recherche de voisins tout en limitant l'usage mémoire. L'information des voisins est stockée dans chaque instance de la classe Particule qui permet d'éviter de recalculer avec le Collider les voisins plusieurs fois par pas de temps. Une fois par pas de temps, avant toutes les autres instructions, le collider utilise une courbe de remplissage dite « en Z » de la simulation pour réordonner les particules et réorganiser la grille.

J.2 Méthode de stockage de données et de compilation

Le développement de Neutrino se fait avec le logiciel Qt. Le programme est en C++. Le compilateur utilisé pour ce programme est Microsoft Visual C++ 2017. Les programmes internes de Qt pour les étapes de préparation de compilation (`qmake.exe`) et d'exécution de compilation (`jom.exe`) sont utilisés avec les options `qtquickcompiler`, qui permet d'accélérer la génération de fichier et l'option de `jom` permettant d'utiliser plusieurs thread¹ pour la compilation.

Pour permettre une résolution parallèle du code, OpenMP [DM98] est utilisé pour paralléliser les boucles des calculs SPH et la recherche de voisins. OpenMP est une interface de programmation pour le calcul parallèle sur architecture à mémoire partagée. Cette API est prise en charge pour les langages de programmation C, C++ et Fortran². OpenMP permet de régler très simplement le nombre de thread à utiliser. Le principe de cette API est d'indiquer dans le code source des blocs de fonctions qui doivent être exécutés en parallèle. La manière dont les blocs sont exécutés en parallèle est gérée par l'API et le plus important pour le programmeur est de vérifier que les blocs de commandes exécutés en parallèle, peuvent l'être et ne génèrent pas de conflits d'accès concurrent aux données. Ce fonctionnement est décrit sur la figure J.2 qui est inspiré du cours de Mattson [Mat]. En anglais, ce fonctionnement s'appelle « fork-join model ». Sur cette figure, on montre que pour chaque bloc d'exécution,

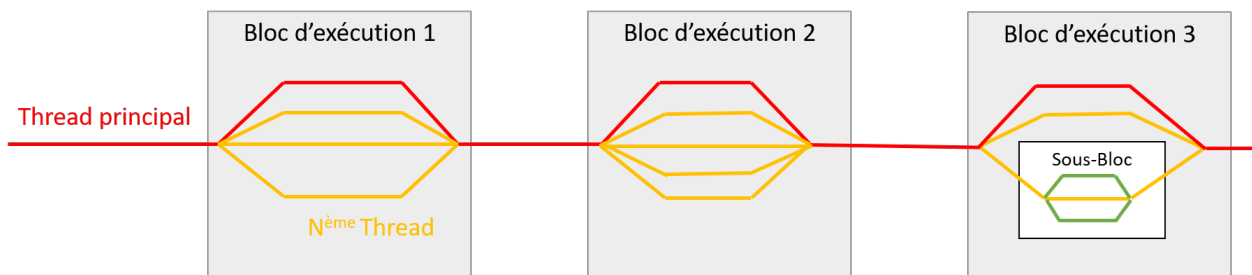


Figure J.2 – Schéma du fonctionnement d'OpenMP. En rouge le fil d'exécution (ou thread) principal, en jaune et vert les fils d'exécution secondaires créés dans les blocs d'exécution.

OpenMP crée différents threads qui vont exécuter les tâches en parallèle. Parfois il est même possible d'imbriquer les blocs d'exécution. OpenMP, s'exécute au moment de la compilation, comme une addition au compilateur traditionnel et va permettre lors de celle-ci de réarranger l'ordre des instructions pour faire un meilleur usage des registres mémoires. De plus, un mécanisme pour gérer l'accès concurrent à une même adresse mémoire y est inclus. OpenMP est utilisé sur toutes les boucles « for each » des algorithmes décrits dans l'annexe E.

La manière dont les algorithmes WCSPH et ISPH (décrits dans l'annexe B) sont codés peut avoir un impact significatif sur le temps de calcul lors de l'exécution de ceux-ci. Au vu de la description faite, il y a un fort potentiel d'amélioration. Notamment il serait intéressant de figer une partie du code source et de le transformer pour qu'il corresponde à un code où

1. Un thread ou fil d'exécution est similaire à un processus. Tous deux représentent l'exécution d'un ensemble d'instructions du langage machine d'un processeur. Du point de vue de l'utilisateur, ces exécutions semblent se dérouler en parallèle. Toutefois, là où chaque processus possède sa propre mémoire virtuelle, les threads d'un même processus se partagent sa mémoire virtuelle. Cependant, tous les threads possèdent leur propre pile d'exécution. Ce terme et sa définition sont normalisés par la norme ISO/IEC 2382 :2015.

2. Initialement cette API avait été développée pour Fortran

les informations sont stockées en structure de tableau. Ceci s'avèrerait notamment nécessaire si l'on souhaite un jour profiter pleinement de la puissance de calcul des cartes GPU.

Glossaire

Dans l'ordre alphabétique.

AUV	De l'anglais : Autonomous Underwater Vehicle. Désigne un véhicule sous-marin autonome.
Cavalement	Le cavalement désigne le déplacement horizontal d'un véhicule marin selon l'axe longitudinal. En anglais : « surge » (voir section 2.1.1)
Embardée	Une embardée désigne le déplacement latéral d'un véhicule marin. L'embardee peut être dans la direction de bâbord ou de tribord. En anglais : « sway » (voir section 2.1.1)
ISPH	Incompressible Smoothed Particles Hydrodynamics
Observateur d'état	En automatique et en théorie de l'information, un observateur d'état est une extension d'un modèle, qui permet de reconstruire l'état à partir d'un modèle du système dynamique et des mesures d'autres grandeurs. C'est à dire d'obtenir une estimation de l'évolution temporelle d'une grandeur physique non mesurable
Lacet	Le lacet est le mouvement de rotation horizontal d'un véhicule marin autour de son axe vertical. En anglais : « yaw » (voir section 2.1.1)
Pilonnement	Le pilonnement désigne le déplacement vertical d'un véhicule marin. En anglais : « heave » (voir section 2.1.1)
Roulis	Le roulis est un mouvement de rotation d'un véhicule marin autour de son axe longitudinal. En anglais : « roll » (voir section 2.1.1)
ROV	De l'anglais : Remotely Operated Vehicle. Désigne un véhicule sous-marin téléguédé.
SPH	Smoothed Particles Hydrodynamics
Surface libre	Surface d'un fluide soumis à une contrainte de cisaillement parallèle nulle, comme par exemple l'interface entre deux fluides homogènes et le plus généralement l'eau et l'air
Tangage	Le tangage est un mouvement de rotation autour de l'axe transversal d'un objet en mouvement. En anglais : « pitch » (voir section 2.1.1)
WCSPH	Weakly Compressible Smoothed Particles Hydrodynamics

Bibliographie

- [ACC⁺15] B. Allotta, A. Caiti, L. Chisci, R. Costanzi, F. Di Corato, C. Fantacci, D. Fenucci, E. Meli, and A. Ridolfi. Development of a navigation algorithm for autonomous underwater vehicles. *IFAC-PapersOnLine*, 48(2) :64–69, 2015.
- [ACG⁺16] Víctor Hugo Andaluz, Fernando A. Chicaiza, Cristian Gallardo, Washington X. Quevedo, José Varela, Jorge S. Sánchez, and Oscar Arteaga. Unity3d-MatLab simulator in real time for robotics applications. In *Lecture Notes in Computer Science*, pages 246–263. Springer International Publishing, 2016.
- [Ach74] Elmar Achenbach. The effects of surface roughness and tunnel blockage on the flow past spheres. *Journal of Fluid Mechanics*, 65(1) :113–125, 1974.
- [ACR⁺14] Corrado Altomare, Alejandro JC Crespo, BD Rogers, JM Dominguez, X Gironella, and M Gómez-Gesteira. Numerical modelling of armour block sea breakwater with smoothed particle hydrodynamics. *Computers & Structures*, 130 :34–45, 2014.
- [AG03] Novodex AG, December 2003.
- [AGHQ12] R.P.G. Andrade, F. Grassi, Y. Hama, and W.-L. Qian. Temporal evolution of tubular initial conditions and their influence on two-particle correlations in relativistic nuclear collisions. *Physics Letters B*, 712(3) :226–230, jun 2012.
- [AHA12] S. Adami, X.Y. Hu, and N.A. Adams. A generalized wall boundary condition for smoothed particle hydrodynamics. *Journal of Computational Physics*, 231(21) :7057–7075, aug 2012.
- [AIA⁺12] Nadir Akinci, Markus Ihmsen, Gizem Akinci, Barbara Solenthaler, and Matthias Teschner. Versatile rigid-fluid coupling for incompressible SPH. *ACM Transactions on Graphics*, 31(4) :1–8, jul 2012.
- [Aki14] Nadir Akinci. *Interface Handling in Smoothed Particle Hydrodynamics*. Phd thesis, Technischen Fakultät, Albert-Ludwigs-Universität Freiburg im Breisgau, May 2014.
- [AKOH00] C E Aguiar, T Kodama, T Osada, and Y Hama. Smoothed particle hydrodynamics for relativistic heavy-ion collisions. *Journal of Physics G : Nuclear and Particle Physics*, 27(1) :75–94, dec 2000.
- [ALR⁺14] Arash Ajoudani, Jinoh Lee, Alessio Rocchi, Mirko Ferrati, Enrico Mingo Hoffman, Alessandro Settini, Darwin. G. Caldwell, Antonio Bicchi, and Nikos G. Tsagarakis. A manipulation framework for compliant humanoid COMAN : Application to a valve turning task. In *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, nov 2014.

- [ASAA08] S. S. Akarca, X. Song, W. J. Altenhof, and A. T. Alpas. Deformation behaviour of aluminium during machining : Modelling by eulerian and smoothed-particle hydrodynamics methods. *Proceedings of the Institution of Mechanical Engineers, Part L : Journal of Materials : Design and Applications*, 222(3) :209–221, jun 2008.
- [Bat10] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, 2010.
- [BB07] Adrian Boeing and Thomas Bräunl. Evaluation of real-time physics simulation systems. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, pages 281–288. ACM, 2007.
- [BCMA13] B. Bouscasse, A. Colagrossi, S. Marrone, and M. Antuono. Nonlinear water wave interaction with floating bodies in SPH. *Journal of Fluids and Structures*, 42 :112–129, oct 2013.
- [Bea11] S. B. Beale. CROSSFLOW. In *A-to-Z Guide to Thermodynamics, Heat and Mass Transfer, and Fluids Engineering*. Begellhouse, 2011.
- [Ber72] Ronald Betts Berklite. *Added mass of submerged objects of arbitrary shapes*. PhD thesis, Monterey, California. Naval Postgraduate School, 1972.
- [BFSO08] Ha H. Bui, Ryoichi Fukagawa, Kazunari Sako, and Shintaro Ohno. Lagrangian meshfree particles method (SPH) for large deformation and failure flows of geomaterial using elastic-plastic soil constitutive model. *International Journal for Numerical and Analytical Methods in Geomechanics*, 32(12) :1537–1570, aug 2008.
- [BGN00] Richard H. Byrd, Jean Charles Gilbert, and Jorge Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1) :149–185, nov 2000.
- [BGPT18] Stefan Band, Christoph Gissler, Andreas Peer, and Matthias Teschner. Mls pressure extrapolation for the boundary handling in divergence-free sph, 2018.
- [BHKB13] S. Braun, C. Höfler, R. Koch, and H.-J. Bauer. Modeling fuel injection in gas turbines using the meshless smoothed particle hydrodynamics method. In *Volume 1A : Combustion, Fuels and Emissions*. ASME, jun 2013.
- [BK00] Javier Bonet and Sivakumar Kulasegaram. Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations. *International journal for numerical methods in engineering*, 47(6) :1189–1214, 2000.
- [BK15] Jan Bender and Dan Koschier. Divergence-free smoothed particle hydrodynamics. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation - SCA '15*. ACM Press, 2015.
- [BK17] Jan Bender and Dan Koschier. Divergence-free SPH for incompressible and viscous fluids. *IEEE Transactions on Visualization and Computer Graphics*, 23(3) :1193–1206, mar 2017.
- [BKR88] J.U. Brackbill, D.B. Kothe, and H.M. Ruppel. Flip : A low-dissipation, particle-in-cell method for fluid flow. *Computer Physics Communications*, 48(1) :25–38, jan 1988.

- [Ble03] Robert D. Blevins. *Applied Fluid Dynamics Handbook*. Krieger Pub Co, 2003.
- [Blu] the blue carbon initiative.
- [BMHDGH03] M. Bursik, B. Martinez-Hackert, H. Delgado, and A. Gonzalez-Huesca. A smoothed-particle hydrodynamic automaton of landform degradation by overland flow. *Geomorphology*, 53(1-2) :25–44, jul 2003.
- [BR86] J.U. Brackbill and H.M. Ruppel. FLIP : A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *Journal of Computational Physics*, 65(2) :314–343, aug 1986.
- [Bre82] CE Brennen. A review of added mass and fluid inertial forces. Technical report, BRENNEN (CE) SIERRA MADRE CA, 1982.
- [BRP05] Javier Bonet and Miguel X. Rodríguez-Paz. Hamiltonian formulation of the variable-h SPH equations. *Journal of Computational Physics*, 209(2) :541–558, nov 2005.
- [BRP⁺13] Christian Boe, Jose Rodriguez, Carlos Plazaola, Ilka Banfield, Amaly Fong, Rony Caballero, and Adan Vega. A hydrodynamic assessment of a remotely operated underwater vehicle based on computational fluid dynamic—part 1—numerical simulation. *Computer Modelling in Engineering & Sciences*, 90, 2013.
- [But16] John Charles Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.
- [CAD⁺17] A.J.C. Crespo, C. Altomare, J.M. Domínguez, J. González-Cao, and M. Gómez-Gesteira. Towards simulating floating offshore oscillating water column converters with smoothed particle hydrodynamics. *Coastal Engineering*, 126 :11–26, aug 2017.
- [CBC99] JK Chen, JE Beraun, and TC Carney. A corrective smoothed particle method for boundary value problems in heat conduction. *International Journal for Numerical Methods in Engineering*, 46(2) :231–252, 1999.
- [CBJ99] J. K. Chen, J. E. Beraun, and C. J. Jih. An improvement for tensile instability in smoothed particle hydrodynamics. *Computational Mechanics*, 23(4) :279–287, may 1999.
- [CCCT07] Hsin-Hung Chen, Hsu-Hui Chang, Chun-Hung Chou, and Pin-Hsien Tseng. Identification of hydrodynamic parameters for a remotely operated vehicle using projective mapping method. In *2007 Symposium on Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies*. IEEE, apr Chen2007.
- [CDFS⁺14] R. Codd-Downey, P. Mojiri Forooshani, A. Speers, H. Wang, and M. Jenkin. From ROS to unity : Leveraging robot and virtual environment middleware for immersive teleoperation. In *2014 IEEE International Conference on Information and Automation (ICIA)*. IEEE, jul 2014.
- [CFL⁺03] A.K. Chaniotis, C.E. Frouzakis, J.C. Lee, A.G. Tomboulides, D. Poulikakos, and K. Boulouchos. Remeshed smoothed particle hydrodynamics for the simulation of laminar chemically reactive flows. *Journal of Computational Physics*, 191(1) :1–17, oct 2003.

- [CHAN02] Paul Cleary, Joseph Ha, Vladimir Alguine, and Thang Nguyen. Flow modelling in casting processes. *Applied Mathematical Modelling*, 26(2) :171–190, feb 2002.
- [CIPT14] Jens Cornelis, Markus Ihmsen, Andreas Peer, and Matthias Teschner. IISPH-FLIP for incompressible fluids. *Computer Graphics Forum*, 33(2) :255–262, may 2014.
- [CIV00] M. Caccia, G. Indiveri, and G. Veruggio. Modeling and identification of open-frame variable configuration unmanned underwater vehicles. *IEEE Journal of Oceanic Engineering*, 25(2) :227–240, April 2000.
- [CL03] Andrea Colagrossi and Maurizio Landrini. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *Journal of Computational Physics*, 191(2) :448–475, nov 2003.
- [Cle98] Paul W. Cleary. Modelling confined multi-material heat and mass flows using SPH. *Applied Mathematical Modelling*, 22(12) :981–993, dec 1998.
- [CLL17] C. S. Chin, W. P. Lin, and J. Y. Lin. Experimental validation of open-frame ROV model for virtual reality simulation and control. *Journal of Marine Science and Technology*, jul 2017.
- [CLW⁺07] Stefano Carpin, Mike Lewis, Jijun Wang, Stephen Balakirsky, and Chris Scrapper. USARSim : a robot simulator for research and education. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, apr 2007.
- [CMK15] Nuttapon Chentanez, Matthias Muller, and Tae-Yong Kim. Coupling 3d eulerian, heightfield and particle methods for interactive simulation of large scale liquid phenomena. *IEEE Transactions on Visualization and Computer Graphics*, 21(10) :1116–1128, oct 2015.
- [CML16] CMLabs. Vortex marine. <http://www.cm-labs.com/>, 2016.
- [Coe13] Ryan Geoffrey Coe. *Improved underwater vehicle control and maneuvering analysis with computational fluid dynamics simulations*. PhD thesis, Virginia Tech, 2013.
- [Cou92] Couchouron. Moyens d’essais français en hydrodynamique. Technical report, Institut Français de Recherche pour l’Exploitation de la Mer, 1992.
- [CPDH12] Paul W. Cleary, Mahesh Prakash, Raj Das, and Joseph Ha. Modelling of metal forging using SPH. *Applied Mathematical Modelling*, 36(8) :3836–3855, aug 2012.
- [CPM⁺95] A.P. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, and R.W. Brodersen. Optimizing power using transformations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(1) :12–31, 1995.
- [CPPH⁺] Krzysztof Choromanski, Aldo Pacchiano, Jack Parker-Holder, Yunhao Tang, Deepali Jain, Yuxiang Yang, Atil Iscen, Jasmine Hsu, and Vikas Sindhwani. Provably robust blackbox optimization for reinforcement learning.
- [CR99] Sharen J Cummins and Murray Rudman. An SPH projection method. *Journal of Computational Physics*, 152(2) :584–607, jul 1999.

- [CVL14] Daniel Cook, Andrew Vardy, and Ron Lewis. A survey of auv and robot simulators for multi-vehicle operations. In *2014 IEEE/OES Autonomous Underwater Vehicles (AUV)*, pages 1–8. IEEE, 2014.
- [CYLS17] Rongxin Cui, Chenguang Yang, Yang Li, and Sanjay Sharma. Adaptive neural network control of AUVs with control input nonlinearities using reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics : Systems*, 47(6) :1019–1029, jun 2017.
- [DA12] Walter Dehnen and Hossam Aly. Improving convergence in smoothed particle hydrodynamics simulations without pairing instability. *Monthly Notices of the Royal Astronomical Society*, 425(2) :1068–1082, aug 2012.
- [dB01] Carl de Boor. *A Practical Guide to Splines*. Springer-Verlag GmbH, 2001.
- [DCVB⁺13] J.M. Domínguez, A.J.C. Crespo, D. Valdez-Balderas, B.D. Rogers, and M. Gómez-Gesteira. New multi-GPU implementation for smoothed particle hydrodynamics on heterogeneous clusters. *Computer Physics Communications*, 184(8) :1848–1860, aug 2013.
- [dDPd08] E. A. de Barros, J. L. D. Dantas, A. M. Pascoal, and E. de Sa. Investigation of normal force and moment coefficients for an auv at nonlinear angle of attack and sideslip range. *IEEE Journal of Oceanic Engineering*, 33(4) :538–549, Oct 2008.
- [DEB19] Yves DEBARD. Résistance des matériaux IUT le mans, 2019.
- [DHD15] Zhipeng Duan, Boshu He, and Yuanyuan Duan. Sphere drag and heat transfer. *Scientific Reports*, 5(1), jul 2015.
- [DM98] L. Dagum and R. Menon. OpenMP : an industry standard API for shared-memory programming. *IEEE Computational Science and Engineering*, 5(1) :46–55, 1998.
- [E.19] Coumans E. Real-time physics simulation. <https://pybullet.org/wordpress/>, 2019.
- [Ear18] Steven Earle. Physical geology. 2018.
- [Ebe93] Rolf Ebert. *Développement d'un environnement de simulation de systèmes complexes. Application aux bâtiments*. Theses, Ecole Nationale des Ponts et Chaussées, November 1993.
- [ELDL11] Gilberto Echeverria, Nicolas Lassabe, Arnaud Degroote, and Séverin Lemaignan. Modular open robots simulation engine : Morse. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 46–51. IEEE, 2011.
- [ELLS08] YH Eng, WS Lau, E Low, and GGL Seet. Identification of the hydrodynamics coefficients of an underwater vehicle using free decay pendulum motion. In *International MultiConference of Engineers and Computer Scientists*, volume 2, pages 423–430, 2008.
- [ES12] Ayssam Elkady and Tarek Sobh. Robotics middleware : A comprehensive literature survey and attribute-based bibliography. *Journal of Robotics*, 2012 :1–15, 2012.

- [ES16] Ole A Eidsvik and Ingrid Schjølberg. Determination of hydrodynamic parameters for remotely operated vehicles. In *ASME 2016 35th International Conference on Ocean, Offshore and Arctic Engineering*, pages V007T06A025–V007T06A025. American Society of Mechanical Engineers, 2016.
- [ETT15] Tom Erez, Yuval Tassa, and Emanuel Todorov. Simulation tools for model-based robotics : Comparison of bullet, havok, mujoco, ode and physx. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4397–4404. IEEE, 2015.
- [FDHB14] Rozita Jalali Farahani, Robert A. Dalrymple, Alexis Hérault, and Giuseppe Bilotta. Three-dimensional SPH modeling of a bar/rip channel system. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 140(1) :82–99, jan 2014.
- [FGRS14] Giancarlo Fortino, Antonio Guerrieri, Wilma Russo, and Claudio Savaglio. Middlewares for smart objects and smart environments : Overview and comparison. In *Internet of Things*, pages 1–27. Springer International Publishing, 2014.
- [Flo11] France Floc’H. *Trajectories prediction of submerged bodies by coupling flow models and Euler-Newton equations*. Theses, Université de Bretagne occidentale - Brest, June 2011.
- [Fos11] Thor I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control : Vademecum de Navium Motu Contra Aquas Et de Motu Gubernando*. JOHN WILEY & SONS INC, 2011.
- [Fou16] Open Source Robotic Foundation. Gazebo : Robot simulation made easy. <http://gazebosim.org>, 2016.
- [FRM⁺18] Ramon A. Suarez Fernandez, E. Andres Parra R., Zorana Milosevic, Sergio Dominguez, and Claudio Rossi. Design, modeling and control of a spherical autonomous underwater vehicle for mine exploration. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, oct 2018.
- [FS91] Thor I. Fossen and Svein I. Sagatun. Adaptive control of nonlinear systems : A case study of underwater robotic systems. *Journal of Robotic Systems*, 8(3) :393–412, jun 1991.
- [FSOM10] Marc Freese, Surya Singh, Fumio Ozaki, and Nobuto Matsuhira. Virtual robot experimentation platform v-rep : a versatile 3d robot simulator. In *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, pages 51–62. Springer, 2010.
- [FSP⁺14] Leandro C. Fernandes, Jefferson R. Souza, Gustavo Pessin, Patrick Y. Shinzato, Daniel Sales, Caio Mendes, Marcos Prado, Rafael Klaser, André Chaves Magalhães, Alberto Hata, Daniel Pigatto, Kalinka Castelo Branco, Valdir Grassi, Fernando S. Osorio, and Denis F. Wolf. CaRINA intelligent robotic car : Architectural design and applications. *Journal of Systems Architecture*, 60(4) :372–392, apr 2014.
- [Gal19] Thierry Gallouët. Normes et conditionnement d’une matrice, chap. 1 systèmes linéaires, 2019.

- [GCBLT13] P.-M. GUILCHER, N. Couty, L. Brosset, and David Le Touzé. Simulations of Breaking Wave Impacts on a Rigid Wall at Two Different Scales with a Two-Phase Fluid Compressible SPH Model. *International Journal of Offshore and Polar Engineering (IJOPE)*, 23(4) :<http://www.isopec.org/publications/journals/journalDecember13.htm>, December 2013.
- [GG17] Shuoxin Gu and Shuxiang Guo. Performance evaluation of a novel propulsion system for the spherical underwater robot (SURIII). *Applied Sciences*, 7(11) :1196, nov 2017.
- [GGBK09] Thomas H. Greif, Simon C. O. Glover, Volker Bromm, and Ralf S. Klessen. Chemical mixing in smoothed particle hydrodynamics simulations. *Monthly Notices of the Royal Astronomical Society*, 392(4) :1381–1387, feb 2009.
- [GGD04] M. Gómez-Gesteira and Robert A. Dalrymple. Using a three-dimensional smoothed particle hydrodynamics method for wave impact on a tall structure. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 130(2) :63–69, mar 2004.
- [GHP01] Etienne Guyon, Jean-Pierre Hulin, and Luc Petit. *Hydrodynamique physique, nouvelle édition*. CNRS Editions, 2001.
- [GM77] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics : theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181(3) :375–389, dec 1977.
- [GMR⁺19] Nicolas Gartner, Niels Montanari, Mathieu Richier, Vincent Hugel, and Ramprasad Sampath. Towards real-time simulation of physically realistic pressure applied to submerged bodies using explicit and semi-implicit SPH algorithms. In IEEE/MTS, editor, *Proceedings IEEE/MTS Oceans, Marseille, France*, 2019.
- [Gol18] F. Goley. MONOLITHIC FLUID STRUCTURE INTERACTION MODEL, APPLICATION TO WATER ENTRY PROBLEM. In *Topical Problems of Fluid Mechanics 2018*. Institute of Thermomechanics, AS CR, v.v.i., 2018.
- [Gra07] William Graebel. *Advanced Fluid Mechanics*. Academic Press, 2007.
- [GRH18] Nicolas Gartner, Mathieu Richier, and Vincent Hugel. Hydrodynamics parameter identification of submerged bodies : numerical methods comparison and friction model analysis. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, October 2018.
- [GSA⁺13] Felipe Albring Guth, Luan Silveira, Marcos Amaral, Silvia Botelho, and Paulo Drews. Underwater visual 3d SLAM using a bio-inspired system. In *2013 Symposium on Computing and Automation for Offshore Shipbuilding*. IEEE, mar 2013.
- [GSBGW99] Domingo García Senz, Eduardo Bravo Guil, and S Woosley. Single and multiple detonations in white dwarfs. *Astronomy and astrophysics*, 349(1) :177–188, 1999.
- [GSTC] Pedro Gonnet, Matthieu Schaller, Tom Theuns, and Aidan B. G. Chalk. Swift : Fast algorithms for multi-resolution sph on multi-core architectures. *8th International SPHERIC Workshop, Trondheim, Norway*.

- [GTB⁺18] Valérie Girard, Mélanie Tron, Coline Brothier, Olivier Samat, Frédéric Guillot, Yann Gayet, and Didier Cassan. Diagnostics d'ouvrages par vecteurs aériens et sous-marins. In *Colloque CFBR : « Méthodes et techniques innovantes dans la maintenance et la réhabilitation des barrages et des digues »*, 2018.
- [GWP⁺18] Ting Gao, Yaxing Wang, Yongjie Pang, Qinglong Chen, and Yuanguai Tang. A time-efficient CFD approach for hydrodynamic coefficient determination and model simplification of submarine. *Ocean Engineering*, 154 :16–26, apr 2018.
- [Har62] Francis H Harlow. The particle-in-cell method for numerical solution of problems in fluid dynamics. Technical report, mar 1962.
- [HBD10] Alexis Hérault, Giuseppe Bilotta, and Robert A. Dalrymple. SPH on GPU with CUDA. *Journal of Hydraulic Research*, 48(sup1) :74–79, jan 2010.
- [HD14] Yu Huang and Zili Dai. Large deformation and failure simulations for geodisasters using smoothed particle hydrodynamics method. *Engineering Geology*, 168 :86–97, jan 2014.
- [Hop15] Philip F. Hopkins. A new class of accurate, mesh-free hydrodynamic simulation methods. *Monthly Notices of the Royal Astronomical Society*, 450(1) :53–110, apr 2015.
- [HP99] John Hultman and Arnaud Pharasyn. Hierarchical, dissipative formation of elliptical galaxies : is thermal instability the key mechanism?. hydrodynamical simulations including supernova feedback, multi-phase gas and metal enrichment in cdm : structure and dynamics of elliptical galaxies. *Astronomy and Astrophysics*, 347 :769–798, 1999.
- [HPR⁺17] Wei Hu, Wenxiao Pan, Milad Rakhsha, Qiang Tian, Haiyan Hu, and Dan Negrut. A consistent multi-resolution smoothed particle hydrodynamics method. *Computer Methods in Applied Mechanics and Engineering*, 324 :278–299, sep 2017.
- [HPW18] Zachary Harris, Tyler Paine, and Louis L. Whitcomb. Preliminary evaluation of null-space dynamic process model identification with application to cooperative navigation of underwater vehicles. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain(October 2018)*, October 2018.
- [HS08] Andreas Hölzer and Martin Sommerfeld. New simple correlation formula for the drag coefficient of non-spherical particles. *Powder Technology*, 184(3) :361–365, jun 2008.
- [HSG16] Eirik Hexeberg Henriksen, Ingrid Schjolberg, and Tor Berge Gjersvik. UW MORSE : The underwater modular open robot simulation engine. In *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*. IEEE, nov 2016.
- [Hum18] *Humanoid Robotics : A Reference*. SPRINGER NATURE, 2018.
- [IABT10] Markus Ihmsen, Nadir Akinci, Markus Becker, and Matthias Teschner. A parallel SPH implementation on multi-core CPUs. *Computer Graphics Forum*, 30(1) :99–112, nov 2010.
- [ICS⁺14] Markus Ihmsen, Jens Cornelis, Barbara Solenthaler, Christopher Horvath, and Matthias Teschner. Implicit incompressible sph. *IEEE Transactions on Visualization and Computer Graphics*, 20(3) :426–435, mar 2014.

- [IFU95] K. Ishii, T. Fujii, and T. Ura. An on-line adaptation method in a neural network based control system for AUVs. *IEEE Journal of Oceanic Engineering*, 20(3) :221–228, jul 1995.
- [IPN14] Serena Ivaldi, Vincent Padois, and Francesco Nori. Tools for dynamics simulation of robots : a survey based on user feedback. *arXiv preprint arXiv :1402.7050*, 2014.
- [IU00] Kazuo Ishii and Tamaki Ura. An adaptive neural-net controller system for an underwater vehicle. *Control Engineering Practice*, 8(2) :177–184, feb 2000.
- [JiRBN11] Taekwon Jang, Roger Blanco i Ribera, Jinhyuk Bae, and Junyong Noh. Simulating SPH fluid with multi-level vorticity. *International Journal of Virtual Reality*, 10(1) :21–27, jan 2011.
- [Joz] Richard Jozsa. Lecture notes 1b methods. PART II : PDEs on bounded domains : separation of variables.
- [JSB] *JSBSim Reference Manual*.
- [KAP11] Phoivos Koukouviniis, John Anagnostopoulos, and Dimitris Papantonis. Sph method used for flow predictions at a turgo impulse turbine : Comparison with fluent. 2011.
- [Ker14] Olivier Kermorgant. A dynamic simulator for underwater vehicle-manipulators. In *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, pages 25–36. Springer, 2014.
- [KGS09] Abbas Khayyer, Hitoshi Gotoh, and Songdong Shao. Enhanced predictions of wave impact pressure by improved incompressible SPH methods. *Applied Ocean Research*, 31(2) :111–131, apr 2009.
- [KH04] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2149–2154. IEEE, 2004.
- [Kon13] Efsthathios Konstantinidis. Added mass of a circular cylinder oscillating in a free stream. *Proceedings of the Royal Society A : Mathematical, Physical and Engineering Sciences*, 469(2156) :20130135, aug 2013.
- [KRR⁺16] Igor Kozhemyakin, Kirill Rozhdestvensky, Vladimir Ryzhov, Nikolay Semenov, and Mikhail Chemodanov. Educational marine robotics in smtu. In *International Conference on Interactive Collaborative Robotics*, pages 79–88. Springer, 2016.
- [KW07] James C. Kinsey and Louis L. Whitcomb. Adaptive identification on the group of rigid-body rotations and its application to underwater vehicle navigation. *IEEE Transactions on Robotics*, 23(1) :124–136, feb 2007.
- [Lab] G. Labonte. Fast adaptive control of a non-linear system by an ADALINE : motion in a fluid. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)*. IEEE.
- [LB16a] Martin Langmajer and Lukáš Bláha. Structural parameter identification of a small robotic underwater vehicle. In *Mechatronics and Robotics Engineering for Advanced and Intelligent Manufacturing*, pages 107–121. Springer International Publishing, aug 2016.

- [LB16b] V. Leble and G.N. Barakos. A coupled floating offshore wind turbine analysis with high-fidelity methods. *Energy Procedia*, 94 :523–530, sep 2016.
- [LDC⁺13] Sylvain Laroche, Daniel Dawes, Steve Cohan, Brian Moore, and Espen Rodsand. Simulation technology to assess ROV operations at an early planning phase. In *Offshore Technology Conference*. Offshore Technology Conference, 2013.
- [LL80] Lev Davidovich Landau and Evgenii Mikhailovich Lifshitz. *Course of theoretical physics*, volume 5. Pergamon Press, 3rd edition, 1980.
- [LL03] G R Liu and M B Liu. *Smoothed Particle Hydrodynamics*. WORLD SCIENTIFIC, oct 2003.
- [LL10] M. B. Liu and G. R. Liu. Smoothed particle hydrodynamics (SPH) : an overview and recent developments. *Archives of Computational Methods in Engineering*, 17(1) :25–76, feb 2010.
- [LLLZ03] M. B. Liu, G. R. Liu, K. Y. Lam, and Z. Zong. Smoothed particle hydrodynamics for numerical simulation of underwater explosion. *Computational Mechanics*, 30(2) :106–118, jan 2003.
- [LMBB07] ETIENNE LAC, ARNAUD MOREL, and DOMINIQUE BARTHÈS-BIESEL. Hydrodynamic interaction between two identical capsules in simple shear flow. *Journal of Fluid Mechanics*, 573 :149–169, feb 2007.
- [LMX⁺08] E.-S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence, and P. Stansby. Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method. *Journal of Computational Physics*, 227(18) :8417–8436, sep 2008.
- [LRR75] B. E. Launder, G. J. Reece, and W. Rodi. Progress in the development of a reynolds-stress turbulence closure. *Journal of Fluid Mechanics*, 68(3) :537–566, apr 1975.
- [LTS⁺17] Zhandong Li, Jianguo Tao, Hao Sun, Yang Luo, Liang Ding, and Zongquan Deng. Hydrodynamic calculation and analysis of a complex-shaped underwater robot based on computational fluid dynamics and prototype test. *Advances in Mechanical Engineering*, 9(11) :1–10, nov 2017.
- [Luc77] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82 :1013, dec 1977.
- [LXL05] M.B. Liu, W.P. Xie, and G.R. Liu. Modeling incompressible flows using a finite particle method. *Applied Mathematical Modelling*, 29(12) :1252–1270, dec 2005.
- [MAC⁺11] S. Marrone, M. Antuono, A. Colagrossi, G. Colicchio, D. Le Touzé, and G. Graziani. delta-SPH model for simulating violent impact flows. *Computer Methods in Applied Mechanics and Engineering*, 200(13-16) :1526–1542, mar 2011.
- [Mar06] Paul Martz. *OpenGL Distilled*. ADDISON WESLEY PUB CO INC, 2006.
- [Mat] Timothy G. Mattson. Introduction to OpenMP. www.openmp.org.
- [MC⁺06] Bernard Multon, Alain H. Clément, Marie Ruellan , Julien Seigneurieux, and Hamid ben Ahmed. *Les Nouvelles Technologies de l’Energie*, chapter

- Systèmes de conversion des ressources énergétiques marines, pages 221–266. Hermès Publishing, 2006.
- [MC17] Jean-Baptiste Mouret and Konstantinos Chatzilygeroudis. 20 years of reality gap. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion on - GECCO '17*. ACM Press, 2017.
- [MCG03] Matthias Müller, David Charypar, and Markus Gross. Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '03*, pages 154–159, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [Men94] F. R. Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32(8) :1598–1605, aug 1994.
- [MH04] Scott McDougall and Oldrich Hungr. A model for the analysis of rapid landslide motion across three-dimensional terrain. *Canadian Geotechnical Journal*, 41(6) :1084–1097, dec 2004.
- [MHL⁺09] Anderson Maciel, Tansel Halic, Zhonghua Lu, Luciana P. Nedel, and Suvranu De. Using the PhysX engine for physics-based virtual surgery with force feedback. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 5(3) :341–353, sep 2009.
- [MjCP⁺12] William A. Mattingly, Dar jen Chang, Richard Paris, Neil Smith, John Blevins, and Ming Ouyang. Robot design using unity for computer games and robotic simulations. In *2012 17th International Conference on Computer Games (CGAMES)*. IEEE, jul 2012.
- [MK99] J. J. Monaghan and A. Kos. Solitary waves on a cretan beach. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 125(3) :145–155, may 1999.
- [MLCP10] Jean-Christophe Marongiu, Francis Leboeuf, JoËlle Caro, and Etienne Parkinson. Free surface flows simulations in pelton turbines using an hybrid sph-ale method. *Journal of Hydraulic Research*, 48(S1) :40–49, 2010.
- [MM98] Parviz Moin and Krishnan Mahesh. DIRECT NUMERICAL SIMULATION : A tool in turbulence research. *Annual Review of Fluid Mechanics*, 30(1) :539–578, jan 1998.
- [MM13] Miles Macklin and Matthias Müller. Position based fluids. *ACM Transactions on Graphics*, 32(4) :1, jul 2013.
- [MMCK14] Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. Unified particle physics for real-time applications. *ACM Trans. Graph.*, 33(4) :153 :1–153 :12, July 2014.
- [Mon90] JJ Monaghan. Modelling the universe. *Publications of the Astronomical Society of Australia*, 8(3) :233–237, 1990.
- [Mon92] J. J. Monaghan. Smoothed particle hydrodynamics. *Annual review of astronomy and astrophysics*, 30 :543–574, 1992.
- [Mon94] J.J. Monaghan. Simulating free surface flows with SPH. *Journal of Computational Physics*, 110(2) :399–406, feb 1994.
- [Mon00] J.J. Monaghan. SPH without a tensile instability. *Journal of Computational Physics*, 159(2) :290–311, apr 2000.

- [Mon05] J J Monaghan. Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68(8) :1703–1759, jul 2005.
- [Mon12] J.J. Monaghan. Smoothed particle hydrodynamics and its diverse applications. *Annual Review of Fluid Mechanics*, 44(1) :323–346, jan 2012.
- [Mou06] Bachir Ben Moussa. On the convergence of SPH method for scalar conservation laws with boundary conditions. *Methods and Applications of Analysis*, 13(1) :29–62, 2006.
- [MSJT10] Matthias Muller, Jos Stam, Doug James, and Nils Thurey. Real time physics class notes. Class notes, NVIDIA and Autodesk and Cornell University and ETH Zürich, 2010.
- [MSK⁺12] Johannes Meyer, Alexander Sendobry, Stefan Kohlbrecher, Uwe Klingauf, and Oskar von Stryk. Comprehensive simulation of quadrotor UAVs using ROS and gazebo. In *Simulation, Modeling, and Programming for Autonomous Robots*, pages 400–411. Springer Berlin Heidelberg, 2012.
- [MSK15] Gergely Magyar, Peter Sinčák, and Zoltán Krizsán. Comparison study of robotic middleware for robotic applications. In *Advances in Intelligent Systems and Computing*, pages 121–128. Springer International Publishing, 2015.
- [MSV⁺16] Musa Morena Marcusso Manhaes, Sebastian A. Scherer, Martin Voss, Luiz Ricardo Douat, and Thomas Rauschenbach. UUV simulator : A gazebo-based package for underwater intervention and multi-robot simulation. In *OCEANS 2016 MTS/IEEE Monterey*. IEEE, sep 2016.
- [MVB11] Virag Mishra, S Vengadesan, and SK Bhattacharyya. Translational added mass of axisymmetric underwater vehicles with forward speed using computational fluid dynamics. *Journal of Ship Research*, 55(3) :185–195, 2011.
- [NVS⁺14] Đula Nađ, Antonio Vasilijević, Nikola Stilinović, Nikola Mišković, and Zoran Vukić. The robot operating system (ros) in simulation, control and navigation of marine vehicles. In *5th Conference on Marine Technology in memory of Academician Zlatko Winkler*, 2014.
- [OKT14] Rafik Ouchene, Mohamed Khalij, and Anne Tanière. Particules non sphériques en écoulement turbulent : quel jeu de forces hydrodynamiques doit-on utiliser ? In *Congrès français de mécanique*. AFM, Maison de la Mécanique, 39/41 rue Louis Blanc, 92400 Courbevoie, France . . . , 2014.
- [OMTdL16] G. Oger, S. Marrone, D. Le Touzé, and M. de Leffe. SPH accuracy improvement through the combination of a quasi-lagrangian shifting transport velocity and consistent ALE formalisms. *Journal of Computational Physics*, 313 :76–98, may 2016.
- [ON16] Pourya Omidvar and Pooyan Nikeghbali. Simulation of violent water flows over a movable bed using smoothed particle hydrodynamics. *Journal of Marine Science and Technology*, 22(2) :270–287, aug 2016.
- [oNAS50] The Society of Naval Architects and Marine Engineers (SNAME). Nomenclature for treating the motion of a submerged body through a fluid. Technical and Research Bulletin No., 1-5, April 1950.
- [PBOW03] Doug Perrault, Neil Bose, Siu O’Young, and Christopher D. Williams. Sensitivity of auv added mass coefficients to variations in hull and control plane geometry. *Ocean Engineering*, 30(5) :645 – 671, 2003.

- [PC86] V C Patel and H C Chen. Flow over tail and in wake of axisymmetric bodies : Review of the state of the art. *Journal of Ship Research*, 30(3), June 1986.
- [PD14] R. Pramanik and D. Deb. Implementation of smoothed particle hydrodynamics for detonation of explosive with application to rock fragmentation. *Rock Mechanics and Rock Engineering*, 48(4) :1683–1698, sep 2014.
- [Phy05] AGEIA PhysX. Ageia physx physics processing unit preview, May 2005.
- [Phy08] Nvidia PhysX. Nvidia completes acquisition of ageia technologies, February 2008.
- [Phy11] PhysXInfo. How to restore ppu support with latest physx drivers, January 2011.
- [Phy17] Nvidia PhysX. Cuda parallel computing platform, 2017.
- [PL10] Gérard Petit and Brian Luzum. IERS conventions (2010). Technical report, BUREAU INTERNATIONAL DES POIDS ET MESURES SEVRES (FRANCE), 2010.
- [PPFS12] Mario Prats, Javier Pérez, J Javier Fernández, and Pedro J Sanz. An open source tool for simulation and supervision of underwater intervention missions. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2577–2582. IEEE, 2012.
- [Pri12] Daniel J. Price. Smoothed particle hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics*, 231(3) :759–794, feb 2012.
- [PT17] Andreas Peer and Matthias Teschner. Prescribed velocity gradients for highly viscous SPH fluids with vorticity diffusion. *IEEE Transactions on Visualization and Computer Graphics*, 23(12) :2656–2662, dec 2017.
- [PTF10] A B Phillips, S R Turnock, and M Furlong. The use of computational fluid dynamics to aid cost-effective hydrodynamic design of autonomous underwater vehicles. *Proceedings of the Institution of Mechanical Engineers, Part M : Journal of Engineering for the Maritime Environment*, 224(4) :239–254, aug 2010.
- [PWW19] Zhouhua Peng, Jiasen Wang, and Jun Wang. Constrained control of autonomous underwater vehicles based on command optimization and disturbance estimation. *IEEE Transactions on Industrial Electronics*, 66(5) :3627–3635, may 2019.
- [PYDSM19] François Pétillon, Camille Yivin, Xavier Del Santo, and Jean-Jacques Maisonneuve. Underwater vehicle manoeuvring : performance prediction tool in constrained and shallow water. In *Proceedings IEEE/MTS Oceans, Marseille, France*, 2019.
- [RCM08] Martin Robinson, Paul Cleary, and Joseph Monaghan. Analysis of mixing in a twin cam mixer using smoothed particle hydrodynamics. *AIChE Journal*, 54(8) :1987–1998, aug 2008.
- [Rey95] O. Reynolds. On the dynamical theory of incompressible viscous fluids and the determination of the criterion. *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences*, 186(0) :123–164, jan 1895.

- [Rig07] Vincent Rigaud. Innovation and operation with robotized underwater systems. *Journal of Field Robotics*, 24(6) :449–459, 2007.
- [RKF⁺13] S. Riviere, S. Khelladi, S. Farzaneh, F. Bakir, and A. Tcharkhtchi. Simulation of polymer flow using smoothed particle hydrodynamics method. *Polymer Engineering & Science*, 53(12) :2509–2518, feb 2013.
- [RMBRV16] Juan A. Ramírez-Macías, Persijn Brongers, Santiago Rúa, and Rafael E. Vásquez. Hydrodynamic modelling for the remotely operated vehicle visor3 using CFD. *IFAC-PapersOnLine*, 49(23) :187–192, 2016.
- [Rob16] Coppelia Robotics. V-rep virtual robot experimentation platform. <http://www.coppeliarobotics.com/>, 2016.
- [ROS] *ROS manual*.
- [RS19] F. Roelofs and A. Shams. CFD—introduction. In *Thermal Hydraulics Aspects of Liquid Metal Cooled Nuclear Reactors*, pages 213–218. Elsevier, 2019.
- [RSF13] Eric Rohmer, Surya PN Singh, and Marc Freese. V-rep : A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1321–1326. IEEE, 2013.
- [RTEF⁺04] P. Ridao, A. Tiano, A. El-Fakdi, M. Carreras, and A. Zirilli. On the identification of non-linear models of unmanned underwater vehicles. *Control Engineering Practice*, 12(12) :1483–1499, dec 2004.
- [RVR⁺] Andrei A. Rusu, Mel Vecerik, Thomas Rothörl, Nicolas Heess, Razvan Pascanu, and Raia Hadsell. Sim-to-real robot learning from pixels with progressive nets.
- [SABW82] William C. Swope, Hans C. Andersen, Peter H. Berens, and Kent R. Wilson. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules : Application to small water clusters. *The Journal of Chemical Physics*, 76(1) :637–649, jan 1982.
- [Sag05] Pierre Sagaut. *Large Eddy Simulation for Incompressible Flows*. Springer-Verlag GmbH, 2005.
- [SCM⁺18] P.N. Sun, A. Colagrossi, S. Marrone, M. Antuono, and A.M. Zhang. Multi-resolution delta-plus-SPH with tensile instability control : Towards high reynolds number flows. *Computer Physics Communications*, 224 :63–80, mar 2018.
- [SCMZ17] P.N. Sun, A. Colagrossi, S. Marrone, and A.M. Zhang. The δ plus-SPH model : Simple procedures for a further improvement of the SPH scheme. *Computer Methods in Applied Mechanics and Engineering*, 315 :25–49, mar 2017.
- [SDFA18] Mohammad Taghi Sabet, Hamidreza Mohammadi Daniali, Alireza Fathi, and Ebrahim Alizadeh. Identification of an autonomous underwater vehicle hydrodynamic model using the extended, cubature, and transformed unscented kalman filter. *IEEE Journal of Oceanic Engineering*, 43(2) :457–467, apr 2018.
- [SEF13] Fabian Spreng, Peter Eberhard, and Florian Fleissner. An approach for the coupled simulation of machining processes using multibody system and smoothed particle hydrodynamics algorithms. *Theoretical and Applied Mechanics Letters*, 3(1) :013005, 2013.

- [Sev17] Josefine Severholt. Generic 6-dof added mass formulation for arbitrary underwater vehicles based on existing semi-empirical methods. Master's thesis, KTH, Naval Systems, 2017.
- [SGNTS14] Philippe St-Germain, Ioan Nistor, Ronald Townsend, and Tomoya Shibayama. Smoothed-particle hydrodynamics numerical modeling of structures impacted by tsunami bores. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 140(1) :66–81, jan 2014.
- [She68] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference on -*. ACM Press, 1968.
- [She84] Donald S. Shepard. Computer mapping : The SYMAP interpolation algorithm. In *Spatial Statistics and Models*, pages 133–145. Springer Netherlands, 1984.
- [Sho12] AD Short. Coastal processes and beaches. *Nature education knowledge*, 3(10) :15, 2012.
- [SIDPRAP06] A. Souto-Iglesias, L. Delorme, L. Pérez-Rojas, and S. Abril-Pérez. Liquid moment amplitude assessment in sloshing type problems with smooth particle hydrodynamics. *Ocean Engineering*, 33(11-12) :1462–1484, aug 2006.
- [SK98] Serge Sutulo and Sun-Young Kim. Systematic approach to pmm/rotating arm experiment planning, parameter estimation, and uncertainty analysis. In *International Symposium and Workshop on Forces Acting on a Maneuvering Vessel*, 1998.
- [SKCG17] Omer Faruk Sukas, Omer Kemal Kinaci, Ferdi Cakici, and Metin Kemal Gokce. Hydrodynamic assessment of planing hulls using overset grids. *Applied Ocean Research*, 65 :35–46, apr 2017.
- [SL03] Songdong Shao and Edmond Y.M. Lo. Incompressible SPH method for simulating newtonian and non-newtonian flows with a free surface. *Advances in Water Resources*, 26(7) :787–800, jul 2003.
- [SLT09] Leonardo Di G. Sigalotti, Hender López, and Leonardo Trujillo. An adaptive SPH method for strong shocks. *Journal of Computational Physics*, 228(16) :5888–5907, sep 2009.
- [Sma08] Tim Smalley. Nvidia finalises ageia deal, details future plans, February 2008.
- [SMA⁺16] Ramprasad Sampath, Niels Montanari, Nadir Akinci, Steven Prescott, and Curtis Smith. Large-scale solitary wave simulation with implicit incompressible sph. *Journal of Ocean Engineering and Marine Energy*, 2(3) :313–329, Aug 2016.
- [SMZ15] Pengnan Sun, Furen Ming, and Aman Zhang. Numerical simulation of interactions between free surface and rigid body using a robust SPH method. *Ocean Engineering*, 98 :32–49, apr 2015.
- [SMZW18] P. N. Sun, F. R. Ming, A. M. Zhang, and B. Wang. Viscous flow past a NACA0012 foil below a free surface through the delta-plus-SPH method. *International Journal of Computational Methods*, 16(02) :1846007, nov 2018.
- [SOT16] M.S. Shadloo, G. Oger, and D. Le Touzé. Smoothed particle hydrodynamics method for fluid flows, towards industrial applications : Motivations, current state, and challenges. *Computers & Fluids*, 136 :11–34, sep 2016.

- [SP09] B. Solenthaler and R. Pajarola. Predictive-corrective incompressible SPH. *ACM Transactions on Graphics*, 28(3) :1, jul 2009.
- [SP17] Sathish K. P. Sanjeevi and Johan T. Padding. On the orientational dependence of drag experienced by spheroids. *Journal of Fluid Mechanics*, 820, may 2017.
- [SSF97] T. Sarkar, P. G. Sayer, and S. M. Fraser. A study of autonomous underwater vehicle hull forms using computational fluid dynamics. *International Journal for Numerical Methods in Fluids*, 25(11) :1301–1313, 1997.
- [SSZ14] Mohammad Taghi Sabet, Pouria Sarhadi, and Mostafa Zarini. Extended and unscented kalman filters for parameter estimation of an autonomous underwater vehicle. *Ocean Engineering*, 91 :329–339, nov 2014.
- [STKPG16] J. Sverdrup-Thygesen, E. Kelasidi, K. Y. Pettersen, and J. T. Gravdahl. The underwater swimming manipulator - a bio-inspired AUV. In *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*. IEEE, nov 2016.
- [SW03] D.A. Smallwood and L.L. Whitcomb. Adaptive identification of dynamically positioned underwater robotic vehicles. *IEEE Transactions on Control Systems Technology*, 11(4) :505–515, jul 2003.
- [SW17] Andrew R. Spielvogel and Louis L. Whitcomb. Adaptive estimation of measurement bias in six degree of freedom inertial measurement units : Theory and preliminary simulation evaluation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, sep 2017.
- [SZPS09] CJ Shi, D Zhao, J Peng, and C Shen. Identification of ship maneuvering model using extended kalman filters. *Marine Navigation and Safety of Sea Transportation*, pages 105–110, 2009.
- [TC01] R. J. Thacker and H. M. P. Couchman. Star formation, supernova feedback, and the angular momentum problem in numerical cold dark matter cosmogony : Halfway there? *The Astrophysical Journal*, 555(1) :L17–L20, jul 2001.
- [TDPTK17] Thien Tran-Duc, Nhan Phan-Thien, and Boo Cheong Khoo. A smoothed particle hydrodynamics (SPH) study of sediment dispersion on the seafloor. *Physics of Fluids*, 29(8) :083302, aug 2017.
- [TDSW11] Hwee-Pink Tan, Roe Diamant, Winston K.G. Seah, and Marc Waldmeyer. A survey of techniques and challenges in underwater localization. *Ocean Engineering*, 38(14-15) :1663–1676, oct 2011.
- [TET12] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo : A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, oct 2012.
- [TM06] Alexandre M. Tartakovsky and Paul Meakin. Pore scale modeling of immiscible and miscible fluid flows using smoothed particle hydrodynamics. *Advances in Water Resources*, 29(10) :1464–1478, oct 2006.
- [TME19] T-mednet, 2019.
- [TMSW07] Alexandre M. Tartakovsky, Paul Meakin, Timothy D. Scheibe, and Rogene M. Eichler West. Simulations of reactive transport and precipitation with smoothed particle hydrodynamics. *Journal of Computational Physics*, 222(2) :654–672, mar 2007.

- [TMV⁺18] Alessandra Tesei, Michele Micheli, Arjan Vermeij, Gabriele Ferri, Marco Mazzi, Gabriel Grenon, Luca Morlando, Stefano Biagini, Kevin D. LePage, Riccardo Costanzi, Davide Fenucci, Andrea Caiti, and Andrea Munafo. Real-time underwater positioning and navigation of an AUV in deep waters. In *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*. IEEE, may 2018.
- [TNTN97] Robert Tramel, Robert Nichols, Robert Tramel, and Robert Nichols. A highly efficient numerical method for overset-mesh moving-body problems. In *13th Computational Fluid Dynamics Conference*, 1997.
- [TSLN07] A. Tiano, R. Sutton, A. Lozowicki, and W. Naeem. Observer kalman filter identification of an autonomous underwater vehicle. *Control Engineering Practice*, 15(6) :727–739, jun 2007.
- [TW13] Giancarlo Troni and Louis Whitcomb. Adaptive estimation of measurement bias in three-dimensional field sensors with angular rate sensors : Theory and comparative experimental evaluation. In *Robotics : Science and Systems IX*. Robotics : Science and Systems Foundation, jun 2013.
- [ULR13] Christian Ulrich, Marzia Leonardi, and Thomas Rung. Multi-physics SPH simulation of complex marine-engineering hydrodynamic problems. *Ocean Engineering*, 64 :109–121, may 2013.
- [VBAMdN07] D. Violeau, C. Buvat, K. Abed-Meraim, and E. de Nanteuil. Numerical modelling of boom and oil spill with SPH. *Coastal Engineering*, 54(12) :895–913, dec 2007.
- [vdVJS⁺07] Pepijn W.J. van de Ven, Tor A. Johansen, Asgeir J. Sørensen, Colin Flanagan, and Daniel Toal. Neural network augmented identification of underwater vehicle models. *Control Engineering Practice*, 15(6) :715–725, jun 2007.
- [VdW11] Jorgen Van der Wardt. *A graphical simulator for the dynamics of marine autonomous surface robots*. PhD thesis, Università di Pisa, 2011.
- [Ver67] Loup Verlet. Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Physical Review*, 159(1) :98–103, jul 1967.
- [Vio12] Damien Violeau. *Fluid Mechanics and the SPH Method : Theory and Applications*. OXFORD UNIV PR, 2012.
- [VN44] John Von Neumann. *Proposal and analysis of a new numerical method for the treatment of hydrodynamical shock problems*. Applied Mathematics Group, Institute for Advanced Study, 1944.
- [VR16] Damien Violeau and Benedict D. Rogers. Smoothed particle hydrodynamics (SPH) for free-surface flows : past, present and future. *Journal of Hydraulic Research*, 54(1) :1–26, jan 2016.
- [Wen95] Holger Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4(1) :389–396, dec 1995.
- [WHK17] Bilal Wehbe, Marc Hildebrandt, and Frank Kirchner. Experimental evaluation of various machine learning regression methods for model identification of autonomous underwater vehicles. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2017.

- [WHK19] Bilal Wehbe, Marc Hildebrandt, and Frank Kirchner. A framework for on-line learning of underwater vehicles dynamic models. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019.
- [Wie22] C Wieselsberger. Further information on the laws of fluid resistance. *Physikalische Zeitschrift*, 23, December 1922.
- [Wil93] David C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, 1993.
- [WK17] Bilal Wehbe and Mario Michael Krell. Learning coupled dynamic models of underwater vehicles using support vector regression. In *OCEANS 2017 - Aberdeen*. IEEE, jun 2017.
- [WL03] Jeen-Shing Wang and C.S.G. Lee. Self-adaptive recurrent neuro-fuzzy control of an autonomous underwater vehicle. *IEEE Transactions on Robotics and Automation*, 19(2) :283–295, apr 2003.
- [WLY] Jeen-Shing Wang, C.S.G. Lee, and Junku Yuh. An online self-organizing neuro-fuzzy control for autonomous underwater vehicles. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*. IEEE.
- [WMNO05] R.A. Waltz, J.L. Morales, J. Nocedal, and D. Orban. An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Mathematical Programming*, 107(3) :391–408, nov 2005.
- [WPR01] John Westwood, Barney Parsons, and Will Rowley. Global ocean markets. *Oceanography*, 14(3) :83–91, 2001.
- [WS06] John Ward-Smith. *Mechanics of Fluids, Eighth Edition*. CRC Press, 2006.
- [XZYC13] Feng Xu, Zao-Jian Zou, Jian-Chuan Yin, and Jian Cao. Identification modeling of underwater vehicles' nonlinear dynamics based on support vector machines. *Ocean Engineering*, 67 :68–76, jul 2013.
- [YAQ17] DSL Yi and HF Al-Qrimli. Identification of hydrodynamics coefficient of underwater vehicle using free decay pendulum method. *Journal of Powder Metallurgy & Mining*, 06(01), 2017.
- [YCM⁺15] Rui Yang, Benoit Clement, Ali Mansour, Ming Li, and Nailong Wu. Modeling of a complex-shaped underwater vehicle for robust control scheme. *Journal of Intelligent & Robotic Systems*, 80(3) :491–506, Dec 2015.
- [YGS13] Chunfeng Yue, Shuxiang Guo, and Liwei Shi. Hydrodynamic analysis of the spherical underwater robot SUR-II. *International Journal of Advanced Robotic Systems*, 10(5) :247, jan 2013.
- [YL13] XiuFeng Yang and MouBin Liu. Numerical modeling of oil spill containment by boom using SPH. *Science China Physics, Mechanics and Astronomy*, 56(2) :315–321, jan 2013.
- [YLJ⁺18] Yao Yan, Yang Liu, Haibo Jiang, Zhike Peng, Alasdair Crawford, James Williamson, Jamie Thomson, Gearoid Kerins, Azat Yusupov, and Sheikh Islam. Optimization and experimental verification of the vibro-impact capsule system in fluid pipeline. *Proceedings of the Institution of Mechanical Engineers, Part C : Journal of Mechanical Engineering Science*, 233(3) :880–894, may 2018.

- [YNV17] H. K. Yoon, T. T. Nguyen, and M. Van Nguyen. Establishment of equations of motion of a roV using cfd technique. In *OCEANS 2017 - Anchorage*, pages 1–6, Sep. 2017.
- [YZS15] P. K. Yeung, X. M. Zhai, and Katepalli R. Sreenivasan. Extreme events in computational turbulence. *Proceedings of the National Academy of Sciences*, 112(41) :12633–12638, sep 2015.
- [ZBE15] Ernani Mehdi Zare, Mohammad Bozorg, and Saeed Ebrahimi. Identification of an autonomous underwater vehicle dynamic using extended kalman filter with arma noise model. *International Journal of Robotics*, 2015.
- [ZDM⁺16] Xihu Zhang, Menglan Duan, Dongfeng Mao, Yang Yu, Jianxing Yu, and Yi Wang. A mathematical model of virtual simulation for deepwater installation of subsea production facilities. *Ships and Offshore Structures*, pages 1–14, 2016.
- [ZF02] Yi Zhu and Patrick J. Fox. Simulation of pore-scale dispersion in periodic porous media using smoothed particle hydrodynamics. *Journal of Computational Physics*, 182(2) :622–645, nov 2002.
- [ZFM99] Yi Zhu, Patrick J Fox, and Joseph P Morris. A pore-scale numerical model for flow through porous media. *International journal for numerical and analytical methods in geomechanics*, 23(9) :881–904, 1999.
- [ZSL⁺] Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Tossingbot : Learning to throw arbitrary objects with residual physics.

Table des figures

1.1	Schéma du fonctionnement d'un bathyscaphe	10
1.2	Un sous-marin de classe Ohio et le Nautille de l'Ifremer	11
1.3	Schéma du sous-marin Nautille de l'Ifremer	12
1.4	Marché mondiale des véhicules sous-marins	12
1.5	Plusieurs ROV commerciaux.	14
1.6	Plusieurs AUV	16
1.7	La morphologie typique d'une plage de sable	17
1.8	Mer calme et mer agitée	18
1.9	Platier de la falaise des Hachettes en Normandie	19
1.10	Zone de déferlement ne débouchant pas sur une plage ou une falaise	20
1.11	Action des vagues dans la zone de surf	21
1.12	Un plongeur évalue l'état du corail en méditerranée	22
1.13	Guerre des mines	22
1.14	Schéma représentant l'architecture d'un environnement de simulation	27
1.15	Simulation d'un balancier avec Nvidia PhysX	32
1.16	Expérience de chute d'un cube dans Nvidia PhysX	34
1.17	Évolution de l'altitude du cube pour un essai de coulée	34
1.18	Simulation d'un balancier sous FleX	35
1.19	Chute de sphère simulée avec FleX	35
1.20	Maillage de l'espace du fluide entourant un véhicule marin pour le calcul de paramètre hydrodynamique	39
1.21	Les différents types de mailles en trois dimensions utilisables pour le maillage	40
1.22	Simulation de la circulation du fluide dans une zone côtière avec un modèle utilisant des particules	41
2.1	Référentiel lié au véhicule sous-marin	46
2.2	Présentation des repères liés au véhicule marin. Le repère fixe par rapport au centre de gravité (CG) décalé du référentiel arbitraire de centre O (CO).	48
2.3	Centre de poussée d'Archimède CB et centre de gravité CG (Source Fossen [Fos11])	51
2.4	Géométrie de la capsule	56
2.5	Paradoxe de d'Alembert	58
2.6	Évolution du coefficient de frottement C_d d'une sphère dans un écoulement de volume infini	60
2.7	Régime sous-critique et super-critique de l'écoulement autour de la sphère	61
2.8	Évolution du coefficient de frottement C_d en fonction du rapport de dimension D/D_0	61

2.9	Vue de la capsule face à l'écoulement avec un angle d'attaque α . (x, y) le repère global où l'écoulement se fait selon x et (x', y') le repère local de la capsule. .	62
2.10	Schéma d'un profil de Joukovski avec les points bord d'attaque et bord de fuite	68
2.11	Effet de la condition de Kutta sur l'écoulement de fluide parfait autour d'un aileron (Source : [Flo11])	68
2.12	Schéma d'un ROV placé sur un mécanisme à mouvement plan (PMM)	69
2.13	Le balancier dans le bassin à houle de l'Université de Toulon	71
2.14	Schéma du balancier étudié	72
2.15	Signal angulaire mesurée et ses dérivées sans filtre	77
2.16	Identification des paramètres (9 essais) sur 2, 3 et 4 périodes. LQ : Modèle de frottement linéaires et quadratiques.	79
2.17	Masse ajoutée en fonction de la vitesse maximale au cours des 9 essais. L, Q, LQ : Modèles de frottement Linéaires, Quadratiques, and Linéaires et Quadratiques	80
2.18	Variations des coûts par rapport aux coefficients pour l'essai 7. L : frottements linéaires, Q : frottements quadratiques	81
2.19	Variation des coûts inférieure à 1 en ce qui concerne les coefficients K_l et K_q - essai 7 - $m_a = 1.57 kg$. LQ : Modèle de frottement linéaire et quadratique.	82
2.20	Évolution du frottement quadratique en fonction de la vitesse la plus élevée dans la plage glissante au cours des 9 essais comparée à deux C_d qui délimitent les plages de Reynolds caractéristiques	84
2.21	Évolution du frottement quadratique en fonction de la vitesse la plus élevée dans la plage glissante au cours des 9 essais	84
2.22	Force de masse ajoutée et force de traînée appliquées sur la sphère pendant l'oscillation à une vitesse maximale donnée. (Modèle Q).	84
3.1	Chronologie du développement des applications de la méthode SPH	89
3.2	Valeur approchée d'un dirac en dimension 1 avec la fonction kernel	94
3.3	Représentation des fonctions $f_W(q)$ et $f'_W(q) sign(q)$ pour le B-spline cubique.	96
3.4	Représentation des fonctions $f_W(q)$ et $f'_W(q) sign(q)$ pour le B-spline quintique.	97
3.5	Approximation discrète d'un champ par un groupe de particules à l'intérieur du rayon d'interaction de la fonction kernel	99
3.6	Particules solides générées (orange) à l'intérieur de la forme solide (vert) avec des particules fluides autour (bleu) et le rayon du kernel pour une particule fluide particulière (rouge).	100
3.7	Évolution des méthodes SPH	102
3.8	Gauche : La scène à simuler avec les différents solides Droite : La scène à simuler avec les différentes particules frontières. Le bassin est en jaune, les particules sont en bleus et la sphère est en rouge.	114
3.9	Evolution de la pression en fonction de la profondeur en utilisant les techniques d'extrapolation et de réflexion de la pression. (WCSPH)	115
3.10	Vue du champ de pression à la profondeur médiane du bassin avec les techniques d'extrapolation et de réflexion de pression. $z = 0.81 m$ and $p_{th} = 7931 Pa$. (WCSPH)	116
3.11	Vue du champ de pression au fond du bassin avec les techniques d'extrapolation et de réflexion de pression. $z = 1.62 m$ and $p_{th} = 15862 Pa$. (WCSPH)	116
3.12	Pression extrapolée sur la sphère avec WCSPH (<i>gauche</i>) et ISPH (<i>droite</i>).	119

3.13	Évolution des coefficients de l'équation 3.78 au cours du temps dans le cas de WCSPH.	120
3.14	Évolution des coefficients de l'équation 3.78 au cours du temps dans le cas de ISPH.	120
4.1	Situation finale après les essais de coulée en cas dynamique.	128
4.2	Expérience numérique d'accélération d'une sphère dans un bassin.	129
4.3	Différentes tailles de sphères et de particules utilisées pour l'identification de paramètres hydrodynamiques.	130
4.4	Évolution temporelle de la vitesse de la sphère de $\varnothing 0,4 m$ avec différentes tailles de particules δ_r	137
4.5	Évolution temporelle de la vitesse des sphères pour une taille de particules δ_r fixe (0.05)	138
4.6	Evolution de la masse ajoutée en fonction de la vitesse pour une sphère de diamètre $\varnothing 0,4 m$ et le modèle de frottement LQ	139
4.7	Evolution de la masse ajoutée en fonction de la vitesse pour une sphère de diamètre $\varnothing 0,4 m$ et le modèle de frottement L	140
4.8	Trajectoire décrite par la sphère au cours de l'expérimentation avec la force d'entrée du balancier hydrodynamique.	142
4.9	Trajectoire estimée avec NIVF pour l'expérimentation avec la force d'entrée du balancier hydrodynamique.	142
4.10	Évolution de la masse ajoutée lors de l'expérience du balancier hydrodynamique et conditionnement de la matrice décrivant le problème d'estimation.	143
4.11	Évolution attendue de la position et de la vitesse de la sphère au cours de l'essai de vitesse maximale	146
4.12	Etat final de la simulation pour l'essai de vitesse maximale avec des particules de taille $\delta_r = 0.025 m$	147
4.13	Courbe de vitesse limite obtenue au cours de la simulation avec deux différentes tailles de discrétisation	147
4.14	Évolution de K_l en fonction de la vitesse avec le modèle linéaire L.	150
4.15	Évolution de C_d en fonction de la vitesse avec le modèle quadratique Q.	151
4.16	Évolution du coefficient de frottement C_d avec différentes tailles de sphère	151
4.17	Évolution des forces de pression et de frottements visqueux s'appliquant sur la sphère au cours de l'essai avec la sphère de $\varnothing 0.4$ et différentes valeurs de δ_r	152
4.18	À gauche : Évolution de la fonction de coût de l'algorithme NIVF avec une sphère de $\varnothing 0,4$, avec la discrétisation $\delta_r = 0.025$ et le modèle de frottement L À droite : Zone où cette fonction de coût est inférieure à 0.75.	157
4.19	Evolution de la fonction de coût de l'algorithme NIVF avec une sphère de $\varnothing 0,4$, avec la discrétisation $\delta_r = 0.025$ et le modèle de frottement L dans la zone où elle est inférieure à 0.75	158
4.20	Forces mises en jeu pour un essai avec une sphère de $\varnothing 0,4$ et avec la discrétisation $\delta_r = 0.025$ en considérant un modèle de frottement linéaire.	158
4.21	À gauche : Évolution de la fonction de coût de l'algorithme NIVF avec une sphère de $\varnothing 0,4$, la discrétisation $\delta_r = 0.025$ et le modèle Q À droite : Zone où cette fonction de coût est inférieure à 0.75.	159

4.22	À gauche : Évolution de la fonction de coût de l'algorithme NIVF avec une sphère de $\varnothing 0,4$, la discrétisation $\delta_r = 0.025$ et le modèle LQ À droite : Zone où cette fonction de coût est inférieure à 0.75.	159
4.23	À gauche : Évolution de la fonction de coût de l'algorithme NIVF avec une sphère de $\varnothing 0,4$ pour un essai de mouvement libre, avec la discrétisation $\delta_r = 0.025$ et le modèle de frottement L À droite : Zone où cette fonction de coût est inférieure à 0.75.	161
5.1	Simulation avec un algorithme multi-résolution. Les particules sont coloriées en fonction de leur vitesse. (Source : Hu [HPR ⁺ 17])	169
5.2	Principe de la méthode de calcul de vorticité de Jang (Source : [JiRBN11]) .	170
5.3	Scène simulée avec la capsule	171
5.4	Évolution temporelle de la vitesse de la capsule au cours du temps comparée à une trajectoire de référence	172
A.1	Présentation des repères liés à la géographie de la terre.	176
B.1	Vue d'une simulation de mission d'un ROV sous Vortex	189
B.2	Image issue de l'environnement de simulation UWMORSE	190
B.3	Capture d'écran d'UWSim	191
B.4	Capture d'écran d'un robot sous-marins dans Gazebo	192
B.5	Capture d'écran de V-REP	193
B.6	Simulation avec un BlueROV 2 sous Unity	194
B.7	Environnement de simulation Neutrino	197
C.1	Méthodes de différentiation	200
C.2	Méthode d'Euler	201
C.3	Intégration Runge-Kutta 2	203
C.4	Comparaison des méthodes d'intégration de Runge-Kutta	205
C.5	Comparaison des méthodes d'intégration de Runge-Kutta	209
C.6	Propagation de l'erreur avec la méthode explicite	210
C.7	Comparaison d'intégration numérique pendant la chute de cube	215
D.1	Calcul de la flèche maximale de la barre de flexion	218
D.2	Déformation de l'axe principal	219
D.3	Répartition des contraintes locales dans l'axe principal (Contrainte en <i>MPa</i>)	219
D.4	Schéma expliquant le lien entre précision angulaire et précision de la position de l'objet à étudier	220
H.1	Calcul de la masse ajoutée d'une sphère	232
H.2	Puits et source de potentiel	233
H.3	Circulation rectiligne uniforme	233
H.4	Circulation du fluide lors d'un écoulement stationnaire autour d'une sphère .	235
H.5	Circulation du fluide autour d'une sphère qui accélère dans un fluide statique	235
J.1	Schéma des interactions entre les classes dans Neutrino	242
J.2	Schéma du fonctionnement d'OpenMP	243

Nicolas Gartner

COncption de Systèmes Mécaniques et Robotiques, Université de Toulon

Identification de paramètres hydrodynamiques par simulation avec Smoothed Particle Hydrodynamics

Cette thèse porte sur les techniques de simulations des interactions dynamiques entre un véhicule sous-marin et l'eau qui l'entoure. L'objectif principal est de proposer une solution satisfaisante pour pouvoir, en amont du processus de conception, tester des algorithmes de contrôle et des formes de coques pour véhicules sous-marins. Il serait alors intéressant de pouvoir simuler en même temps la dynamique du solide et celle du fluide. L'idée développée dans cette thèse est d'utiliser la technique Smoothed Particles Hydrodynamics (SPH), qui est très récente et qui modélise le fluide comme un ensemble de particules sans maillage. Afin de valider les résultats de simulations une première étude a été réalisée avec un balancier hydrodynamique. Cette étude a permis la mise au point d'une méthode innovante d'estimation de paramètre hydrodynamique (forces de frottement et masse ajoutée) qui est plus robuste que les méthodes existantes lorsqu'il est nécessaire d'utiliser des dérivées numériques du signal mesuré. Ensuite, l'utilisation de deux types de solveur SPH : Weakly Compressible SPH et Incompressible SPH, est validée en suivant la démarche de validation proposée dans cette thèse. Sont étudiés, premièrement, le comportement du fluide seul, deuxièmement, un cas hydrostatique, et enfin un cas dynamique. L'utilisation de deux méthodes de modélisation de l'interaction fluide-solide : la méthode de réflexion de la pression et la méthode d'extrapolation est étudiée. La capacité d'atteindre une vitesse limite due aux forces de frottement est démontrée. Les résultats d'estimation des paramètres hydrodynamiques à partir des essais de simulation est finalement discutée. La masse ajoutée simulée du solide s'approche de la réalité, mais les forces de frottement semblent actuellement ne pas correspondre à la réalité. Des pistes d'améliorations pour pallier à ce problème sont proposées.

mots clés : Paramètres hydrodynamiques, SPH, Méthode numérique, Interaction, Fluide-solide, Fluide-structure, écoulement incompressible, Robotique sous-marine

Identification of hydrodynamic parameters by simulation with Smoothed Particle Hydrodynamics

This thesis focuses on techniques that allows the simulation of dynamic interactions between an underwater vehicle and the surrounding water. The main objective is to propose a satisfactory solution to be able to test control algorithms and hull shapes for underwater vehicles upstream of the design process. In those cases, it would be interesting to be able to simulate solid and fluid dynamics at the same time. The idea developed in this thesis is to use the Smoothed Particles Hydrodynamics (SPH) technique, which is very recent, and which models the fluid as a set of particles without mesh. In order to validate the simulation results a first study has been performed with a hydrodynamic pendulum. This study allowed the development of an innovative method for estimating the hydrodynamic parameters (friction forces and added mass) which is more robust than previous existing methods when it is necessary to use numerical derivatives of the measured signal. Then, the use of two types of SPH solver : Weakly Compressible SPH and Incompressible SPH, is validated following the validation approach proposed in this thesis. Firstly, the behaviour of the fluid alone is studied, secondly, a hydrostatic case, and finally a dynamic case. The use of two methods for modelling the fluid-solid interaction : the pressure mirroring method and the extrapolation method is studied. The ability to reach a limit velocity due to friction forces is demonstrated. The results of the hydrodynamic parameters estimation from simulation tests are finally discussed. The simulated added mass of the solid approaches reality, but the friction forces currently seem not to correspond to reality. Possible improvements to overcome this problem are proposed.

keywords: Hydrodynamic parameters, SPH, Numerical method, Interaction, Fluid-solid, Fluid-structure, Incompressible flow, Underwater robotics

