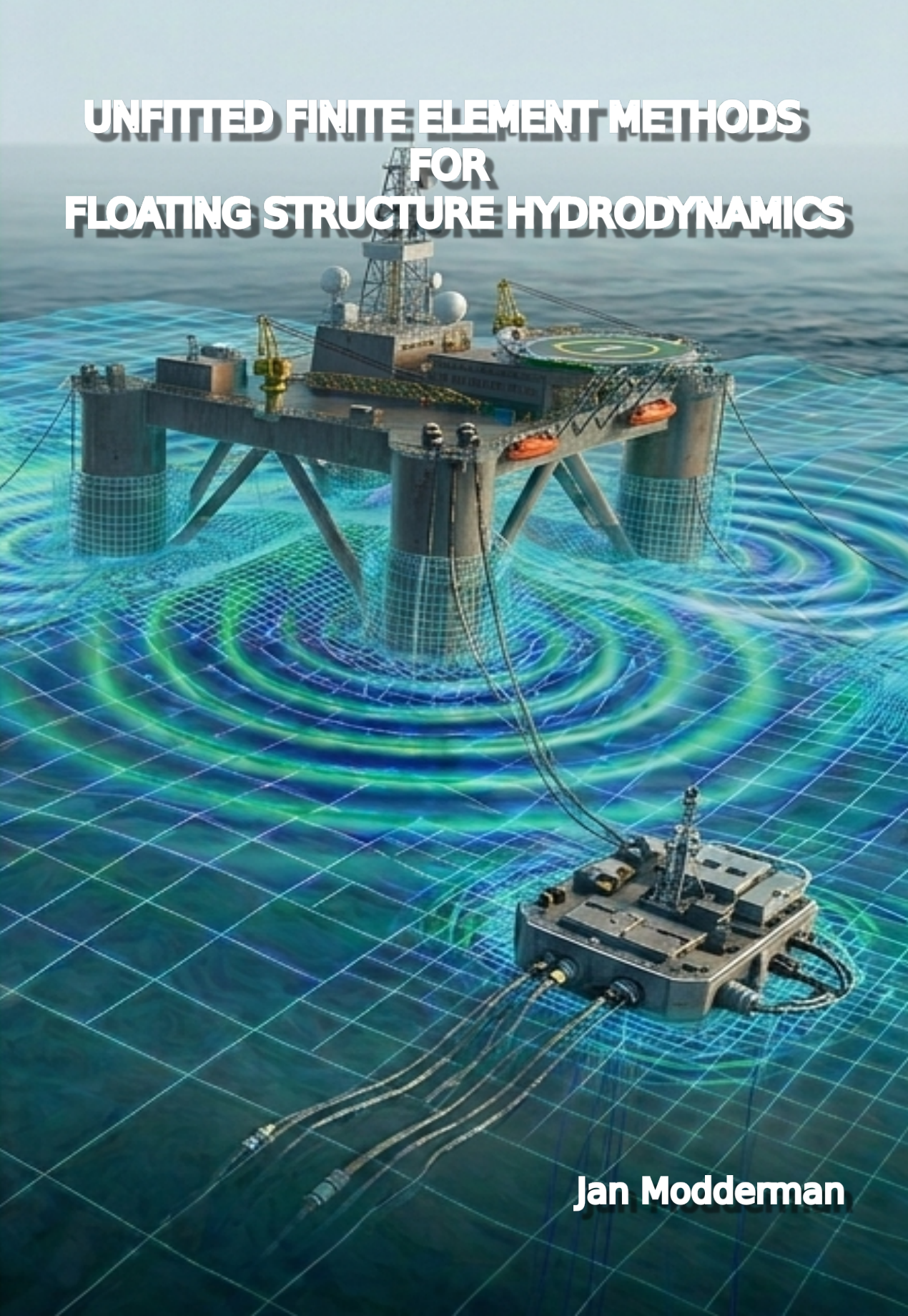
UNFITTED FINITE ELEMENT METHODS FOR FLOATING STRUCTURE HYDRODYNAMICS



Jan Modderman

Unfitted Finite Element Methods for Floating Structure Hydrodynamics

Unfitted Finite Element Methods for Floating Structure Hydrodynamics

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. H. Bijl,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen
op maandag 15 juni 2026 om 15:00 uur

door

Jan MODDERMAN

Dit proefschrift is goedgekeurd door de promotor en de copromotor.

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. A.V. Metrikine,	Technische Universiteit Delft, promotor
Dr. J.O. Colomes Gene,	Technische Universiteit Delft, copromotor
Onafhankelijke leden:	
Prof. dr. ir. L.J. Sluys,	Technische Unversiteit Delft
Prof. dr. ir. C. Vuik,	Technische Unversiteit Delft
Dr. A.P. Engsig-Karup,	Technical University of Denmark
Prof. dr. G. Scovazzi,	Duke University
Prof. dr. E. Bachynski-Polić,	Norwegian University of Science and Technology

Keywords: unfitted finite element methods, Cut FEM, Aggregated unfitted FEM, shifted boundary method, weighted shifted boundary method, linearized potential flow, fluid-structure interaction, floating structures, free-surface waves, adaptive mesh, refinement, added mass, added damping, benchmarking, free decay studies, hydrodynamics

Printed by: Gildeprint - Enschede

Cover: by J. Modderman using A.I.

ISBN 978-94-6518-348-0

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

"It is not knowledge, but the act of learning, not the possession of but the act of getting there, which grants the greatest enjoyment."

Carl Friedrich Gauß

Contents

Summary

There is a growing need for new floating structures for renewable energy and offshore infrastructure, driven by abundant offshore resources and limited land space. Current design processes rely heavily on physical experiments and conventional grid-based numerical tools, which require mesh generation for each design iteration. This meshing step is identified as a major bottleneck, as seen in aerospace and automotive industries, due to its cost and complexity. The same challenge is presumed to apply to offshore floating-structure design. To circumvent this, one can use unfitted, embedded, or immersed boundary methods that reduce human intervention in the meshing step. These unfitted methods have seen limited application in modeling offshore structures. In this thesis specifically, we investigate unfitted finite element methods as an alternative simulation approach for floating offshore structures.

We introduce and justify the fundamental assumptions of the physical modeling for both the fluid and the structure. Specifically, we derive the linearized potential flow formulation in reduced form for the fluid and model the structure as a rigid body. We demonstrate how we couple the fluid and the structure in a single system of equations in the time domain. In this work, the structure is only incorporated as a boundary condition, due to the rigid body assumption. As we work with linearized potential flow, we also derive the system of equations in the frequency domain representation. We transform the systems of equations to their corresponding weak formulations such that we can deal with a challenging integral on the dynamic boundary condition for the structure more easily by assuming a constant wetted surface of the structure.

The traditional finite element method is introduced, and the time and frequency domain weak formulations are transformed to their corresponding discrete weak formulations suitable for the finite element method. A concise historical overview of unfitted, embedded and immersed boundary methods is presented, which highlights the motivation behind their development and challenges encountered. We then compare the newest unfitted methods and select the cut finite element method (CutFEM), aggregated unfitted finite element method (AgFEM), shifted boundary method (SBM), and weighted shifted boundary method (WSBM), because of their suitable properties for moving domains and complex geometries. After which, each of the four aforementioned methods is explained in detail. We describe the discrete spaces and discrete weak forms for each method. And we derive the shifted boundary condition for the kinematic and dynamic boundary conditions at the structural boundary.

To assess and compare the four unfitted methods, we perform a benchmarking study in 2D and 3D, using both implicit and explicit geometry representations. Convergence rates, condition numbers, computational performance, and implementation complexity are compared. CutFEM and AgFEM exhibit the expected convergence rates, whereas SBM and WSBM converge one order lower than CutFEM and AgFEM because no gradient recovery is used. Note that this is only because the problem considered here is Neumann

type boundaries. For Dirichlet type boundaries, all four methods should have optimal convergence rates. Regarding numerical conditioning, SBM has the smallest condition numbers overall; for $p_e = 1$, AgFEM is comparable to SBM, and for $p_e = 2$, AgFEM's condition numbers are similar to CutFEM's. WSBM has the largest condition numbers. Performance tests in 2D show consistent trends in runtime and memory allocations: SBM is fastest and most memory-efficient, followed by CutFEM, then WSBM, and finally AgFEM. AgFEM's main bottleneck is the costly initialization of the finite element space due to its aggregation algorithm. In terms of implementation complexity, AgFEM is the most complex. CutFEM and WSBM both pose challenges, with CutFEM further complicated by its reliance on a tessellation algorithm for integration. SBM is easiest to implement, although its discrete weak formulation is relatively more involved.

We demonstrate the time domain capabilities of the numerical framework for both 2D and 3D problems, including multi-body simulations, using simple and realistic floating structures. All four unfitted finite element methods are shown to agree with a body-fitted method and to reproduce experimental heave free-decay results, provided sufficient mesh resolution and polynomial order. An adaptive mesh refinement strategy is proposed, because Cartesian grids cannot efficiently represent fine geometric details in 3D. This strategy is completely implemented for CutFEM and AgFEM, but for SBM and WSBM, the Hessian operator on the reference element for adaptively refined grids is still missing, so a term in the shifting operator is currently neglected. With appropriate mesh refinement and polynomial degree, all unfitted methods give consistent results; CutFEM and AgFEM are less sensitive to coarse meshes, while SBM and WSBM may require gradient recovery to reach high accuracy with fewer elements. WSBM in particular, can show spurious oscillations or divergence on non-adaptive meshes, highlighting the critical role of mesh adaptivity and the need to complete the Hessian operator definition for a full comparison of SBM and WSBM.

The numerical framework's performance for frequency domain problems is demonstrated by applying CutFEM, AgFEM, and SBM to 2D cases to compute added mass and added damping coefficients, and extending AgFEM to 3D realistic floating structures using unstructured refined meshes from GMSH. The methods successfully reproduce these coefficients for simple geometries and, with AgFEM, for realistic ones. The accuracy of added mass solutions appears more sensitive to mesh refinement near the structure than that of added damping. A remaining challenge is optimally balancing domain size and local mesh refinement for each wavelength, rather than relying on a single background mesh.

For future research, the dissertation's completeness can be improved by implementing gradient recovery for SBM and WSBM and by defining the Hessian on the reference element. For frequency domain problems, an optimal, wavelength-dependent mesh strategy should be developed using the adaptive approach from the time domain studies, balancing domain size, refinement level, and automated background grid generation. An extension of the solvers to parallel computing is underway to enable large-scale multi-structure simulations, the challenge of which is parallal adaptive AgFEM. The numerical framework should have validated wave generation and damping modules, enabling longer simulations within a limited domain. For large-scale simulations, the first priority is validating the multi-body frequency domain formulation. Comparable large-scale validation studies must also be identified and performed for time domain simulations. For the time domain studies,

mooring systems should be incorporated, either via linear stiffness matrices or more advanced models. The current framework can be expanded upon by incorporating higher fidelity models. This also introduces the challenges of moving free surfaces and moving structures that require remeshing. One step avoiding free surfaces with higher fidelity models, is modeling of fluid structure interaction of submerged risers or mooring lines which have marine growth. Lastly, we have proposed the generalized SBM (GSBM) to address challenges encountered during this project. This method should be subjected to the benchmark initiative and it can be a promising tool for the time domain simulations.

Samenvatting

Er is een groeiende behoefte aan nieuwe drijvende constructies voor hernieuwbare energie en offshore-infrastructuur, gedreven door de grote hoeveelheid offshore-bronnen en de beperkte beschikbaarheid van land. De huidige ontwerpprocessen zijn afhankelijk van fysieke experimenten en conventionele, op roostergebaseerde numerieke hulpmiddelen, waarbij voor elke ontwerpiteratie een nieuw rooster moet worden gegenereerd. In de auto- en luchtvaartindustrie, wordt het genereren van zo'n rooster gezien als een belangrijk knelpunt vanwege de kosten en complexiteit die daarbij gepaard gaan. Wij doen de aanname dat eenzelfde uitdaging geldt voor het ontwerpen van drijvende offshore-constructies. Om dit te omzeilen, kan gebruik worden gemaakt van unfitted, embedded of immersed boundary-methoden, die menselijke interventie bij het maken van een rooster verminderen. Deze unfitted methoden zijn tot nu toe slechts beperkt toegepast bij de modellering van offshore-constructies. In dit proefschrift onderzoeken we specifiek unfitted eindige-elementmethoden als een alternatief simulatieraamwerk voor drijvende offshore-constructies.

We introduceren en motiveren de fundamentele aannames voor het fysisch modelleren van zowel de vloeistof als de structuur. We leiden in het bijzonder de gelineariseerde potentiaalstroming in gereduceerde vorm af voor de vloeistof, en modelleren de structuur als een star lichaam. We laten zien hoe we de vloeistof en structuur koppelen in één systeem van vergelijkingen in het tijdsdomein. In dit werk wordt de structuur uitsluitend opgenomen als een randvoorwaarde, vanwege de aanname van een star lichaam. We leiden het systeem van vergelijkingen ook af in de frequentiedomeinrepresentatie, omdat we werken met gelineariseerde potentiaalstroming. We transformeren de systemen naar de bijbehorende zwakke formuleringen, zodat we de complexe integraal op de dynamische randvoorwaarde eenvoudiger kunnen behandelen door een constant nat oppervlak van de structuur aan te nemen.

We introduceren de traditionele eindige-elementmethode en de zwakke formuleringen in tijd en frequentie worden omgezet naar hun bijbehorende discrete zwakke formuleringen, geschikt voor eindige-elementdiscretisatie. Een beknopt historisch overzicht van unfitted, embedded en immersed boundary-methoden wordt gegeven, waarin de drijfveren voor hun ontwikkeling en de bijbehorende uitdagingen worden besproken. Vervolgens vergelijken we de nieuwste unfitted methoden en selecteren we de cut eindige-elementmethode (CutFEM), de aggregated unfitted eindige-elementmethode (AgFEM), de shifted boundary method (SBM) en de weighted shifted boundary method (WSBM), vanwege hun geschikte eigenschappen voor bewegende domeinen en complexe geometrieën. Daarna worden de vier methoden in detail uitgelegd. We beschrijven de discrete ruimten en de discrete zwakke vormen voor elke methode. Ook leiden we de shifted boundary-conditie af voor de kinematische en dynamische randvoorwaarden op de structurele grens.

Om de vier unfitted methoden te evalueren en te vergelijken, voeren we een referentieonderzoek uit in 2D en 3D, met zowel impliciete als expliciete geometrierepresentaties.

Convergentiesnelheden, conditiegetallen, rekentijd en implementatiecomplexiteit worden vergeleken. CutFEM en AgFEM laten de verwachte convergentiesnelheden zien, terwijl SBM en WSBM convergeren één orde lager dan CutFEM en AgFEM, omdat er geen gradientherstelmethode wordt gebruikt. Dit komt door het feit dat het ons probleem bestaat uit Neumann type grensvoorwaarden. In het geval van Dirichlet type grensvoorwaarden, zullen SBM en WSBM optimale convergentiesnelheden moeten behalen met de huidige implementatie. Wat numerieke conditionering betreft heeft SBM de kleinste conditiegetallen; voor $p_e = 1$ is AgFEM vergelijkbaar met SBM, en voor $p_e = 2$ zijn AgFEM's conditiegetallen vergelijkbaar met die van CutFEM. WSBM heeft de grootste conditiegetallen. Prestatiemetingen in 2D laten consistente trends zien in rekentijd en geheugengebruik: SBM is het snelst en meest geheugenefficiënt, gevolgd door CutFEM, daarna WSBM, en tot slot AgFEM. Het belangrijkste knelpunt van AgFEM is de dure initialisatie van de eindige-elementruimte door het aggregatie algoritme. Qua implementatiecomplexiteit is AgFEM het meest uitdagend. CutFEM en WSBM brengen eveneens moeilijkheden met zich mee, waarbij CutFEM verder wordt gecompliceerd door zijn afhankelijkheid van een tessellatie algoritme voor integratie. SBM is het eenvoudigst te implementeren, hoewel de discrete zwakke formulering relatief uitgebreid is.

We demonstreren de tijdsdomeincapaciteiten van het numerieke raamwerk voor zowel 2D- als 3D-problemen, inclusief simulaties met meerdere structuren met eenvoudige en realistische drijvende constructies. Alle vier unfitted eindige-elementmethoden komen overeen met een traditionel eindige-elementmethode en reproduceren experimentele resultaten in de deinrichting, mits voldoende roosterresolutie en polynoomorde worden gebruikt. Een adaptieve roosterverfijningsstrategie wordt voorgesteld, omdat Cartesiaanse roosters inefficiënt zijn in het representeren van geometrisch fijne details in 3D. Deze strategie is volledig geïmplementeerd voor CutFEM en AgFEM, maar voor SBM en WSBM ontbreekt nog de Hessiaanoperator op het referentie-element voor adaptieve roosters; hierdoor wordt een term in de shifting-operator momenteel verwaarloosd. Met geschikte adaptieve verfijning en polynoomorde leveren alle unfitted methoden consistente resultaten; CutFEM en AgFEM zijn minder gevoelig voor lagere resolutie roosters, terwijl SBM en WSBM mogelijk een gradientherstelmethode vereisen om hoge nauwkeurigheid te behalen met weinig elementen. WSBM kan in het bijzonder onechte oscillaties of divergentie vertonen op niet-adaptieve roosters, wat de kritieke rol van adaptiviteit onderstreept en het belang om de Hessiaan volledig te definiëren voor een volledige vergelijking van SBM en WSBM.

De prestaties van het numerieke raamwerk in het frequentiedomein worden gedemonstreerd door CutFEM, AgFEM en SBM toe te passen op 2D-scenarios voor het berekenen van toegevoegde massa- en toegevoegde dempingcoëfficiënten, en door AgFEM uit te breiden naar realistische 3D constructies met gebruik van ongestructureerde, lokaal verfijnde roosters uit GMSH. De methoden reproduceren deze coëfficiënten succesvol voor eenvoudige geometrieën en, in het geval van AgFEM, ook voor realistische geometrieën. De nauwkeurigheid van toegevoegde massaoplossingen lijkt gevoeliger voor roosterverfijning nabij de structuur dan die van toegevoegde demping. Een resterende uitdaging is het optimaal balanceren van domeingrootte en lokale roosterverfijning voor elke golflengte, in plaats van te vertrouwen op één enkel achtergrondrooster.

Voor toekomstig onderzoek kan de volledigheid van het proefschrift worden verbeterd door een gradientherstelmethode te implementeren voor SBM en WSBM en door de

Hessiaan op het referentie element te definiëren. Voor frequentiedomeinproblemen moet een optimale, golflengte-afhankelijke roosterstrategie worden ontwikkeld, gebaseerd op de adaptieve aanpak uit de tijdsdomeinstudies, waarbij domeingrootte, verfijningsniveau en automatische roostergeneratie in balans worden gebracht. Een uitbreiding van de oplossers naar parallel rekenen is gaande om grootschalige meerdere-structuursimulaties mogelijk te maken, waarbij parallel adaptief AgFEM de belangrijkste uitdaging vormt. Het numerieke raamwerk moet gevalideerde modules bevatten voor golfgeneratie en demping, zodat langere simulaties in een beperkt domein mogelijk zijn. Voor grootschalige simulaties heeft de validatie van de meerdere-structuursformulering in het frequentiedomein de hoogste prioriteit. Vergelijkbare grootschalige validatiestudies moeten ook worden geïdentificeerd en uitgevoerd voor tijdsdomeinsimulaties. Voor de tijdsdomeinstudies moeten meerlijnen worden meegenomen, hetzij via lineaire stijfheidsmatrices, hetzij via geavanceerdere modellen. Het huidige raamwerk kan worden uitgebreid met modellen van hogere nauwkeurigheid, wat bijkomende uitdagingen introduceert zoals bewegende vrij-oppervlakken en bewegende structuren die remeshing vereisen. Een stap om vrij-oppervlakken te vermijden bij modellen van hogere nauwkeurigheid is het modelleren van vloeistof-structuur-interactie van ondergedompelde risers of meerlijnen met aangroei. Tot slot introduceren we de generalized SBM (GSBM) om uitdagingen binnen dit project te adresseren. Deze methode moet worden opgenomen in het referentieonderzoek en kan een veelbelovend hulpmiddel zijn voor tijdsdomeinsimulaties.

Preface

This dissertation presents the results of research carried out at Delft University of Technology between 2021 and 2025, within the field of computational methods for ocean and offshore engineering. The work was conducted as part of a broader research programme on digital twins, focussing on the development and application of numerical methods for the simulation of complex wave–structure interaction problems.

Although both of my parents pursued a PhD, I long considered this path unlikely for myself, with the intention of obtaining a degree and transitioning directly into professional practice. This perspective changed during my master's thesis, where I discovered a strong interest in research and problem-driven exploration. Towards the end of that project, I was encouraged to consider a PhD position on digital twins for offshore engineering, a topic sufficiently broad to allow the freedom to gradually shape my own research direction.

Over the course of the project, this freedom led to an increasing exposure to the finite element method and, in particular, to the study of embedded and unfitted formulations. What initially started as an application-orientated research topic gradually evolved into a methodological and computational endeavour, with a strong emphasis on the development, implementation, and validation of novel numerical techniques. As a result, this thesis positions itself at the intersection of computational fluid dynamics, numerical analysis, and scientific software development, aiming to bridge theoretical concepts with practical large-scale simulations.

The intellectual journey underlying this work involved both productive phases and inevitable setbacks. Looking back, one of the most valuable outcomes of the PhD has been the confidence gained through the versatility required to navigate unfamiliar topics, develop new tools, and adapt to changing research directions. While the process is often associated with challenges, it is ultimately the accumulation of experiences, first attempts, collaborations, and gradual personal growth that has defined this trajectory.

This thesis reflects that process, presenting not only the technical results but also the evolution from applying existing numerical methods to actively contributing to their development within the context of offshore engineering. Dear reader, I would like to invite you to follow this journey through the exposure to the methods and models presented here. To view the results not only as technical contributions but as the outcome of a process of exploration and refinement.

Jan
Rotterdam, January 2026

1

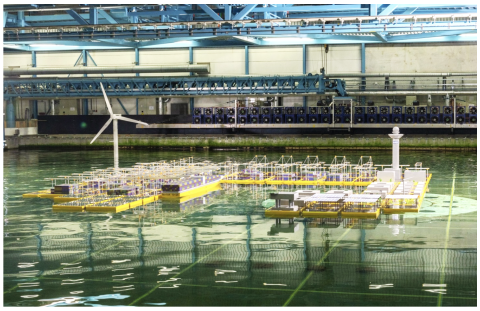
Introduction

Marine hydrodynamics is a foundational discipline that underpins the understanding of fluid behavior in the ocean and its interaction with maritime structures [1, 2]. Its study is critical across a wide range of applications, including offshore engineering, ship design, renewable energy, and naval operations.

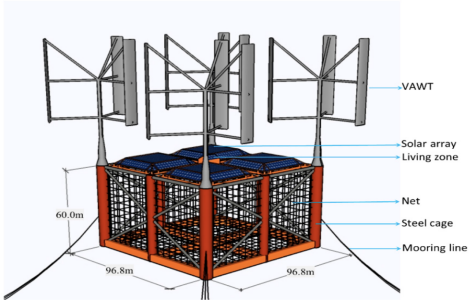
In offshore engineering, hydrodynamics govern the design and operation of platforms, floating structures, and subsea installations. Accurate prediction of wave, current, and wind forces is essential to ensure structural stability, operational reliability, and the longevity of offshore assets. The growing focus on marine renewable energy underscores the continued relevance of marine hydrodynamics as a foundational discipline. The European Commission has articulated a target to achieve climate neutrality by the year 2050, which involves generating energy predominantly from offshore renewable sources due to spatial constraints onshore and the higher efficiency of energy extraction offshore [3]. During the years 1995 to 2025, Europe has seen substantial development in its offshore wind industry, with numerous commercial bottom-founded wind projects already established [4]. Additionally, devices such as wave energy converters (WECs), floating photovoltaic (FPV) platforms, and floating offshore wind turbines (FOWTs) have been gaining increased interest. These structures rely on precise knowledge of fluid-structure interactions to optimize energy extraction while maintaining structural safety under harsh ocean conditions.

In addition to floating platforms for renewable energy, there is a rising interest in floating aquaculture, coastal infrastructure, and floating urban environments [5]. Moreover, a promising concept known as multi-purpose platforms (MPP) is gaining traction in ocean engineering. Such platforms cater to the needs of multiple offshore industries; for example, energy harvesting from both wind and waves, as well as aquaculture and wind energy harvesting [6]. The advantages of MPPs include modularity, reduced costs, and enhanced system integration. Overall, there is an increasing call for research and innovation in the domain of floating structures [7, 8], as demonstrated by several European projects, such as MERMAID, MUSES [9], and Space@Sea [10]. Examples of such floating platforms are depicted in Figure 1.1; specifically, on the left, the wave basin test conducted for the Space@Sea project [11], and on the right, a concept of a wind-solar-aquaculture system [12].

1



(a) *Source:* Otto and Hüsken [11]



(b) *Source:* Zheng et al. [12]

Figure 1.1: Examples of floating offshore structures, on the left the Space@Sea project wave basin test conducted at MARIN [11]. On the right, a concept of a Wind–Solar–Aquaculture System [12].

Nevertheless, several challenges persist. The ocean environment is inherently harsh and variable, characterized by extreme waves, strong currents, and multi-scale interactions. Nonlinear fluid-structure interactions, such as wave slamming and ringing, complicate modeling [13], as does operation near resonance, where even small nonlinearities can have significant consequences [14]. Moreover, translating laboratory and computational findings to real-world scenarios still presents several challenges, i.e., long-term statistics and extended simulation times, realistic treatment of absorption and reflection on possibly partially submerged shorelines, consideration of basin-scale effects, and precise characterization of wave breaking. Addressing these challenges requires continued advancements in computational modeling, experimental techniques, and integrated design approaches [15]. While experimental testing in wave basins and towing tanks has traditionally served as a cornerstone of hydrodynamic research, such approaches are costly and often limited in scale and scope [16]. Especially when there is an interest in farm-scale hydrodynamics, e.g., for arrays of WECs [17]. As a result, there is a growing reliance on computational tools, which provide a more flexible and cost-effective means of investigating and optimizing marine systems under a wide range of operating conditions. [18, 19]

The current standard in the design of floating structures is the use of numerical tools that tend to be case-specific implementations, which rely on calibration using model tests and the experience of the operator for the trade-off between computational cost and the neglect of physical phenomena [20]. For FOWTs, see the work by Yu et al. [21] and Thomsen et al. [22] for comparative studies on the hydrodynamic modeling approaches for the UMaine VolturnUS-S semi-submersible platform [23] and the TetraSpar Demo platform [24] respectively. The work by Nair et al. [25] reviews hydrodynamic modeling approaches for FPVs. Leary et al. [26] conducted a comparative study on the hydrodynamic modeling of WECs. Numerical modeling for offshore hydrodynamics now involves diverse techniques that balance accuracy, efficiency, and robustness. Engineers use multi-fidelity toolchains, selecting from linear models and nonlinear strategies while neglecting the physical properties of the fluid based on the design phase and specific phenomena of interest [19].

Physical models

When modeling the physics, the following hierarchy of model assumptions is generally applied. At the lowest level are the linear potential-flow (LPF) models, which are the industry benchmark for initial design and system-level evaluation of floating structures [21, 22]. LPF models are applied in both frequency domain and time domain to efficiently and reliably predict radiation, diffraction, added mass, and added damping phenomena. These techniques are thoroughly integrated into coupled aero-hydro-servo-elastic simulation frameworks and are crucial for both preliminary design and control analyses [27, 28]. Nevertheless, their fundamental constraint is linearization of the free-surface boundary conditions and wave kinematics, which restricts their capability to accurately forecast nonlinear free-surface dynamics, such as wave breaking, slamming, or green water deck events [29]. It is important to note that this kinematic linearization is distinct from the linearization of hydrodynamic loads: viscous drag, which is inherently quadratic in velocity, can still be approximated within a linear wave kinematics framework, for instance through the drag term of the Morison equation [29]. However, because LPF models are grounded in inviscid potential flow, viscous drag is absent from the formulation entirely and must be introduced through semi-empirical corrections rather than emerging naturally from the governing equations.

To overcome some of these limitations, fully nonlinear potential-flow (FNPF) methods have been developed [30]. By retaining the irrotational-flow assumption while solving nonlinear free-surface boundary conditions, FNPF models can capture large-amplitude waves and strongly nonlinear effects [31] at lower computational costs than solvers that include viscous effects [32].

At the high-fidelity end of the spectrum are the Navier-Stokes equations (NSE), which include rotational flow, viscous effects, and compressibility. Depending on the computational requirements, the turbulence can be fully resolved, also known as Direct Numerical Simulation (DNS), or turbulence models can be applied, i.e., Reynolds-Averaged Navier-Stokes (RANS) [33], large-eddy (LES) [34], or detached-eddy (DES) [35] approaches. These turbulence models can provide detailed insight into flow separation, vortex shedding, and impact phenomena. The models are capable of predicting slamming loads, run-up, and spray, which cannot be accurately predicted by potential flow formulations. The computational expense remains a primary bottleneck, often rendering long-duration simulations of realistic sea states impractical [36].

Another approach is the shallow water equations (SWE), for which the main assumption is that the horizontal length scale is much larger than the vertical length scale. The SWE are derived from the NSE and depth integrated, thereby reducing the problem size by one dimension. In the classical derivation of the SWE, the fluid is assumed to be inviscid, although extensions with viscosity are possible [37]. This model occupies a middle ground: it is more general than linear potential flow, capturing nonlinear, large-amplitude, long-wave dynamics. Yet, it is simpler than the NSE, trading vertical detail and possibly viscous effects for computational efficiency [38]. Expanding on the SWE, we can also consider the Boussinesq equations [39]; these account for weakly dispersive and nonlinear wave effects in finite water depth. These models are widely used to model coastal wave propagation, shoaling, and wave–structure interactions where the wavelength is not extremely large compared to water depth [40].

1

Recognizing the complementary strengths of these models, hybrid approaches have emerged that couple potential-flow solvers in the far field whilst resolving the Navier-Stokes equations around the body. These methods achieve a compromise between efficiency and accuracy, enabling the study of localized viscous effects in realistic sea states [41, 42]. In Figure 1.2, an overview of the varying levels of modeling fidelity is displayed versus approximate computational cost, accompanied by key physical properties.

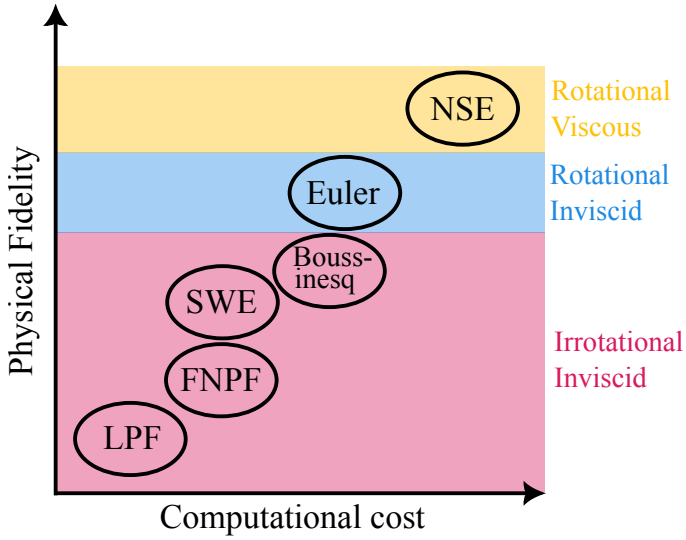


Figure 1.2: Conceptual representation of varying level of modeling fidelity versus computational cost.

Discretization methods

Apart from adjusting the physical fidelity or governing equations, it is imperative to consider that a compromise in the discretization method of the model will also impact the cost and accuracy. Broadly speaking, we can consider the following discretization categories: traditional grid-based (Eulerian), particle-based or mesh-free (Lagrangian), boundary-focused, and hybrid or higher-order methods.

Traditional grid-based: Finite difference methods (FDM) [43] approximate spatial and temporal derivatives on structured grids and remain attractive for canonical problems with simple domains, such as regular wave propagation or idealized wave tanks. Finite volume methods (FVM) [44], which enforce integral conservation laws over control volumes, constitute the foundation of most Computational Fluid Dynamics (CFD) solvers used to predict free-surface flows, violent slamming, and multiphase phenomena. Finite element methods (FEM) [45] employ variational formulations and unstructured meshes, enabling high-order accuracy and local adaptivity. FEM is widely adopted for complex geometries and for coupled fluid-structure interaction analyses.

Particle-based or mesh-free: Violent free-surface motions, wave breaking, and large structural displacements often motivate the use of Lagrangian particle formulations. Prominent examples include smoothed particle hydrodynamics (SPH) [46] and the moving particle

semi-implicit (MPS) [47] method, which avoid mesh distortion and are well suited to green-water loading and sloshing phenomena. Their natural handling of breaking and splashing is appealing, but high computational costs and numerical dissipation requirements for stability limit their use to targeted scenarios [48, 49].

Boundary-focused: For problems governed by the Laplace equation, boundary element methods (BEM) [50] reduce the dimensionality of the domain by confining the discretization to the wetted boundaries. Panel methods, a specialized form of BEM, have been a mainstay of linear and weakly nonlinear seakeeping analysis. Recent accelerations, such as the fast multipole BEM, broaden their applicability to large three-dimensional domains [51].

Hybrid and high-order: these approaches continue to expand the modeling toolbox. The harmonic polynomial cell (HPC) [52] method represents the potential field in overlapping cells using harmonic polynomial bases, ensuring local satisfaction of the Laplace equation and offering high accuracy for transient wave-structure interaction. Variants of FEM, such as the spectral element method (SEM) [53], extend FEM using higher-order basis functions and enhanced conservation properties.

Offshore Simulation Tools

Having outlined the hierarchy of physical models and the range of discretization approaches available, it is instructive to examine how these choices manifest in the tools that currently dominate offshore engineering practice. Table 1.1 provides a structured overview of widely used simulation tools, organized by physical fidelity and discretization method. As the table illustrates, the discretization landscape is heavily concentrated around a small set of fitted methods: finite volume methods (FVM) for high-fidelity Navier–Stokes solvers, boundary element methods (BEM) for potential-flow solvers, and finite difference methods (FDM) for mid-fidelity and research-oriented tools. These methods share a common requirement: the computational mesh must conform to the geometry of the domain boundaries and, in the case of moving structures, must be updated or regenerated as the geometry evolves. For offshore problems involving large-amplitude motions, complex floating body geometries, or multiple interacting structures, this mesh conformity requirement introduces significant practical overhead and can become a bottleneck in simulation turnaround time. The mainstream tools, including OpenFOAM-based frameworks [54–56], STAR-CCM+ [57], ANSYS Fluent and Aqwa [58], DualSPHysics [59], OrcaFlex [60], COMSOL [61], and MIKE 21/3 [62], are supported by extensive validation studies and active user communities across both academia and industry. Several more research-oriented solvers (Reef3D [63], OceanWave3D [64], FUNWAVE [65], Basilisk [66], NEMOH [67], SWASH [68], HOS-NWT [69]) are primarily developed within single groups or for specialized investigations, and collectively reflect the breadth of numerical approaches that have been explored. The concentration of body-fitted discretization methods across this landscape raises a practical question: what does this requirement for mesh conformity cost in the context of offshore engineering design workflows? This is examined in the following section.

From Design to Analysis

The majority of tools in Table 1.1 rely on discretization methods that require the computational mesh to conform to the boundaries of the domain and the geometry of any structures

Table 1.1: Overview of offshore simulation tools, their model fidelity, and discretization methods

Tool	Model Fidelity	Discretization Method
OpenFOAM-based [54–56]	NSE (RANS, LES, DNS)	FVM
STAR-CCM+ [57]	NSE (RANS, LES)	FVM
ANSYS Fluent/Aqwa [58]	NSE (RANS, LES, DES)	FVM / BEM
DualSPHysics [59]	NSE	SPH
Basilisk [66]	NSE	FVM
COMSOL [61]	NSE, FNPF, LPF	FEM
Reef3D [63]	NSE, FNPF, SWE	FDM
OceanWave3D [64]	FNPF	FDM / SEM
HOS-NWT [69]	FNPF	Pseudo-spectral
OrcaFlex [60]	FNPF, LPF	BEM / FDM
NEMOH [67]	LPF	BEM
FUNWAVE [65]	Boussinesq	FDM
SWASH [68]	SWE	FDM
MIKE 21/3 [62]	SWE	FVM

present. For these body-fitted methods, mesh quality has a direct and well-documented impact on both solution accuracy and solver efficiency [70]. Two aspects of mesh quality are relevant here. The first is solution quality: the degree to which the mesh enables a numerical PDE simulation to be executed efficiently while maintaining the integrity of the underlying physics and meeting the accuracy requirements of the problem [70]. The second is the á priori geometric quality of the elements: their shape, aspect ratio, and Jacobian ratio, which together determine the interpolation and discretization errors introduced by the mesh independently of the solution. Poor quality elements, such as those with small internal angles, high aspect ratios, or near-zero Jacobian ratios, introduce large interpolation errors and ill-conditioning that degrade both accuracy and solver performance [71].

When designing and evaluating new concepts such as novel floating structures, development proceeds through an iterative loop between Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE). In the CAD phase, a geometrical representation is produced; in the CAE phase, this geometry is evaluated using a numerical wave tank or equivalent solver. For fitted discretization methods, a conforming mesh must be constructed for each design iteration, creating a coupling between geometric change and computational overhead.

For high-fidelity Navier–Stokes solvers, this coupling is acute. Volumetric, boundary-conforming meshes with adequate resolution near walls, free surfaces, and wake regions are expensive to generate for complex geometries; geometry preprocessing and grid generation have been identified as significant bottlenecks that pace CFD workflows, particularly for novel configurations [72, 73]. This challenge is referred to hereafter as the *meshing bottleneck*. For potential-flow BEM solvers, the discretization requirement is fundamentally different: only a surface panelization of the body is needed rather than a volumetric mesh of the surrounding fluid domain [29, 74]. For typical offshore geometries, this surface discretization is a comparatively straightforward preprocessing step, as reflected by the

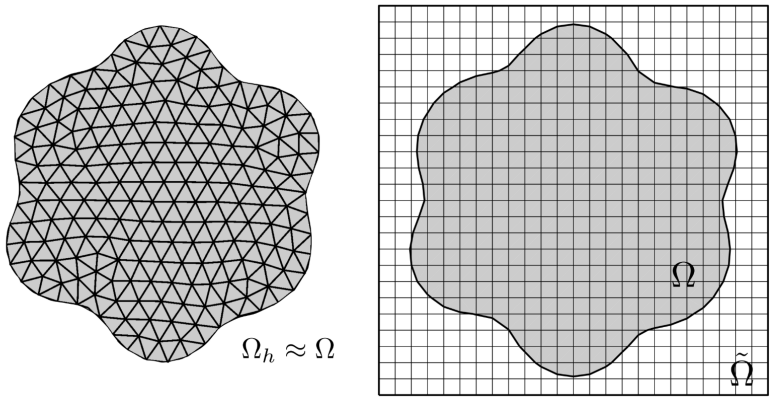


Figure 1.3: Body-fitted grid of a 2D popcorn shape (left) and an unfitted grid of the same geometry (right). Image courtesy: Eric Neiva (See Gridap tutorial 20: Poisson on unfitted meshes).

availability of lightweight dedicated tools developed specifically for this purpose within the hydrodynamics community [75]. The meshing bottleneck, in the strict sense applicable to volumetric CFD discretizations, is therefore not the primary motivation for exploring unfitted methods at the potential-flow fidelity level.

The motivation shifts, however, when the problem involves moving or deforming boundaries. Fully nonlinear potential-flow formulations track the free surface as a moving boundary, and fluid–structure interaction problems involving large body motions impose remeshing or mesh deformation requirements that grow increasingly costly as geometric complexity increases. It is precisely in this regime, where moving boundaries, large-amplitude free-surface dynamics, and complex evolving geometries, that unfitted methods become most attractive. By decoupling the computational mesh from the physical geometry, these methods handle moving boundaries without remeshing, represent complex boundaries implicitly, and are naturally suited to the class of problems that fitted methods handle least gracefully.

Conceptually, the distinction between fitted and unfitted approaches is illustrated in Figure 1.3. On the left, a body-fitted grid of a 2D popcorn shape is shown, in which all elements are oriented to align with the domain boundary. On the right, the unfitted approach allows the computational grid and the geometric boundary to be non-aligned; the geometry is embedded within a background grid, and boundary conditions are enforced either weakly or through additional constraints [76–78].

Beyond their suitability for moving-boundary problems, unfitted methods offer additional advantages: they leverage structured background grids that support high-order approximation spaces and efficient solvers, and they eliminate the geometric approximation error introduced when curved boundaries are represented by low-order polynomial elements [79]. Challenges remain, including the accurate treatment of cut cells, stable enforcement of boundary conditions, and the conditioning of the resulting linear systems [80], and these continue to drive active research in the field.

The present work focuses on the potential-flow setting as a necessary and tractable

starting point. Rather than immediately targeting the full moving-boundary or fully nonlinear regime, this work establishes unfitted discretizations within the linear potential-flow framework, examining their accuracy, stability, and practical behaviour across a range of problem types. This constitutes a foundational step toward the longer-term goal of applying unfitted methods to fully nonlinear potential-flow and fluid–structure interaction problems in offshore engineering, where their advantages over fitted approaches are most pronounced.

In the offshore engineering domain, the use of unfitted methods has so far been limited. The OceanWave3D package [64] includes capabilities for wave–structure interaction with nonlinear waves using an immersed boundary method in 2D [81, 82]. An immersed boundary approach has been applied to nonlinear free-surface waves over non-uniform seabed geometry using a high-order finite difference scheme [83]. The HPC method has been combined with an immersed boundary formulation for wave–structure interaction in 2D using adaptive quadtrees [84] and extended to 3D using adaptive octrees [85]. The extended finite element method has been applied to estimate added mass and damping coefficients and second-order mean wave loads for simple 2D geometries [86]. More recently, an immersed boundary method has been implemented within REEF3D::CFD to model wave–structure interaction for a floating offshore wind turbine [87]. Collectively, these contributions demonstrate that unfitted methods are viable in the offshore context, while also confirming that systematic development and validation across the relevant range of physical models and problem types remains an open research challenge.

The preceding sections have outlined the societal need for reliable offshore wave simulation tools, surveyed existing numerical wave tank software, and identified the moving-boundary and fully nonlinear problem regime as the setting in which unfitted methods offer the greatest practical advantage over fitted alternatives. Finite element methods are particularly well-suited as the underlying framework for this development: they accommodate complex geometries, support unstructured background grids, and provide well-developed formulations for free-surface and structural coupling that are essential for realistic large-scale offshore problems.

Research Objectives

Building on the aforementioned insights, the objective of this dissertation is to:

> **Thesis objective:** Develop a numerical wave tank based on unfitted finite element methods to efficiently simulate large-scale offshore wave–structure interactions while eliminating the meshing bottleneck.

To achieve this objective, a number of sub-objectives are defined as follows:

- **i.** Formulate a mathematical model for wave–structure interaction suitable for an unfitted finite element framework (chapter 2);
- **ii.** identify different types of unfitted finite element methods suitable for a numerical wave tank (chapter 3);
- **iii.** develop and implement an unfitted finite element discretization capable of handling complex offshore geometries without mesh conformity (chapter 3);

iv. compare performance of the unfitted finite element methods (chapter 4);

v. validate the methodologies through benchmark case studies (chapter 5 and 6).

Thesis outline

In chapter 2, the physical problem is reduced to a set of governing equations for both the fluid and the structure. Specifically, a linear potential flow formulation for free surface flows is coupled to a rigid structure in both time domain and frequency domain representations. In preparation for the next chapter, the governing equations are rewritten into an equivalent weak formulation. In chapter 3, the finite element method is briefly introduced, and a historical literature review of immersed boundary, embedded, and unfitted methods is provided. An argument is constructed to select certain unfitted finite element methods, and a common terminology is introduced across the unfitted methods. Then, an in depth discussion of the selected unfitted finite element methods is presented, and the discrete weak formulations are derived. In chapter 4, the performance benchmark between unfitted methods is conducted. Four test cases consisting of three geometries using both implicit and explicit geometrical representations are conducted. The accuracy and condition numbers for each test case are compared. For the first test case, the computational performance is compared, and the chapter is concluded with a comparison of implementation complexity across unfitted methods. In chapter 5, four time domain case studies are conducted. Starting with a simple geometry in 2D, followed by a simple geometry in 3D, then a realistic geometry in 3D, and lastly a proof of concept multi-body study with realistic geometries in 3D. Results are presented, and bottlenecks for large-scale simulations in the numerical wave tank are addressed and discussed. In chapter 6, the frequency domain case studies are described, starting with simple geometries in 2D and extending the method for realistic geometries in 3D. The results are presented and validated. In chapter 7, conclusions and future recommendations are made.

Reader's guide

At the start of each chapter, there is a one-page introduction that briefly states the topic, its relevance within this work, and defines the novel contributions that this work brings to the wider scientific community. Afterward, a formal outline of the chapter's contents is included. This work is not a linear build-up of topics; for this reason, an overview of the connectivity between chapters is depicted in Figure 1.4.

1

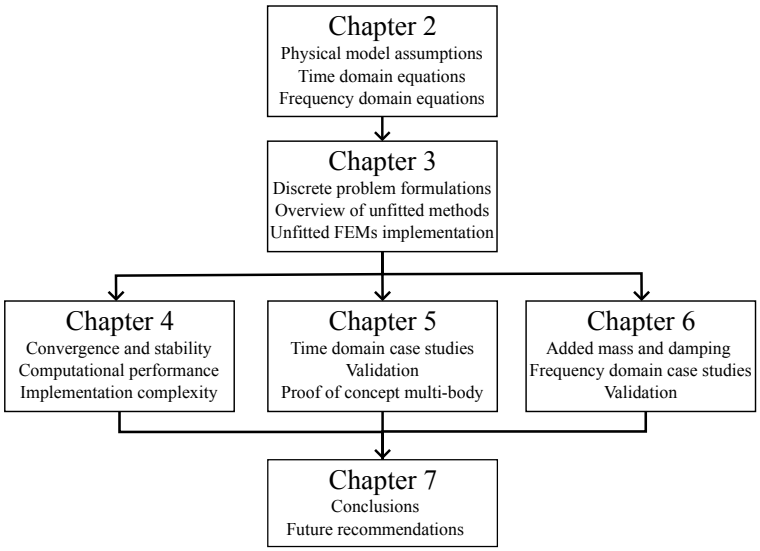


Figure 1.4: Overview of the connectivity between and topics of the chapters in this dissertation.

2

Problem Setting and Governing Equations

Offshore structures are continually exposed to harsh environments, including waves, currents, and wind. Understanding the fundamental physical principles governing fluid motion and its interaction with structures is essential for the safe and efficient design of offshore systems. Within this work, we are only concerning ourselves with the interaction between waves and offshore structures. The theory introduced in this chapter is not novel, but the purpose is rather to set the stage by introducing definitions and modeling assumptions, as well as consolidating and formalizing the governing equations that serve as the basis for the numerical methods and analyses developed in subsequent chapters. More explicitly, this chapter relates the physical reality of the interaction between ocean waves and floating structures to a mathematical formulation for wave–structure interaction problems. It begins by defining the problem domain and discussing the assumptions that determine the level of physical fidelity considered, leading to a linear potential flow model for free surface flows in reduced notation. The dynamics of a rigid body interacting with the surrounding fluid are introduced, and the coupling between the two systems is set in a strong formulation. Both the time and frequency domain representations of the governing equations are presented and their corresponding continuous weak formulations are derived.

2.1 Domain Definition

The general problem description considered in this work is depicted in Figure 2.1. The interior of the domain Ω comprises the entire fluid region bounded by

- the seabed, denoted by Γ_{sb},
- the body surface of the floating structure, denoted by Γ,
- the free surface, denoted by Γ_f, and
- an open or radiation boundary, denoted by Γ_w, at large distances to represent outgoing waves and prevent reflections [1, 29].

We consider the origin, in 2D O_{xz} or in 3D O_{xyz}, of the domain to be located at the center of the structure at the free surface level in the vertical direction, unless specified otherwise. When considering a domain in 3D, the free surface boundary Γ_f coincides with the O_{xy}-plane.

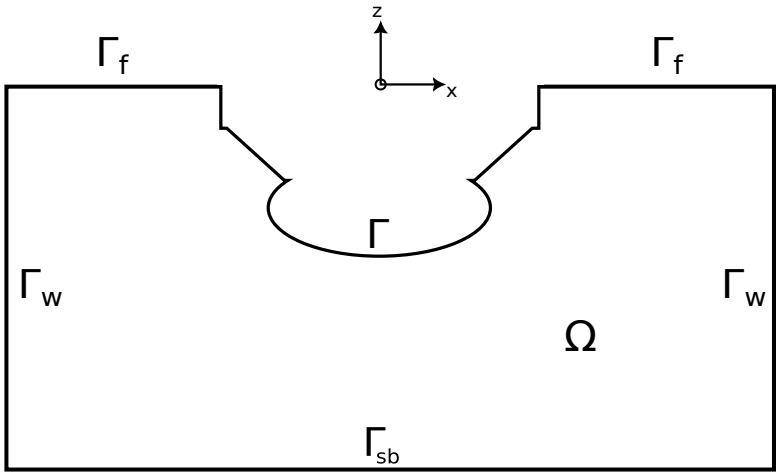


Figure 2.1: General description of the domain with interior Ω and boundaries Γ, Γ_f, Γ_w and Γ_{sb}.

2.2 Hydrodynamic Model

As discussed in the introduction, the highest fidelity model to represent a fluid can be derived by conservation laws. Considering waves in the ocean, our first assumption is that the variation of water density is insignificant [88].

Assumption 1 *Incompressibility of the fluid.*

Therefore, we can represent the fluid using the incompressible Navier-Stokes equations. This set of equations is given in Equation 2.1 and consists of the conservation of mass and the conservation of momentum.

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \qquad \text{in} \quad \Omega, \\ \rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) &= -\nabla p + \mu\nabla^2\mathbf{u} + \rho\mathbf{g} \qquad \text{in} \quad \Omega, \end{aligned} \tag{2.1}$$

where $\mathbf{u}$ is the fluid velocity vector, ρ is the fluid density, p is the pressure, μ the dynamic viscosity, and $\mathbf{g}$ the gravitational acceleration.

Viscous effects are confined to boundary layers of thickness, typically of the order of $\mathcal{O}(10)$ cm for wave periods of several seconds [88]. Therefore, the second assumption is introduced, stating that the flow of the fluid is inviscid.

Assumption 2 *The fluid flow is inviscid.*

Consequently, the incompressible Navier–Stokes equations reduce to the Euler form in the fluid interior:

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \qquad \text{in} \quad \Omega, \\ \rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) &= -\nabla p + \rho\mathbf{g} \qquad \text{in} \quad \Omega. \end{aligned} \tag{2.2}$$

Finally, the third assumption is that the flow is irrotational, i.e. $\nabla \times \mathbf{u} = \mathbf{0}$, as the fluid is inviscid and there is no mechanism to generate or sustain vorticity in the wave interior [1, 88].

Assumption 3 *The fluid flow is irrotational.*

These three assumptions allow us to rewrite the fluid velocity vector as $\mathbf{u} = \nabla\phi$, where ϕ is the scalar velocity potential.

The conservation of mass is then rewritten into Laplace's equation, and the conservation of momentum leads to the unsteady Bernoulli equation [1, 29]:

$$\begin{aligned} \nabla^2\phi &= 0 \qquad \text{in} \quad \Omega, \\ \frac{\partial \phi}{\partial t} + \frac{1}{2}|\nabla\phi|^2 + \frac{p}{\rho} + gz &= C(t) \qquad \text{on} \quad \Gamma_\text{f}, \end{aligned} \tag{2.3}$$

where z is the vertical coordinate and $C(t)$ is some arbitrary function of time. For more expansive derivations of the potential flow formulation, the reader is referred to the textbooks by Journée and Massie [1], Newman [29], and Mei et al. [88].

To close the problem, boundary conditions must be prescribed on each part of the domain boundary $\partial\Omega = \Gamma_\text{f} \cup \Gamma_\text{w} \cup \Gamma_\text{sb} \cup \Gamma$. These are introduced in the time-domain formulation in subsection 2.4.1, where the governing equations are linearized for small-amplitude free-surface motions.

2.3 Structural Model

In this work, we assume that the floating structure behaves like a rigid body, i.e. we neglect elastic deformation of the structure.

Assumption 4 *The structure is rigid.*

Lastly, it is assumed that there is no structural damping.

Assumption 5 *There is no structural damping within the structure.*

Under these assumptions, the equations of motion of the body follow from Newton's second law:

$$\boldsymbol{M}\frac{\partial^2 \mathbf{u}}{\partial t^2} = \boldsymbol{f}_{ext}. \tag{2.4}$$

Here, $\boldsymbol{M}$ is the mass matrix, $\mathbf{u}$ is the generalized displacement vector, and $\boldsymbol{f}_{ext}$ is the generalized external forces vector. Throughout this work, we mention the displacement vector and external forces vector, which refer to the generalized displacement vector and generalized external forces vector respectively. The directions of motion of a single rigid structure are represented by three physical degrees of freedom in 2D and by six physical degrees of freedom in 3D. The dimensions of the aforementioned mass matrix, displacement vector, and external forces vector are equal to the number of physical degrees of freedom. In this study, external forces $\boldsymbol{f}_{ext}$ follow from the integration of pressure on the wet surface of the body; no additional forces are considered. In particular, gravitational loading on the structure is not included as a separate body force in Equation 2.4. This is justified by assuming that the structure is initially in hydrostatic equilibrium, so that the gravitational force and the hydrostatic buoyancy force cancel exactly in the equilibrium state. Perturbations about this equilibrium produce a net hydrostatic restoring force that enters the equations of motion through the pressure integral in Equation 2.5. Gravity therefore enters the coupled system implicitly through the fluid pressure rather than as an explicit structural body force. These governing equations are defined solely on the boundary of the object, which is denoted by Γ, explicitly without a subscript.

$$\boldsymbol{f}_{ext} = -\int_{\Gamma} p\,\boldsymbol{n}_{\Gamma,j}\,d\Gamma. \tag{2.5}$$

Here $\boldsymbol{n}_{\Gamma,j}$ is a vector that represents both the translational and rotational normal vectors. Meaning in 2D it is defined as $\boldsymbol{n}_{\Gamma,j} = \{\boldsymbol{n}_\Gamma, \boldsymbol{n}_\Gamma, \mathbf{r}\times\boldsymbol{n}_\Gamma\}$ for physical degrees of freedom $j = 1,2,3$ corresponding to surge, heave, and pitch respectively. The rotational arm is defined as $\mathbf{r} = \mathbf{x} - \mathbf{x}_0$, where $\mathbf{x}$ is a coordinate on the boundary and $\mathbf{x}_0$ is taken as the point of hydrostatic equilibrium. In 3D this notation is extended to $\boldsymbol{n}_{\Gamma,j} = \{\boldsymbol{n}_\Gamma, \boldsymbol{n}_\Gamma, \boldsymbol{n}_\Gamma, \mathbf{r}\times\boldsymbol{n}_\Gamma, \mathbf{r}\times\boldsymbol{n}_\Gamma, \mathbf{r}\times\boldsymbol{n}_\Gamma\}$ for physical degrees of freedom $j = 1,2,3,4,5,6$ corresponding to surge, sway, heave, roll, pitch, and yaw. An overview of the six physical degrees of freedom is depicted in Figure 2.2.

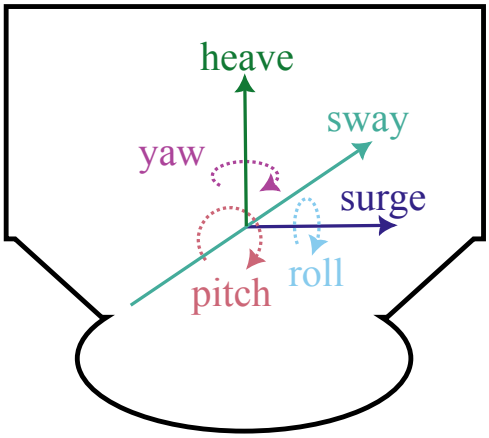


Figure 2.2: Directions of motion for the six physical degrees of freedom of the rigid structure.

2.4 Time Domain

To describe the evolution of the wave–structure system over time, the governing equations are first formulated in the time domain. In this setting, the velocity potential ϕ satisfies the Laplace equation within the fluid domain, while the free surface and body boundaries impose time dependent kinematic and dynamic conditions. This formulation captures the instantaneous radiation and diffraction of waves generated by arbitrary motions or incident disturbances and serves as the starting point for deriving the corresponding weak form and subsequent numerical discretization.

2.4.1 Strong Formulation

Starting with the model representation for the fluid as defined in Equation 2.3, we introduce the strong formulation of the hydrodynamic governing equations as stated in Equation 2.6. The body boundary condition is intentionally omitted from Equation 2.6 here, as it is introduced and discussed in detail below in the context of the fluid-structure coupling.

$$\Delta\phi = 0 \quad \text{in} \quad \Omega, \tag{2.6a}$$

$$\frac{\partial \eta}{\partial t} + \tilde{\nabla}\phi \cdot \tilde{\nabla}\eta - \nabla\phi \cdot \boldsymbol{n}_z = 0 \quad \text{on} \quad \Gamma_{\mathrm{f}}, \tag{2.6b}$$

$$\frac{\partial \phi}{\partial t} + \frac{1}{2}|\nabla\phi|^2 + \frac{p_{\mathrm{atm}}}{\rho} + g\eta = 0 \quad \text{on} \quad \Gamma_{\mathrm{f}}, \tag{2.6c}$$

$$\nabla\phi \cdot \boldsymbol{n}_z = 0 \quad \text{on} \quad \Gamma_{\mathrm{sb}}, \tag{2.6d}$$

$$\nabla\phi \cdot \boldsymbol{n}_{\mathrm{w}} = V_{\mathrm{w}} \quad \text{on} \quad \Gamma_{\mathrm{w}}. \tag{2.6e}$$

Here, we introduced the notation $\Delta := \nabla^2$, η is the scalar free surface elevation, the operator $\tilde{\nabla} = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$ is the gradient operator in the horizontal plane, $\boldsymbol{n}$ denotes the normal vector to the respective surfaces or in the vertical z-direction, the atmospheric $p_{\mathrm{atm}} = 0$, which is

set to zero [1, 88], and some predefined fluid velocity V_w on the surrounding boundaries. Currents are not accounted for in this formulation.

We solve the Laplace equation in the interior Ω of the domain in Equation 2.6d. The free surface boundary Γ_f is governed by a dynamic boundary condition, Equation 2.6c, which is obtained using the unsteady Bernoulli equation and a kinematic condition, Equation 2.6b, such that the velocity of the free surface elevation is bounded by the velocity potential. Intuitively, these conditions can be viewed as pressure continuity with atmospheric pressure and no flow across the free surface. On the seabed Γ_{sb}, we require a no-leak boundary condition, see Equation 2.6d, meaning the velocity of the water particles in the direction normal to the seabed is zero. Note that we only consider uniform flat seabed geometry in this work. On the vertical boundaries surrounding the domain Γ_w, we require either open boundary conditions, inflow and outflow conditions, or non-reflective wave absorption conditions, as depicted in Equation 2.6e. Note that in the case of non-reflecting boundary conditions, additional terms should be added, as noted in [89, 90].

Generally, considering potential flow for offshore engineering applications, it is common to consider only first-order effects [91]. Linearization is accomplished by neglecting the nonlinear terms from Equation 2.6. Furthermore, both conditions are transferred from the instantaneous free surface to the mean water level $z = 0$, since deviations of the free surface from its rest position are assumed small. After linearization, Equation 2.6b and Equation 2.6c are rewritten to:

$$\frac{\partial \eta}{\partial t} - \nabla\phi \cdot \boldsymbol{n}_z = 0 \quad \text{on} \quad \Gamma_f, \tag{2.7a}$$

$$\frac{\partial \phi}{\partial t} + g\eta = 0 \quad \text{on} \quad \Gamma_f, \tag{2.7b}$$

The so-called reduced formulation for linear potential flow can then be introduced [92] by rewriting equation (2.7b) and taking a partial time derivative on both sides, we obtain:

$$\frac{\partial \eta}{\partial t} = -\frac{1}{g}\frac{\partial^2 \phi}{\partial t^2} \qquad \text{on} \quad \Gamma_f. \tag{2.8}$$

Equation (2.8) is then substituted into the kinematic boundary condition (2.7a). This eliminates the unknown scalar free surface elevation η from the problem description, leaving us with Equation 2.9.

$$\Delta\phi = 0 \quad \text{in} \quad \Omega, \tag{2.9a}$$

$$\frac{\partial^2 \phi}{\partial t^2} + g\nabla\phi \cdot \boldsymbol{n}_z = 0 \quad \text{on} \quad \Gamma_f, \tag{2.9b}$$

$$\nabla\phi \cdot \boldsymbol{n}_z = 0 \quad \text{on} \quad \Gamma_{sb}, \tag{2.9c}$$

$$\nabla\phi \cdot \boldsymbol{n}_w = V_w \quad \text{on} \quad \Gamma_w. \tag{2.9d}$$

The rigid structure is then coupled to the fluid via kinematic and dynamic conditions. Similarly to the free surface boundary conditions, we consider a kinematic condition such that the velocity of the fluid at the structure boundary matches the velocity of the structure. Therefore, on the structural boundary Γ we set:

$$\nabla\phi \cdot \boldsymbol{n}_\Gamma = \frac{\partial \mathbf{u}}{\partial t} \cdot \boldsymbol{n}_{\Gamma,j} \quad \text{on} \quad \Gamma. \tag{2.10}$$

The imposition of the dynamic boundary condition is achieved by coupling the unsteady Bernoulli equation to Newton's second law. Starting with the Bernoulli equation from Equation 2.3, we linearize the equation by neglecting the nonlinear term and setting $C(t) = 0$, leaving us with:

$$\frac{\partial \phi}{\partial t} + \frac{p}{\rho} + gz = 0 \qquad \text{on} \quad \Gamma. \tag{2.11}$$

Additionally, the arbitrary unknown vertical displacement denoted by the z-coordinate in Equation 2.11 is set to the unknown displacement vector $\mathbf{u}$ of the structure in the vertical direction. For readability and future substitution, we also rewrite the equation such that we isolate p.

$$p = -\rho\left(\frac{\partial \phi}{\partial t} + g\mathbf{u} \cdot \boldsymbol{n}_z\right) \qquad \text{on} \quad \Gamma. \tag{2.12}$$

Newton's second law as defined in Equation 2.4 and Equation 2.5 is given as:

$$\boldsymbol{M}\frac{\partial^2 \mathbf{u}}{\partial t^2} = -\int_\Gamma p\,\boldsymbol{n}_{\Gamma,j}\, d\Gamma \qquad \text{on} \quad \Gamma. \tag{2.13}$$

Here, we have unknown displacement vector $\mathbf{u}$ and pressure p. The rewritten linearized unsteady Bernoulli equation, see Equation 2.12, is then substituted for the pressure p in Equation 2.13. This leaves us with the system of equations given in Equation 2.14, which consists of both the kinematic and dynamic constraints.

$$\nabla\phi \cdot \boldsymbol{n}_\Gamma - \frac{\partial \mathbf{u}}{\partial t} \cdot \boldsymbol{n}_{\Gamma,j} = 0 \quad \text{on} \quad \Gamma, \tag{2.14a}$$

$$\frac{\boldsymbol{M}}{\rho}\frac{\partial^2 \mathbf{u}}{\partial t^2} + \int_\Gamma \left(\frac{\partial \phi}{\partial t} + g\mathbf{u} \cdot \boldsymbol{n}_z\right) \boldsymbol{n}_{\Gamma,j}\, d\Gamma = \mathbf{0} \quad \text{on} \quad \Gamma. \tag{2.14b}$$

As this is a linear problem, it should be stated that the boundary Γ is known a-priori and it is independent of the displacement $\mathbf{u}$.

The fluid and the structure are coupled by merging Equation 2.9 and Equation 2.14, meaning we solve a single system of equations with two unknowns: the velocity potential ϕ and the displacement vector $\mathbf{u}$. The coupled system of equations is presented in Equation 2.15.

$$\Delta\phi = 0 \quad \text{in} \quad \Omega, \tag{2.15a}$$

$$\frac{\partial^2\phi}{\partial t^2} + g\nabla\phi\cdot\boldsymbol{n}_z = 0 \quad \text{on} \quad \Gamma_{\text{f}}, \tag{2.15b}$$

$$\nabla\phi\cdot\boldsymbol{n}_z = 0 \quad \text{on} \quad \Gamma_{\text{sb}}, \tag{2.15c}$$

$$\nabla\phi\cdot\boldsymbol{n}_{\text{w}} = V_{\text{w}} \quad \text{on} \quad \Gamma_{\text{w}}, \tag{2.15d}$$

$$\nabla\phi\cdot\boldsymbol{n}_\Gamma - \frac{\partial\mathbf{u}}{\partial t}\cdot\boldsymbol{n}_{\Gamma,j} = 0 \quad \text{on} \quad \Gamma, \tag{2.15e}$$

$$\frac{\boldsymbol{M}}{\rho}\frac{\partial^2\mathbf{u}}{\partial t^2} + \int_\Gamma\left(\frac{\partial\phi}{\partial t} + g\mathbf{u}\cdot\boldsymbol{n}_z\right)\boldsymbol{n}_{\Gamma,j}d\Gamma = \mathbf{0} \quad \text{on} \quad \Gamma. \tag{2.15f}$$

2.4.2 Weak Formulation

We now transform the strong form to a weak form such that we incorporate the boundary conditions in a natural manner and set up the coupled system of equations for the proposed numerical methods following in the next chapter. However, before deriving the weak form of the coupled system in Equation 2.15, let us define some notation that will be used hereafter. We abuse the notation of p, which in this subsection does not refer to pressure, but to the function space definition of Lebesgue spaces. These are denoted by $L^p(\Omega)$, $1 \leq p < \infty$, the spaces of functions such that their p-th power is absolutely integrable in Ω. For the case in which $p = 2$, we have a Hilbert space with a scalar product

$$(u,v)_\Omega \equiv (u,v) := \int_\Omega u(\mathbf{x})v(\mathbf{x})d\Omega \tag{2.16}$$

and induced norm $\|u\|_{L^2(\Omega)} \equiv \|u\| = (u,u)^{1/2}$. Abusing the notation, the same symbol as in Equation 2.16 will be used for the integral of the product of two functions, even if these are not in $L^2(\Omega)$, and both for scalar and vector fields. The space of functions whose distributional derivatives up to order m are in $L^2(\Omega)$ are denoted by $H^m(\Omega)$. We will focus on the case of $m = 1$, which is also a Hilbert space, $H^1(\Omega)$. Given a Banach space X, $L^p(0,T;X)$ is the space of time-dependent functions such that their X-norm is in $L^p(0,T)$.

The weak form of Equation 2.15 consists of finding $[\phi,\mathbf{u}] \in L^2(0,T;\mathcal{W})\times L^2(0,T;\mathcal{V})$ such that

$$a([\phi,\mathbf{u}],[w,\mathbf{v}]) = b([w,\mathbf{v}]) \qquad \forall[w,\mathbf{v}] \in \mathcal{W}\times\mathcal{V}, \tag{2.17}$$

With $\mathcal{W} := H^1(\Omega)$ and $\mathcal{V} := \mathbb{R}^{d_j}$ with d_j being the number of physical degrees of freedom. The bilinear and linear forms are given by

$$a([\phi,\mathbf{u}],[w,\mathbf{v}]) := (\nabla w,\nabla\phi)_\Omega + \left(w,\frac{1}{g}\frac{\partial^2\phi}{\partial t^2}\right)_{\Gamma_{\text{f}}} - \left(w,\frac{\partial\mathbf{u}}{\partial t}\cdot\boldsymbol{n}_{\Gamma,j}\right)_\Gamma + \left(\mathbf{v},\frac{\boldsymbol{M}}{\rho|\Gamma|}\frac{\partial^2\mathbf{u}}{\partial t^2} + \frac{\partial\phi}{\partial t}\boldsymbol{n}_{\Gamma,j} + g(\mathbf{u}\cdot\boldsymbol{n}_z)\boldsymbol{n}_{\Gamma,j}\right)_\Gamma, \tag{2.18}$$

and

$$b([w,\mathbf{v}]) := (w,V_{\text{w}})_{\Gamma_{\text{w}}}. \tag{2.19}$$

In the definitions of the bilinear form, $a([\phi,\mathbf{u}],[w,\mathbf{v}])$, and the linear form, $b([w,\mathbf{v}])$, described in Equation 2.18 and Equation 2.19 we have used the kinematic boundary conditions from Equation 2.15c - Equation 2.15e. The last term in the expression of Equation 2.18 enforces the dynamic interface condition from Equation 2.15f between fluid and solid, noting that the mass matrix is introduced inside the integral averaged with the boundary measure $|\Gamma| := \int_\Gamma d\Gamma$. It is important to highlight that the formulation presented in equation Equation 2.17 is described for structures with rigid body motions, but the same framework could be adopted for elastic structures [89, 93]. We also reiterate that in this work we use a monolithic coupling between fluid and structure, i.e. both fluid and solid degrees of freedom are solved in a unique coupled system of equations.

2.5 Frequency Domain

As we have already derived the coupled system of equations using linear potential flow theory, we can now rewrite the system of equations such that we can analyze it in the frequency domain.

That is, we assume the potential field and displacement can be decomposed in a Fourier series,

$$\phi(\mathbf{x},t) = \sum_{k=1}^{\infty} \hat{\phi}_k(\mathbf{x}) \exp(-i\omega_k t), \tag{2.20}$$
$$\mathbf{u}(t) = \sum_{k=1}^{\infty} \hat{\mathbf{u}}_k \exp(-i\omega_k t).$$

Where $\hat{\phi}_k$ and $\hat{\mathbf{u}}_k$ are complex-valued time-independent amplitudes associated with each frequency ω_k. Applying the superposition principle, we can solve independent linear problems for each frequency using the Fourier decompositions Equation 2.20. Thus, for a given term of the Fourier expansion with associated frequency ω_k, we can rewrite Equation 2.15 such that the coupled system in the frequency domain reads

$$\Delta\hat{\phi}_k = 0 \quad \text{in} \quad \Omega, \tag{2.21a}$$
$$-\omega_k^2\hat{\phi}_k + g\nabla\hat{\phi}_k\cdot\boldsymbol{n}_z = 0 \quad \text{on} \quad \Gamma_\text{f}, \tag{2.21b}$$
$$\nabla\hat{\phi}_k\cdot\boldsymbol{n}_z = 0 \quad \text{on} \quad \Gamma_\text{sb}, \tag{2.21c}$$
$$\nabla\hat{\phi}_k\cdot\boldsymbol{n}_\text{w} = V_\text{w}^k \quad \text{on} \quad \Gamma_\text{w}, \tag{2.21d}$$
$$\nabla\phi\cdot\boldsymbol{n}_\Gamma + i\omega_k\hat{\mathbf{u}}_k\cdot\boldsymbol{n}_{\Gamma,j} = 0 \quad \text{on} \quad \Gamma, \tag{2.21e}$$
$$-\omega_k^2\frac{\boldsymbol{M}}{\rho}\hat{\mathbf{u}}_k + \int_\Gamma\left[\left(-i\omega\hat{\phi}_k + g\hat{\mathbf{u}}_k\cdot\boldsymbol{n}_z\right)\boldsymbol{n}_{\Gamma,j}\right]d\Gamma = \mathbf{0} \quad \text{on} \quad \Gamma. \tag{2.21f}$$

We transform the strong form of the coupled system in Equation 2.21 using notation introduced in subsection 2.4.2.

The weak form of the coupled problem in free surface linear potential flow in Equation 2.21 reads: find $[\phi,\mathbf{u}] \in \mathcal{W}\times\mathcal{V}$ such that

$$a_\omega([\phi,\mathbf{u}],[w,\mathbf{v}]) = b_\omega([w,\mathbf{v}]) \qquad \forall[w,\mathbf{v}] \in \mathcal{W}\times\mathcal{V}, \tag{2.22}$$

where $\mathcal{W} := H^1(\Omega)$ and $\mathcal{V} := \mathbb{C}^{d_j}$. In this case, since we consider translational and rotational motion, d_j is equal to $d_j = 3$ in 2D and $d_j = 6$ in 3D. The bilinear and linear forms are defined as follows:

$$a_\omega([\phi,\mathbf{u}],[w,\mathbf{v}]) := (\nabla w, \nabla\phi)_\Omega - \frac{\omega^2}{g}(w,\phi)_{\Gamma_f} + i\omega\left(w, \mathbf{u}\cdot\boldsymbol{n}_{\Gamma,j}\right)_\Gamma + \\ -\omega^2\frac{\boldsymbol{M}_\rho}{|\Gamma|}(\mathbf{v},\mathbf{u})_\Gamma - i\omega\left(\mathbf{v}, \phi\boldsymbol{n}_{\Gamma,j}\right)_\Gamma + g\left(\mathbf{v}, (\boldsymbol{n}_z\cdot\mathbf{u})\,\boldsymbol{n}_{\Gamma,j}\right)_\Gamma, \tag{2.23}$$

and

$$b_\omega([w,\mathbf{v}]) := (w, V_\mathrm{w})_{\Gamma_\mathrm{w}}. \tag{2.24}$$

In Equation 2.23, $\boldsymbol{M}_\rho$ is the relative mass with respect to the fluid density, i.e. $\boldsymbol{M}_\rho := \frac{\boldsymbol{M}}{\rho}$. In the definitions of the bilinear form $a_\omega([\phi,\mathbf{u}],[w,\mathbf{v}])$ and the linear form $b_\omega([w,\mathbf{v}])$ in Equation 2.23 and Equation 2.24, the kinematic boundary conditions Equation 2.21c-Equation 2.21e are used. The last term in Equation 2.23 enforces the dynamic interface condition Equation 2.21f between the fluid and solid, where the mass term is averaged over the boundary measure $|\Gamma| := \int_\Gamma d\Gamma$.

2.6 Conclusions

To conclude this chapter, Table 2.1 provides a quick reference to all governing equations. It lists the strong and weak formulations for both the time and frequency domain problems together with their equation numbers for easy cross-referencing.

Table 2.1: Summary of governing equations and their locations within this chapter.

Problem type	Formulation	Reference
Time	Strong	Equation 2.15
Time	Weak	Equation 2.18 and Equation 2.19
Frequency	Strong	Equation 2.21
Frequency	Weak	Equation 2.23 and Equation 2.24

In this chapter, we introduced the problem setting for floating structures in the offshore environment. The domain and boundaries are defined, and the varying levels of modeling fidelity for fluids are discussed. Starting with the incompressible Navier-Stokes equations and then reducing the model fidelity to linear potential flow for free surface flows in reduced notation. We introduced the assumptions posed on both the fluid and structure and the corresponding governing equations. We showed how the governing equations are coupled and introduced weak formulations for the time and frequency domain problems considered in this work.

3

Unfitted Finite Element Methods for Floating Structures

From an application perspective, the methods discussed in this chapter address one of the challenges in marine and offshore engineering: efficiently simulating fluid–structure interaction for geometries that undergo large motions or are difficult to mesh. By enabling accurate and flexible numerical analysis of complex floating structures, these approaches contribute to the development and optimization of next-generation offshore systems, such as floating wind turbines and wave energy devices. The novelty presented in this chapter consists of a common description and implementation of four different unfitted finite element methods within a single computational framework. For each method, the formulation for two-way coupled linear potential flow for free surface flows in reduced notation with a rigid object is introduced in both time and frequency domain forms. Furthermore, for the SBM and WSBM, the derivation of the formulation is novel. As such, the methods outlined in this work do not consider large motions and are restricted to facilitating the mesh generation process. This chapter provides the theoretical and methodological foundation for the unfitted finite element methods employed for simulating floating structures. We begin with a concise introduction to the finite element method and the corresponding discrete formulations for body-fitted approaches. A historical overview of the development of unfitted finite element methods is presented, culminating in a synthesis of their main principles and common challenges. Based on this overview, we justify the selection of specific unfitted formulations considered in this dissertation and introduce the notation and common descriptors used to describe them consistently. The latter part of the chapter presents the four individual methods in detail, outlining their core ideas, distinguishing features, and underlying assumptions. While the unfitted methods and body-fitted formulations themselves follow from established literature, the synthesis provided here aims to create a unified framework that facilitates comparison, implementation, and evaluation in the context of simulations of the hydrodynamics of floating structures.

3.1 Introduction to Unfitted FEMs

This section starts with a brief introduction to the finite element method. Then, a historical overview of immersed boundary, embedded, or unfitted methods is presented, leading to common challenges.

3.1.1 The Finite Element Method

The preceding chapter presented the governing equations in their continuous weak form. For numerous engineering applications, these variational formulations prove to be unsolvable through analytical methods when confronted with complex geometries. The finite element method offers a systematic approach to approximating the weak solution within a finite-dimensional space [45, 94]. This section provides a brief introduction to the finite element method. For a more formal introduction to the finite element method, the reader is referred to [45, 94–96]. It also introduces the semi-discrete and discrete weak forms for the continuous weak forms from chapter 2.

Domain Partitioning and Mesh Requirements

In order to numerically approximate the weak solution, the physical domain Ω is divided into a finite set of non-overlapping elementary units referred to as elements; an element is denoted by K. Common choices for elements include triangles, tetrahedra, quadrilaterals, and hexahedra. In Figure 3.1 an illustration of such a mesh with triangles is depicted. The physical domain Ω is discretized by elements that approximately conform to the physical boundary Γ. The set of these elements is the mesh $\mathcal{T}_h = K_1, ..., K_m$, where the subscript h represents some characteristic element size, serving as the fundamental metric of resolution. In a similar fashion, the domain boundary is discretized by assigning the union of those edges of $\mathcal{T}_h$ that approximately coincide with Γ to Γ_h. For the finite element method to be effectively implemented, the mesh is required to adhere to several properties:

- **Conformity:** The union of the closures of all elements is precisely equivalent to Ω_h or $\Omega_h = \cup_{K \in \mathcal{T}_h} K$. The intersection of any two elements results in either an empty set, a shared vertex, a common edge, or a common face.
- **Element quality:** It is imperative that elements are neither excessively distorted nor elongated. Conditions related to shape regularity, such as maintaining bounded aspect ratios, are essential to guarantee the stability of the discrete problem and achieve optimal rates of convergence. A review on mesh and element quality is given in [71].
- **Mesh size:** The largest element diameter $h_e = \max_{K_i \in \mathcal{T}_h} h_{K_i}$ controls the discretization error; smaller h_e yields higher accuracy at increased computational cost.

Finite-Dimensional Function Spaces

The continuous weak formulation of our problem seeks a solution for the velocity potential ϕ in an infinite-dimensional Sobolev space $\mathcal{W}$. In order to compute this solution, we must introduce the finite-dimensional, also known as the discrete subspace $\mathcal{W}_h \subset \mathcal{W}$ defined on the discrete domain Ω_h. Similarly, for the rigid body displacements $\mathbf{u}$, we should identify

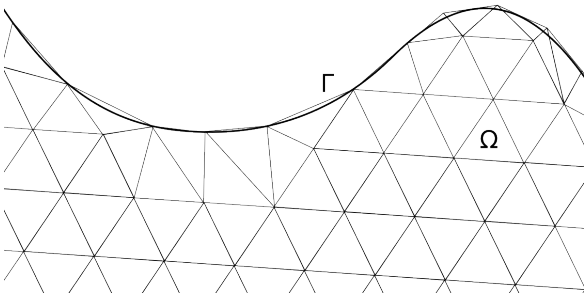


Figure 3.1: Reference conformal domain Ω with arbitrary shaped boundary Γ.

a finite-dimensional subspace. Here, we introduce $\mathcal{V}_h \subset \mathcal{V}$. Note that $\mathcal{V}_h$ is a vector of d_j physical degrees of freedom, which could be understood as a vector-valued constant FE space over any given domain; in that case, it is sufficient to have $\mathcal{V}_h$ defined over Γ_h, which is defined by the set of edges of $\mathcal{T}_h$ belonging to Γ.

A typical choice for $\mathcal{W}_h$ is the space of piecewise polynomial functions of degree p_e that are globally continuous:

$$\mathcal{W}_h := \left\{ w_h \in C^0(\Omega_h) \,:\, w_h|_K \in \mathbb{P}_{p_e}(K) \,\forall K \in \mathcal{T}_h \right\}. \tag{3.1}$$

Here $\mathbb{P}_{p_e}(K)$ denotes the set of polynomials of total degree p_e on each element K. The most common finite elements are the Lagrange family, where the polynomial degree p_e can be chosen according to the desired accuracy. For example:

- $p_e = 1$: linear elements with nodal basis functions that are nonzero only on elements sharing the node.
- $p_e = 2$: quadratic elements, which add mid-edge nodes and achieve higher-order accuracy.

In this work, we use order $p_e = 1$ and $p_e = 2$. Extensions to higher-order elements are possible in theory, yet not applied.

The global finite element basis denoted by $\{w_i\}_{i=1}^N$ has two important characteristics:

Continuity. Each w_i is continuous across inter-element boundaries, ensuring that $\mathcal{W}_h \subset H^1(\Omega)$ and that the discrete weak form is well posed.

Local support. Each basis function is nonzero only on a small patch of elements adjacent to its associated node. This locality yields sparse matrices and efficient assembly.

The discrete solution is then expressed as a linear combination

$$\phi_h(\boldsymbol{x}) = \sum_{i=1}^{N} \Phi_i \, w_i(\boldsymbol{x}), \tag{3.2}$$

where the coefficients Φ_i are the unknown nodal values to be determined by solving the global finite element system.

Boundary Conditions

Boundary conditions are incorporated either strongly or weakly. For Dirichlet, or essential, conditions, the standard approach is to enforce the prescribed nodal values directly, thereby restricting $\mathcal{W}_h$. Neumann, or natural, conditions appear as boundary integrals in the weak form from integration by parts and require no further modification. In the problems considered in this work, we generally enforce the kinematic boundary conditions via Neumann boundary conditions. The treatment of boundary conditions becomes more involved when working with unfitted methods. More advanced weak imposition techniques, e.g. Nitsche's method [97], can be particularly relevant when we later consider unfitted finite element methods.

Discrete Weak Formulation

Replacing the infinite-dimensional trial and test spaces in the weak formulations from chapter 2 with the finite-dimensional subspaces gives the semi-discrete and discrete weak forms for the time- and frequency-domain problems, respectively.

Time-domain semi-discrete weak form

For the time-domain problem, we introduce the semi-discrete, continuous in time and discrete in space, weak form as:

find $[\phi_h, \mathbf{u}_h] \in L^2(0,T;\mathcal{W}_h) \times L^2(0,T;\mathcal{V}_h)$ such that

$$a_h([\phi_h, \mathbf{u}_h],[w_h, \mathbf{v}_h]) = b_h([w_h, \mathbf{v}_h]) \qquad \forall [w_h, \mathbf{v}_h] \in \mathcal{W}_h \times \mathcal{V}_h, \tag{3.3}$$

with

$$\begin{aligned} a_h([\phi_h, \mathbf{u}_h],[w_h, \mathbf{v}_h]) :=& (\nabla w_h, \nabla \phi_h)_{\Omega_h} + \left(w_h, \frac{1}{g}\frac{\partial^2 \phi_h}{\partial t^2} \right)_{\Gamma_{\mathrm{f},h}} - \\ & \left(w_h, \frac{\partial \mathbf{u}_h}{\partial t} \cdot \boldsymbol{n}_{\Gamma,j} \right)_{\Gamma_h} + \left(\mathbf{v}_h, \frac{\boldsymbol{M}}{\rho |\Gamma_h|} \frac{\partial^2 \mathbf{u}_h}{\partial t^2} + \left(\frac{\partial \phi_h}{\partial t} + g(\mathbf{u}_h \cdot \boldsymbol{n}_z) \right) \boldsymbol{n}_{\Gamma,j} \right)_{\Gamma_h}, \end{aligned} \tag{3.4}$$

and

$$b_h([w_h, \mathbf{v}_h]) := (w_h, \boldsymbol{V}_{\mathrm{w}} \cdot \boldsymbol{n}_{\mathrm{w}})_{\Gamma_{\mathrm{w},h}} \tag{3.5}$$

In equations (3.4) and (3.5), $\Gamma_{f,h}$, and $\Gamma_{\mathrm{w},h}$ denote the set of facets of dimension $d_n - 1$, i.e. edges in 2D or faces in 3D, of Ω_h that belong to Γ_{f}, and Γ_{w}, respectively.

Frequency-domain discrete weak form

For the frequency-domain boundary-value problem, we introduce the discrete weak form as:

find $[\phi_h, \mathbf{u}_h] \in \mathcal{W}_h \times \mathcal{V}_h$ such that

$$a_{\omega,h}([\phi_h, \mathbf{u}_h],[w_h, \mathbf{v}_h]) = b_{\omega,h}([w_h, \mathbf{v}_h]) \qquad \forall [w_h, \mathbf{v}_h] \in \mathcal{W}_h \times \mathcal{V}_h \tag{3.6}$$

with

$$a_{\omega,h}([\phi_h,\mathbf{u}_h],[w_h,\mathbf{v}_h]) := (\nabla w_h,\nabla\phi_h)_{\Omega_h} - \frac{\omega^2}{g}(w_h,\phi_h)_{\Gamma_{\mathrm{f},h}} + i\omega(w_h,\mathbf{u}_h\cdot\boldsymbol{n}_\Gamma)_{\Gamma_h} + \\ -\omega^2\frac{\boldsymbol{M}_\rho}{|\Gamma_h|}(\mathbf{v}_h,\mathbf{u}_h)_{\Gamma_h} - i\omega\left(\mathbf{v}_h,\phi_h\boldsymbol{n}_{\Gamma,j}\right)_{\Gamma_h} + g\left(\mathbf{v}_h,(\boldsymbol{n}_z\cdot\mathbf{u}_h)\,\boldsymbol{n}_{\Gamma,j}\right)_{\Gamma_h} \tag{3.7}$$

and

$$b_{\omega,h}([w_h,\mathbf{v}_h]) := (w_h,\boldsymbol{V}_{\mathrm{w}}\cdot\boldsymbol{n}_{\mathrm{w}})_{\Gamma_{\mathrm{w},h}} \tag{3.8}$$

The domain and boundaries are discretized as before. Note that the test and trial spaces $\mathcal{W}_h$ and $\mathcal{V}_h$ for the frequency discrete weak form consist of complex-valued functions $\mathbb{C}$.

Element Matrices and Assembly

The bilinear and linear forms can be decomposed into element contributions, because basis functions have local support. For each element K we compute the bilinear and linear form using numerical integration, i.e. Gaussian quadrature rules. The global stiffness matrix $\mathbb{A}$ and global load vector $\mathbb{b}$ are obtained by assembly: summing all element contributions according to the global degrees of freedom numbering. This means that for each integral term in the weak forms, i.e. Equation 3.3 and Equation 3.6, we rewrite the integral over the discrete domain as the sum of integrals over each element using Equation 3.9 and Equation 3.10.

$$(\cdot,\cdot)_{\Omega_h} := \sum_{K\in\mathcal{T}_h}(\cdot,\cdot)_K \tag{3.9}$$

$$(\cdot,\cdot)_{\Gamma_h} := \sum_{K\in\mathcal{T}_h}(\cdot,\cdot)_{K\cap\Gamma} \tag{3.10}$$

Summarizing, this subsection on the finite element method introduced the pipeline from the continuous weak formulation to a discrete algebraic system. In the next sections, the topic of unfitted finite element methods will be expanded upon.

3.1.2 Historical Overview of Unfitted Methods

In this section, a concise historical review of unfitted methods is presented to demonstrate awareness of the broader class of techniques beyond those applied in this dissertation. The overview situates the selected methods within the wider development of unfitted approaches, highlighting key concepts and representative strategies. The development of numerical methods for solving partial differential equations has seen a significant evolution, particularly concerning the integration of complex geometries into computational models. Over the decades, several distinct yet interconnected methodologies have emerged to address the challenges posed by meshes that do not conform to the exact shape of physical domains, often referred to as immersed, embedded, or unfitted methods. While this review does not aim to be exhaustive, it aims to provide sufficient context to understand the motivation for focusing on the unfitted methods applied in the following sections.

Early Concepts and Foundations (1960s–1970s)

The foundational ideas for some of these approaches can be traced back to the early 1960s. Saulev introduced the term *fictitious domain methods* during the early 1960s [98]. This approach involves embedding the original, often complex, physical domain into a larger, simpler auxiliary domain, typically one that can be readily discretized with structured or Cartesian grids. This simplification of mesh generation allowed for the use of fast solvers, but necessitated new techniques for incorporating the original boundary conditions, such as Lagrange multipliers or penalty terms [99] [100]. In 1963, Noh introduced an early coupled Eulerian-Lagrangian numerical scheme for simulating time-dependent, two-dimensional compressible fluid dynamics, combining a Lagrangian mesh update with an Eulerian remap step to handle large deformations without mesh tangling [101]. Independently, Sirovich in 1967 utilized a concept similar to the later Immersed Boundary method, employing Lagrangian markers tied by stiff springs to target positions to approximate fixed boundaries [102]. A pivotal development in this era was the introduction of the *Immersed Boundary (IB) method* by Peskin in 1972. Initially developed to simulate cardiac mechanics and associated blood flow, the IB method's distinguishing feature was its use of a fixed Cartesian grid for Eulerian fluid variables, while the immersed boundary itself (e.g., elastic heart muscle fibers) was represented by a moving curvilinear mesh of Lagrangian points [103]. The interaction between the fluid and the boundary was mediated through a smoothed approximation to the Dirac delta function. This method was particularly suited for flows with immersed elastic boundaries [104]. Additionally, a well-deserved mention in the 1970s, J. Nitsche proposed a variational approach enabling Dirichlet boundary conditions to be imposed in a weak sense, thus avoiding the need for them to be essential boundary conditions within the finite element space [97]. This concept proved to be highly influential in the later development of unfitted finite element methods [105].

Emergence of Enriched Methods and Level Set (1980s–1990s)

The 1980s saw further advancements in fictitious domain methods, with contributions such as those by Barrett and Elliott, who introduced the *unfitted finite element method* designed for elliptic equations [106–108]. This method aimed to overcome limitations of classical fictitious domain approaches by incorporating a weight function within the variational formulation. Towards the end of the decade, the *level set method*, pioneered by Osher and Sethian [109], emerged as a powerful tool for implicitly describing evolving interfaces, which would later become integral to many unfitted methods, e.g. [100, 110–112].

The 1990s marked a period of significant diversification and refinement for these techniques:

- **Immersed Boundary Methods (continued)**: Peskin and others explored implicit and semi-implicit versions of the immersed boundary method to alleviate severe time step restrictions commonly encountered with explicit schemes. Their work demonstrated that implicit methods offered superior stability, while explicit methods remained the most economical per time step. Applications of the IB method expanded to encompass a wide array of biological and engineering problems, including blood flow dynamics, aquatic animal locomotion, and simulations involving suspended particles and platelet aggregation [102, 104, 113].

- **Fictitious Domain Methods (continued)**: fictitious domain methods were further developed for Dirichlet problems, leveraging Lagrange multipliers to enable the use of regular grids and fast solvers even for complex geometries [99].
- **Partition of Unity Methods (PUM):** The concept of enriching approximation spaces gained prominence with the *Partition of Unity Method* (PUM). PUM allowed the inclusion of a priori analytical knowledge about a problem into the finite element space, leading to highly effective and robust methods, even for problems where classical FEM might struggle or be computationally prohibitive [114].
- **Generalized Finite Element Methods (GFEM)**: The *Generalized Finite Element Method* (GFEM) [115], was seen as a blend of classical FEM and PUM, utilizing standard finite element shape functions to construct the partition of unity, thereby maintaining computational efficiency [116, 117].
- **eXtended Finite Element Methods (XFEM)**: A significant specialization within the GFEM family was the *eXtended Finite Element Method* (XFEM) [118, 119]. XFEM's core strength lies in its ability to model discontinuities, such as cracks, independently of the underlying mesh, eliminating the need for costly remeshing during crack growth. It achieves this by enriching the polynomial approximation space locally with functions that capture specific non-smooth features, such as discontinuous fields and near-tip asymptotic crack fields. The level set method proved to be an ideal complement for describing these interfaces in XFEM [110, 111].
- **Ghost Fluid Method (GFM)**: An early approximate domain method was the *Ghost Fluid Method* (GFM) [112]. In summary, this method enables multiphase flow simulation by leveraging a standard single-phase solver together with ghost cells. Each fluid is advanced independently using standard time integration, and the level set function selects the valid fluid at each point. The key is that all interface handling and boundary conditions are managed through the ghost cell routine, which defines ghost values to enforce the Euler equations properly. This makes the method simple to implement and adaptable, requiring only the addition of a ghost cell subroutine to an existing solver.

Diversification and Refinement (2000s - 2010s)

This era saw the maturation of unfitted methods, with a focus on stability, accuracy, and practical implementation for complex geometries and evolving interfaces, addressing challenges like ill-conditioning.

- **Nitsche's method for unfitted meshes**: Up to this point in history, the general approach for the weak enforcement of Dirichlet boundary conditions was using Lagrange multipliers [99]. A drawback in computational efficiency of Lagrange multipliers is the need to introduce additional multiplier variables, instead Nitsche's method [97] from the 1970s was re-introduced for the weak imposition of boundary and interface conditions. Hansbo and Hansbo [105] were among the first to demonstrate its practical application for interface problems on unfitted meshes, showing how to enforce boundary and interface conditions in a weak sense without using

Lagrange multipliers. By adding consistent penalty and flux terms to the variational formulation.

- **Finite Cell Method (FCM)**: The *Finite Cell Method* (FCM) is a fictitious domain approach developed to handle complex geometries by combining the fictitious domain concept with high-order finite elements [100, 120]. In FCM, the physical domain is embedded in a larger, simple computational mesh that extends beyond the true boundary. The method integrates over both the physical and fictitious parts, applying a penalization to the fictitious region to effectively deactivate it. This allows the use of structured grids and high-order basis functions for problems with implicit (level set) or explicit (CAD-based, i.e. STL file-type) geometrical representations, avoiding complex meshing while retaining accuracy. Accurate numerical integration near cut cells is critical and is typically addressed through adaptive quadrature or sub-cell integration.

- **Trace Finite Element Method (TraceFEM)**: The *Trace Finite Element Method* (TraceFEM) [121] is an unfitted approach developed to solve partial differential equations on surfaces or interfaces implicitly defined within a bulk mesh. Instead of explicitly meshing the surface, TraceFEM represents it, e.g. by a level set function, and constructs the finite element space by taking the trace of the background mesh basis functions onto the embedded surface. This allows accurate discretization of surface or interface problems on fixed, structured meshes without requiring mesh conformity to the geometry. Stabilization techniques are often incorporated to address challenges related to small cut elements and ensure numerical robustness. TraceFEM has become a widely used framework for solving problems such as surface diffusion, fluid–interface coupling, and other PDEs posed on evolving interfaces.

- **Cut Finite Element Method (CutFEM)**: The *Cut Finite Element Method* (CutFEM) [122] is an unfitted finite element framework designed to solve partial differential equations on domains with complex or evolving boundaries that do not conform to the computational mesh. CutFEM extends the classical finite element method by allowing elements to be intersected, by an implicitly defined boundary or interface. To handle the resulting small cut cells, which can cause ill-conditioning and loss of stability, CutFEM combines weak enforcement of boundary or interface conditions, typically using Nitsche's method[97, 105] with additional stabilization terms, such as ghost penalty [123] techniques. For an extensive review of the core concepts and key developments the reader is referred to [124].

- **Aggregated unfitted Finite Element Method (AgFEM)**: The *Aggregated unfitted Finite Element Method* (AgFEM) is an unfitted finite element technique designed to address the ill-conditioning and stability issues caused by small cut cells in embedded or immersed domain problems [77]. AgFEM mitigates these challenges by aggregating or merging cut elements with neighboring elements to form larger, more regular computational cells, thereby improving the conditioning of the resulting linear systems. This aggregation preserves the accuracy of the solution while enabling the use of enhanced standard finite element spaces and solvers on fixed background meshes. By reducing the sensitivity to arbitrarily small intersections between the mesh and

the physical domain, AgFEM facilitates robust and scalable simulations of complex geometries without the need for boundary-fitted remeshing.

- **Shifted Boundary Method (SBM)**: The *Shifted Boundary Method* (SBM) is an unfitted finite element technique developed to simplify the treatment of complex or non-conforming boundaries without explicitly cutting mesh elements [125]. Instead of directly resolving the true boundary within the mesh, SBM introduces a surrogate boundary that is shifted slightly inside the background mesh while maintaining a consistent distance to the actual geometry. The governing equations and boundary conditions are modified accordingly to account for this shift, ensuring accuracy and stability without the complications of cut cells. This strategy avoids the need for complex quadrature over irregular cut elements and naturally integrates with standard finite element formulations on structured or unstructured grids.

Taken together, these developments illustrate the main strategies for handling non-conforming geometries in numerical analysis. Early fictitious domain and immersed boundary methods focused on enforcing boundary and interface conditions in extended domains, and on capturing the immersed boundary. Later methods introduced varying stabilization approaches, local enrichment, or techniques to handle cut cells to improve accuracy, conditioning, and ease of implementation. This progression highlights the current key challenges of unfitted methods, namely:

Challenge 1 ***(C1)*** *integration over intersected elements,*

Challenge 2 ***(C2)*** *conditioning of small intersections (small cut cell problem), and*

Challenge 3 ***(C3)*** *consistent enforcement of boundary and interface conditions.*

3.1.3 Synthesis and Common Challenges

The methods described in the preceding overview represent key branches of the unfitted finite element family. They share the common goal of simplifying mesh generation for problems with complex or evolving geometries by using fixed background meshes that do not conform to domain boundaries or interfaces. These methods could remove the need for frequent remeshing, which is especially valuable for problems involving moving interfaces or complex or more realistic geometries.

The mitigation strategies for the integration of cut-cells (**C1**) generally consist of two strategies, namely *subtriangulation*, which is a method that entails segmenting elements that are crossed by a boundary or interface into smaller sub-cells or sub-triangles. This approach is specifically used for precise numerical integration over intricate geometric regions while maintaining the conformity of the finite element space [77, 121, 122]. It should be noted that the subtriangulation approach discussed here partially undermines the appeal of unfitted methods, since for imperfect geometries it effectively requires local grid generation and thus reintroduces some of the complexity that unfitted methods are designed to avoid. The second strategy is *adaptive quadrature rules*, which involve non-uniform refinement of integration sub-cells to cluster quadrature points for enhanced accuracy in regions where the integrand is discontinuous or varies strongly, such as near boundaries and interfaces [120, 121]. Additionally, a third option worth mentioning is

the *moment-fitting* integration approach, which constructs weights and quadrature points based on the idea of fitting specific integral moments [110]. This summarized overview is, in a sense, an oversimplification, as other integration approaches are possible, as well as combinations of integration methods [120].

The second challenge considers matrix conditioning when small intersections are present (**C2**). These small cut cells manifest when the physical domain intersects only a minimal portion of a background element. Basis functions that are predominantly supported on these slivers contribute negligibly small entries to the local stiffness matrices, as the integrals are scaled according to the volume fraction of the cut cell. When these matrices are assembled globally, they result in rows and columns with near-zero contributions, leading to eigenvalues that approach zero within the system matrix. Consequently, the condition number inversely correlates with the smallest cut volume, significantly impairing the efficacy of linear solvers. Therefore, several mitigation strategies are introduced. FCM has *no explicit* stabilization for cut cells; it applies higher-order basis functions, direct solvers, and uses an artificial small density outside of the immersed domain, to stabilize the base formulation [100, 120]. For TraceFEM, *several stabilization techniques* are mentioned, particularly various forms of penalty and volume-based gradient controls [121]. CutFEM mainly employs the *ghost penalty* stabilization [122, 123]. AgFEM avoids the need for stabilization of small cut cells by excluding the ill-posed degrees of freedom using a *cell aggregation* algorithm and an extension operator that replaces the ill-posed degrees of freedom with a linear combination of well-posed degrees of freedom defined in their assigned root cell [77].

The third challenge considers the enforcement of boundary conditions (**C3**). These are typically imposed in one of three ways. The most robust modern approach is *Nitsche's method*, which weakly enforces the boundary condition by adding consistent and symmetric terms to the variational form, ensuring stability without introducing additional unknowns [97, 105]. An earlier, simpler alternative is the *penalty method*, which adds a large penalty term to discourage deviations from the prescribed condition, though this can degrade conditioning and lacks variational consistency [100, 120]. A third option is the use of *Lagrange multipliers*, which introduces extra unknowns along the unfitted boundary to exactly enforce constraints in a weak sense [99, 126]. In the original work, FCM applies a penalty method for the enforcement of boundary conditions [120], whereas TraceFEM, CutFEM, AgFEM, and SBM apply Nitsche's method [77, 121, 122, 125]. More recent implementations of FCM apply Nitsche's method as well [127].

In Table 3.1 an overview is depicted of the three challenges and the mitigation strategies imposed by each unfitted method. It should be noted that for SBM, challenges (**C1**) and (**C2**) are not applicable as this unfitted method is an approximate domain method and does not introduce cut cells.

This section concludes the brief historical overview of unfitted methods. It is acknowledged that not all immersed, embedded, or unfitted methods have been covered here. Not all details regarding the challenges of these methods are discussed. The overview shows each method and the challenges as stated in their original works, and there have been numerous contributions over the years to improve each of these unfitted methods. See, for example, the reviews for FCM [76], and for CutFEM and the historical overview [124], regarding stability and conditioning (**C2**) [80]. A brief overview of the development of

C	FCM	TraceFEM	CutFEM	AgFEM	SBM
1	adaptive	subtriangulation	subtriangulation	subtriangulation	-
2	not explicit	several	ghost penalty	aggregation	-
3	Nitsche	Nitsche	Nitsche	Nitsche	Nitsche

Table 3.1: Overview of general challenges, denoted by the first column, of unfitted methods and mitigation strategies per specific unfitted method: FCM, TraceFEM, CutFEM, AgFEM, and SBM.

AgFEM is given in the introduction by Badia et al. [128]. For a more complete list of citations for a historical overview and for an overview of developments of the SBM, the reader is referred to the introduction sections of [129, 130].

3.2 Unfitted FEM for Potential Flow FSI

The following section justifies the focus on several of these methods, outlining their relevance to the objectives of this work and their suitability for addressing the particular demands. It describes the differences and commonalities in notation between the unfitted methods. Thereafter, a common notation across unfitted methods is introduced, which is used throughout this work. After which the unfitted methods considered in this work are introduced. Using potential flow for free surface flows with the finite element method is no new feat; see [131–139]; the novelty of this work lies in the coupling of a linear potential flow formulation for free surface flows to a rigid floating structure using several novel unfitted finite element methods to incorporate the structure. For each unfitted method, we introduce a linear potential flow formulation for free surface waves in a reduced form, coupled to a rigid object in both the time and frequency domains. To the authors' knowledge, this has not been demonstrated before. Specifically, the shifted boundary definition is similar to the Neumann-boundary condition applied in solid mechanics [140] and to the shallow water equations [141], but it is significantly different due to the dynamic boundary condition between the fluid and the structure. The potential flow formulation for WSBM is entirely novel, as this formulation has been exclusively defined for incompressible Navier-Stokes [129].

3.2.1 Justification for Unfitted Methods

The numerical modeling of offshore hydrodynamics, including free surface flows and fluid–structure interaction, places stringent demands on mesh flexibility, interface resolution, and numerical stability. Additionally, these unfitted methods should all be implemented in a single finite element framework such that a comparison across methods can be conducted more fairly.

Embedded approaches, such as the Finite Cell Method and the Trace Finite Element Method, are not pursued here, as they are naturally more aligned with solid mechanics or surface-bound partial differential equations rather than with bulk free surface flows and large-scale fluid-structure interaction problems central to offshore applications.

At the start of this PhD project in November 2021, there were several works on unfitted finite element methods targeting free-surface flows or fluid-structure interaction, particularly emphasizing the suitability of these methods for such applications. The SBM

has been applied to the shallow water equations [141], and to free surface flows [129] using the weighted SBM. CutFEM has been applied to fluid-structure interaction in 2D for elastic structures [142], in 3D for hyperelastic structures [143], in 3D for elastic structures using various domain decomposition techniques [144], in 3D for poroelasticity [145], in 2D for a massless immersed membrane [146], and in 2D non-Newtonian free surface flows for glacier modeling [147]. AgFEM has not been applied to free surface flows or to fluid-structure interactions as of this starting date. None of the unfitted methods have been applied yet to incompressible potential flow formulations at this time.

Considering the requirements for numerical wave tanks for offshore applications, i.e., focusing on free surface flows and fluid-structure interaction with possibly large deformations, and taking into account the promises of the unfitted finite element methods for handling such problems. Additionally, noting that an incompressible potential flow model for free surface flows and fluid-structure interaction does not exist when coupled with AgFEM, CutFEM, or SBM. The decision has been made to incorporate these three unfitted methods into the proposed numerical wave tank. Additionally, the WSBM is considered a fourth unfitted method, as it addresses issues with the SBM when applied to free surface flows.

Developments outside this dissertation during the project

During the time of this project, progress has been made in developing and applying unfitted finite element methods for potential flow with free surfaces, general free surface problems, and fluid–structure interaction. We summarize key developments for each method, focusing on their evolution within these themes with respect to the dates at which the justifications have been made.

For potential flow with free surface flows, there has been limited development. Visbech et al. show preliminary results using a high-order spectral element method with an SBM type unfitted method [148]. For fluid-structure interaction, Kees et al. propose a CutFEM formulation using the notion of equivalent polynomials [149], and for free surface flows, the Cut-PFEM is proposed by Zorilla et al. [150]. Xu et al. show improvements for the WSBM with respect to the original work [129] with the inclusion of Dirichlet-type boundary conditions [130] and the full Navier-Stokes equations [151]. For AgFEM, the developments have been orientated towards space-time formulations for moving domains; see Badia et al. [128, 152]. Additionally, AgFEM has been implemented for coupled surface-bulk problems in time-dependent domains, focusing on fluid-fluid interactions by Neiva and Turlier [153].

Finally, a promising approach that aligns well with the research direction of this work is the recently introduced formulation of the SBM, known as the generalized SBM (GSBM). This approach provides a finite element formulation that is decoupled from the underlying geometry and does not require any redefinition of finite element spaces, integration procedures, or geometric entities [154].

3.2.2 Notations and Definitions

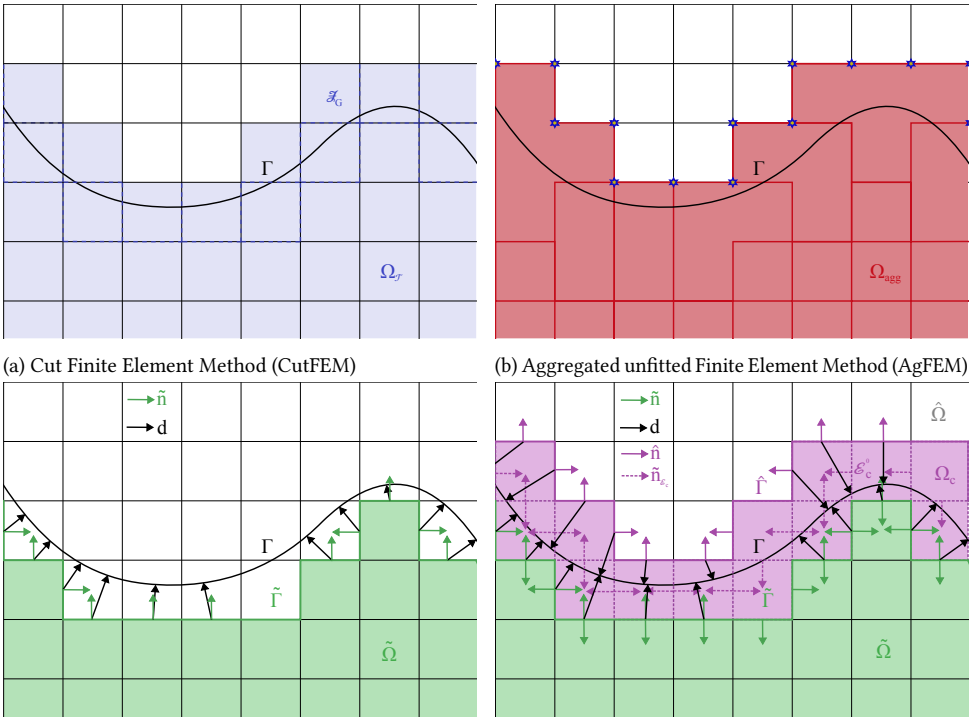
The four unfitted finite element methods considered in this work are the Cut Finite Element Method [122], the Aggregated unfitted Finite Element Method [77], the Shifted Boundary Method [125], and the Weighted SBM [129]. Broadly speaking, these methods can be divided into two categories: methods that integrate in the physical domain, i.e., CutFEM

and AgFEM, and approximate boundary type methods, i.e., SBM and WSBM. Due to this division of methods, there is also different terminology between these methods for similar concepts. In this section, we will describe the commonalities and differences between these methods and adopt consistent terminology across methods.

Method	Integration Domains	FE space	Stabilization region
CutFEM	$\Omega_{\text{cut},h}, \Gamma_h$	$\Omega_{\mathcal{T}}$	$\mathcal{F}_{G,h}$
AgFEM	$\Omega_{\text{cut},h}, \Gamma_h$	$\Omega_{\text{agg},h}$	-
SBM	$\tilde{\Omega}_h, \tilde{\Gamma}_h$	$\tilde{\Omega}_h$	-
WSBM	$\tilde{\Omega}_h \cup \Omega_{\text{c},h}, \tilde{\Gamma}_h \cup \mathcal{E}^0_{\text{c},h} \cup \hat{\Gamma}_h$	$\tilde{\Omega}_h \cup \Omega_{\text{c},h}$	$\tilde{\Gamma}_h \cup \mathcal{E}^0_{\text{c},h}$

Table 3.2: Overview of different domains used by the CutFEM, AgFEM, SBM, and WSBM in their corresponding original works.

In Figure 3.2, we show the different domains required for the FE spaces when using the various unfitted finite element methods described in this thesis. The notation in this figure is consistent with the original works.



(a) Cut Finite Element Method (CutFEM)

(b) Aggregated unfitted Finite Element Method (AgFEM)

(c) Shifted Boundary Method (SBM)

(d) Weighted Shifted Boundary Method (WSBM)

Figure 3.2: Overview of the general concept of four types of embedded methods discussed in this work using notation from each original work.

The embedded and fitted domains and boundaries of the different methods have varying

naming conventions in the respective original papers, see Table 3.2. For the sake of readability of the coming sections, here we summarize the different domains of integration and domains required for interpolation used by the different approaches, establishing a unified notation that will be used hereinafter, see Table 3.3. We distinguish between domains of integration, i.e. the domain where the numerical integration of the weak form is performed, for both interior and boundary integrals, the finite element space domain, i.e. the domain over which the basis functions are defined, and the stabilization integration domain for cut cells, where additional stabilization terms are integrated.

Method	Integration Domains	FE space	Stabilization region
CutFEM	$\Omega_{\text{cut},h}$, Γ_h	$\Omega_{\text{act},h}$	$\mathcal{E}_h^0$
AgFEM	$\Omega_{\text{cut},h}$, Γ_h	$\Omega_{\text{agg},h}$	-
SBM	$\tilde{\Omega}_h$, $\tilde{\Gamma}_h$	$\tilde{\Omega}_h$	-
WSBM	$\Omega_{\text{act},h}$, $\mathcal{E}_h^0 \cup \hat{\Gamma}_h$	$\Omega_{\text{act},h}$	$\mathcal{E}_h^0$

Table 3.3: Overview of different domains used by the CutFEM, AgFEM, SBM, and WSBM with common domain descriptors that are used throughout this work.

The list below is supplementary to Table 3.3, it names and describes conceptually what is referred to by our common domain descriptors. We consider the domains, represented by Ω, the boundaries, represented by Γ, and the interfaces, represented by $\mathcal{E}$.

- $\Omega_{\text{cut},h}$: The cut domain, also known as the physical domain, approximately fits the physical boundary.
- $\Omega_{\text{act},h}$: The active domain consists of all interior and cut cells; this domain does not fit the physical boundary.
- $\Omega_{\text{agg},h}$: The aggregated domain is similar to the active domain, but the degrees of freedom (dofs) of the cut cells are aggregated to the interior cells.
- $\tilde{\Omega}_h$: The inner surrogate domain consists of all interior cells only; this domain does not fit the physical boundary.
- Γ_h: The unfitted boundary approximately fits the physical boundary.
- $\tilde{\Gamma}_h$: The inner surrogate boundary corresponds to the interface between the interior and cut cells; note this is a boundary and not an interface.
- $\hat{\Gamma}_h$: The outer surrogate boundary corresponds to the interface between the cut and exterior cells; note this is a boundary and not an interface.
- $\mathcal{E}_h^0$: The cut cell edges are the set of edges corresponding to the interface between the interior and cut cells, as well as the edges between the cut cells.

Common domain definitions

Let us consider Ω the domain where the problem is defined, and Γ the embedded boundary. We denote by Γ_h a discrete representation of the boundary Γ over which we can define

integration rules. Let us consider a non-conforming background triangulation $\mathcal{T}_h$. The set of elements of a background triangulation interior to Ω and non-cut by Γ are denoted by $\mathcal{T}_{\text{in}}$, that is $\mathcal{T}_{\text{in}} := \{K \in \mathcal{T}_h : K \subset \Omega\}$. Then, the interior domain $\tilde{\Omega}_h$ will be composed of the union of elements in the interior triangulation $\tilde{\Omega}_h := \left(\cup_{K \in \mathcal{T}_{\text{in}}} K\right)$. Let us define the triangulation composed of the set of elements intersected by Γ_h, denoted as $\mathcal{T}_\Gamma := \{K \in \mathcal{T}_h : K \cap \Gamma_h \neq \emptyset\}$, resulting in the intersected domain $\Omega_{\text{c},h} := \left(\cup_{K \in \mathcal{T}_\Gamma} K\right)$. The union of interior and intersected cells composes the active domain $\Omega_{\text{act},h} := \tilde{\Omega}_h \cup \Omega_{\text{c}}$. Each element of the intersected triangulation $\mathcal{T}_\Gamma$ can be decomposed into a set of sub-cell elements defined by convex polyhedra that conform to the discrete boundary Γ_h, which will be denoted by $\mathcal{T}_{\Gamma,\text{cut}}$, see [155] for an in-depth description of how to define such triangulation. We denote by cut interior domain, $\Omega_{\text{cut},h}$, the union of elements in $\mathcal{T}_{\text{in}}$ and $\mathcal{T}_{\Gamma,\text{cut}}$, $\Omega_{\text{cut},h} := \left\{\cup_{K \in \mathcal{T}_{\text{in}} \cup \mathcal{T}_{\Gamma,\text{cut}}} K\right\}$. Let us denote by $\tilde{\Gamma}$ the set of facets of dimension $d_n - 1$ of $\mathcal{T}_{\text{in}}$ that belong to $\tilde{\Omega}_h$, with d being the topological dimension of Ω, closest to Γ. Note that $\tilde{\Omega}_h$ and $\tilde{\Gamma}_h$ are denoted by surrogate domain and surrogate boundary, respectively, in the SBM-related works [125, 140]. We define Γ_{cut} as the set of facets of dimension $d_n - 1$ of $\mathcal{T}_{\text{in}}$ that are cut by Γ and by $\hat{\Gamma}_h$, also denoted as outer surrogate boundary, the set of facets of dimension $d_n - 1$ of $\mathcal{T}_\Gamma$ that do not belong to $\tilde{\Gamma}_h$ nor Γ_{cut}, i.e. $\hat{\Gamma}_h := \left\{E \in \mathcal{T}_\Gamma : E \notin (\tilde{\Gamma} \cup \Gamma_{\text{cut}})\right\}$.

Let us finally consider non-overlapping cell aggregates A_K composed of cut cells and one interior cell $K \in \mathcal{T}_{\text{in}}$ such that the aggregate is connected. This leads to another triangulation $\mathcal{T}_{\text{ag}}$, given by the aggregations of cells in $\Omega_{\text{act},h}$, see [77] for details on the aggregation algorithm. Then, the corresponding domain is $\Omega_{\text{agg},h} := \left(\cup_{A \in \mathcal{T}_{\text{ag}}} A\right)$.

In this work, we assume that the integration on the cut cells, $\Omega_{\Gamma,\text{cut}}$, is done by constructing an explicit representation of the discrete domain. Alternatively, one could also use adaptive quadrature rules to integrate on this domain without the need of tessellating the cut cells, see [76, 156].

Common implementational pipeline

Now, we introduce a brief list of common steps for each unfitted method. This list serves as an enumerated reference, such that we can explain the different methods and discuss differences at a later stage.

- **I1 Geometry representation**: define the physical domain using implicit or explicit representation.
- **I2 Generate background mesh**: create a fixed mesh such that it serves as a bounding box or relevant domain of interest.
- **I3 Cut cell and interface detection**: locate elements intersected by the boundary or interface using the level-set or advanced algorithms.
- **I4 Weak formulation**: introduce the variational problem and generate boundary and interface terms appropriate for the method.
- **I5 Discretization**: introduce discretized domain required for the finite element space and for the integration space.
- **I6 Method-specific step**: this step differs depending on the unfitted method. Note that this can include modifications or additions to the weak form defined in **I4**.

I7 **Assembly and solve**: assemble global matrices and vectors, and solve discretized problem.

In this work, steps **I1**-**I3** and **I7** are exactly the same across unfitted methods. The weak form in step **I4** is taken from Equation 3.3 and Equation 3.6 for each specific problem, but for some unfitted methods the bilinear form requires modifications that are to be discussed. In step **I5** the first differences across methods are introduced and discussed. Step **I6** is the central point of discussion in the next sections.

3.2.3 Cut Finite Element Method

The concept underpinning the Cut Finite Element Method [122] focuses on achieving a discretization process that is independent of the geometric description, aiming to reduce the challenges of mesh generation. This approach seeks to maintain the precision and stability characteristic of a traditional finite element method and proposes stabilization techniques, specifically ghost penalty stabilization, to counteract the small-cut cell problem [122]. In the original work [122], only geometrical representations using level-set functions are considered, but a claim is made that CutFEM is easily extended to explicit geometry representations.

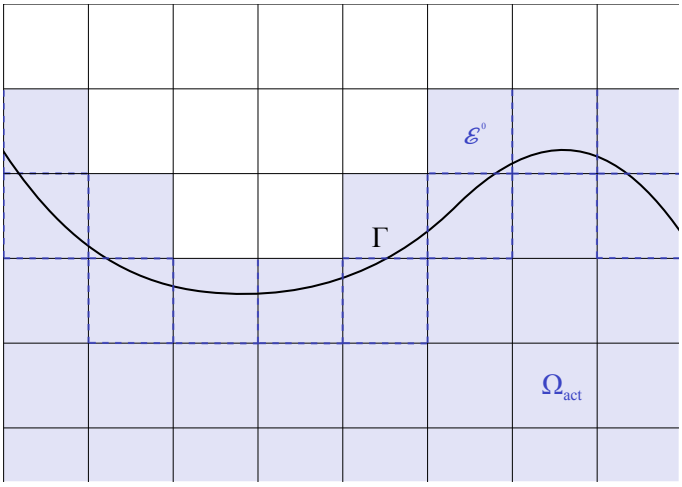


Figure 3.3: CutFEM with naming conventions used throughout this work, domain for basis function is the active domain Ω_{act}, interfaces for ghost penalty stabilization $\mathcal{E}^0$, and embedded boundary Γ.

Considering the discretization step **I5**, it is required for the full discretized domain to be reduced to the active domain using the active cells as determined by **I3**. The finite element spaces are defined on this active domain, denoted by $\Omega_{\mathcal{T}}$ in the original work; this corresponds to the blue area in Figure 3.2a. We denote the active domain by $\Omega_{act,h}$. For the integration space, standard integration rules are applied for the non-cut elements, but the cut elements require subtriangulation algorithms, such that the integration approximately conforms to the physical boundary Γ. For reference, the notation in this work is depicted in Figure 3.3. The integration domain is denoted by $\Omega_{cut,h}$. Similarly, the embedded boundary

Γ_h is assigned integration quadratures. The discrete FE space is defined only on the active domain, which leads to the following definition

$$\mathcal{W}_h^{\text{act}} := \left\{ w_h \in C^0(\Omega_{\text{act},h}) \,:\, w_h|_K \in \mathbb{P}_p(K)\ \forall K \in \mathcal{T}_h \right\}. \tag{3.11}$$

Then, the method-specific step **I6** requires us to include the edges of the cut cells. These are denoted by $\mathcal{F}_G$ in the original work and highlighted in Figure 3.2a by the dashed blue lines. In this work, the edges of cut cells are denoted by $\mathcal{E}_h^0$. On these cut edges, ghost penalty stabilization [123] is applied. The ghost penalty stabilization terms are dependent on the order of the discrete spaces p_e, an algorithmic constant parameter $\gamma_g > 0$, and the element size h_e. For this problem, the embedded boundary is a Neumann-type boundary, and hence the stabilization terms of ghost penalty should be appropriately chosen [157].

Time-domain semi-discrete weak form

With these considerations, the semi-discrete weak form using CutFEM is stated as:

find $[\phi_h, \mathbf{u}_h] \in L^2(0,T;\mathcal{W}_h^{\text{act}}) \times L^2(0,T;\mathcal{V}_h)$ such that

$$a_h^{\text{cut}}([\phi_h,\mathbf{u}_h],[w_h,\mathbf{v}_h]) = b_h([w_h,\mathbf{v}_h]) \qquad \forall [w_h,\mathbf{v}_h] \in \mathcal{W}_h^{\text{act}} \times \mathcal{V}_h, \tag{3.12}$$

with

$$\begin{aligned} a_h^{\text{cut}}([\phi_h,\mathbf{u}_h],[w_h,\mathbf{v}_h]) :=& (\nabla w_h, \nabla \phi_h)_{\Omega_{\text{cut},h}} + \left(w_h, \frac{1}{g}\frac{\partial^2 \phi_h}{\partial t^2} \right)_{\Gamma_{\text{f},h}} - \\ & \left(w_h, \frac{\partial \mathbf{u}_h}{\partial t} \cdot \boldsymbol{n}_{\Gamma,j} \right)_{\Gamma_h} + \left(\mathbf{v}_h, \frac{\boldsymbol{M}}{\rho|\Gamma_h|}\frac{\partial^2 \mathbf{u}_h}{\partial t^2} + \left(\frac{\partial \phi_h}{\partial t} + g(\mathbf{u}_h \cdot \boldsymbol{n}_z) \right) \boldsymbol{n}_{\Gamma,j} \right)_{\Gamma_h} + \\ & \sum_{i=1}^{p_e} \langle [\![\nabla^{(i)} w_h]\!], \gamma_g h_e^{(2i+1)} [\![\nabla^{(i)} \phi_h]\!] \rangle_{\mathcal{E}_h^0}. \end{aligned} \tag{3.13}$$

Note that there are two main differences between the bilinear form a_h in equations (3.13) and (3.4). Firstly, the integration domains of the interior and boundary terms differ, $\Omega_{\text{cut},h}$ and Ω_h, respectively. The difference is that the integration domain and integration boundary are algorithmically tessellated instead of generated in a body-fitted manner.

Secondly, there are the additional ghost penalty stabilization terms, see the last term in equation (3.13). Here, we introduce the jump operator $[\![\cdot]\!] = (\cdot)^+ \boldsymbol{n}^+ + (\cdot)^- \boldsymbol{n}^-$, where the superscripts + and − denote both sides of an interface. Additionally, we introduce the notation $\langle \cdot \rangle$ to denote that the integration domain consists of interfaces. Note that different alternatives to this stabilization term can be considered [123, 158]. In [159] the authors discuss such alternatives and the link to a weak enforcement of the AgFEM approach. The linear form remains equal as defined in Equation 3.5. It is also important to highlight that the space for the rigid body motion degrees of freedom remains the same as it only consists of constant domain-independent values.

Frequency-domain discrete weak form

For the frequency-domain problem, the weak form is defined as:

find $[\phi_h, \mathbf{u}_h] \in \mathcal{W}_h^{\text{act}} \times \mathcal{V}_h$ such that

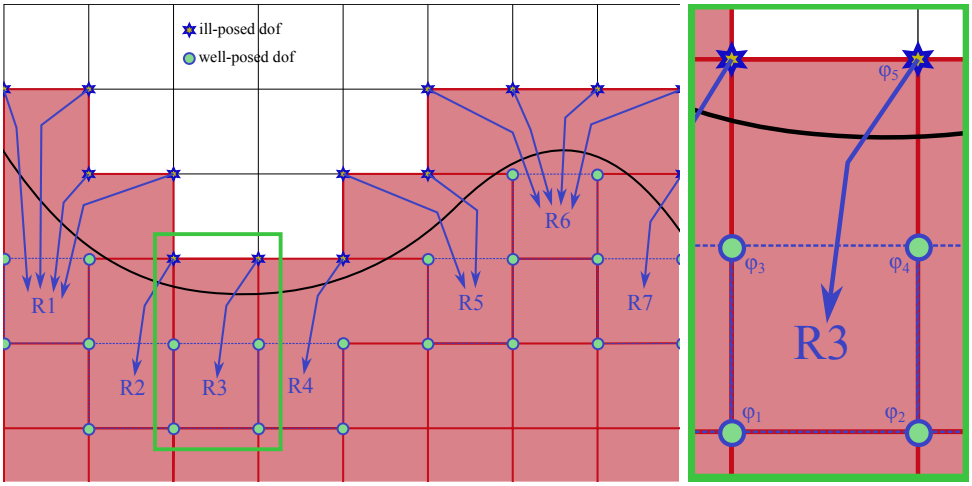
$$a_{\omega,h}^{\text{cut}}([\phi_h,\mathbf{u}_h],[w_h,\mathbf{v}_h]) = b_{\omega,h}([w_h,\mathbf{v}_h]) \qquad \forall [w_h,\mathbf{v}_h] \in \mathcal{W}_h^{\text{act}} \times \mathcal{V}_h, \tag{3.14}$$

with the bilinear form defined as:

$$
\begin{aligned}
a^{\text{cut}}_{\omega,h}([\phi_h,\mathbf{u}_h],[w_h,\mathbf{v}_h]) := &(\nabla w_h,\nabla\phi_h)_{\Omega_{\text{cut},h}} - \frac{\omega^2}{g}(w_h,\phi_h)_{\Gamma_{\text{f},h}} + \\
&i\omega\left(w_h,\mathbf{u}_h\cdot\boldsymbol{n}_{\Gamma,j}\right)_{\Gamma_h} - i\omega\left(\mathbf{v}_h,\phi_h\boldsymbol{n}_{\Gamma,j}\right)_{\Gamma_h} - \\
&\omega^2\frac{\boldsymbol{M}_\rho}{|\Gamma_h|}(\mathbf{v}_h,\mathbf{u}_h)_{\Gamma_h} + g\left(\mathbf{v}_h,(\boldsymbol{n}_z\cdot\mathbf{u}_h)\,\boldsymbol{n}_{\Gamma,j}\right)_{\Gamma_h} \\
&\sum_{i=1}^{p_e}\langle[\![\nabla^{(i)}w_h]\!],\gamma_g h_e^{(2i+1)}[\![\nabla^{(i)}\phi_h]\!]\rangle_{\mathcal{E}_h^0}.
\end{aligned}
\tag{3.15}
$$

3.2.4 Aggregated unfitted Finite Element Method

The main idea that has driven the development of the Aggregated unfitted Finite Element Method [77] is similar to that of CutFEM. With respect to CutFEM, the AgFEM does not require additional stabilization to deal with the small cut cell problem. Instead, it proposes an algebraic aggregation of basis functions in small cut cells. The main working principle of AgFEM is to define a modified discrete space through the use of an extension operator, which extends basis functions of the non-cut cells to the cut cells.



(a) Assigning ill-posed degrees of freedom (dof) to root cells with well-posed degrees of freedom.

(b) Close up of the case for root cell denoted by R3

Figure 3.4: Conceptual representation of the ill-posed degree of freedom mitigation strategy for AgFEM.

The discretization step **I5** for AgFEM is similar to that of CutFEM, but there is one crucial difference which we could classify as the method-specific step **I6**. First, just like for CutFEM, it is required to identify the active domain $\Omega_{\text{act},h}$ as the first step in the construction of the aggregated space. Then, for the cut cells, the ill-posed degrees of freedom are identified, and using an aggregation algorithm, these ill-posed degrees of freedom are aggregated to cells with well-posed degrees of freedom. These cells to which they are aggregated are called root cells. Figure 3.4a depicts the process of assigning ill-posed degrees of freedom,

denoted by the star symbols, to root cells, denoted by R1 to R7, which are highlighted in blue dashed lines and surrounded by well-posed degrees of freedom denoted by the circles. Note that it is assumed that the degrees of freedom coincide with the vertices for this example; for higher-order elements, these degrees of freedom could be located on the element boundaries. For application purposes, there needs to be a choice of a cell aggregation scheme to be imposed: which is chosen to be the one applied in the original paper [77].

The ill-posed degrees of freedom are then reformulated using algebraic combinations of the degrees of freedom in their corresponding root cells. This construction is made using the extension operator. In Figure 3.4b, the case for root cell R3 is highlighted, to give a conceptual understanding of how the ill-posed degrees of freedom are mitigated. The ill-posed degree of freedom, denoted by φ_5, is constrained using the extension operator, which is a linear combination of the well-posed degrees of freedom of the root cell. This means we rewrite for the ill-posed degree of freedom $\varphi_5 = \sum_{i=1}^{4} c_i \varphi_i$, with c_i the constraining coefficients. These coefficients are obtained from information of the aggregates. After this process is applied to all ill-posed degrees of freedom, we are left with the aggregated space, which ensures that all degrees of freedom are well-posed and, as such, the small cut cell problem is alleviated. A distinctive trait of the AgFEM is the finite element space definition, $\mathcal{W}_h^{\text{ag}}$, which is formally given by

$$\mathcal{W}_h^{\text{ag}} := \left\{ w_h \in \mathcal{C}^0(\Omega_{\text{act},h}) \,:\, w_h \,|_A \in \mathbb{P}^r(A), \forall A \in \mathcal{T}_{\text{ag}} \right\}. \tag{3.16}$$

An in-depth explanation of the workings and mathematical proofs of AgFEM is outside the scope of this work, and the interested reader is referred to [77]. The discretization step **I5** is then continued as for CutFEM, tessellating the cut domain $\Omega_{\text{cut},h}$ and embedded boundary Γ_h for integration.

Time-domain semi-discrete weak form

The semi-discrete weak form for the AgFEM case is stated as:

find $\phi_h \in L^2(0,T;\mathcal{W}_h^{\text{ag}})$, and $\mathbf{u}_h \in L^2(0,T;\mathcal{V}_h)$ such that

$$a_h^{\text{ag}}((\phi_h,\mathbf{u}_h),(w_h,\mathbf{v}_h)) = b_h(w_h,\mathbf{v}_h) \qquad \forall w_h \in \mathcal{W}_h^{\text{ag}} \text{ and } \mathbf{v}_h \in \mathcal{V}_h, \tag{3.17}$$

with

$$\begin{aligned} a_h^{\text{ag}}([\phi_h,\mathbf{u}_h],[w_h,\mathbf{v}_h]) :=& (\nabla w_h, \nabla \phi_h)_{\Omega_{\text{cut},h}} + \left(w_h, \frac{1}{g}\frac{\partial^2 \phi_h}{\partial t^2} \right)_{\Gamma_{\text{f},h}} - \left(w_h, \frac{\partial \mathbf{u}}{\partial t} \cdot \boldsymbol{n}_{\Gamma,j} \right)_{\Gamma_h} + \\ & \left(\mathbf{v}_h, \frac{\boldsymbol{M}}{\rho|\Gamma_h|}\frac{\partial^2 \mathbf{u}_h}{\partial t^2} + \frac{\partial \phi_h}{\partial t} + g(\mathbf{u}_h \cdot \boldsymbol{n}_z)\boldsymbol{n}_{\Gamma,j} \right)_{\Gamma_h}. \end{aligned} \tag{3.18}$$

Note that the only difference between Equation 3.18 and Equation 3.4 is the integration domain of the interior term $\Omega_{\text{cut},h}$ and the boundary Γ_h.

Frequency-domain discrete weak form

For the frequency-domain problem, the weak form is defined:

find $[\phi_h, \mathbf{u}_h] \in \mathcal{W}_h^{\text{ag}} \times \mathcal{V}_h$ such that

$$a_{\omega,h}^{\text{ag}}([\phi_h, \mathbf{u}_h], [w_h, \mathbf{v}_h]) = b_{\omega,h}([w_h, \mathbf{v}_h]) \qquad \forall [w_h, \mathbf{v}_h] \in \mathcal{W}_h^{\text{ag}} \times \mathcal{V}_h, \tag{3.19}$$

with the bilinear form defined as:

$$\begin{aligned} a_{\omega,h}^{\text{ag}}([\phi_h, \mathbf{u}_h], [w_h, \mathbf{v}_h]) := & (\nabla w_h, \nabla \phi_h)_{\Omega_{\text{cut},h}} - \frac{\omega^2}{g} (w_h, \phi_h)_{\Gamma_{\text{f},h}} + \\ & i\omega \left(w_h, \mathbf{u}_h \cdot \boldsymbol{n}_{\Gamma,j} \right)_{\Gamma_h} - i\omega \left(\mathbf{v}_h, \phi_h \boldsymbol{n}_{\Gamma,j} \right)_{\Gamma_h} - \\ & \omega^2 \frac{\boldsymbol{M}_\rho}{|\Gamma_h|} (\mathbf{v}_h, \mathbf{u}_h)_{\Gamma_h} + g \left(\mathbf{v}_h, (\boldsymbol{n}_z \cdot \mathbf{u}_h) \boldsymbol{n}_{\Gamma,j} \right)_{\Gamma_h}. \end{aligned} \tag{3.20}$$

3.2.5 Shifted Boundary Method

The Shifted Boundary Method [125] presents itself as an alternative unfitted finite element framework designed to ease the complexity of geometry management without compromising accuracy or conditioning quality. The key motivation is to circumvent the integration process over cut cells and the poor conditioning commonly associated with small cut cells. This is achieved by relocating the implementation of boundary conditions to an approximate surrogate boundary that aligns with the underlying mesh.

The SBM avoids the small cut cell problem by integrating only in non-cut cells, i.e. in the surrogate domain $\tilde{\Omega}$, as denoted in the original work [125]. The boundary conditions are enforced at the surrogate boundary, $\tilde{\Gamma}$, shifting the boundary condition value from the physical boundary to the surrogate boundary through a Taylor series expansion. Therefore, during the discretization step **I5**, we are required to identify only the non-cut elements of interest. The cut cell detection algorithm assigns one of three flags to the elements: `IN`, `CUT`, or `OUT`. This notation is also used for the naming convention of the surrogate domain in this work, considering only the elements flagged as `IN`. The surrogate boundary $\tilde{\Gamma}$ is then identified as the interface between the surrogate domain and the domain considering only the `CUT` elements. In Figure 3.2c, this is depicted using the naming conventions from the original work [125]. The method-specific step **I6** then requires us to modify the weak form such that we take into account the shift from the surrogate boundary to the true boundary.

For a formal definition, let us consider a given map $\mathcal{M}$ that maps points from the surrogate boundary to the true boundary:

$$\begin{aligned} \mathcal{M} : \tilde{\Gamma} &\longrightarrow \Gamma \\ \tilde{\mathbf{x}} &\longrightarrow \mathcal{M}(\tilde{\mathbf{x}}) = \mathbf{x} := \mathbf{d} + \tilde{\mathbf{x}}. \end{aligned} \tag{3.21}$$

Where $\mathbf{d} := \mathbf{x} - \tilde{\mathbf{x}}$ is a distance vector that could be obtained, for example, through closest point projection algorithms. Then, any function evaluated at the true boundary, Γ, can be approximated by a Taylor series expansion, in this case truncated after the second term, evaluated at the surrogate boundary, $\tilde{\Gamma}$:

$$f(\mathbf{x}) = f(\mathcal{M}(\tilde{\mathbf{x}})) = f(\tilde{\mathbf{x}}) + \nabla f(\tilde{\boldsymbol{x}}) \mathbf{d} + \mathcal{O}(\mathbf{d}^2). \tag{3.22}$$

For higher order approximations, it is required to increase the number of terms in the Taylor series expansion.

The governing equations are integrated on the surrogate domain, $\tilde{\Omega}$, and surrogate boundary, $\tilde{\Gamma}$. One challenge identified here for the proposed weak formulation is that the pressure force acting on the object should be integrated over the physical boundary, see Equation 2.15f. Therefore, to account for the mismatch of integration boundaries for the dynamic boundary condition on the fluid-solid interface, the change in surface area of this integral, i.e. last term in the weak form stated in Equation 3.4 needs to be accounted for using the absolute value of the Jacobian of the map, $\mathcal{M}$, denoted by J_Γ. As the displacement $\mathbf{u}$ of the object is relative to its rest position, it will also be relative to its shifted rest position and thus does not need to be shifted. The hydrodynamic pressure is dependent on the velocity potential ϕ, which is defined on the surrogate boundary and thus this term needs to be shifted using Equation 3.22. Thus, the approximated dynamic boundary condition term becomes

$$\left(\mathbf{v}_h, \frac{\boldsymbol{M}}{\rho|\Gamma_h|}\frac{\partial^2 \mathbf{u}_h}{\partial t^2} + \left(\frac{\partial \phi_h}{\partial t} + g(\mathbf{u}_h \cdot \boldsymbol{n}_z)\right)\boldsymbol{n}_{\Gamma,j}\right)_{\Gamma_h} \approx \tag{3.23}$$
$$\left(\mathbf{v}_h, \frac{\boldsymbol{M}}{\rho J_\Gamma\left|\tilde{\Gamma}_h\right|}\frac{\partial^2 \mathbf{u}_h}{\partial t^2} + \left(\frac{\partial \phi_h}{\partial t} + \nabla(\frac{\partial \phi_h}{\partial t})\cdot \mathbf{d} + g(\mathbf{u}_h \cdot \boldsymbol{n}_z)\right)\boldsymbol{n}_{\Gamma,j}\right)_{\tilde{\Gamma}_h}.$$

Let us denote the normal vector with respect to the surrogate boundary by $\tilde{\boldsymbol{n}}_\Gamma$. In order to impose the shifted boundary conditions, we start by rewriting the normal vector of the surrogate boundary in terms of the physical normal vector $\boldsymbol{n}_\Gamma$ and the physical tangential vector $\boldsymbol{\tau}_\Gamma$. In Figure 3.5 these quantities are depicted; for the sake of clarity, the Γ subscript is omitted from the figure.

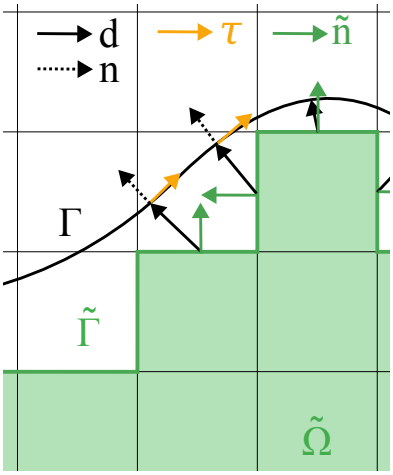


Figure 3.5: Definition of the distance vector $\boldsymbol{d}$ in black, the true normal vector $\boldsymbol{n}$ in dotted black, the true tangential vector $\boldsymbol{\tau}$ in yellow and the surrogate normal vector $\tilde{\boldsymbol{n}}$ in green. Note that the normal and tangential vectors are unit normal and unit tangential vectors, which might not be apparent from the figure.

$$\tilde{\boldsymbol{n}}_\Gamma = (\tilde{\boldsymbol{n}}_\Gamma \cdot \boldsymbol{\tau}_\Gamma)\boldsymbol{\tau}_\Gamma + (\tilde{\boldsymbol{n}}_\Gamma \cdot \boldsymbol{n}_\Gamma)\boldsymbol{n}_\Gamma \tag{3.24}$$

Then, the normal gradient of the solution ϕ at any point of the surrogate boundary can be decomposed as

$$\nabla\phi_h \cdot \tilde{\boldsymbol{n}}_\Gamma = \nabla\phi_h \cdot (\tilde{\boldsymbol{n}}_\Gamma \cdot \boldsymbol{\tau}_\Gamma)\boldsymbol{\tau}_\Gamma + \nabla\phi_h \cdot (\tilde{\boldsymbol{n}}_\Gamma \cdot \boldsymbol{n}_\Gamma)\boldsymbol{n}_\Gamma. \tag{3.25}$$

Using the Taylor series expansion, the approximated kinematic boundary condition, see Equation 2.15e, evaluated at the surrogate boundary reads

$$\frac{\partial \mathbf{u}_h}{\partial t} \cdot \boldsymbol{n}_{\Gamma,j} = (\nabla\phi_h \cdot \boldsymbol{n}_\Gamma)|_{\Gamma_h} \approx (\nabla\phi_h \cdot \boldsymbol{n}_\Gamma)|_{\tilde{\Gamma}_h} + ((\nabla\nabla\phi_h \cdot \boldsymbol{n}_\Gamma) \cdot \mathbf{d})|_{\tilde{\Gamma}_h}. \tag{3.26}$$

Substituting equation (3.26) into equation (3.25), we are left with the following contribution of the Neumann-type boundary condition to the weak form

$$\left(w_h, \nabla\phi_h \cdot (\tilde{\boldsymbol{n}}_\Gamma \cdot \boldsymbol{\tau}_\Gamma)\boldsymbol{\tau}_\Gamma + \frac{\partial \mathbf{u}_h}{\partial t} \cdot (\tilde{\boldsymbol{n}}_\Gamma \cdot \boldsymbol{n}_\Gamma)\boldsymbol{n}_{\Gamma,j} - (\nabla(\nabla\phi_h) \cdot \boldsymbol{d}) \cdot (\tilde{\boldsymbol{n}}_\Gamma \cdot \boldsymbol{n}_\Gamma)\boldsymbol{n}_\Gamma) \right)_{\tilde{\Gamma}_h}. \tag{3.27}$$

Using equation (3.24), we can rewrite equation (3.27) such that we remove $\boldsymbol{\tau}_\Gamma$

$$\left(w_h, \nabla\phi_h \cdot \tilde{\boldsymbol{n}}_\Gamma + \frac{\partial \mathbf{u}_h}{\partial t} \cdot (\tilde{\boldsymbol{n}}_\Gamma \cdot \boldsymbol{n}_\Gamma)\boldsymbol{n}_{\Gamma,j} - (\nabla(\nabla\phi_h) \cdot \boldsymbol{d} + \nabla\phi_h) \cdot (\tilde{\boldsymbol{n}}_\Gamma \cdot \boldsymbol{n}_\Gamma)\boldsymbol{n}_\Gamma) \right)_{\tilde{\Gamma}_h}. \tag{3.28}$$

We are now left with the shifted kinematic and dynamic boundary conditions for the structure, and only require the definition of the discrete space for SBM. Similarly to CutFEM, it is defined as

$$\mathcal{W}_h^{\text{in}} := \left\{ w_h \in C^0(\tilde{\Omega}_h) \,:\, w_h|_K \in \mathbb{P}_p(K)\ \forall K \in \mathcal{T}_h \right\}, \tag{3.29}$$

where the only difference with respect to the former definition, see Equation 3.11, is that the space is defined on the surrogate domain, instead of the active domain.

Time-domain semi-discrete weak form

Then finally, the the semi-discrete weak form for the SBM reads:

find $[\phi_h, \mathbf{u}_h] \in L^2(0,T;\mathcal{W}_h^{\text{in}}) \times L^2(0,T;\mathcal{V}_h)$ such that

$$a_h^{\text{sbm}}([\phi_h, \mathbf{u}_h], [w_h, \mathbf{v}_h]) = b_h([w_h, \mathbf{v}_h]) \qquad \forall [w_h, v_h] \in \mathcal{W}_h^{\text{in}} \times \mathcal{V}_h. \tag{3.30}$$

In that case, the bilinear form is given by

$$\begin{aligned} a_h^{\text{sbm}}([\phi_h, \mathbf{u}_h], [w_h, \mathbf{v}_h]) := {} & (\nabla w_h, \nabla\phi_h)_{\tilde{\Omega}_h} + \left(w_h, \frac{1}{g}\frac{\partial^2 \phi_h}{\partial t^2} \right)_{\tilde{\Gamma}_{\text{f},h}} - \\ & \left(w_h, \nabla\phi_h \cdot \tilde{\boldsymbol{n}}_\Gamma + (\tilde{\boldsymbol{n}}_\Gamma \cdot \boldsymbol{n}_\Gamma)\boldsymbol{n}_{\Gamma,j} \cdot \frac{\partial \mathbf{u}_h}{\partial t} - (\tilde{\boldsymbol{n}}_\Gamma \cdot \boldsymbol{n}_\Gamma)\boldsymbol{n}_\Gamma \cdot (\nabla(\nabla\phi_h) \cdot \boldsymbol{d} - \nabla\phi_h) \right)_{\tilde{\Gamma}_h} + \\ & \left(\mathbf{v}_h, \frac{\boldsymbol{M}}{J_\Gamma \rho \left|\tilde{\Gamma}_h\right|} \frac{\partial^2 \mathbf{u}_h}{\partial t^2} + \left(\frac{\partial \phi_h}{\partial t} + \nabla(\frac{\partial \phi_h}{\partial t}) \cdot \mathbf{d} + g(\mathbf{u}_h \cdot \boldsymbol{n}_z) \right) \boldsymbol{n}_{\Gamma,j} \right)_{\tilde{\Gamma}_h}. \end{aligned} \tag{3.31}$$

Frequency-domain discrete weak form

For the frequency problem, the same shifting operators are applied, and it results in the following discrete weak form:

find $[\phi_h, \mathbf{u}_h] \in \mathcal{W}_h^{\text{in}} \times \mathcal{V}_h$ such that

$$a^{\text{sbm}}_{\omega,h}([\phi_h, \mathbf{u}_h], [w_h, \mathbf{v}_h]) = b_{\omega,h}([w_h, \mathbf{v}_h]) \qquad \forall [w_h, \mathbf{v}_h] \in \mathcal{W}_h^{\text{in}} \times \mathcal{V}_h, \tag{3.32}$$

with the bilinear form given by:

$$\begin{aligned} a^{\text{sbm}}_{\omega,h}([\phi_h, \mathbf{u}_h], [w_h, \mathbf{v}_h]) := (\nabla w_h, \nabla \phi_h)_{\tilde{\Omega}_h} - \frac{\omega^2}{g} (w_h, \phi_h)_{\tilde{\Gamma}_{\text{f},h}} - \\ \omega^2 \frac{\boldsymbol{M}_\rho}{J_\Gamma \left|\tilde{\Gamma}_h\right|} (\mathbf{v}_h, \mathbf{u}_h)_{\tilde{\Gamma}_h} + g \left(\mathbf{v}_h, (\mathbf{u}_h \cdot \boldsymbol{n}_z)\, \boldsymbol{n}_{\Gamma,j}\right)_{\tilde{\Gamma}_h} - \\ i\omega \left(\mathbf{v}_h, (\phi_h + \nabla \phi_h \cdot \mathbf{d}) \boldsymbol{n}_{\Gamma,j}\right)_{\tilde{\Gamma}_h} - \\ (w_h, \nabla \phi_h \cdot \tilde{\boldsymbol{n}}_\Gamma)_{\tilde{\Gamma}_h} - i\omega \left(w_h, \mathbf{u}_h \cdot (\boldsymbol{n}_\Gamma \cdot \tilde{\boldsymbol{n}}_\Gamma) \boldsymbol{n}_{\Gamma,j}\right)_{\tilde{\Gamma}_h} + \\ (w_h, (\boldsymbol{n}_\Gamma \cdot \tilde{\boldsymbol{n}}_\Gamma) \boldsymbol{n}_\Gamma \cdot (\nabla (\nabla \phi_h) \cdot \mathbf{d} + \nabla \phi_h))_{\tilde{\Gamma}_h}. \end{aligned} \tag{3.33}$$

3.2.6 Weighted Shifted Boundary Method

The Weighted Shifted Boundary Method is an extension of the SBM, originally derived for free surface flows by Colomes et al. [129]. The purpose of this method is to address spurious pressure oscillations that appear due to lack of mass conservation of the SBM in problems with moving domains, i.e. wetting and drying of cells. The main idea of the WSBM is to weight the test function by the volume fraction of cells, solving the problem in the full active domain $\Omega_{\text{act},h}$, in the original work this is denoted as $\tilde{\Omega}_h \cup \Omega_{\text{c},h}$.

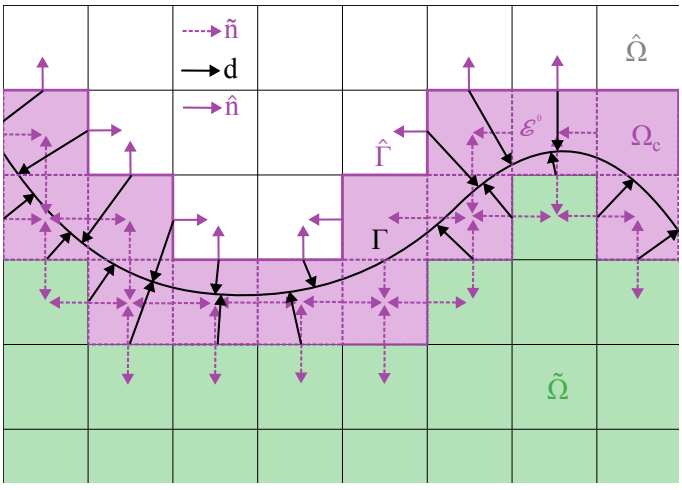


Figure 3.6: WSBM naming conventions used throughout this work, we consider two domains, namely $\tilde{\Omega}$ and Ω_c. One boundary $\hat{\Gamma}$ and one interface $\mathcal{E}^0$.

Considering the discretization step **I5**, we now need the surrogate domain $\tilde{\Omega}_h$, and a domain corresponding to all cut elements; we denote this as $\Omega_{c,h}$. The union of these two domains corresponds to the active domain $\Omega_{act,h} = \tilde{\Omega}_h \cup \Omega_{c,h}$ as defined previously. The reason we identify these two domains separately is that we only want to apply the weighted test functions exclusively in the $\Omega_{c,h}$ domain for computational efficiency. The gain achieved in computational performance because of this decision is up for debate, theoretically less evaluations should indicate better performance, but since the weights are pre-calculated, the difference might be small. On the other hand, when the domain is large and the number of cut cells is relatively small, a significant gain could be made. Additionally, we identify the inner surrogate boundary $\tilde{\Gamma}_h$ as done for SBM; we identify the outer surrogate boundary $\hat{\Gamma}_h$ by finding the intersection between $\Omega_{c,h}$ and the domain consisting of all cells marked as OUT, denoted by $\hat{\Omega}_h$, and we identify the set of edges $\mathcal{E}^0_{c,h}$ that are in the $\Omega_{c,h}$ domain and **not** part of $\tilde{\Gamma}_h$ or $\hat{\Gamma}_h$. Summarizing, as depicted in Figure 3.2d, for discretization, we require two domains: $\tilde{\Omega}_h$ and $\Omega_{c,h}$, two interfaces: $\tilde{\Gamma}_h$ and $\mathcal{E}^0_{c,h}$, and one boundary: $\hat{\Gamma}_h$. Note that the two interfaces of the WSBM correspond to the interface required in CutFEM for the ghost penalty stabilization, or $\mathcal{E}^0_h = \tilde{\Gamma}_h \cup \mathcal{E}^0_{c,h}$. For summary, the naming conventions used throughout this work are displayed in Figure 3.6

Then, for the method-specific step **I6**, the contribution from the cut cells is weighted by the volume fraction; the small cut cell problem can lead to stability issues. For this reason, the same ghost penalty technique [123] used in the CutFEM approach is used here. This will be enforced on the discretized interface $\mathcal{E}^0_h$. Additionally, the WSBM introduces weight functions to the test functions, which require a modification to the finite element spaces. Lastly, as it is an SBM-type method, we again require modifying the weak form to shift the boundary and interface conditions. The derivations for these shifted boundaries are equivalent to those of the SBM as defined in the previous section. Some minor changes are made on the interfaces, which will be explained. Let us start from the Laplacian in the continuous weak form defined over some domain Ω.

$$(\Delta\phi, w)_\Omega = 0. \tag{3.34}$$

After discretization, this should be approximated by the sum of integrals over all elements.

$$(\Delta\phi, w_h)_\Omega = \sum_\kappa (\Delta\phi, w_h)_\kappa \simeq \sum_\kappa (\alpha_\kappa \Delta\phi, w_h)_\kappa \tag{3.35}$$

where α_κ is the volume fraction of an element κ that is defined as

$$\alpha_\kappa := \frac{\text{meas}(\kappa_{\text{cut}} \cup \kappa)}{\text{meas}(\kappa)}. \tag{3.36}$$

The subscript cut denotes the cut mesh region of an element κ. The method assumes that for a given element κ, the following holds

$$\int_{\kappa_{\text{cut}}} f \, d\kappa \simeq \int_\kappa \alpha_\kappa f \, d\kappa. \tag{3.37}$$

Then, integration by parts is applied to simplify the Laplacian.

$$(\Delta\phi, w_h)_\Omega = \sum_\kappa [(\alpha_\kappa \nabla\phi, \nabla w_h)_\kappa - \sum_{\partial\kappa} (\alpha_\kappa \nabla\phi \cdot \boldsymbol{n}, w_h)_{\partial\kappa}]. \tag{3.38}$$

The test function w_h is redefined using the approximation $w_{h,\alpha} = \alpha w_h$ with $w_{h,\alpha} \in \mathcal{W}_{h,\alpha} := \{w_h \in \mathcal{W}_h \,|\, w_{h,\alpha} = \alpha_\kappa w_h \,\forall\kappa\}$. So, we are left with a Petrov-Galerkin discretization as the test function $w_{h,\alpha}$ is discontinuous. Thereby Equation 3.38 can be rewritten to

$$(\Delta\phi, w_{h,\alpha})_\Omega = (\nabla\phi, \nabla w_{h,\alpha})_\Omega - (\{\nabla\phi\}, [\![w_{h,\alpha}]\!])_{\partial\Omega} - ([\![\nabla\phi]\!], \{w_{h,\alpha}\})_{\partial\Omega} \tag{3.39}$$

denoting that $\partial\Omega$ is the boundary surrounding the domain Ω. Here, operators are defined as: the mean operator $\{\cdot\} = \frac{1}{2}(\cdot)^+ + \frac{1}{2}(\cdot)^-$ and the jump operator $[\![\cdot]\!] = (\cdot)^+ \boldsymbol{n}^+ + (\cdot)^- \boldsymbol{n}^-$, where the superscripts + and − denote both sides of an edge. This jump operator is the same as the one for the ghost penalty stabilization. In this work, we want to enforce the gradient of ϕ to be continuous, and therefore we will enforce that $[\![\nabla\phi]\!] = 0$, which results in the last term of Equation 3.39 dropping out. The weak form of the Laplacian is then given by

$$(\Delta\phi, w_{h,\alpha})_\Omega = (\nabla\phi, \nabla w_{h,\alpha})_\Omega - (\{\nabla\phi\}, [\![w_{h,\alpha}]\!])_{\partial\Omega} \tag{3.40}$$

Note that for the problems considered in this work we strictly enforce $[\![\nabla\phi]\!] = 0$ and therefore in the shifted terms as derived for the SBM we will apply the jump operator. Specifically, the jump condition is enforced on the last three terms of Equation 3.28.

The test function w_h is replaced by the weighted test function $w_{h,\alpha}$ in domain $\Omega_{c,h}$, the governing equation of the object is shifted to the exterior surrogate boundary $\hat{\Gamma}_h$ and the shifted Neumann-type boundary is applied to the exterior surrogate boundary $\hat{\Gamma}_h$, the interior surrogate interface $\tilde{\Gamma}_h$ and to the interface in the cut domain $\mathcal{E}^0_{c,h}$. Lastly, ghost penalty stabilization is applied to the interface of $\mathcal{E}^0_{c,h}$ and of the interior surrogate interface $\tilde{\Gamma}_h$. Reiterating, that for the weak formulations we define the aforementioned interfaces as $\mathcal{E}^0_h = \tilde{\Gamma}_h \cup \mathcal{E}^0_{c,h}$. Let us formally introduce the discrete space as

$$\mathcal{W}^{\text{act}}_{h,\alpha} := \{w_h \in \mathcal{W}^{\text{act}}_h \,|\, w_{h,\alpha} = \alpha_\kappa w_h \,\forall\kappa\} \tag{3.41}$$

Time-domain semi-discrete weak form

Then the semi-discrete weak form is stated as:

find $[\phi_h, \mathbf{u}_h] \in L^2(0,T;\mathcal{W}^{\text{act}}_{h,\alpha}) \times L^2(0,T;\mathcal{V}_h)$ such that

$$a^{\text{wsbm}}_h([\phi_h, \mathbf{u}_h], [w_h, \mathbf{v}_h]) = b_h([w_h, \mathbf{v}_h]) \qquad \forall [w_h, v_h] \in \mathcal{W}^{\text{act}}_{h,\alpha} \times \mathcal{V}_h. \tag{3.42}$$

with the bilinear form given as

$$
\begin{aligned}
a_h^{\text{wsbm}}((\phi_h,\mathbf{u}_h),(w_h,\mathbf{v}_h)) := {} & (\nabla w_h,\nabla\phi_h)_{\tilde{\Omega}_h} + (\nabla w_{h,\alpha},\nabla\phi_h)_{\Omega_{c,h}} + (w_{h,\alpha},\frac{1}{g}\frac{\partial^2\phi_h}{\partial t^2})_{\hat{\Gamma}_f,h} + \\
& (\mathbf{v}_h,\frac{\boldsymbol{M}}{J_\Gamma\rho\left|\hat{\Gamma}_h\right|}\frac{\partial^2\mathbf{u}_h}{\partial t^2} + \left(\frac{\partial\phi_h}{\partial t} + \nabla(\frac{\partial\phi_h}{\partial t})\cdot\mathbf{d} + g(\mathbf{u}_h\cdot\boldsymbol{n}_z)\right)\boldsymbol{n}_{\Gamma,j})_{\hat{\Gamma}_h} - \\
& (w_{h,\alpha},\nabla\phi_h\cdot\hat{\boldsymbol{n}}_\Gamma)_{\hat{\Gamma}_h} - (w_{h,\alpha},\frac{\partial\mathbf{u}_h}{\partial t}\cdot(\hat{\boldsymbol{n}}_\Gamma\cdot\boldsymbol{n}_\Gamma)\boldsymbol{n}_{\Gamma,j})_{\hat{\Gamma}_h} + \\
& (w_{h,\alpha},\nabla\phi_h\cdot(\hat{\boldsymbol{n}}_\Gamma\cdot\boldsymbol{n}_\Gamma)\boldsymbol{n}_\Gamma)_{\hat{\Gamma}_h} + (w_{h,\alpha},(\nabla(\nabla\phi_h)\cdot\boldsymbol{d})\cdot(\hat{\boldsymbol{n}}_\Gamma\cdot\boldsymbol{n}_\Gamma)\boldsymbol{n}_\Gamma)_{\hat{\Gamma}_h} - \\
& \langle[\![w_{h,\alpha}]\!],\{\nabla\phi_h\}\rangle_{\mathcal{E}_h^0} - \langle\widetilde{[\![w_{h,\alpha}]\!]}_j,\{\frac{\partial\mathbf{u}_h}{\partial t}\}\rangle_{\mathcal{E}_h^0} + \\
& \langle\widetilde{[\![w_{h,\alpha}]\!]},\{\nabla\phi_h\}\rangle_{\mathcal{E}_h^0} + \langle\widetilde{[\![w_{h,\alpha}]\!]},\{\nabla(\nabla\phi_h)\cdot\boldsymbol{d}\}\rangle_{\mathcal{E}_h^0} - \\
& \langle\{w_{h,\alpha}\},\widetilde{[\![\frac{\partial\mathbf{u}_h}{\partial t}]\!]}_j\rangle_{\mathcal{E}_h^0} + \\
& \langle\{w_{h,\alpha}\},\widetilde{[\![\nabla\phi_h]\!]}\rangle_{\mathcal{E}_h^0} + \langle\{w_{h,\alpha}\},\widetilde{[\![\nabla(\nabla\phi_h)\cdot\boldsymbol{d}]\!]}\rangle_{\mathcal{E}_h^0} + \\
& \sum_{i=1}^{p_e}\langle[\![\nabla^{(i)}w_h]\!],\gamma_g h_e^{(2i+1)}[\![\nabla^{(i)}\phi_h]\!]\rangle_{\mathcal{E}_h^0}
\end{aligned}
\tag{3.43}
$$

Here, we introduce a new notation only applicable on the interfaces of $\mathcal{E}_h^0$, we define $\widetilde{[\![w_{h,\alpha}]\!]} = (w_{h,\alpha})^+(\tilde{\boldsymbol{n}}^+_{\mathcal{E}_h^0}\cdot\boldsymbol{n}_\Gamma)\boldsymbol{n}_\Gamma + (w_{h,\alpha})^-(\tilde{\boldsymbol{n}}^-_{\mathcal{E}_h^0}\cdot\boldsymbol{n}_\Gamma)\boldsymbol{n}_\Gamma$ with $\boldsymbol{n}_\Gamma$ being the true normal vector oriented to the physical boundary Γ_h and $\tilde{\boldsymbol{n}}_{\mathcal{E}_h^0}$ being the normal vector with respect to the interface defined by $\mathcal{E}_h^0$. Similarly, for the one term that has a dependency on $\mathbf{u}_h$ we introduce a similar notation with a subscript j as it has a dependency on the number of physical degrees of freedom. Formally, we define $\widetilde{[\![w_{h,\alpha}]\!]}_j = (w_{h,\alpha})^+(\tilde{\boldsymbol{n}}^+_{\mathcal{E}_h^0}\cdot\boldsymbol{n}_\Gamma)\boldsymbol{n}_{\Gamma,j} + (w_{h,\alpha})^-(\tilde{\boldsymbol{n}}^-_{\mathcal{E}_h^0}\cdot\boldsymbol{n}_\Gamma)\boldsymbol{n}_{\Gamma,j}$. The only difference with respect to the other shifted jump operator is that the true normal outside the normal product is given by $\boldsymbol{n}_{\Gamma,j}$.

Briefly discussing this weak form and comparing it to that of the SBM. We see that the first four rows correspond well to the semi-discrete weak form of SBM, see Equation 3.31. The difference is the definition on the outer surrogate boundary $\hat{\Gamma}_h$ instead of the (inner) surrogate boundary $\tilde{\Gamma}_h$ of the SBM. The bottom three rows of equations are added, which are all defined on the interface $\mathcal{E}_h^0$. On the bottom row is the ghost penalty stabilization, similar to the CutFEM case. The remaining terms on the fifth and sixth row correspond to the shifting operators applied to the interface $\mathcal{E}_h^0$ given the definitions introduced in this section.

Frequency-domain discrete weak form

For completeness, we also introduce a frequency-domain discrete weak form, although this formulation will not be used in the case studies in later chapters.

find $[\phi_h,\mathbf{u}_h]\in\mathcal{W}^{\text{act}}_{h,\alpha}\times\mathcal{V}_h$ such that

$$
a^{\text{wsbm}}_{\omega,h}([\phi_h,\mathbf{u}_h],[w_h,\mathbf{v}_h]) = b_{\omega,h}([w_h,\mathbf{v}_h]) \qquad \forall[w_h,\mathbf{v}_h]\in\mathcal{W}^{\text{act}}_{h,\alpha}\times\mathcal{V}_h, \tag{3.44}
$$

with the bilinear form given by:

$$
\begin{aligned}
a^{\text{wsbm}}_{\omega,h}([\phi_h,\mathbf{u}_h],[w_h,\mathbf{v}_h]) := {} & (\nabla w_h,\nabla\phi_h)_{\tilde{\Omega}_h} + \left(\nabla w_{h,\alpha},\nabla\phi_h\right)_{\Omega_{\text{c},h}} - \frac{\omega^2}{g}\left(w_{h,\alpha},\phi_h\right)_{\hat{\Gamma}_{\text{f},h}} - \\
& \omega^2\frac{\boldsymbol{M}_\rho}{J_\Gamma\left|\hat{\Gamma}_h\right|}(\mathbf{v}_h,\mathbf{u}_h)_{\hat{\Gamma}_h} + g\left(\mathbf{v}_h,(\mathbf{u}_h\cdot\boldsymbol{n}_z)\,\boldsymbol{n}_{\Gamma,j}\right)_{\hat{\Gamma}_h} - \\
& i\omega\left(\mathbf{v}_h,(\phi_h+\nabla\phi_h\cdot\mathbf{d})\boldsymbol{n}_{\Gamma,j}\right)_{\hat{\Gamma}_h} - \\
& \left(w_{h,\alpha},\nabla\phi_h\cdot\hat{\boldsymbol{n}}_\Gamma\right)_{\hat{\Gamma}_h} - i\omega\left(w_{h,\alpha},\mathbf{u}_h\cdot(\boldsymbol{n}_\Gamma\cdot\hat{\boldsymbol{n}}_\Gamma)\boldsymbol{n}_{\Gamma,j}\right)_{\hat{\Gamma}_h} + \\
& \left(w_{h,\alpha},(\boldsymbol{n}_\Gamma\cdot\hat{\boldsymbol{n}}_\Gamma)\boldsymbol{n}_\Gamma\cdot(\nabla(\nabla\phi_h)\cdot\mathbf{d}+\nabla\phi_h)\right)_{\hat{\Gamma}_h} - \\
& \langle\widetilde{[\![w_{h,\alpha}]\!]},\{\nabla\phi_h\}\rangle_{\mathcal{E}^0_h} - i\omega\langle\widetilde{[\![w_{h,\alpha}]\!]}_j,\{\mathbf{u}_h\}\rangle_{\mathcal{E}^0_h} + \\
& \langle\widetilde{[\![w_{h,\alpha}]\!]},\{\nabla(\nabla\phi_h)\cdot\mathbf{d}+\nabla\phi_h\}\rangle_{\mathcal{E}^0_h} - \\
& i\omega\langle\{w_{h,\alpha}\},\widetilde{[\![\mathbf{u}_h]\!]}_j\rangle_{\mathcal{E}^0_h} + \\
& \langle\{w_{h,\alpha}\},[\![\nabla(\widetilde{\nabla\phi_h)\cdot\mathbf{d}}+\nabla\phi_h]\!]\rangle_{\mathcal{E}^0_h} + \\
& \sum_{i=1}^{p_e}\langle[\![\nabla^{(i)}w_h]\!],\gamma_g h_e^{(2i+1)}[\![\nabla^{(i)}\phi_h]\!]\rangle_{\mathcal{E}^0_h}.
\end{aligned}
\tag{3.45}
$$

3.3 Conclusions

In this chapter, a brief introduction is provided to the finite element method. The discrete weak forms of both time and frequency domain problems are displayed for a body-fitted finite element formulation. Then, a historical overview of unfitted methods is presented, covering the motivation for their development and key milestones over the years. The main challenges for unfitted methods are presented as follows: integration over cut cells, stability issues due to small cut cells, and difficulties in the consistent imposition of boundary conditions. The justification is made to exclude the Finite Cell Method and the Trace Finite Element Method from this work and to continue with the Cut Finite Element Method, Aggregated unfitted Finite Element Method, Shifted Boundary Method, and Weighted Shifted Boundary Method. Afterwards, the commonalities and differences between the unfitted methods are discussed, and the common definitions used throughout this work are described. Then, the four considered unfitted methods are introduced, highlighting details about their workings and the modifications required concerning the traditional body-fitted finite element method. All methods require modifications to the computational domain and modifications to the weak forms. For AgFEM, SBM, and WSBM, the potential flow formulation is novel in itself. Although AgFEM and SBM have been applied to the Poisson problem in the literature [77, 125], the inclusion of the free surface boundary conditions is novel. The WSBM has only been applied to the Stokes and Navier-Stokes problems; therefore, this formulation is entirely novel. Additionally, within the context of a potential flow fluid, the coupling to rigid structures is an entirely novel application for these unfitted methods. An overview of the discrete bilinear forms for each problem is depicted in Table 3.4 for cross-referencing.

Table 3.4: Overview of the bilinear forms of the semi-discrete and discrete weak forms for the time- and frequency-domain representations and their locations within this chapter.

Method	**Section**	**Time-domain**	**Frequency-domain**
Body-fitted	subsection 3.1.1	Equation 3.4	Equation 3.7
CutFEM	subsection 3.2.3	Equation 3.13	Equation 3.15
AgFEM	subsection 3.2.4	Equation 3.18	Equation 3.20
SBM	subsection 3.2.5	Equation 3.31	Equation 3.33
WSBM	subsection 3.2.6	Equation 3.43	Equation 3.45

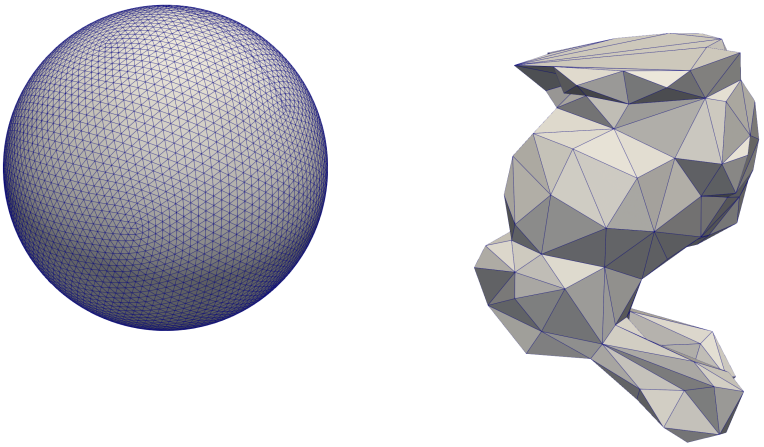
4

Benchmarking Unfitted Finite Element Methods

Benchmarking is crucial in this study as we consider several unfitted finite element methods that allow meshes to be non-conforming to geometric boundaries. This flexibility benefits complex or changing geometries, dynamic interfaces, and multiphysics problems, but it presents numerical issues such as small cut cells, numerical stability, and integration problems. Stabilization strategies are often necessary, and their success varies based on the specific problem and parameter choices. Systematic benchmarking rigorously evaluates these methods against known analytical or high-fidelity solutions, examining their accuracy, stability, and computational costs. Comparing different unfitted method variants reveals their strengths and weaknesses, aiding practitioners in choosing suitable methods for their cases of interest and underscoring the contributions of this work. Transparent benchmarking also supplies reference results for the wider research community. The novelty in this chapter is the comparison of computational performance between the four unfitted methods. The comparison is assumed to be fair, as all methods are implemented in a unified computational framework. Additionally, a discussion is presented based on both results and theory, allowing others to gain insight into the advantages and disadvantages of each method. The benchmark cases considered in this work are specifically tailored with the concept of offshore floating structures in mind; hence, for other applications or a more generalized benchmarking effort, this work can serve as a suitable starting point. In this chapter, the method of manufactured solutions is applied to verify each unfitted method by comparing exact and approximate solutions on successively refined meshes for varying polynomial orders to evaluate numerical accuracy. For stability, the condition numbers across methods are compared. Computational performance is compared based on time-to-solution and the number of allocations required, with nine categories across unfitted methods considered for discussion. Lastly, a discussion is presented on the implementation complexity, which is based on the theory of each unfitted method and the challenges encountered during implementation.

4.1 Test Cases: Definitions and Geometries

In this benchmarking study, we consider three geometries; however, we have four test cases. Firstly, we consider a case in 2D with a square domain that is intersected at the top by a horizontal cylinder, which is represented implicitly using a level set function. Secondly, we extend the study to 3D and consider a cubic domain that is intersected at the top by a sphere, which is represented implicitly using a level set function. Thirdly, the same case study is conducted, except now the sphere is represented using a stereolithographic CAD file format, or STL, meaning the difference is the transition from implicit geometry representation to explicit. Fourthly, we consider the low-poly Stanford bunny to be a complex, explicitly defined geometry. The two STL geometries in their corresponding orientations are depicted in Figure 4.1, both of which intersect the cubic domain at the top surface.



(a) Sphere STL. (b) Low poly Stanford bunny STL, downward oriented.

Figure 4.1: Two explicit geometrical representations (STLs) for the test cases, on the left the sphere, and on the right the low poly Stanford bunny mirrored across the horizontal plane.

Domain and Governing Equation

We consider the domain of the first test, as depicted in Figure 4.2. A square domain with a length and height of $L_x = 1.0$ m, and an intersection with a horizontal cylinder through the top of the domain. The cylinder has a radius $R = 0.25$ m. There are three boundaries present: the object boundary Γ_1, the top surface Γ_2, and the remaining boundaries Γ_3, which correspond to the left, right, and bottom. On the boundaries Γ_1 and Γ_2, we enforce Neumann type boundary conditions, and on Γ_3, we enforce a Dirichlet type boundary condition. This choice ensures that we maintain the Neumann type boundaries on the top surface and on the structure, as is done throughout this dissertation. For the other test cases, the domain remains similar: all side boundaries except for the top of the domain are denoted by Γ_3, the top surface is denoted by Γ_2 except where it intersects with the geometrical boundary, which is represented by Γ_1. For the 3D studies, we introduce an

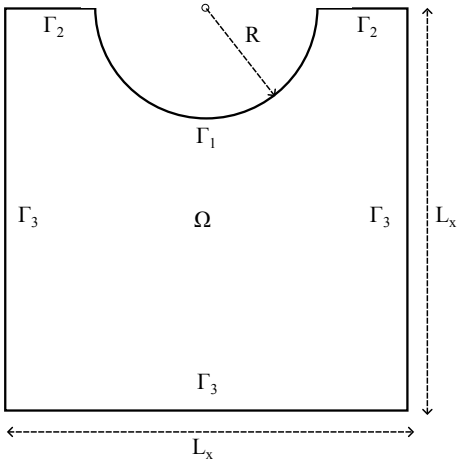


Figure 4.2: Domain Ω for the 2D benchmarking study with boundaries Γ_1, Γ_2, and Γ_3. The dimensions corresponding to a square domain of length $L_x = 1.0$ m and intersected circle at the top with radius $R = 0.25$ m.

additional dimension of size $L_x = 1.0$ m and the additional boundaries are part of boundary Γ_3. Note that for all test cases, we intersect the geometry with the top boundary Γ_2 of the domain Ω.

For this study, we use the method of manufactured solutions; we predefine an analytical solution that is enforced through a forcing term. Note that this chapter is devoid of physics, and the problems considered are completely fabricated. We select the governing equation and analytical solutions such that they are closely related to the problems solved in the later chapters in which the physics do matter. The governing equation for a linear potential flow problem is given by the Laplacian set equal to zero. Therefore, we select a Laplacian as the governing equation in Equation 4.1, and we introduce the forcing term $f(x, z, t)$ on the right-hand side.

$$\Delta\phi = f(x, z, t) \quad \text{in} \quad \Omega \tag{4.1}$$

The analytical solution is set as stated in Equation 4.2.

$$\phi^*(x, z, t) = \frac{g\eta_0}{\omega} \frac{\cosh(k(z+d)}{\cosh(kd)} \sin(kx - \omega t) \tag{4.2}$$

Here, ω is the wave frequency which is derived via the dispersion relation $\omega = \sqrt{gk\tanh(kd)}$ with $g = 9.81$ m/s^2, $\eta_0 = 0.05$ m is the wave amplitude, k is the wave number that is set to $k = \frac{2\pi}{L_x}$ and $d = L_x$ is the depth.

In the 3D test cases, the governing equation and the solution are defined as Equation 4.3 and Equation 4.4, respectively.

$$\Delta\phi = f(x, y, z, t) \quad \text{in} \quad \Omega \tag{4.3}$$

$$\phi^*(x,y,z,t) = \frac{g\eta_0}{\omega}\frac{\cosh(k(z+d)}{\cosh(kd)}\sin(kx+ky-\omega t) \tag{4.4}$$

Discrete Weak Formulation

Given these governing equations, we can formulate the discrete weak form of the benchmarking case studies as follows: find $\phi_h \in \mathcal{W}_h$ such that

$$a_h(\phi_h, w_h) = b_h(w_h) \qquad \forall w_h \in \mathcal{W}_h, \tag{4.5}$$

with

$$a_h(\phi_h, w_h) := (\nabla w_h, \nabla \phi_h)_{\Omega_h}, \tag{4.6}$$

and

$$b_h(w_h) := (w_h, f)_{\Omega_h} + (w_h, \nabla\phi^* \cdot \boldsymbol{n}_1)_{\Gamma_{1,h}} + (w_h, \nabla\phi^* \cdot \boldsymbol{n}_2)_{\Gamma_{2,h}}. \tag{4.7}$$

With the discrete space $\mathcal{W}_h$ defined as in chapter 3 and enforcing a Dirichlet type boundary conditions on Γ_3 where we set $\phi = \phi^*$ on Γ_3 in a strong sense. Reiterating that Neumann type boundary conditions are enforced on Γ_1 and Γ_2.

We will now briefly state the discrete weak forms required for each unfitted method. The derivations, which are similar to those described in chapter 3, are presented in Appendix A for completeness. The definitions of the discrete spaces $\mathcal{W}_h^{\text{act}}$, $\mathcal{W}_h^{\text{ag}}$, and $\mathcal{W}_h^{\text{in}}$ are all equal to those defined in chapter 3.

CutFEM Weak Form

find $\phi_h \in \mathcal{W}_h^{\text{act}}$ such that

$$a_h^{\text{cut}}(\phi_h, w_h) = b_h^{\text{cut}}(w_h) \qquad \forall w_h \in \mathcal{W}_h^{\text{act}}, \tag{4.8}$$

with

$$\begin{aligned} a_h^{\text{cut}}(\phi_h, w_h) := &(\nabla w_h, \nabla \phi_h)_{\Omega_{\text{cut}}} + \\ &\sum_{i=1}^{p} \langle [\![\nabla^{(i)} w_h \cdot \boldsymbol{n}_{\mathcal{E}_h^0}]\!], \gamma_g h_e^{(2i+1)} [\![\nabla^{(i)} \phi_h \cdot \boldsymbol{n}_{\mathcal{E}_h^0}]\!] \rangle_{\mathcal{E}_h^0}, \end{aligned} \tag{4.9}$$

and

$$b_h^{\text{cut}}(w_h) := (w_h, f)_{\Omega_{\text{cut}}} + (w_h, \nabla\phi^* \cdot \boldsymbol{n}_1)_{\Gamma_{1,h}} + (w_h, \nabla\phi^* \cdot \boldsymbol{n}_2)_{\Gamma_{2,h}}. \tag{4.10}$$

AgFEM Weak Form

find $\phi_h \in \mathcal{W}_h^{\text{ag}}$ such that

$$a_h^{\text{ag}}(\phi_h, w_h) = b_h^{\text{ag}}(w_h) \qquad \forall w_h \in \mathcal{W}_h^{\text{ag}}, \tag{4.11}$$

with

$$a_h^{\text{ag}}(\phi_h, w_h) := (\nabla w_h, \nabla \phi_h)_{\Omega_{\text{cut}}}, \tag{4.12}$$

and

$$b_h^{\text{ag}}(w_h) := (w_h, f)_{\Omega_{\text{cut}}} + (w_h, \nabla\phi^* \cdot \boldsymbol{n}_1)_{\Gamma_{1,h}} + (w_h, \nabla\phi^* \cdot \boldsymbol{n}_2)_{\Gamma_{2,h}}. \tag{4.13}$$

SBM Weak Form
find $\phi_h \in \mathcal{W}_h^{\text{in}}$ such that

$$a_h^{\text{sbm}}(\phi_h, w_h) = b_h^{\text{sbm}}(w_h) \qquad \forall w_h \in \mathcal{W}_h^{\text{in}}, \tag{4.14}$$

with

$$\begin{aligned} a_h^{\text{sbm}}(\phi_h, w_h) :=& (\nabla w_h, \nabla \phi_h)_{\tilde{\Omega}_h} - (w_h, \nabla \phi_h \cdot \tilde{\boldsymbol{n}}_1)_{\tilde{\Gamma}_{1,h}} + \\ & (w_h, (\boldsymbol{n}_1 \cdot \tilde{\boldsymbol{n}}_1)\boldsymbol{n}_1 \cdot [\nabla(\nabla \phi_h) \cdot \mathbf{d} + \nabla \phi_h])_{\tilde{\Gamma}_{1,h}}, \end{aligned} \tag{4.15}$$

and

$$\begin{aligned} b_h^{\text{sbm}}(w_h) :=& (w_h, f)_{\tilde{\Omega}_h} + (w_h, \nabla \phi^* \cdot \boldsymbol{n}_2)_{\tilde{\Gamma}_{2,h}} + \\ & (w_h, \nabla \phi^* \cdot (\boldsymbol{n}_1 \cdot \tilde{\boldsymbol{n}}_1)\boldsymbol{n}_1)_{\tilde{\Gamma}_{1,h}}. \end{aligned} \tag{4.16}$$

WSBM Weak Form
find $\phi_h \in \mathcal{W}_h^{\text{act}}$ such that

$$a_h^{\text{wsbm}}(\phi_h, w_h) = b_h^{\text{wsbm}}(w_h) \qquad \forall w_h \in \mathcal{W}_h^{\text{act}}, \tag{4.17}$$

with

$$\begin{aligned} a_h^{\text{wsbm}}(\phi_h, w_h) :=& (\nabla w_h, \nabla \phi_h)_{\tilde{\Omega}_h} + \left(\nabla w_{h,\alpha}, \nabla \phi_h\right)_{\Omega_{\text{c},h}} - \left(w_{h,\alpha}, \nabla \phi_h \cdot \hat{\boldsymbol{n}}_1\right)_{\hat{\Gamma}_{1,h}} + \\ & \left(w_{h,\alpha}, (\boldsymbol{n}_1 \cdot \hat{\boldsymbol{n}}_1)\boldsymbol{n}_1 \cdot [\nabla(\nabla \phi_h) \cdot \mathbf{d} + \nabla \phi_h]\right)_{\hat{\Gamma}_{1,h}} - \\ & \langle [\![w_{h,\alpha}]\!], \{\nabla \phi_h\} \rangle_{\mathcal{E}_h^0} + \\ & \langle \widetilde{[\![w_{h,\alpha}]\!]}, \{\nabla(\nabla \phi_h) \cdot \mathbf{d} + \nabla \phi_h\} \rangle_{\mathcal{E}_h^0} + \\ & \langle \{w_{h,\alpha}\}, [\![\nabla(\widetilde{\nabla \phi_h) \cdot \mathbf{d}} + \nabla \phi_h]\!] \rangle_{\mathcal{E}_h^0} + \\ & \sum_{i=1}^{p} \langle [\![\nabla^{(i)} w_h]\!], \gamma_g h_e^{(2i+1)} [\![\nabla^{(i)} \phi_h]\!] \rangle_{\mathcal{E}_h^0}, \end{aligned} \tag{4.18}$$

and

$$\begin{aligned} b_h^{\text{wsbm}}(w_h) :=& (w_h, f)_{\tilde{\Omega}_h} + \left(w_{h,\alpha}, f\right)_{\Omega_{\text{c},h}} + \left(w_{h,\alpha}, \nabla \phi^* \cdot \hat{\boldsymbol{n}}_2\right)_{\hat{\Gamma}_{2,h}} + \\ & \left(w_{h,\alpha}, \nabla \phi^* \cdot (\boldsymbol{n}_1 \cdot \hat{\boldsymbol{n}}_1)\boldsymbol{n}_1\right)_{\hat{\Gamma}_{1,h}} + \\ & \langle \widetilde{[\![w_{h,\alpha}]\!]}, \{\nabla \phi^*\} \rangle_{\mathcal{E}_h^0} + \langle \{w_{h,\alpha}\}, \widetilde{[\![\nabla \phi^*]\!]} \rangle_{\mathcal{E}_h^0}. \end{aligned} \tag{4.19}$$

Notation for the Weak Forms
Here, we consider $\Omega_{\text{cut},h}$, $\Gamma_{1,h}$, and $\Gamma_{2,h}$ to be the domain and boundaries that are algorithmically tessellated to fit the physical domain Ω and boundaries Γ_1 and Γ_2. The active domain is denoted by $\Omega_{\text{act},h}$ and the surrogate domain is denoted by $\tilde{\Omega}_h$; we re-use the notation $\Omega_{\text{c},h}$ for the domain consisting only of cut elements. The inner and outer surrogate boundaries

are denoted by $\tilde{\Gamma}_h$ and $\hat{\Gamma}_h$, respectively. The interface required for CutFEM and WSBM is denoted by $\mathcal{E}_h^0$.

There is quite a bit of notation introduced for the normal vectors, especially with the SBM-type formulations. We make a distinction here between the true normal vector $\boldsymbol{n}_1$, which is orientated toward the circle, and the surrogate normal vectors $\tilde{\boldsymbol{n}}_1$ and $\hat{\boldsymbol{n}}_1$, which are normal vectors with respect to the surrogate boundaries on which they are defined. For reference, the reader is referred to Figure 3.2d. The normal vector defined on the interface is denoted by $\boldsymbol{n}$ for the true normal vector and $\tilde{\boldsymbol{n}}$ for the surrogate normal vector.

Error and Condition Number Calculation

To assess the accuracy at each grid resolution, the L2 norm of the error $\|\varepsilon\|_2$ is calculated by Equation 4.20. The error norms for each method are calculated in the surrogate domain $\tilde{\Omega}$, as this domain is present for all four methods.

$$\|\varepsilon\|_2 = \sqrt{\int (\phi - \phi_h)^2 d\Omega_{\text{in}}} \tag{4.20}$$

The condition number κ is calculated using the `cond` function, available in the Julia programming language from the `LinearAlgebra` package. Formally, it is calculated as

$$\kappa(\mathbb{A}) = \|\mathbb{A}\|_p \|\mathbb{A}^{-1}\|_p, \tag{4.21}$$

where $\|\cdot\|_p$ is the p-norm, and $\mathbb{A}$ is the linear system matrix. Although the default function is set to calculate the condition number with the 2-norm, we use the 1-norm in this study due to computational constraints.

4.2 Numerical Conditioning

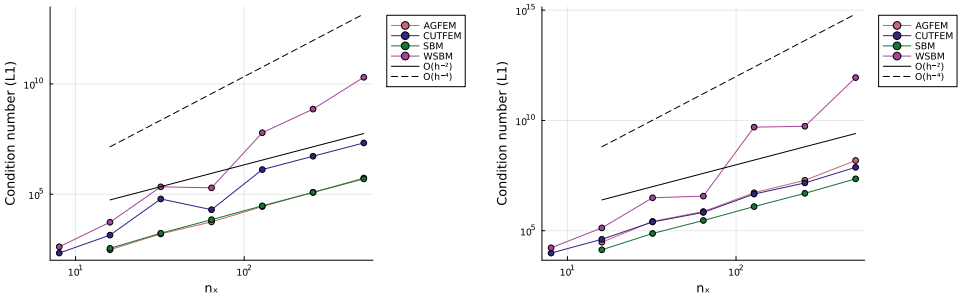
In this section, we inspect the condition number κ of the system matrix, which governs its practical numerical behavior. The condition number of a matrix intuitively indicates its proximity to singularity. A high condition number usually means that numerical algorithms for inverting the matrix become increasingly inaccurate. Such matrices correspond to linear transformations where minor perturbations in the input lead to significant variations in the output vector's norm, which is typically undesirable. As discussed before, it is expected that a large fraction of small cut cells will increase the condition number; therefore, it is anticipated that SBM will have the lowest condition numbers for all test cases. AgFEM also removes all ill-posed degrees of freedom and therefore should not have large condition numbers. Both CutFEM and WSBM apply ghost penalty stabilization on the region where small cut cells can exist, and the choice of the penalty parameter will influence the condition number. For the CutFEM cases in both 2D and 3D we will continue given the standard value of $\gamma_g = 0.1$, as applied in the original works [122]. For WSBM, there is no available standard value, and it is empirically decided that for 2D we choose $\gamma_g = 0.0$ or no ghost penalty stabilization, and for 3D $\gamma_g = 10.0$. The results from the sensitivity analysis and justification is presented in Appendix B.

Figure 4.3 shows the condition number κ as a function of the number of elements n for the 2D cylindrical geometry. For first-order elements ($p_e = 1$), AgFEM and SBM yield the smallest condition numbers, with CutFEM slightly larger and exhibiting mild oscillations

attributable to small cut cells near the immersed boundary. WSBM produces the largest condition numbers and similarly shows oscillations, consistent with the presence of small weights in the surrogate boundary formulation. At second order ($p_e = 2$), SBM retains the smallest condition numbers while WSBM grows significantly larger with refinement, diverging from the other methods. Crucially, WSBM is the only method in the 2D case that does not recover the theoretically expected $\mathcal{O}(h^{-2})$ scaling, instead exhibiting steeper growth. As discussed in Appendix B, no tested value of the ghost penalty parameter γ_g recovers this scaling for 2D WSBM, suggesting that the ghost penalty stabilization does not effectively control the strip between the true and surrogate boundaries in this configuration.

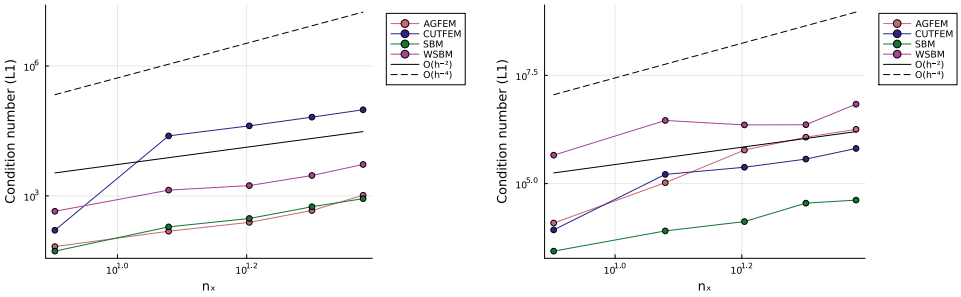
For the 3D test cases — the implicit sphere (Figure 4.4) and the explicit sphere (Figure 4.5) — the two geometries produce qualitatively similar condition number behavior, indicating that the representation of the geometry (implicit vs. explicit) has little influence on conditioning for smooth surfaces. At $p_e = 1$, CutFEM yields the largest condition numbers, while AgFEM and SBM share the smallest, with WSBM intermediate. At $p_e = 2$, AgFEM grows to become comparable to CutFEM, WSBM produces the largest condition numbers, and SBM retains the smallest. Compared to the 2D case, all methods in 3D show condition number scaling closer to $\mathcal{O}(h^{-2})$, consistent with the ghost penalty stabilization being more effective in 3D as discussed in Appendix B.

The explicit low-poly Stanford bunny geometry (Figure 4.6) introduces additional geometric complexity and yields some notable differences relative to the sphere cases. For WSBM, the coarsest mesh ($n = 8$) could not represent the outer surrogate boundary and is therefore excluded. At $p_e = 1$, AgFEM produces the smallest condition numbers, followed by WSBM, SBM, and CutFEM. However, at $p_e = 2$ this ranking changes markedly: AgFEM now yields the largest condition numbers, with CutFEM producing the smallest. This reversal is likely attributable to the geometric complexity of the bunny surface, which introduces more irregular cut configurations and a more variable surrogate boundary than the sphere geometries. AgFEM's agglomeration strategy, which performs well for smooth geometries at higher order, may be less effective when element agglomeration is constrained by the complex topology of the bunny surface, leading to larger condition numbers at $p_e = 2$. The sensitivity of condition number rankings to geometry highlights that no single unfitted method dominates across all configurations and that the choice of method should account for the complexity of the immersed geometry.



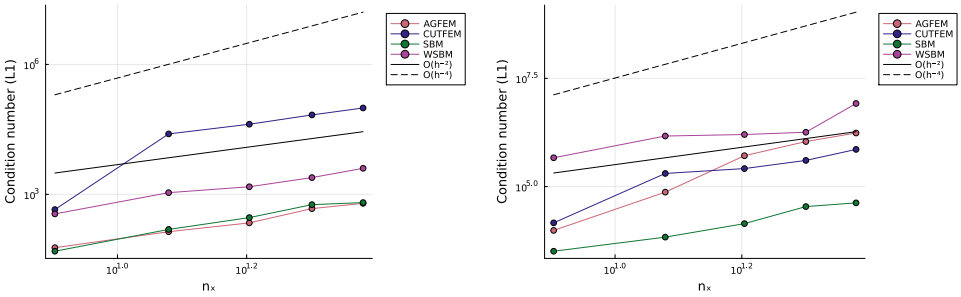
(a) Order $p_e = 1$ (b) Order $p_e = 2$

Figure 4.3: Condition number study for varying grid resolutions via method of manufactured solutions for four types of embedded methods with varying orders for the 2D cylinder implicit geometry.



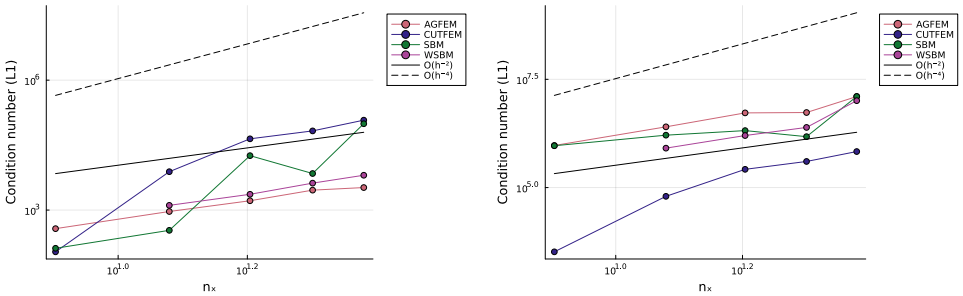
(a) Order $p_e = 1$ (b) Order $p_e = 2$

Figure 4.4: Condition number study for varying grid resolutions via method of manufactured solutions for four types of embedded methods with varying orders for the 3D sphere implicit geometry.



(a) Order $p_e = 1$ (b) Order $p_e = 2$

Figure 4.5: Condition number study for varying grid resolutions via method of manufactured solutions for four types of embedded methods with varying orders for the 3D sphere explicit geometry.



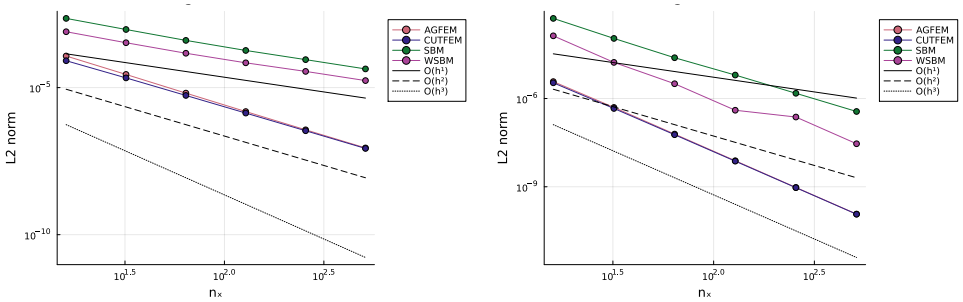
(a) Order $p_e = 1$ (b) Order $p_e = 2$

Figure 4.6: Condition number study for varying grid resolutions via method of manufactured solutions for four types of embedded methods with varying orders for the 3D low poly Stanford bunny explicit geometry.

4.3 Convergence

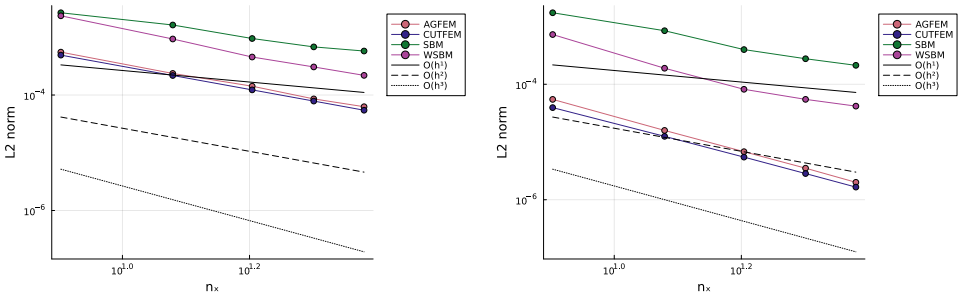
The theoretical convergence rates for unfitted finite element methods follow from standard approximation theory: for polynomial order p_e, the L^2 error is expected to converge at a rate of $\mathcal{O}(h^{p_e+1})$, corresponding to slopes of -2 and -3 for $p_e = 1$ and $p_e = 2$, respectively. The well-posedness proofs underpinning these rates are deemed out of scope for this work, but a starting point from which these proofs can be derived is presented by Akkerman et al. [92].

The convergence plot for the 2D cylinder case for the four unfitted methods is depicted in Figure 4.7. For CutFEM and AgFEM, the rate of convergence corresponds correctly to the theoretical order of convergence; that is, for first and second order elements, the rate of convergence is second and third order, respectively. For the SBM and WSBM, the rate of convergence is limited by the Taylor series expansion; as we employ only a first order Taylor series expansion, it is expected that this will decrease the rate of convergence by one order. According to theory, this limit can be superseded using a gradient recovery technique that has not been implemented.



(a) Order $p_e = 1$ (b) Order $p_e = 2$

Figure 4.7: Grid convergence study via method of manufactured solutions for four types of embedded methods with varying orders for the 2D cylinder implicit geometry.



(a) Order $p_e = 1$

(b) Order $p_e = 2$

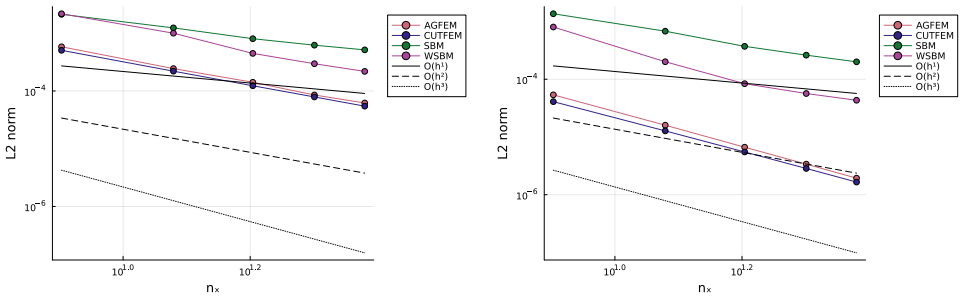
Figure 4.8: Grid convergence study via method of manufactured solutions for four types of embedded methods with varying orders for the 3D sphere implicit geometry.

The 3D convergence results for the implicit sphere (Figure 4.8), explicit sphere (Figure 4.9), and explicit low-poly Stanford bunny (Figure 4.10) are broadly consistent with the 2D findings. The implicit and explicit sphere geometries produce nearly identical convergence curves for all methods and both polynomial orders, confirming that the geometric representation — implicit level set versus explicit surface mesh — does not introduce meaningful differences for smooth geometries. This agreement serves as a cross-verification of the two geometry handling pipelines.

For the Stanford bunny case, the WSBM curve at $p_e = 2$ appears to plateau at the largest n, this could be a one off value or pose an actual plateau. In order to address this, cases with larger number of elements n should be considered.

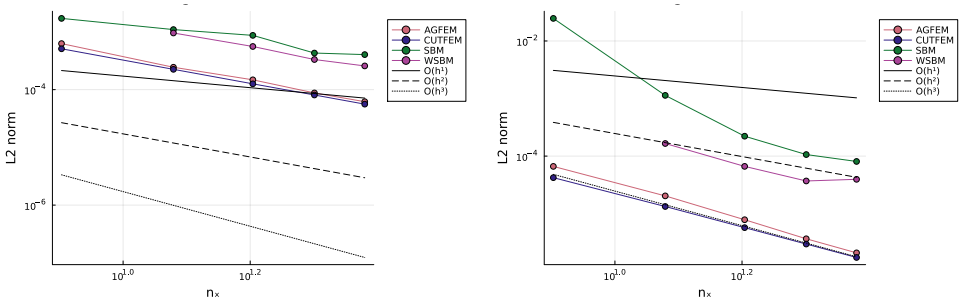
Across all 3D test cases, CutFEM yields the lowest L^2 errors, followed by AgFEM, then WSBM, and finally SBM. It is worth noting that SBM produces the largest errors despite having the smallest condition numbers across most configurations. This reflects the fact that conditioning and accuracy are governed by different aspects of the formulation: SBM's small condition numbers stem from its stable boundary enforcement, but its accuracy is limited by the Taylor expansion as we do not consider a gradient recovery technique. For Dirichlet type boundary conditions, the SBM and WSBM should reproduce optimal convergence rates similar to AgFEM and CutFEM.

The 3D benchmark cases are run in a serial setup and are therefore memory-limited, resulting in a relatively coarse range of refinement levels. This restricts the number of data points available for establishing asymptotic convergence rates and is a limitation of the present study. A parallel implementation would allow finer refinement levels and a more conclusive assessment of asymptotic behavior in 3D and is recommended as a priority for follow-up work.



(a) Order $p_e = 1$ (b) Order $p_e = 2$

Figure 4.9: Grid convergence study via method of manufactured solutions for four types of embedded methods with varying orders for the 3D sphere explicit geometry.

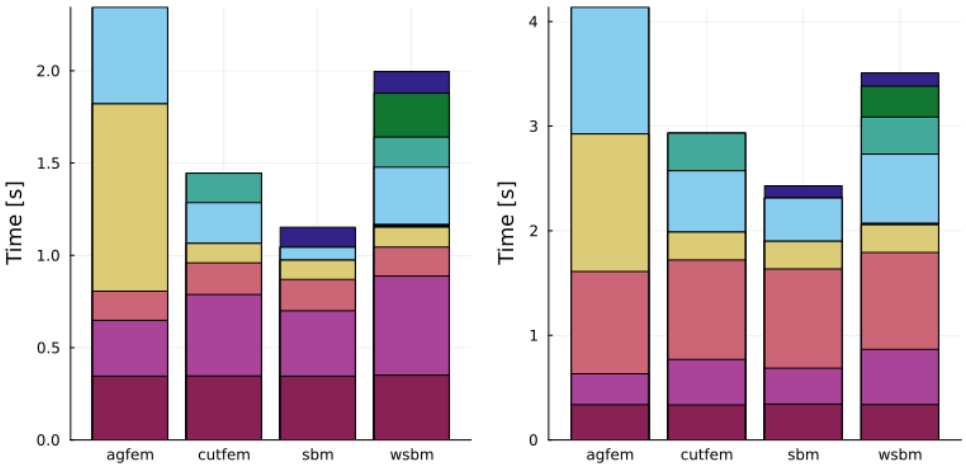


(a) Order $p_e = 1$ (b) Order $p_e = 2$

Figure 4.10: Grid convergence study via method of manufactured solutions for four types of embedded methods with varying orders for the 3D low poly Stanford bunny explicit geometry.

4.4 Computational Performance

The performance of each method and the relative performance between methods are investigated. Performance is quantified by the time-to-solution and memory allocations, which are obtained through the `@time` macro built into Julia. For this section, we only consider the 2D horizontal cylindrical test case. It should be noted that these studies have been conducted on Julia version 1.8.5 and `Gridap` version 0.18.4.

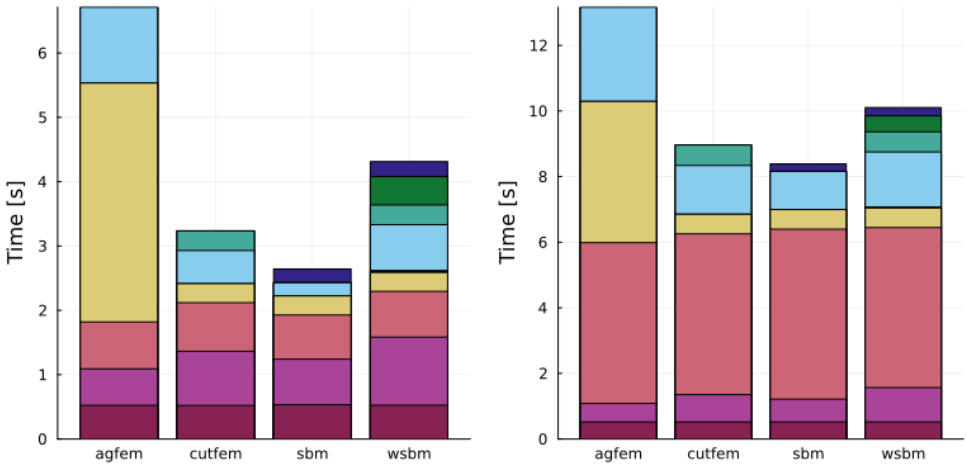


(a) Order $p_e = 1$ (b) Order $p_e = 2$

Figure 4.11: Comparison of time performance between AgFEM, CutFEM, SBM and WSBM for a 2D cylinder case for varying orders and number of elements of n by n with $n = 256$. Legend located in Table 4.1.

In general, time distributions for benchmark studies consistently show a right skew. This effect can be explained by recognizing that the machine noise impacting benchmarking is inherently positive: there are no typical noise factors that would cause a machine to execute instructions faster than the theoretical "ideal" time set by the hardware. For this reason, we consider the minimum to be a robust estimator for time benchmarking. Allocations are obtained from the same macro, and the same logic as for time benchmarking is applied regarding why we take the minimum number of allocations. The number of allocations is obtained directly from Julia's garbage collector, using the change in the internal allocation counter before and after the expression is executed. Before benchmarking using the macro, we conduct a warm-up run, after which 100 cases are executed. The warm-up run is necessary, as Julia uses a Just In Time compiler, meaning that the first execution of the code will also include the compilation time.

In Figure 4.11 and Figure 4.12, bar plots of the time-to-solution for AgFEM, CutFEM, SBM, and WSBM are depicted. We consider cases $n = 256$ and $n = 512$ with varying orders $p_e = 1$ and $p_e = 2$. The time-to-solution is divided into nine categories. **Not** all of these categories are present for each unfitted method. An overview of the categories is depicted in Table 4.1.



(a) Order $p_e = 1$

(b) Order $p_e = 2$

Figure 4.12: Comparison of time performance between AgFEM, CutFEM, SBM and WSBM for a 2D cylinder case for varying orders and number of elements of n by n with $n = 512$. Legend located in Table 4.1.

	(1). Consists of setting up the model, quadrature rules, weak form and functions
	(2). Embedding the geometry into the computational domain.
	(3). Solving the linear system of equations.
	(4). Setting up the FE spaces with trial and test functions.
	(5). Calculating the volume fraction α.
	(6). Global matrix contributions from the interior terms
	(7). Global matrix contributions from the ghost penalty stabilization terms
	(8). Global matrix contributions from the shifted edges (WSBM only)
	(9). Global matrix contributions from the shifted boundary

Table 4.1: Legend on categories for time-to-solution and allocations benchmarking.

From Figure 4.11 and Figure 4.12, the time-to-solution for the category setup **(1)** is about equal across the methods, as expected. The domain setup **(2)** is fastest for AgFEM, followed by SBM, CutFEM, and WSBM, respectively. SBM is slightly slower due to the approach used in this study, where we interface between the SBM domain and the non-SBM section of the domain to obtain the boundary. CutFEM and WSBM are likely slower due to the inclusion of edges in the domain setup. The solving step **(3)** is comparable across methods as well; here, there might be small differences that are expected to be matrix dependent. For the space category **(4)**, CutFEM, SBM, and WSBM are equivalent, as expected. At this step, the AgFEM starts to lag behind because of the cell-agglomeration algorithm required for this method. With the current code, this category is a significant bottleneck in the comparison between methods. For the interior matrix category **(6)**, SBM is the fastest, followed by CutFEM, WSBM, and AgFEM, respectively. Theoretically, SBM should be the fastest, which is clearly visible for first-order elements; however, for second-order

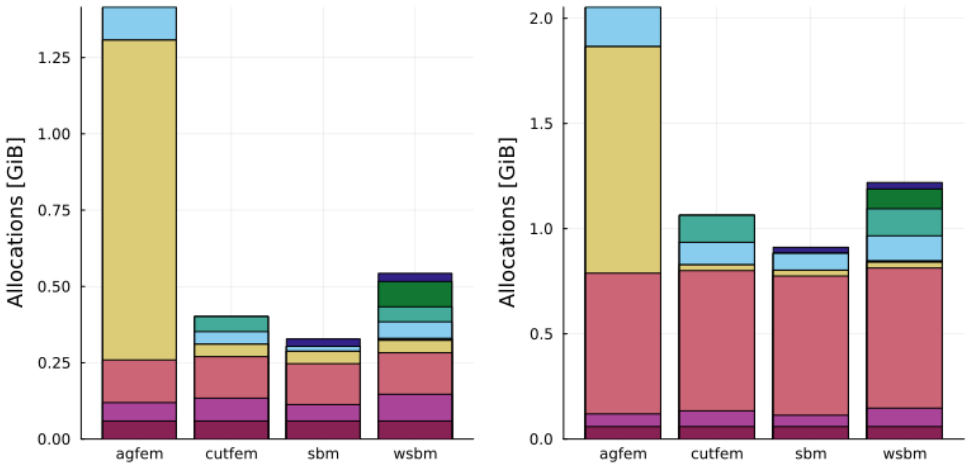
elements, these differences are smaller. It was expected that WSBM would outperform CutFEM, as WSBM has simpler quadrature rules and should be comparable to SBM. One key performance indicator for the WSBM in the current software setup is the manner in which the volume fraction weight is evaluated; we perform a cell-wise volumetric division of the integrated physical cell over the integrated active cell. Alternative methods could be introduced here to reduce the overall time. The remaining categories are not present in all methods, so a strict comparison is not possible; yet these categories provide insight into how the workload is being divided and are therefore of interest. The WSBM has a volume fraction category **(5)**, which has an almost negligibly small contribution; for second-order elements, it cannot be distinguished in the figure. This is because it is currently dependent on steps **(1)** and **(2)** to obtain the arrays containing the required information for the cell-wise division. The contributions of the ghost penalty stabilization on the edges **(7)** in the cut mesh region are equivalent for WSBM and CutFEM, as expected. The application of the shifting operator on these edges **(8)** for WSBM is relatively large for the first-order but decreases for the second-order. The shifting operator applied to the surrogate boundary **(9)** for SBM and WSBM is comparable, with WSBM being slightly slower. This is likely due to the evaluation of the volume fraction weight as well.

Overall, the difference between $p_e = 1$ and $p_e = 2$ is the increase in total time-to-solution and an increased time for solving **(3)** with respect to the setup **(1)** and domain **(2)** steps. For $p_e = 2$, the spaces **(4)** step is relatively smaller than those for $p_e = 1$. The differences between $n = 256$ and $n = 512$ elements are also reflected in an increased total time, and the trends across all methods appear to be similar.

In Figure 4.13 and Figure 4.14, the number of allocations for each category is compared. The study considered $n = 256$ and $n = 512$ with varying order $p_e = 1$ and $p_e = 2$. The legend is the same as described before in Table 4.1. The results for the allocations are similar to those for the time-to-solution. The main difference is the relatively larger allocations that are used for the aggregated FE space **(4)** for the AgFEM. Increasing the number of elements n or the order p_e increases the allocation size as well. For higher order, a relatively larger number of allocations is used for the solving step **(3)**.

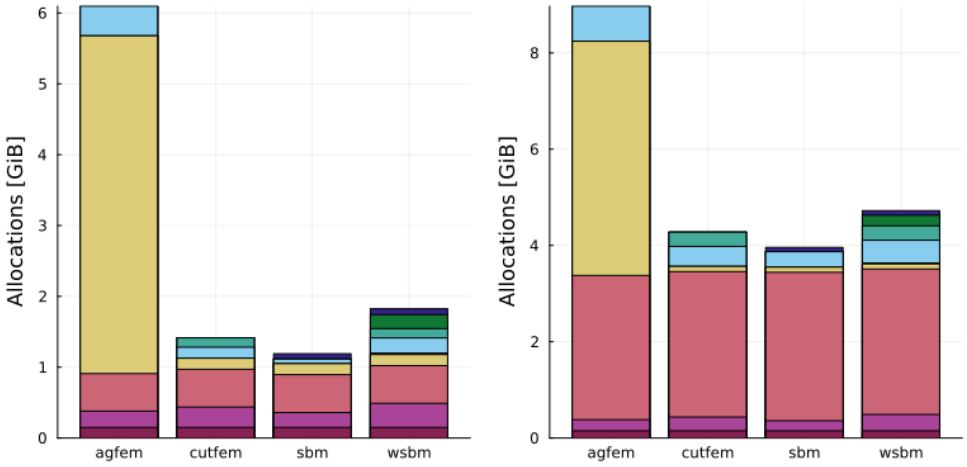
In terms of both time-to-solution and allocations, SBM outperforms all other methods, followed by CutFEM, WSBM, and AgFEM, respectively. For higher order, the same trend remains visible. In order to make AgFEM more competitive, an optimization study on the creation of the enhanced finite element spaces should be performed.

There is room for improvement in the current study for SBM and WSBM in the domain category by optimizing the creation of the surrogate boundaries, as these are generated from the same algorithms used for CutFEM and AgFEM. Additionally, the inclusion of a gradient recovery method might negatively impact performance for SBM and WSBM, and this should be studied.



(a) Order $p_e = 1$ (b) Order $p_e = 2$

Figure 4.13: Comparison of data allocation performance between AgFEM, CutFEM, SBM and WSBM for a 2D semicircle case for varying orders and number of elements of n by n with $n = 256$. Legend located in Table 4.1.



(a) Order $p_e = 1$ (b) Order $p_e = 2$

Figure 4.14: Comparison of data allocation performance between AgFEM, CutFEM, SBM and WSBM for a 2D semicircle case for varying orders and number of elements of n by n with $n = 512$. Legend located in Table 4.1.

4.5 Implementation Complexity

Implementation complexity is a rather vague term, and the first step is to define and quantify what is considered implementation complexity in this work. We consider a standard finite element library, without any unfitted finite element tools, as the baseline. The implementation complexity is then defined as all additional modifications required

	CutFEM	AgFEM	SBM	WSBM
IN OUT CUT algorithm	yes	yes	yes	yes
Triangulation portion	yes	yes	yes	yes
Cut interfaces	yes	no	no	yes
Specialized integration	yes	yes	no	no
Cell aggregation	no	yes	no	no
Number of additions	4	4	2	3

Table 4.2: Overview of the implementation complexity for each unfitted finite element method.

for implementing the four unfitted finite element methods. We quantify it by counting the number of additional data structures or algorithms required to facilitate one of the proposed unfitted finite element methods concerning a baseline finite element framework.

In that sense, we start with the fundamental step that is present for all unfitted methods, namely the intersection algorithm required to flag the elements as IN, OUT, or CUT. Alongside this algorithm, all methods require a modification of the triangulations that allows for the representation of a portion of the background triangulation that corresponds to the active or surrogate triangulation. On these triangulation portions, regular integration rules should be applied.

On top of the active triangulation, we are required to integrate on the interfaces of the cut elements for both CutFEM and WSBM to apply the ghost penalty stabilization. These need to be identified and require standard integration rules for interfaces.

For both CutFEM and AgFEM, there is a requirement for an integration domain that fits the embedded boundary, which requires specialized integration rules. Note that for the WSBM implementation in this dissertation, we also use this to obtain the cell-wise weights; other strategies are also possible [129]. Lastly, AgFEM requires a cell aggregation algorithm [77], which should be constructed on top of the active domain and return a constrained finite element space.

The implementation complexity is summarized in Table 4.2. Based on the total number of required additions, both CutFEM and AgFEM are the most complex. Although the implementation of a cell aggregation algorithm is likely to be more involved than the identification and integration over cut cell interfaces. The WSBM follows, with three additional features, and the method requiring the least modification to implement is the SBM.

4.6 Conclusions

In this chapter, we introduced a benchmarking effort for four types of unfitted finite element methods, namely CutFEM, AgFEM, SBM, and WSBM. We consider three geometries: a horizontal cylinder in 2D, and in 3D, a sphere and a low poly Stanford bunny geometry. We use implicit and explicit geometrical representations via a level set function and STL file types, respectively. To evaluate each geometry, we consider four test cases, namely one in 2D with the horizontal cylinder and three cases in 3D. Initially, a sphere is represented by a level set function, followed by a sphere represented using an STL, and lastly, the low poly Stanford bunny as a complex explicit geometry. We verify each method's implementation

	CutFEM	AgFEM	SBM	WSBM
Accuracy	1	2	4	3
Condition number	3	2	1	4
Computational performance	2	4	1	3
Implementation complexity	3	4	1	2

Table 4.3: Ranking of the unfitted finite element methods in each category presented in this benchmarking initiative.

using the method of manufactured solutions via a grid convergence study for varying numbers of polynomial orders p_e. It is shown that, for all cases, the expected order of convergence is recovered for CutFEM, AgFEM, SBM, and WSBM. We consider the condition number of the discrete system of equations for each case study and for each method. SBM is expected to produce the lowest condition numbers due to the absence of cut cells, which is also confirmed by the data. WSBM has the largest condition numbers across methods, irrespective of polynomial order p_e, and CutFEM is generally lower than WSBM. AgFEM performs similarly to SBM for order $p_e = 1$, but it increases the condition number in a manner similar to CutFEM for order $p_e = 2$. Additionally, for the 2D horizontal cylinder case, a performance review is conducted for $n = 256$ and $n = 512$, with varied polynomial orders $p_e = 1$ and $p_e = 2$. We propose a division of nine categories across unfitted methods and show block diagrams denoting computational time and allocations. The overall trends between time-to-solution and allocations are similar for all four unfitted methods. SBM outperforms all other methods and requires the least time and the fewest allocations. CutFEM is slightly slower and requires more allocations than WSBM. AgFEM is the slowest in performance and requires the most allocations; this is due to the cell aggregation algorithm used whilst setting up the finite element space. Lastly, a discussion is presented on the complexity of implementation regarding each of the four unfitted methods, assigning the following ranking from least to most complex to implement: SBM, WSBM, CutFEM, AgFEM. In Table 4.3, a ranking is depicted for the four unfitted methods as they are assessed within our proposed framework.

5

Floating Structures in the Time Domain

Time domain simulations play a central role in the analysis and design of floating offshore structures because they have the ability to capture the full, nonlinear, transient response of these systems under realistic environmental loading. This chapter does not address the nonlinear response, nor does it demonstrate realistic environmental loading. Instead, it serves as a foundational step, beginning with linear potential flow coupled with a rigid structure and demonstrating how CutFEM, AgFEM, SBM, and WSBM can be applied to model interactions between floating structures and water. The novelty in this chapter lies in the demonstration of the successful implementation of the aforementioned unfitted methods to conduct time domain simulations of floating structures in 2D and 3D. Additionally, it is shown that simulations can be conducted for both simple and realistic offshore structures. It is important to note that, although the results are for linear potential flow, an extension to higher fidelity models is relatively straightforward. Before the first section, the discretization of the time domain applied in this work is introduced. Then, in the first section, a 2D verification study is conducted to verify that the solutions of unfitted and body-fitted methods match. Dependency on the number of elements and polynomial order is discussed in the context of unfitted methods. Then, the problem is extended to 3D, and a comparison is made between implicit and explicit geometrical representations. Thereafter, the 3D simulations are validated against the experimental work by Kramer et al. [160]. An adaptive mesh refinement strategy is proposed to make more efficient use of the available computational resources. The implementation has been validated, and the future steps of development are discussed. The adaptive grid approach is then applied to a realistic offshore geometry: the OC4 DeepCWind semisubmersible, such that the natural period of oscillation is recovered. Lastly, the third case study is extended to a multi-body simulation, and it is demonstrated that the displacement of the first structure influences the displacements of the second structure.

Discrete Time Stepping

In this chapter, we start with the semi-discrete weak formulations as defined in chapter 3. For discrete time stepping, the generalized alpha method [161] is used. This is an implicit integration scheme with the ability to control numerical dissipation while preserving second-order accuracy. It introduces two parameters: α_f and α_m, that control how the governing equations are evaluated in time. Both of these parameters depend on the spectral radius at infinite frequency $\rho_\infty \in [0,1]$. Setting $\rho_\infty = 1$ results in the Newmark-β method [162], and decreasing ρ_∞ will increase high frequency damping. To progress in time, a linear interpolation of variables at the quadrature points is conducted, specifically for the velocity potential ϕ at discrete time step n_t:

$$\ddot{\phi}_{n_t+1-\alpha_m} = (1-\alpha_m)\ddot{\phi}_{n_t+1} + \alpha_m \ddot{\phi}_{n_t} \tag{5.1}$$

$$\dot{\phi}_{n_t+1-\alpha_f} = (1-\alpha_f)\dot{\phi}_{n_t+1} + \alpha_f \dot{\phi}_{n_t} \tag{5.2}$$

$$\phi_{n_t+1-\alpha_f} = (1-\alpha_f)\phi_{n_t+1} + \alpha_f \phi_{n_t}. \tag{5.3}$$

5

Here, the dots indicate the order of the time derivative. For the displacement vector $\mathbf{u}$, equivalent definitions are used. Then, Equation 5.3 and Equation 5.2 are defined using the Newmark time integration scheme:

$$\dot{\phi}_{n_t+1} = \dot{\phi}_{n_t} + \Delta t((1-\gamma)\ddot{\phi}_{n_t} + \gamma \ddot{\phi}_{n_t}) \tag{5.4}$$

$$\phi_{n_t+1} = \phi_{n_t} + \Delta t \dot{\phi}_{n_t} + \frac{\Delta t^2}{2}((1-2\beta)\ddot{\phi}_{n_t} + 2\beta \ddot{\phi}_{n_t}), \tag{5.5}$$

given the time step size Δt and parameters β and γ. The four parameters can then be rewritten such that they are all dependent on ρ_∞ as:

$$\alpha_m = \frac{2\rho_\infty - 1}{\rho_\infty + 1} \tag{5.6}$$

$$\alpha_f = \frac{\rho_\infty}{\rho_\infty + 1} \tag{5.7}$$

$$\gamma = \frac{1}{2} + \alpha_f - \alpha_m \tag{5.8}$$

$$\beta = \frac{1}{4}(1 + \alpha_f - \alpha_m)^2 \tag{5.9}$$

In the following sections, an overview of the case studies considered in this dissertation is presented. We consider four case studies that are structured progressively, such that we start with a simple verification scenario and work towards more realistic validation scenarios. The first case study is a verification study designed to demonstrate that the governing equations are correctly solved and that they match the solution of a body-fitted or conformal FEM. The second case study demonstrates that an extension to 3D is possible, and we establish the equivalence between implicit and explicit geometrical representations. In the third case study, we propose a validation case study via a numerical free decay study

for heave oscillations using realistic offshore geometries. This structure is freely floating, and mooring systems are not considered. In the fourth study, we demonstrate the capability of one realistic offshore geometry inducing motion on another offshore geometry that rests at equilibrium. This last test serves as a proof of concept for multi-body simulations and is not a validated study.

5.1 Case 1: Horizontal Cylinder

The first case study considers the experimental work done by Ito [163], in which the transient heave oscillation of a horizontal cylinder is studied. In this work, the cylinder was constrained to pure heave by an air-bearing heave rod, the displacement was measured with a linear variable differential transformer, and that the cylinder was ballasted to achieve exactly the half-submerged condition. The main objective of this case study is the numerical verification between body-fitted, also known as conformal, and unfitted methods. The reason for selecting Ito's work for the verification study is twofold: we want the initial verification study to be simple; therefore, a two-dimensional heaving horizontal cylinder is a strong choice. Because of the circular shape, we enforce that the unfitted methods produce varying shapes of cut cells, but we do not require additional mesh refinement to capture local features, which is anticipated for complex-shaped geometries. Additionally, Ito's work considers small initial displacements and allows for linear potential-flow assumptions to be sufficient.

From the experimental work, we consider the domain as depicted in Figure 5.1, with the domain Ω bounded by the structure boundary Γ, the free surface boundary Γ_{f}, wall boundaries Γ_{w}, and the seabed boundary Γ_{sb}.

The domain dimensions of the original work are in imperial units. These parameters are taken in this work as the depth $d = 1.22$ m, the radius of the cylinder as $R = 7.62$ cm, and the initial displacement as $h_0 = 2.54$ cm. The domain length is taken to be $L_x = 6.0$ m. The point of hydrostatic equilibrium is such that the center of the cylinder coincides with the level of the free surface at rest. The simulation runs for a total of $T_e = 2.0$ s with a timestep of $\Delta t = 1/50$ s. No numerical wave damping is implemented; as such, $\nabla\phi_h \cdot \boldsymbol{n}_{\mathrm{w}} = 0$ on Γ_{w}. Therefore, the choice of domain length L_x and simulation time T_e is decided empirically so that we can minimize the domain size while still capturing the solution without wave reflections.

The semi-discrete weak forms for each method are defined in chapter 3, and an overview of their locations within this work is presented in section 3.3. The problem is discrete in time using the generalized alpha time stepping method with Newmark-β set up, i.e., $\rho_\infty = 1$. Additionally, only the heave directional motion is considered in this study. For this reason, the unknown displacement $\mathbf{u}$ is a single degree of freedom only existing on the structure boundary, which has an initial displacement of $\mathbf{u} = 0.0254$ m and an initial velocity of $\dot{\mathbf{u}} = 0.0$ m/s. The velocity potential throughout the domain has an initial value of $\phi = 0.0$ m^2/s, and the initial value of the first time derivative of the velocity potential is $\dot{\phi} = 0.0$ m^2/s^2. The application of these initial conditions is the same across all four unfitted methods considered in this study. For AgFEM and CutFEM, this is easy to understand as the integration domain matches closely to the physical domain. Whereas for SBM and WSBM, it might be unclear whether the initial distance vector should be included for the initial displacement. In this case, as the domain does not actually deform, we can consider

the displacement to be relative to either the inner surrogate, outer surrogate, or physical boundary; therefore, it will not make a difference.

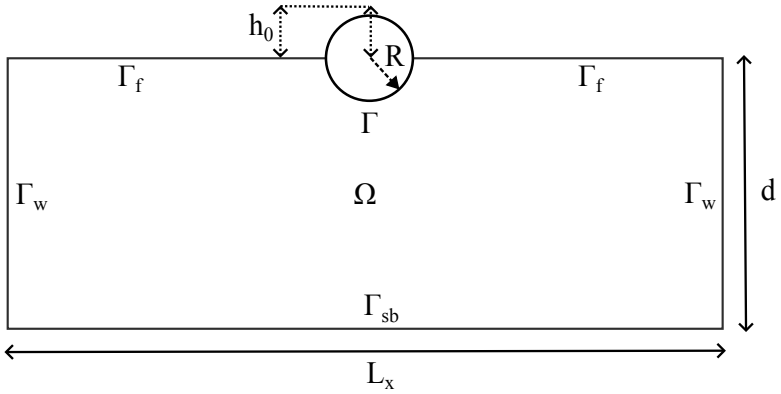
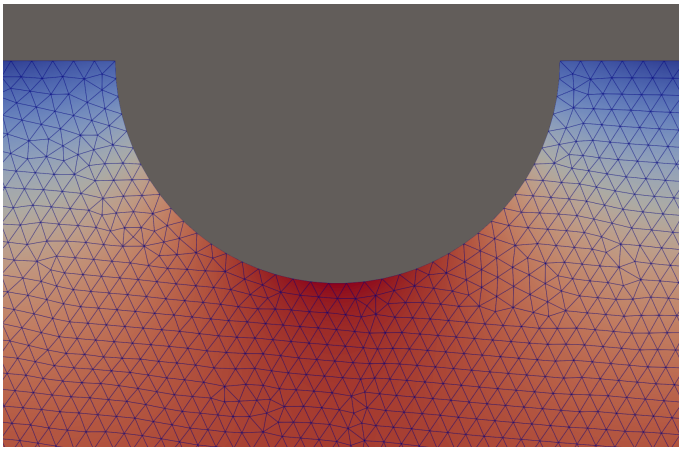


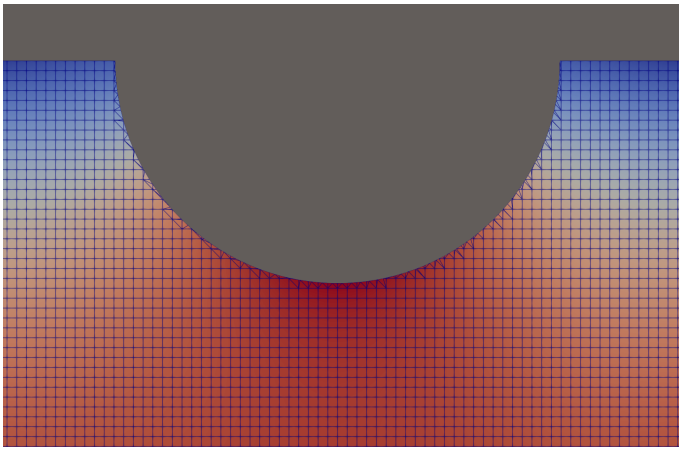
Figure 5.1: Case study 1: floating horizontal cylinder in 2D as described by Ito [163]. The depth $d = 1.22$ m, the radius $R = 7.62$ cm and the initial displacement $h_0 = 2.54$ cm. We choose the domain length $L_x = 6.0$ m and consider a total simulation time of $T_e = 2.0$ s. The domain Ω, structure boundary Γ, free surface boundary Γ_f, wall boundary Γ_w, and seabed boundary Γ_{sb}.

To study and compare the unfitted methods, we also vary the number of elements n and the polynomial order p_e of the elements. As we have a rectangular domain, the number of elements is distributed to approximate perfectly square elements by taking $5n$ in the horizontal direction for each element in the vertical direction. Realistically, because of the length over depth ratio of $\frac{L_x}{d} \approx 4.92$, the elements are slightly larger in the vertical direction than in the horizontal direction. In Figure 5.2, several discrete domains are depicted; in Figure 5.2a, a body-fitted or conformal domain is depicted, corresponding to a traditional FEM that is used as the ground truth. In Figure 5.2b, the integration space for both CutFEM and AgFEM is depicted, and in Figure 5.2c, the surrogate domain is illustrated as used for the SBM-type methods. Note that for the WSBM, one additional layer of elements is present at the cylinder boundary over which the test functions are weighted.

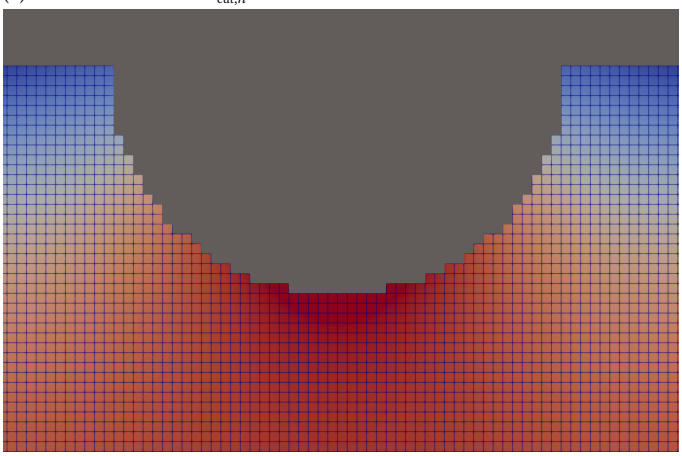
In the result plots, the displacement of the cylinder is normalized using the initial displacement h_0, and the time is normalized using the nondimensional time $\bar{t} = t\sqrt{\frac{g}{R}}$, where $g = 9.81$ m/s^2 is the gravitational constant and R is the cylinder radius. The results are displayed in Figure 5.3-Figure 5.7 and overall show the same trends. We confirm that given sufficiently high number of elements and/or polynomial order all unfitted methods match closely to a traditional conformal FEM. CutFEM and AgFEM are close to indistinguishable in all cases; both SBM and WSBM follow the trend of oscillation but achieve more accurate solutions as the number of elements and polynomial order increases. Additionally, the slight frequency difference that is present for SBM and WSBM in cases with order $p_e = 1$ appears to reduce for order $p_e = 2$. The likely reason for this is that at polynomial order $p_e = 1$, the Hessian operator and product with the distance function in the weak formulation return zero, or simply put, we do not shift the terms using the distance function. For polynomial order $p_e = 2$, the Hessian operator does not strictly return zero and therefore shifts the contributions on the boundary and interface conditions



(a) Body-fitted or conformal domain Ω_h.



(b) Tessellated domain $\Omega_{\text{cut},h}$.



(c) Inner surrogate domain $\tilde{\Omega}_j$.

Figure 5.2: Three discrete domain representations for the horizontal cylinder case study displaying the velocity potential solution at time step $t = 1.0$ s.

using the distance vector. This limitation occurs for purely Neumann boundary conditions, which is the case in this work and requires a gradient recovery technique [140].

For the case with 300 x 60 elements, the WSBM is most different from the solution, even diverging for order $p_e = 2$. It is unclear why exactly the unstable behaviour of the WSBM occurs.

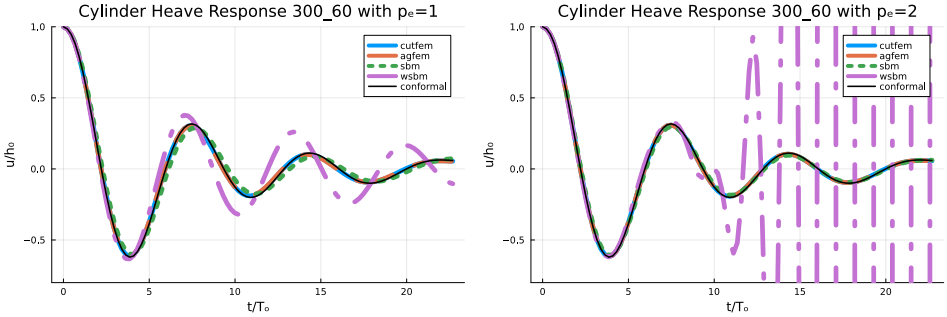


Figure 5.3: Non-dimensional heave displacement versus non-dimensional time for the four unfitted methods with mesh 300 x 60 for order $p_e = 1$ on the left and order $p_e = 2$ on the right.

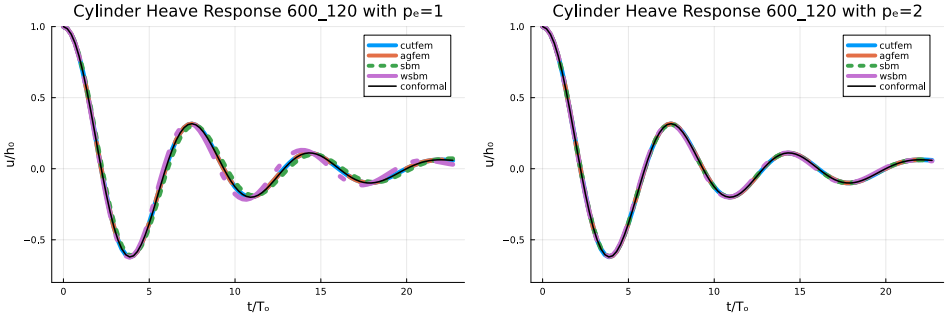


Figure 5.4: Non-dimensional heave displacement versus non-dimensional time for the four unfitted methods with mesh 600 x 120 for order $p_e = 1$ on the left and order $p_e = 2$ on the right.

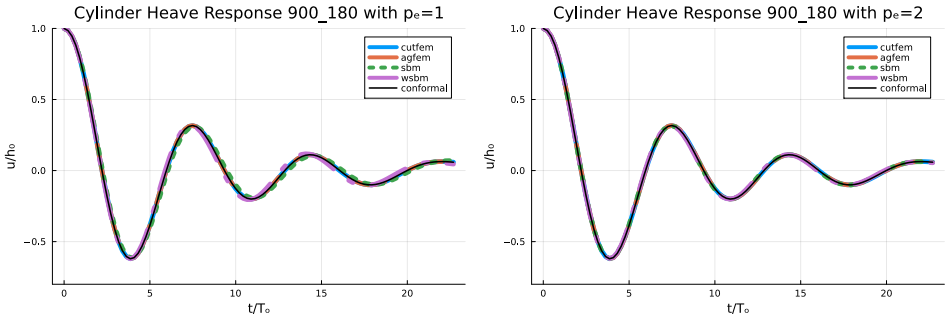


Figure 5.5: Non-dimensional heave displacement versus non-dimensional time for the four unfitted methods with mesh 900 x 180 for order $p_e = 1$ on the left and order $p_e = 2$ on the right.

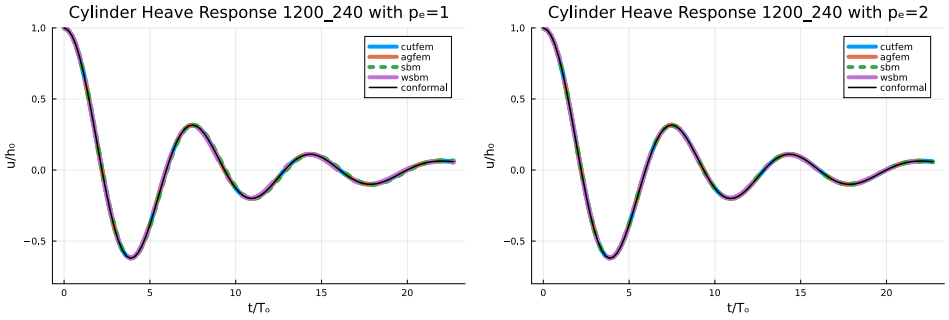


Figure 5.6: Non-dimensional heave displacement versus non-dimensional time for the four unfitted methods with mesh 1200 x 240 for order $p_e = 1$ on the left and order $p_e = 2$ on the right.

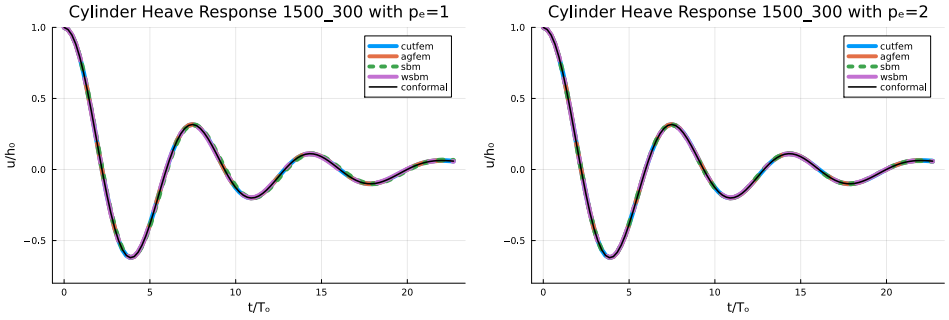


Figure 5.7: Non-dimensional heave displacement versus non-dimensional time for the four unfitted methods with mesh 1500 x 300 for order $p_e = 1$ on the left and order $p_e = 2$ on the right.

5.2 Case 2: Sphere

The second case study considers a sphere in heave motion and is based on the experimental and numerical work by Kramer et al. [160]. The main difference with respect to the previous case study is the extension to three dimensions. Considering the numerical framework

applied in this work, few changes are required to facilitate 3D simulations. Additionally, we conduct a study on the equivalency of implicit and explicit geometrical representations within the presented framework. The extension to 3D significantly increases the computational cost, which poses limitations for the Cartesian grids that have been used up to this point. Therefore, an adaptive grid refinement strategy based on a forest of octrees is also presented.

Experimental and numerical reference data

The benchmark experiment by Kramer et al. [160] was conducted at the wave basin of the Ocean and Coastal Engineering Laboratory at Aalborg University, Denmark, with basin dimensions of 13.0 m by 8.44 m and a water depth of 0.9 m. Heave decay was recorded using an optical motion capture system (Qualisys) at 300 fps. The sphere was freely floating with six degrees of freedom; roll and pitch rotations remained below 0.5° and were accounted for in the uncertainty analysis. The sphere was ballasted to the half-submerged condition, with the waterline at the equator. The initial displacement of $h_0 = 0.03$ m was applied via a line attached to the top of the sphere and released by an electrical actuator, with pre-release measurements confirming a mean initial velocity of 0.0 m/s with a standard uncertainty of 0.0004 m/s. The test duration was chosen to capture approximately eight natural heave periods, corresponding to $T_e = 6.0$ s given a natural period of $T_0 = 0.76$ s. In addition to the experimental data, Kramer et al. [160] provide numerical reference data generated through a blind test involving multiple research institutions and industry partners, who independently formulated models based on the shared physical parameters. The reference dataset comprises eleven models spanning three levels of fidelity: five Reynolds-Averaged Navier–Stokes (RANS) models, one fully nonlinear potential flow (FNPF) model, and five linear potential flow (LPF) models. The RANS models, implemented in OpenFOAM [54] and StarCCM+ [57], showed deviations of $0 - -4$ mm at crests and troughs for the largest drop height, while the FNPF model showed deviations of up to 8 mm. The LPF models showed the largest deviations, up to 50 mm at the largest drop height, primarily due to phase errors arising from linearization. The present method is a linear potential flow formulation and is therefore compared primarily against the experimental data and the LPF reference models, with the RANS and FNPF results included to contextualize the role of viscosity and nonlinearity in the benchmark.

Domain and parameters

In Figure 5.8, the computational domain is depicted; it corresponds to the dimensions of the wave tank as described by Kramer et al. [160]. We consider only the heave free decay test from an initial displacement of $h_0 = 0.03$ m. The sphere has a radius of $R = 0.15$ m, and the domain dimensions are length $L_x = 13.0$ m, breadth $L_y = 8.44$ m, and depth $d = 0.9$ m. For the implicit geometrical representation of the sphere, a level set function is defined, and for the explicit geometrical representation, a sphere of radius $R = 0.15$ is created using the standard geometries available in the free software Meshmixer [164], developed by AutoDesk. All parameters of interest are depicted in Table 5.1. The total simulation time is $T_e = 6.0$ s, and we choose a time step size of $\Delta t = 0.05$ s. No numerical wave damping is implemented, and on the surrounding wall boundaries Γ_{w}, we enforce that $\nabla \phi_h \cdot \boldsymbol{n}_{\mathrm{w}} = 0$.

Domain length: L_x	13.0 m
Domain breadth: L_y	8.44 m
Domain depth: d	0.9 m
Sphere radius: R	0.15 m
Density of water: ρ	998.2 kg/m^3
Gravitational constant: g	9.82 m/s^2
Mass: m	7.0559 kg
Moment of inertia (roll): I_{xx}	9.8251 kg·m^2
Moment of inertia (pitch): I_{yy}	9.8524 kg·m^2
Moment of inertia (yaw): I_{zz}	7.3052 kg·m^2

Table 5.1: Dimensions and relevant parameters for the second case study

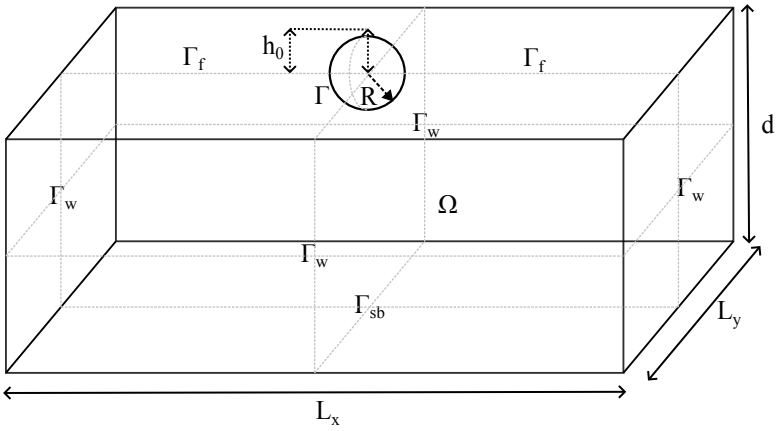


Figure 5.8: Case study 2: floating sphere in 3D as described by Kramer et al. [160]. The depth $d = 0.9$ m, the radius $R = 0.15$ m and the initial displacement $h_0 = 0.03$ m. The domain lengths $L_x = 13.0$ m and $L_y = 8.44$ m and a total simulation time of $T_e = 6.0$ s. The domain Ω, structure boundary Γ, free surface boundary Γ_f, wall boundary Γ_w, and seabed boundary Γ_sb.

Implicit versus explicit geometrical representation

The goal of this section is to verify that, within the computational framework we employ, there are equivalent solutions between using a level set function to represent the geometry and using an .STL file format to represent the geometry, i.e., implicit and explicit geometrical representation, respectively. For explicit geometrical representations, there might be a difference due to slight misalignment of the distance and normal vectors with respect to the implicit cases. It is not expected that there will be differences in the cut volume, as the error in volume is of machine precision [165].

For this numerical experiment, we employ a Cartesian grid with a chosen number of elements set to $n = 50$ and a distribution of elements in each direction (x, y, z) as $(7n, 4n, n)$, such that the total number of elements in the domain is 3.5 million. In Figure 5.9 on the left, the velocity potential in the computational domain for the SBM is depicted at time $t = 2.2$ s, and on the right, a y-plane cross-section at $x = 0$ m is shown.

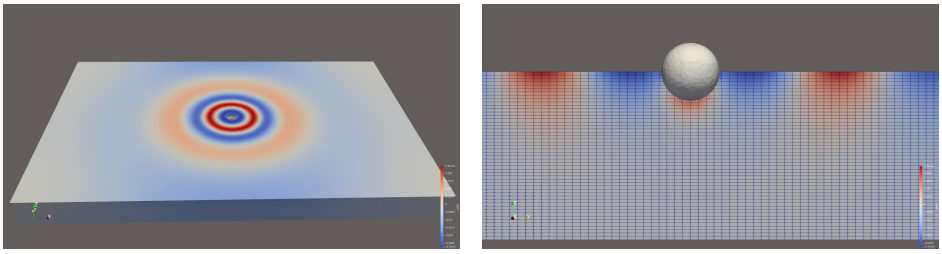


Figure 5.9: Depiction of sphere case study simulation for SBM type approach at time step $t = 2.2$ s, full domain on the left, and close up at the y-plane cross-section on the right.

Figure 5.10 depicts the results for the implicit versus explicit geometrical representations of the heave decay of a sphere. The heave is normalized using the initial heave displacement, and the time is normalized using the period of oscillation $T_o = 0.76$ s, as described by Kramer et al. [160]. From Figure 5.10, we note that for CutFEM, AgFEM, and SBM, there is no visible difference between the implicit and explicit geometrical representations. For WSBM, both the explicit and implicit geometrical representations exhibit divergence within the depicted time window. The explicit representation diverges immediately, while the implicit representation diverges with a smaller magnitude. The difference between the two is likely attributable to the misalignment of distance and normal vectors in the explicit representation, which worsens the geometric accuracy of the surrogate boundary and may exacerbate the underlying instability.

The cause of this divergence is not fully understood and has been an open issue throughout the development of this framework. As an initial step, it has been verified that all weights in the WSBM formulation are positive and nonzero throughout the simulation, ruling out degenerate weight values as the direct cause of the instability. Several hypotheses have been identified. First, the active domain used for WSBM is inherited from the AgFEM and CutFEM implementations and may include elements with extremely small support near the surrogate boundary. This also implies that for such cells the distance function is

comparatively large with respect to other distance vectors from cells with larger weights. Preliminary investigations filtering out such elements showed some improvement, but no robust and well-tested solution has been established. Possibly, there is room for improvement on the surrogate boundary selection, similar to the work by Yang et al. [166] for the SBM. Second, the ghost penalty stabilization approach used here may not be appropriate stabilization for the WSBM formulation in the transient setting. Third, it could be that instability is occurring at the intersection of the free surface boundary and the cut cell region due to improper enforcement of the boundary condition. The weak form itself is not suspected to contain errors. This instability is consistent with the divergence observed for the coarsest mesh in the 2D cylinder transient simulations, suggesting it is a systematic issue related to mesh resolution rather than geometry-specific behavior. Resolving this instability is identified as a priority for future work, and the WSBM results presented in the remainder of this section should be interpreted with this limitation in mind.

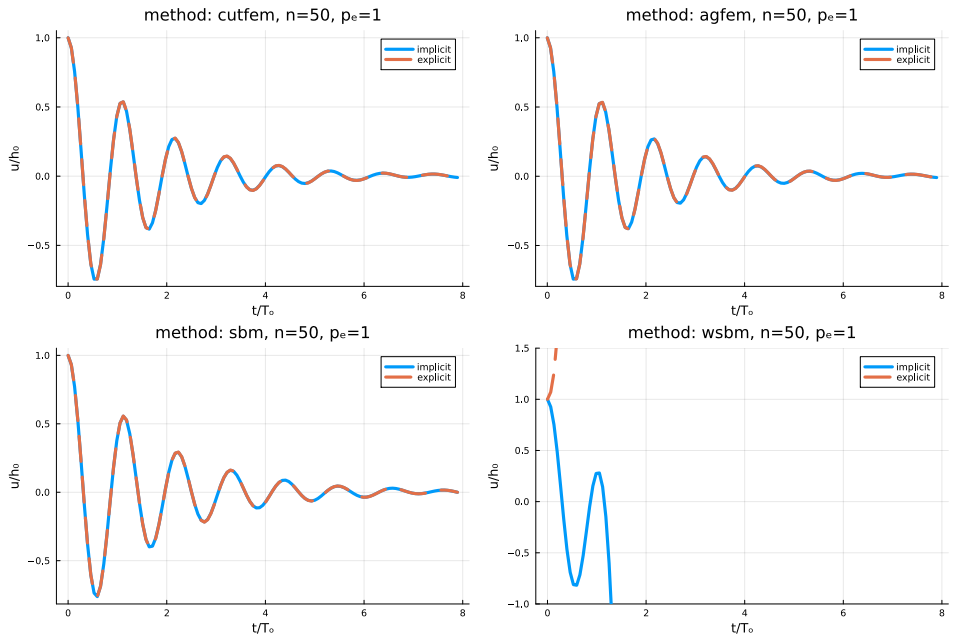


Figure 5.10: Non-dimensional heave displacement versus non-dimensional time for the four unfitted methods with mesh 350 x 200 x 50 for order $p_e = 1$. Axes are normalized using initial displacement $h_0 = 0.03$ m and period $T_o = 0.76$ s.

Validation

To validate the correctness of the framework, the numerical results are compared against the experimental dataset from Kramer et al. [160]. This benchmark was specifically designed to provide a high-quality public reference for the validation and calibration of numerical models, with emphasis on estimating the precision of physical decay tests and providing a reliable real-world reference for fluid-structure interaction simulations. The authority of

this dataset is further evidenced by the eleven numerical methods validated against it in the original publication.

The present framework employs a linear potential flow formulation in a volumetric finite element setting, which does not correspond directly to any of the LPF0–3 models referenced in Kramer et al. [160] Consequently, a direct numerical-to-numerical comparison against those models would be complicated by differences in modeling assumptions — particularly regarding hydrostatic treatment and hydrodynamic coefficient evaluation — rather than reflecting the accuracy of the present implementation alone. The experimental data, therefore, serves as the most appropriate and unambiguous reference, as it is independent of such modeling assumptions. At the drop height of $h_0 = 0.1D$ used in this work, the benchmark operates in the nominally linear regime, where Kramer et al. [160] demonstrate that higher-fidelity numerical models agree closely with the experimental data, confirming that the experiment provides a reliable reference at this drop height.

In Figure 5.11, the implicit geometrical representation resulting from the unfitted methods in the previous section is depicted, along with the experimental results. It is observed that there is a slight frequency difference and smaller amplitudes than those observed in the experimental results, see Figure 5.11. More discussion on the discrepancy between the results of the numerical methods and the experiment is presented in the validation with grid adaptivity subsection. Overall, the decay trend matches the experimental results, indicating that an increase in the number of elements, an increase in polynomial order, or a decrease in time step size could improve the accuracy of the solution.

As we observe instabilities for the WSBM, which were also noted in the 2D verification study, it is suspected that increasing the number of elements will improve solution quality. For the results depicted in Figure 5.11, AgFEM and CutFEM are visually indistinguishable, and SBM has a slight shift that is also observed in the 2D verification study for polynomial order $p_e = 1$.

Among the stable methods, SBM exhibits a slight phase shift relative to CutFEM and AgFEM, which is also observed in the 2D verification study at $p_e = 1$. Two contributing factors are identified. First, the boundary condition is enforced by integration over the surrogate boundary $\tilde{\Gamma}$ rather than the true boundary Γ, introducing a geometric error that decreases with mesh refinement as the surrogate boundary more accurately approximates the true boundary. Second, at $p_e = 1$, the distance vector term in the Taylor expansion is not included as the Hessian is zero, meaning the boundary condition is enforced without the distance vector contribution that would otherwise compensate for the displacement between Γ and $\tilde{\Gamma}$. In the 2D verification study, the phase shift reduces significantly at $p_e = 2$, confirming that the absence of the distance vector term is a meaningful contributor to the error at first order. For the 3D cases considered here, only $p_e = 1$ is used due to computational constraints. At this order, the Hessian is identically zero, and the distance vector term is not employed, meaning the 3D SBM results are subject to the same two sources of error as the 2D $p_e = 1$ case. The phase shift is therefore expected to decrease with mesh refinement or upon extension to $p_e = 2$, both of which are recommended for future work.

Grid Adaptivity

The Cartesian grid simulation in 3D from the previous subsection takes hours to run a single simulation. This makes the current computational framework unattractive for large

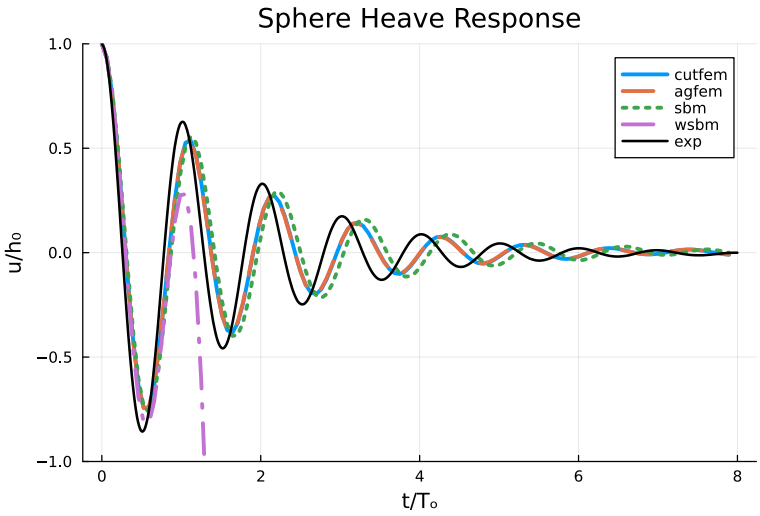


Figure 5.11: Results for Cartesian grid with $n = 50$ with distribution $(7n, 4n, n)$ and timestep $\Delta t = 0.05$ s using implicit geometrical representation.

scale simulations, and it forces us to use many more computational resources than we need. To alleviate this disadvantage, we aim to improve the framework by implementing a parallel approach and incorporating mesh adaptivity. Mesh adaptivity allows us to make more efficient use of the available computational resources by increasing the number of elements only in the regions in which we are interested. In this work, this will be at the free surface and close to the floating structure. To achieve this, our proposed framework is coupled with the `p4est` library; this is a software package designed for managing parallel adaptive mesh refinement on forests of octrees (in 3D) or quadtrees (in 2D). Its core purpose is to provide a scalable and efficient way to represent and refine meshes for scientific computing applications that require dynamic resolution [167].

As a first disclaimer, **no** results in this dissertation have been generated using parallel implementations. The development of the parallel implementation, coupled with the unfitted methods available in `Gridap`, occurred simultaneously with the development of this work; however, there was not enough time in the project to carry out this task. The mesh adaptivity implementation for serial simulations has been developed and is used to generate the results presented in this work.

For adaptive mesh refinement using a forest of octrees, each cubic element is divided into eight equally sized cubic elements. In doing so, there will be an introduction of so called hanging nodes, edges, or faces. These lie on the interface between a coarse element and a refined element. The degrees of freedom within these hanging edges and faces are constrained linearly using the degrees of freedom of the neighboring coarse cell. The procedure for assigning linear constraints becomes quite involved for AgFEM, as we now consider both well-posed and ill-posed hanging degrees of freedom.

The adaptive mesh refinement procedure is defined a priori; we want to capture the solution with high accuracy near the free surface and close to the structure. Therefore, we

choose to refine all cut elements and the elements neighboring the cut elements n_{ref} times. Additionally, the free surface is refined only when the iteration is larger than $n_{\text{ref}} - 2$ times, so that the free surface, far away from the object, is not overly refined. The reason for this refined strategy is empirically based, and a more optimal approach may be possible. This process is summarised in pseudocode in Algorithm 1.

Algorithm 1 Refinement algorithm

```
Require: domain_model, geometry, flags
  for i in 1:n_ref do
      generate cell neighbor map
      cut(geometry)
      find indices of cut cells
      find indices of neighbors of cut cells using neighbor map
      set REFINE flag for indices
      if i > n_ref − 2 then
          generate face to cells map
          find face indices of faces at the free surface
          use face to cells map to identify indices of cells at the free surface
          set REFINE flag for indices
      end if
      new_model = adapt mesh using flags
      domain_model = new_model
  end for
  background_model = domain_model
```


Note that for the results with adaptive mesh refinement, we only consider the explicit geometrical representation. Additionally, for SBM and WSBM, we exclude the product of the distance function and the Hessian of the velocity potential, as this Hessian operator is currently not defined on the reference element. An alternative approach is to generate an unstructured mesh in GMSH [168], which is refined close to the free surface and structure as the Hessian is defined for that element type. Although this will require manual intervention to optimize the mesh for each case.

Validation with grid adaptivity

The domain dimensions and parameters are the same as those previously defined in Table 5.1. As an initial number of elements, we set the initial number of elements to $n = 8$ with a distribution of $(7n, 4n, n)$ for each direction. The number of refinement iterations is set to $n_{\text{ref}} = 4$, and a y-plane cross-section at $x = 0$ m, with a close up near the sphere, is depicted in Figure 5.12. The total number of elements in this adaptively refined mesh is 155071, which is a significant decrease compared to the 3.5 million for the Cartesian grid case.

The number of refinement iterations n_{ref} and time step size Δt are varied. In Figure 5.13, we depict the case with $n_{\text{ref}} = 4$ and $\Delta t = 0.0125$ s, and in Figure 5.14, the case with $n_{\text{ref}} = 5$ and $\Delta t = 0.05$ s is depicted. When comparing the results between these two plots with respect to the experimental solution, it becomes apparent that the time step size likely plays a larger role in the correctness of the solution than we had anticipated for the Cartesian grid

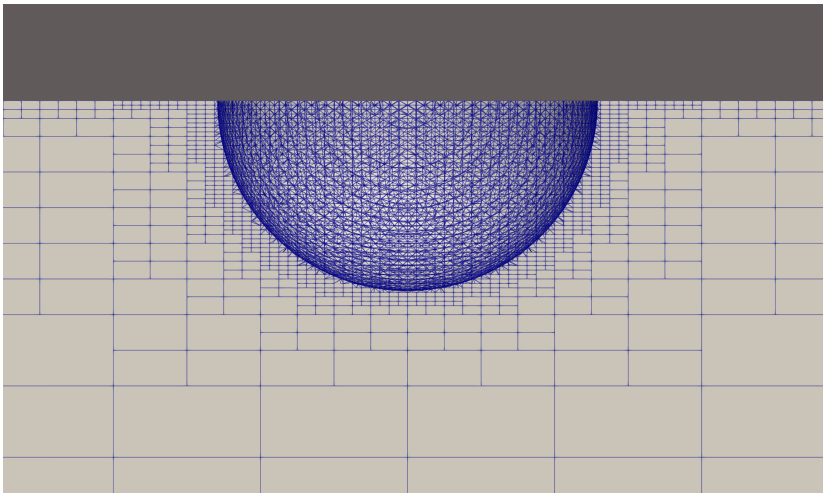


Figure 5.12: Adaptively refined mesh for AgFEM method, initial number of elements $n = 8$ and number of refinement iterations $n_{\text{ref}} = 4$.

case. Due to the large computational cost of the Cartesian grid approaches, the decision was made not to redo these simulations. We test the influence of the time step size on the solution accuracy by decreasing the time step size for the case with $n_{\text{ref}} = 5$ and thus setting $\Delta t = 0.0125$ s. The resulting plot is depicted in Figure 5.15, and it shows a closer resemblance to the experimental solution.

When comparing the unfitted methods, it is observed that, for all three plots, the WSBM has a larger amplitude than all the other unfitted methods. SBM, on the other hand, has a smaller amplitude, but the period of oscillations matches more closely to that of CutFEM and AgFEM. For the cases with $\Delta t = 0.0125$, there appears to be a slight frequency difference between CutFEM and AgFEM, which becomes more emphasized at the later time steps. Given the current setup of the framework, it is expected that both SBM and WSBM will differ from the true solutions as we miss the Hessian distance vector product, as well as gradient recovery techniques. For both CutFEM and AgFEM, the implementation should be complete and thus the results should not differ, except where method specific differences arise. Therefore, it might be interesting to understand why there is a slight frequency difference between the two.

For CutFEM and AgFEM, the implementation is complete, and no systematic error from missing terms is expected, unlike SBM and WSBM. Nevertheless, a small phase drift between the two methods accumulates over time and becomes visible at later time steps for $\Delta t = 0.0125$ s. This difference is not observed in the Cartesian grid results and appears consistent across refinement levels in the adaptive mesh setting, suggesting it is not a spatial resolution effect. The most likely cause is the ghost penalty stabilization, which is applied in CutFEM but not in AgFEM. Ghost penalty adds a mesh-dependent perturbation

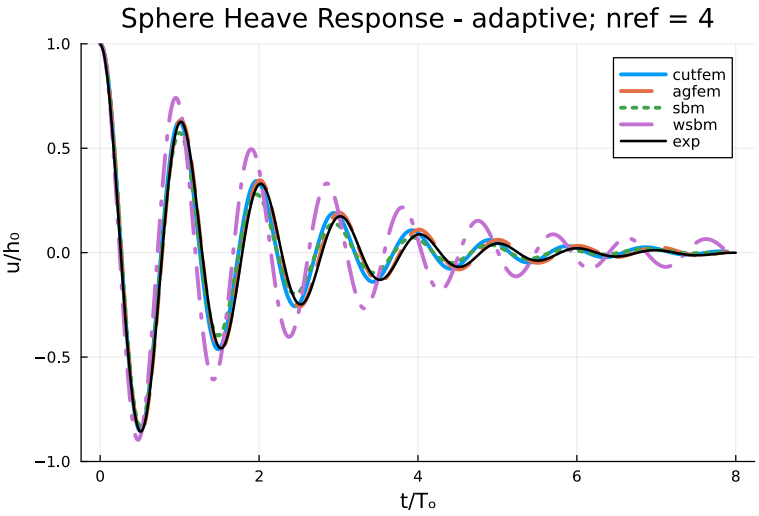


Figure 5.13: Results for adaptively refined grid starting with $n = 8$ and number of refinement iterations $n_{\text{ref}} = 4$ with time step size $\Delta t = 0.0125$ s.

to the bilinear form that may introduce a slight spurious modification to the effective stiffness, leading to a small shift in the observed frequency of oscillation that accumulates over time. AgFEM, which instead handles small cut cells through element aggregation, does not introduce this perturbation and produces a frequency closer to the experimental result. It should be noted, however, that the adaptive mesh implementation used here is still under active development and has not been fully validated; some part of the observed difference may therefore reflect implementation artefacts specific to the adaptive setting rather than a fundamental difference between the methods. A dedicated investigation comparing CutFEM with and without ghost penalty in the adaptive mesh setting is recommended to confirm this hypothesis.

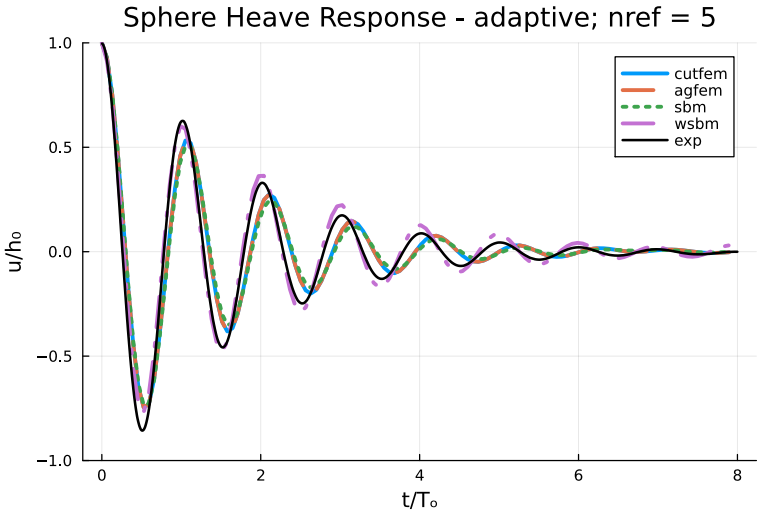


Figure 5.14: Results for adaptively refined grid starting with $n = 8$ and number of refinement iterations $n_{\text{ref}} = 5$ with time step size $\Delta t = 0.05$ s.

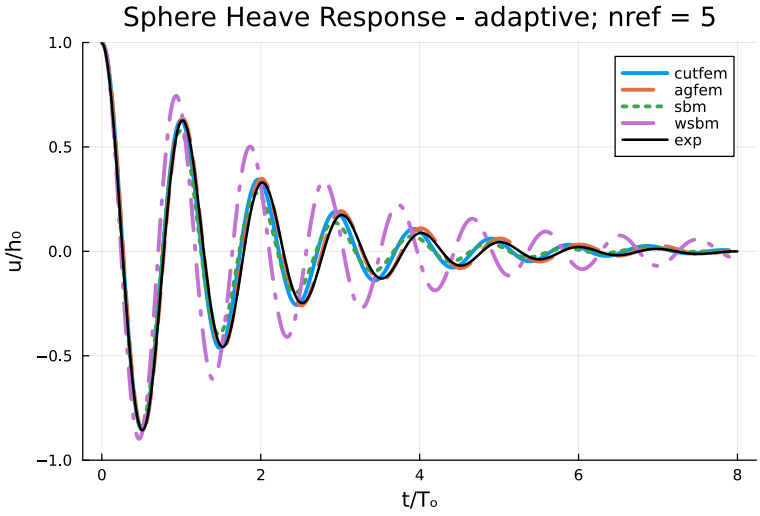


Figure 5.15: Results for adaptively refined grid starting with $n = 8$ and number of refinement iterations $n_{\text{ref}} = 5$ with time step size $\Delta t = 0.0125$ s.

5.3 Case 3: OC4 Phase II DeepCWind Semisubmersible

In this third case study, we extend the 3D problem from a simple geometry to a realistic geometry. The case considered is the OC4 DeepCWind Phase II semisubmersible, as defined by Robertson et al. [169]. The structure consists of three outer columns with heave plates at the bottom and a single main column in the center. The columns are connected via horizontal and diagonal cross members. We study free decay in the heave direction without considering the mooring configuration. The formulations in this work are based on linear potential flow; for this reason, the goal of this study is to recover the period of oscillation in free decay and not to match the displacements, as the amplitude is heavily influenced by viscous damping, which a linear potential flow formulation cannot capture without modifications, i.e., Morison elements or additional quadratic drag terms [169].

Domain and parameters

The domain is depicted in Figure 5.16; the dimensions are the same as those in the work by Wang et al. [170]. The domain length $L_x = 400$ m, breadth $L_y = 200$ m, and depth $d = 180$ m. The exact dimensions of the semisubmersible are presented by Robertson et al. [169], in this work, we use an STL to represent the semisubmersible. The corresponding mass and moments of inertia are as presented by Wang et al. [170]. All relevant parameters are defined in Table 5.2. The adaptive mesh refinement, as described in the previous section, is applied to this case study. We start with a number of elements of $n = 8$ with an initial distribution of $(2n, n, n)$ in (x, y, z) directions, after which a number of refinement iterations $n_{\text{ref}} = 4$ are applied. For the initial conditions, we consider $\phi = 0.0$ m^2/s, $\dot{\phi} = 0.0$ m^2/s^2, $h_0 = -2.2$ m and $\dot{h}_0 = 0.0$ m/s.

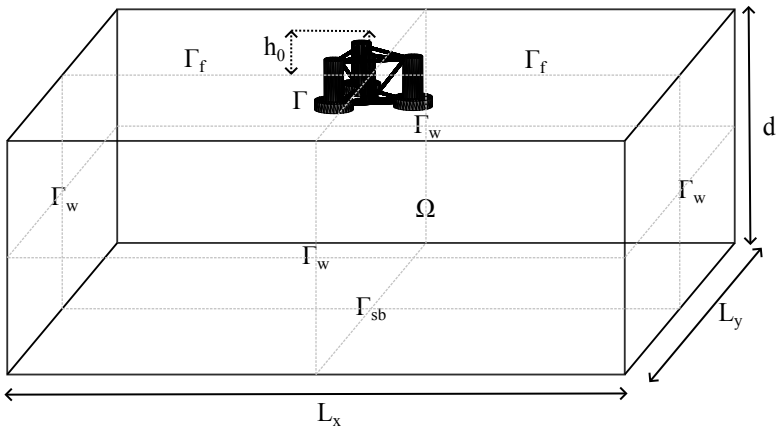


Figure 5.16: Case study 3: floating semisubmersible in 3D as described by Robertson et al. [169]. The depth $d = 180$ m, and the initial displacement $h_0 = -2.2$ m. The domain length $L_x = 400$ m, breadth $L_y = 200$ m and consider a total simulation time of $T_e = 100$ s. The domain Ω, structure boundary Γ, free surface boundary Γ_{f}, wall boundary Γ_{w}, and seabed boundary Γ_{sb}.

Validation

The validation study aims to reproduce the natural period in the heave direction for the DeepCWind semisubmersible, which is given as $T_o = 17.5$ s [170, 171]. Note that in these

Domain length: L_x	400 m
Domain breadth: L_y	200 m
Domain depth: d	180 m
Density of water: ρ	1025 kg/m^3
Gravitational constant: g	9.81 m/s^2
Mass: m	1.4268e7 kg
Moment of inertia (roll): I_{xx}	1.2898e10 kg·m^2
Moment of inertia (pitch): I_{yy}	1.2851e10 kg·m^2
Moment of inertia (yaw): I_{zz}	1.4189e10 kg·m^2

Table 5.2: Dimensions and relevant parameters for the third case study

references, the natural period includes contributions from the mooring lines, which we neglect in this work, such that no modifications to the weak formulations of Chapter 3 have to be made. According to Robertson et al. [169], a linearized three lined catenary mooring system can be included as an additional linearized restoring matrix. In the heave direction of motion, the load-displacement contribution of this restoring matrix is given as $1.91e4$ N/m, which we compare against the contribution of the hydrostatic restoring matrix given by $3.836e6$ N/m [169]. In the heave direction, the contribution of the linearized mooring system is about 0.5% that of the hydrostatic restoring force; we therefore justify that we can neglect the mooring line contribution. The increase in stiffness from the mooring system is small, and the heave natural period will be slightly shorter with mooring than without it.

$$T_n = 2\pi \sqrt{\frac{m + A_{33,\infty}}{C_{33}}} \tag{5.10}$$

Neglecting viscous effects impacts both both natural period and the damping ratio. To better understand this, we consider the difference between the undamped and damped natural periods. The theoretical undamped natural period $T_n = 17.02$ s is obtained using Equation 5.10, where m corresponds to the mass of the floater, $A_{33,\infty} = 1.47e7$ kg is the infinite frequency limit added mass in heave [169], and $C_{33} = 3.836e6$ N/m is the hydrostatic restoring component in heave. This constitutes a 3% difference between the theoretical undamped natural period and the reported reference natural period that includes mooring and viscous effects. Here, we assume that the reference natural period $T_o = 17.5$ s corresponds to the theoretical damped natural period. The relationship between the damped T_d and undamped T_n natural periods is given by Equation 5.11.

$$T_d = \frac{T_n}{(1 - \zeta^2)} \tag{5.11}$$

Here, ζ is the damping ratio, corresponding to a dimensionless damping strength. We obtain $\zeta \approx 0.23$ given the undamped and damped natural periods and using Equation 5.11, consistent with the significant viscous damping expected for this geometry due to flow separation at the heave plates. Note that our model will not show this damping ratio due to neglected mooring and viscous effects, and for this reason, we only aim to reproduce the reference natural period.

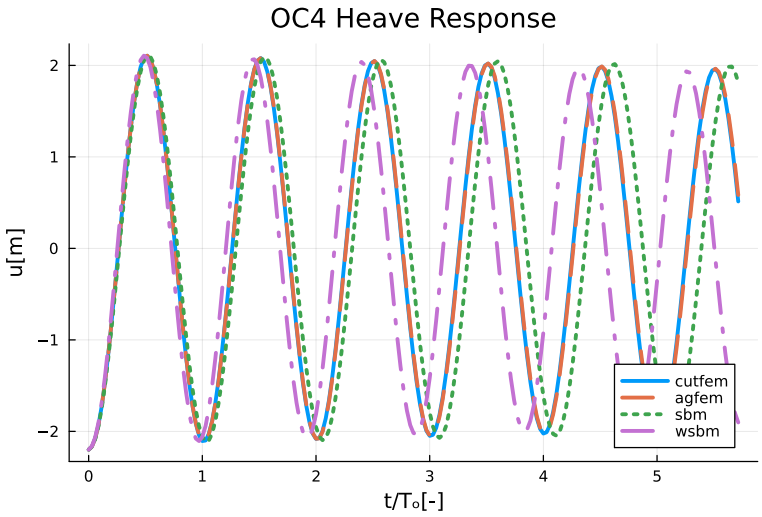


Figure 5.17: Free decay test heave displacement with initial displacement $h_0 = -2.2$ m and total simulation time of $T_e = 100$ s. The time step is $\Delta t = 0.5$ s and the time axis is normalized using the natural heave period $T_o = 17.5$ s.

In Figure 5.17, the free decay heave response is depicted, showing that both CutFEM and AgFEM approximately match the target period of oscillation. The SBM has a frequency difference that is slightly larger than the period of oscillation, while the WSBM has a slightly smaller frequency difference. This behaviour is also observed in the case studies on the cylinder and sphere and may be attributed to the missing shifting contribution with adaptive grids. These results indicate that for CutFEM and AgFEM, we are able to conduct simulations for realistic offshore geometries. For SBM and WSBM, a strong setup is presented, for which the required steps to improve the tool are known. It is further noted that the simulated response shows very little amplitude decay over the duration of the simulation. This is consistent with the linear potential flow formulation, which only accounts for radiation damping. For the OC4 semisubmersible, radiation damping in the heave direction is small relative to the viscous contributions arising from flow separation at the heave plates, meaning the total damping is significantly underestimated. Capturing the observed amplitude decay would require the inclusion of Morison elements or additional quadratic drag terms [169].

5.4 Case 4: Multi-Body Concept

In this section, we present a non-validated case study as proof of concept that this framework can be applied to simulations with multiple structures. Due to time constraints within this project, we are unable to identify and conduct a validation study; therefore, we can only demonstrate that we can simulate multi-body scenarios, showing that an influence is exerted on the second structure due to an initial displacement of the first structure.

The previous case study with the OC4 Phase II DeepCWind semisubmersible has been replicated, and the domain is extended by 200 m with the inclusion of a second

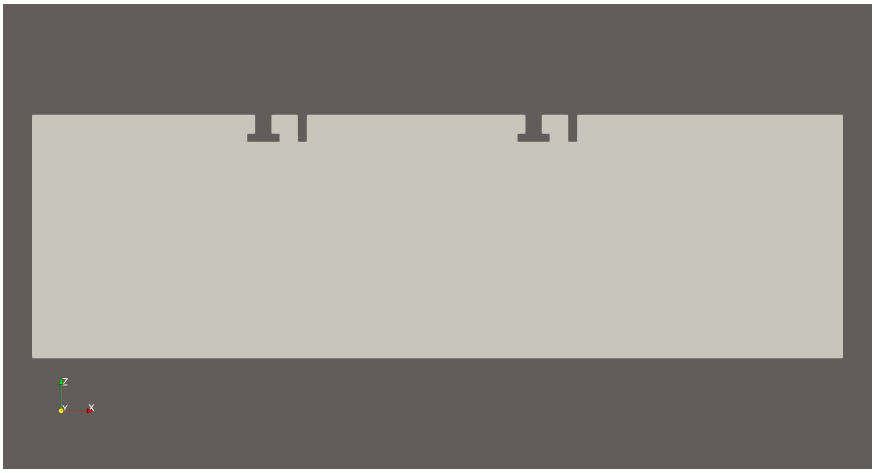


Figure 5.18: Slice of computational domain at $y = 0$ for the multi-body simulation case.

semisubmersible located at position $(x, y, z) = (200, 0, 0)$ m, as depicted in Figure 5.18. The first semisubmersible is the one on the left, and the second is the one on the right. The initial element distribution is given as $(3n, n, n)$ with $n = 2$ and $n_{\text{ref}} = 6$. With respect to the dimensions and parameters defined in Table 5.2, the only change is the domain length $L_x = 600$ m. The total simulation time is set to $T_e = 500$ s, with a time step of $\Delta t = 0.5$ s. Note that this simulation is only conducted using the AgFEM, and no comparisons across methods are made.

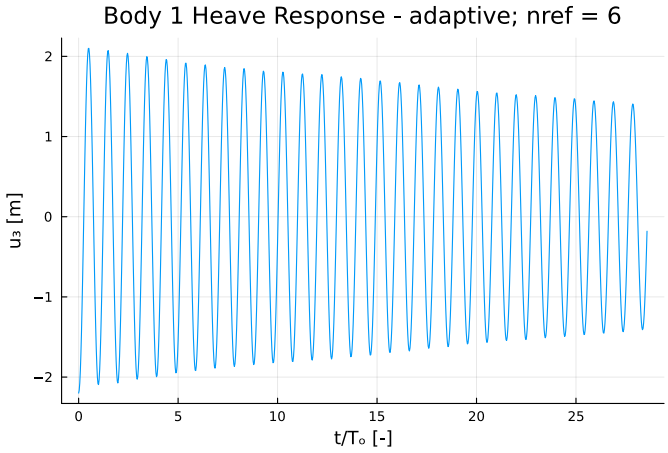


Figure 5.19: Heave displacement of the first semisubmersible given initial displacement $h_0 = -2.2$ m. Time is normalized using $T_o = 17.5$ s.

We disturb the first semisubmersible using the heave displacement $h_0 = -2.2$ m, as done before. Additionally, we consider $\phi = 0.0$ m^2/s, $\dot{\phi} = 0.0$ m^2/s^2 and $\dot{h}_0 = 0.0$ m/s. The

second semisubmersible is at equilibrium and is not displaced in any way. In doing so, the expectation is that the heaving motion of the first semisubmersible will generate waves that radiate towards the second semisubmersible and cause noticeable motion in either surge, heave, pitch, or any combination of those three directions. It is expected that there will be motion in the other directions, but it should be of negligible size.

In Figure 5.19, the heave displacement following the free decay test of the first semisubmersible is depicted. We normalize again the time again using the natural period of oscillation in heave $T_o = 17.5$ s, and we observe that the number of oscillations matches that of case study 3, but we also note a slight frequency difference that becomes more apparent at the later time steps, as the lowest point of each oscillation becomes more misaligned with the normalized time axis.

Due to the displacement of the first semisubmersible, we observe the displacements in the six degrees of freedom of the second semisubmersible, as depicted in Figure 5.20. The sway, roll, and yaw motions have significantly smaller amplitude signals than the surge, heave, and pitch motions, which thus match the expected behaviour. For the surge, heave, pitch, and yaw motions, it appears that the time signal oscillates around an offset from the zero position. The envelope of all signals appears to be consistent across the degrees of freedom.

In this proof of concept case study, we show that we can solve in the time domain for two floating structures that influence one another's motion via the fluid. Using this numerical framework, arrays of floating structures could be studied in the time domain, although a validation study should still be conducted.

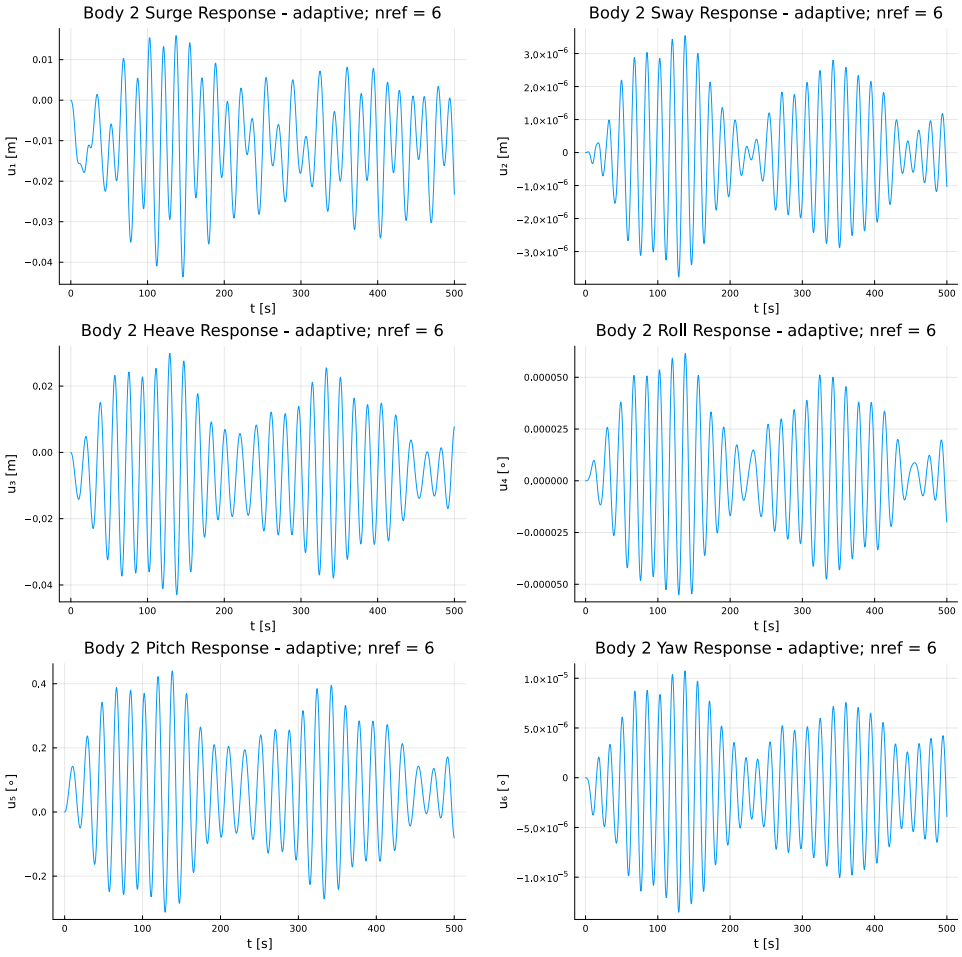


Figure 5.20: Six degrees of freedom displacements of the second semisubmersible, due to the free heave decay of the first semisubmersible with a spacing of 200 m between the two structures.

5.5 Conclusions

In this chapter, we present a computational framework that can perform time domain simulations for two-way coupled fluid-structure interaction using linear potential flow, with four unfitted methods to incorporate the structure. The framework can simulate in 2D using implicit geometrical representations; it can also simulate in 3D using both implicit and explicit geometrical representations. We show that similar solutions are obtained when these representations are compared. Additionally, in 3D, the computational requirements become significantly large when using Cartesian grids; therefore, we propose an adaptively refined mesh using the `p4est` library [167], and we validate its implementation using the spherical case study. We show that both adaptive mesh refinement and finer time step sizes correspond to solutions that closely resemble the experimental results from Kramer et al.

[160]. Then, we show that for a realistic offshore floater geometry, the DeepCWind OC4 semisubmersible [169], we can recover the natural period of oscillation. We are unable to recover the amplitude of oscillation, as the geometry is such that viscous damping effects cannot be neglected. Additionally, we demonstrate that within the computational framework, we can conduct multi-body simulations, although a validation study should still be conducted.

When comparing the unfitted methods, it is noted that, with sufficient mesh refinement and polynomial order, similar results are obtained across methods. CutFEM and AgFEM are least affected by the differences introduced by a coarser mesh. For SBM and WSBM, it is essential to employ the gradient recovery technique to improve solution quality with fewer elements; this should be implemented and studied. For WSBM, it is observed that spurious oscillations and divergence can occur; however, for grids that are adaptively refined, this behaviour is not observed. The cause of this behaviour has not been studied, but it might be related to element size. In computational terms, mesh adaptivity is an absolute must have improvement; therefore, it is important that the Hessian operator on the reference element is defined in such a way that a complete comparison with fully implemented SBM and WSBM can be conducted.

Lastly, as mentioned, no results here are generated using parallel solvers; however, development has been ongoing in this direction, and it is a logical next step to implement, so that this serves as a complete foundation for a numerical toolbox for conducting time domain simulations with linear potential flow.

6

Estimation of Hydrodynamic Coefficients

In offshore hydrodynamics, the accurate prediction of the motion response of floating structures critically depends on the characterization of hydrodynamic coefficients, particularly the added mass and added damping. These coefficients describe how the surrounding fluid modifies the inertia and damping of a body as it moves through water: added mass accounts for the additional inertia associated with accelerating the fluid, while added damping represents the energy dissipated through radiated waves. Together, these are used when determining the natural frequencies, motion amplitudes, and wave-induced loads on submerged systems, such as offshore platforms or floating wind turbines. Reliable estimation of these quantities is therefore essential for the safe design, performance assessment, and dynamic analysis of offshore structures. Using the finite element method to obtain these coefficients is not strictly novel; the work by Wang et al. [86] used an extended finite element method to obtain these coefficients for simple geometries in 2D. In this work, the novelty lies in reproducing these results in 2D using three different unfitted finite element methods. Additionally, an extension to 3D is introduced for complex geometries. In this chapter, the unfitted finite element methods discussed in previous chapters are applied to estimate the hydrodynamic added mass and added damping coefficients. In the first section, the derivation of these coefficients is described for both single- and multi-body problems. Then, a verification study is conducted in 2D for simple geometries using CutFEM, AgFEM, and SBM. Next, the AgFEM is applied to realistic offshore geometries and verified using the OC3 Hywind SPAR support structure and the OC4 DeepCWind support structure.

6.1 Estimation of Added Mass and Added Damping

6.1.1 Single Body with Six Degrees of Freedom

The discrete form of Equation 3.6 can be expressed in an equivalent block matrix form as follows:

$$\begin{pmatrix} \mathbb{A}_{w\phi} & \mathbb{A}_{wu} \\ \mathbb{A}_{v\phi} & \mathbb{A}_{vu} + \mathbb{C}_{vu} \end{pmatrix} \begin{pmatrix} \boldsymbol{\Phi} \\ \boldsymbol{U} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F}_{\phi} \\ \boldsymbol{F}_{u} \end{pmatrix}, \tag{6.1}$$

with

$$\mathbb{A}_{w\phi} := (\nabla w_h, \nabla \phi_h)_{\Omega_h} - \frac{\omega^2}{g} (w_h, \phi_h)_{\Gamma_{\mathrm{f},h}}, \tag{6.2a}$$

$$\mathbb{A}_{wu} := i\omega \left(w_h, \mathbf{u}_h \cdot \boldsymbol{n}_{\Gamma,j} \right)_{\Gamma_h}, \tag{6.2b}$$

$$\mathbb{A}_{v\phi} := -i\omega \left(\mathbf{v}_h, \phi_h \boldsymbol{n}_{\Gamma,j} \right)_{\Gamma_h}, \tag{6.2c}$$

$$\mathbb{A}_{vu} := -\omega^2 \frac{\boldsymbol{M}_{\rho}}{|\Gamma|} (\mathbf{v}_h, \mathbf{u}_h)_{\Gamma_h}, \tag{6.2d}$$

$$\mathbb{C}_{vu} := g \left(\mathbf{v}_h, (\boldsymbol{n}_z \cdot \mathbf{u}_h) \boldsymbol{n}_{\Gamma,j} \right)_{\Gamma_h}, \tag{6.2e}$$

$$\boldsymbol{F}_{\phi} := (w_h, V_{\mathrm{w}})_{\Gamma_{\mathrm{w},h}}, \tag{6.2f}$$

$$\boldsymbol{F}_{u} := \mathbf{0}. \tag{6.2g}$$

In Equation 6.1, $\boldsymbol{\Phi}$ is the complex-valued block vector of degrees of freedom associated with the discrete velocity potential, and $\boldsymbol{U} := \mathbf{u}_h$ is the complex-valued block vector associated with the body displacement. We use bold symbols to denote vectorial quantities and blackboard bold to denote matrices. Note that the matrices $\mathbb{A}_{vu}$ and $\mathbb{C}_{vu}$, and the vector $\boldsymbol{F}_u$ are of dimensions $d_j \times d_j$ and $d_j \times 1$, respectively, with $d_j \leq 6$. As, in 2D, these terms consist of two translational and one rotational degree of freedom, i.e., surge, heave, and pitch; and in 3D, there are three translational and three rotational degrees of freedom allowed, i.e., surge, sway, heave, roll, pitch, and yaw.

We can reduce the algebraic system Equation 6.1 to a single equation, condensing the contribution from $\boldsymbol{\Phi}$, which leads to

$$\left(\mathbb{A}_{vu} + \mathbb{C}_{vu} - \mathbb{A}_{v\phi} \mathbb{A}_{w\phi}^{-1} \mathbb{A}_{wu} \right) \boldsymbol{U} = \boldsymbol{F}_u - \mathbb{A}_{v\phi} \mathbb{A}_{w\phi}^{-1} \boldsymbol{F}_{\phi}. \tag{6.3}$$

$$\left[-\omega^2 (\mathbf{M} + \mathbf{A}) - i\omega \mathbf{B} + \mathbf{C} \right] \mathbf{u} = \mathbf{f} \tag{6.4}$$

Here, Equation 6.3 is equivalent to the system of equations of motion for a spring-mass-damper system with six degrees of freedom, as given in Equation 6.4. The first term A_{vu} corresponds to the inertia term of the body; the second term C_{vu} is the spring stiffness given by the hydrostatic restoring force; and the third term $\mathbb{A}_{v\phi} \mathbb{A}_{w\phi}^{-1} \mathbb{A}_{wv}$ corresponds to the added inertia and damping experienced by the object due to the surrounding fluid. The equivalency is represented in Equation 6.5.

$$\mathbb{A}_{v\phi} \mathbb{A}_{w\phi}^{-1} \mathbb{A}_{wu} = \omega^2 \mathbf{A} + i\omega \mathbf{B} \tag{6.5}$$

For the radiation problem considered here, we set the right-hand side of Equation 6.3 to zero. Therefore, we obtain the equation for added mass in Equation 6.6 and the equation for added damping in Equation 6.7.

$$\mathbf{A} = \frac{Re\left(\mathbb{A}_{v\phi}\mathbb{A}_{w\phi}^{-1}\mathbb{A}_{wu}\right)}{\omega^2} \tag{6.6}$$

$$\mathbf{B} = \frac{Im\left(\mathbb{A}_{v\phi}\mathbb{A}_{w\phi}^{-1}\mathbb{A}_{wu}\right)}{\omega} \tag{6.7}$$

The non-dimensionalized added mass and added damping are stated in Equation 6.8 and Equation 6.9. Here we have radial frequency ω, density ρ, and submerged volume or area $\forall$ in three and two-dimensions, respectively.

$$\bar{\mathbf{A}} = \frac{\mathbf{A}}{\rho\forall} \tag{6.8}$$

$$\bar{\mathbf{B}} = \frac{\mathbf{B}}{\omega\rho\forall} \tag{6.9}$$

6.1.2 Multi-Body

For completeness, we propose a multi-body formulation: we can extend the block matrix form depicted in Equation 6.1 to the following system of equations:

$$\begin{pmatrix} \mathbb{A}_{w\phi} & \mathbb{A}_{wu_1} & \cdots & \mathbb{A}_{wu_q} & \cdots & \mathbb{A}_{wu_N} \\ \mathbb{A}_{v_1\phi} & \mathbb{A}_{v_1u_1} + \mathbb{C}_{v_1u_1} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & & & \vdots \\ \mathbb{A}_{v_q\phi} & 0 & & \mathbb{A}_{v_qu_q} + \mathbb{C}_{v_qu_q} & \cdots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ \mathbb{A}_{v_N\phi} & 0 & \cdots & 0 & \cdots & \mathbb{A}_{v_Nu_N} + \mathbb{C}_{v_Nu_N} \end{pmatrix} \begin{pmatrix} \boldsymbol{\Phi} \\ \boldsymbol{U}_1 \\ \boldsymbol{U}_q \\ \vdots \\ \boldsymbol{U_N} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F}_\phi \\ \boldsymbol{F}_{u_1} \\ \boldsymbol{F}_{u_q} \\ \vdots \\ \boldsymbol{F}_{u_N} \end{pmatrix}, \tag{6.10}$$

with

$$\mathbb{A}_{w\phi} := (\nabla w_h, \nabla\phi_h)_{\Omega_h} - \frac{\omega^2}{g}(w_h, \phi_h)_{\Gamma_{\mathrm{f},h}}, \tag{6.11a}$$

$$\mathbb{A}_{wu_q} := i\omega\left(w_h, \mathbf{u}_{h,q}\cdot\boldsymbol{n}_{\Gamma,j}\right)_{\Gamma_h}, \tag{6.11b}$$

$$\mathbb{A}_{v_q\phi} := -i\omega\left(\mathbf{v}_{h,q}, \phi_h\boldsymbol{n}_{\Gamma,j}\right)_{\Gamma_h}, \tag{6.11c}$$

$$\mathbb{A}_{v_qu_q} := -\omega^2\frac{M_\rho}{|\Gamma|}\left(\mathbf{v}_{q,h}, \mathbf{u}_{q,h}\right)_{\Gamma_h}, \tag{6.11d}$$

$$\mathbb{C}_{v_qu_q} := g\left(\mathbf{v}_{q,h}, \left(\boldsymbol{n}_z\cdot\mathbf{u}_{q,h}\right)\boldsymbol{n}_{\Gamma,j}\right)_{\Gamma_h}, \tag{6.11e}$$

$$\boldsymbol{F}_\phi := (w_h, V_{\mathrm{w}})_{\Gamma_{\mathrm{w},h}}, \tag{6.11f}$$

$$\boldsymbol{F}_{u,q} := \mathbf{0}. \tag{6.11g}$$

Here, we use the iterator $q = 1, ..., N$ for the number of rigid bodies. Additionally, we have already set all off-diagonal terms between the bodies $\mathbb{A}_{v_{q_1} u_{q_2}} = 0 \quad \forall q_1 \neq q_2$ as these are only coupled by the fluid via the velocity potential. Similarly, to Equation 6.3, the contribution assumed to be of Φ is substituted into the system of equations for the bodies:

$$\begin{pmatrix} \mathbb{A}_{v_1 u_1} + \mathbb{C}_{v_1 u_1} - \mathbb{X}_{11} & -\mathbb{X}_{1q} & \cdots & -\mathbb{X}_{1N} \\ -\mathbb{X}_{q1} & \mathbb{A}_{v_q u_q} + \mathbb{C}_{v_q u_q} - \mathbb{X}_{qq} & \cdots & -\mathbb{X}_{qN} \\ \vdots & \vdots & \ddots & \vdots \\ -\mathbb{X}_{N1} & -\mathbb{X}_{Nq} & \cdots & \mathbb{A}_{v_N u_N} + \mathbb{C}_{v_N u_N} - \mathbb{X}_{NN} \end{pmatrix} \begin{pmatrix} \boldsymbol{U}_1 \\ \boldsymbol{U}_q \\ \vdots \\ \boldsymbol{U_N} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F}_{u_1} - \mathbb{A}_{v_1 \phi} \mathbb{A}_{w\phi}^{-1} \boldsymbol{F}_{\phi} \\ \boldsymbol{F}_{u_q} - \mathbb{A}_{v_q \phi} \mathbb{A}_{w\phi}^{-1} \boldsymbol{F}_{\phi} \\ \vdots \\ \boldsymbol{F}_{u_N} - \mathbb{A}_{v_N \phi} \mathbb{A}_{w\phi}^{-1} \boldsymbol{F}_{\phi} \end{pmatrix}, \tag{6.12}$$

with

$$\mathbb{X}_{q_1 q_2} = \mathbb{A}_{v_{q_1} \phi} \mathbb{A}_{w\phi}^{-1} \mathbb{A}_{w u_{q_2}}. \tag{6.13}$$

The system of equations in Equation 6.12 is equivalent to N six degree of freedom spring-mass-damper systems. Hence, we can rewrite the definitions that describe the dynamics of added mass and added damping as follows:

$$\mathbf{A}_{q_1 q_2} = \frac{Re\left(\mathbb{X}_{q_1 q_2}\right)}{\omega^2} \tag{6.14}$$

$$\mathbf{B}_{q_1 q_2} = \frac{Im\left(\mathbb{X}_{q_1 q_2}\right)}{\omega} \tag{6.15}$$

Thus, we end up with N^2 added mass and added damping matrices of size 6×6, where the diagonal terms $q_1 = q_2$ consist of the added mass and added damping experienced by each structure due to the surrounding fluid, and the off-diagonal terms $q_1 \neq q_2$ represent the added mass and added damping that the structure experiences via the surrounding fluid because of the other structures. The non-dimensional added mass and added damping are generally obtained using the same procedure as in equations Equation 6.8 and Equation 6.9. In some works, the added mass and added damping are also divided by the total number of structures, which corresponds to the summed total displaced volume of all structures. The computational bottleneck in this approach is the matrix inversion in Equation 6.13.

6.2 Single Body - Simple Geometries

In this section, CutFEM, AgFEM, and SBM are employed to estimate the added mass and added damping for simple geometries, namely a heaving rectangle and cylinder. Our approach is verified using analytical methods and numerical experiments.

Ghost penalty for Neumann-type boundary

In the case of purely Neumann-type boundary conditions, ghost penalty stabilization terms may be unnecessary. Thus, an initial CutFEM study examines two scenarios: one without and one with these stabilization terms, varying the parameter γ. The impact on the condition number via the L1 norm is analyzed for increasing elements n and varying submergence depth H. The geometry is a horizontal cylinder with a submergence depth adjusted to create small cut cells. The mesh is a Cartesian grid with n elements and a rectangular domain setup, having $(3n, n)$ elements in x and z directions.

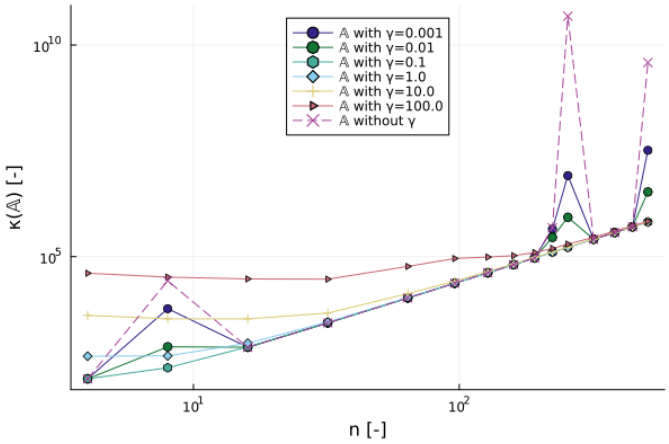


Figure 6.1: The condition number $\kappa(\mathbb{A})$ for varying number of elements n between a CutFEM approach with varying ghost penalty parameters and one without ghost penalty stabilization.

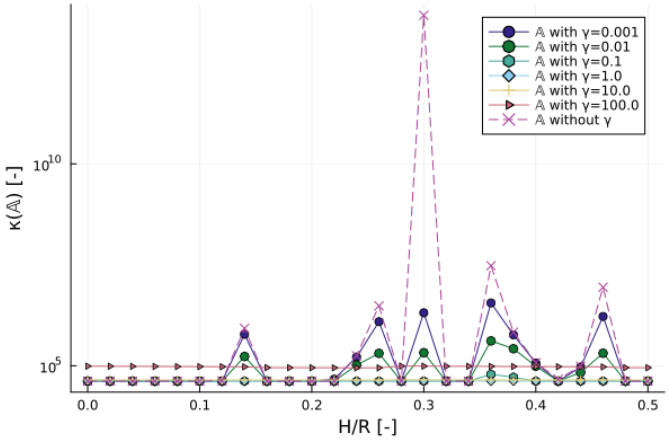


Figure 6.2: The condition number $\kappa(\mathbb{A})$ for varying submergence depth ratios H/R between a CutFEM approach with ghost penalty parameters and one without ghost penalty parameters.

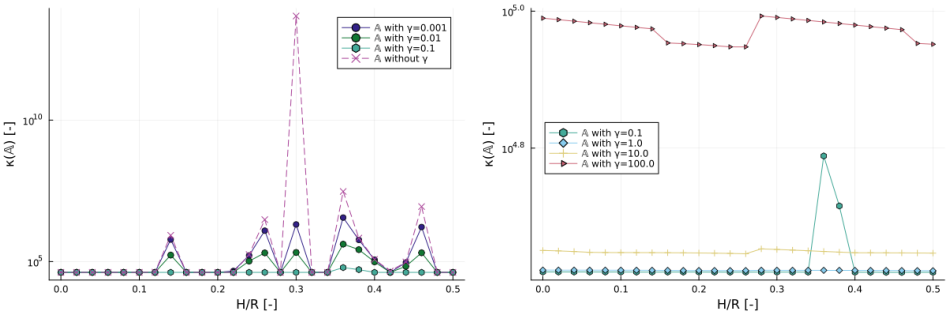


Figure 6.3: The condition number $\kappa(\mathbb{A})$ for varying submergence depth ratios H/R with a close up view on the smaller (left) and larger (right) ghost penalty parameters γ.

In Figure 6.1, the relationship between the number of elements n and the condition number $\kappa(\mathbb{A})$ is illustrated, demonstrating that the ghost penalty effectively reduces the condition number in certain instances, particularly those involving smaller cut cells. It is observed that the larger stabilization parameters $\gamma > 1.0$ generally result in elevated condition numbers. This is particularly evident for both lower and higher numbers of elements, although the disparity diminishes with an increase in elements. Conversely, the smaller stabilization parameters $\gamma < 0.1$ exhibit a greater susceptibility to the ill-conditioning associated with small cut cells. Analysis of Figure 6.1 allows the conclusion that $\gamma = 0.1$ and $\gamma = 1.0$ exhibit optimal performance. In Figure 6.2, the condition number is depicted as a function of varying submergence distances in the context of $n = 128$. Ideally, when the submergence distance approximates a multiple of the cell length scale, one anticipates relatively higher condition numbers, as evidenced by the five peaks in Figure 6.2. The susceptibility of smaller stabilization parameters to ill-conditioning is further underscored in Figure 6.3. Conversely, the larger stabilization parameters appear to avoid ill-conditioning yet tend to exhibit higher condition numbers, as emphasized in Figure 6.3.

The present study verifies the necessity of employing ghost penalty stabilization to mitigate the potential for matrix ill-conditioning. Furthermore, it suggests that the original stabilization penalty parameter, as proposed in $\gamma = 0.1$, is adequate for the scenarios analyzed in this chapter.

6.2.1 Heaving Rectangle

A two-dimensional surface-piercing rectangle is examined, characterized by a draft D and a breadth B. This study follows the same geometrical setup as previously investigated by Wang et al. [86] and Liang et al. [172]. The ratio of draft to breadth considered is $D/B = 2$. The domain dimensions consist of a depth $d = 40D$ and a length $L_x = 2\lambda_{max}$ for the minimum non-dimensional wave number $\bar{k} = 0.1$. No alterations in the submergence distance are contemplated in this analysis, signified by $H = 0.0$.

Figure 6.4 shows the mesh for calculating hydrodynamic coefficients using the methods described in chapter 3. The background mesh, refined near the object, does not conform to the rectangle. Figure 6.5 offers a close-up of the heaving rectangle triangulations: the

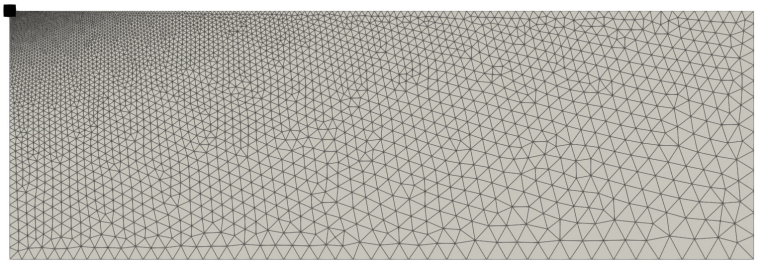


Figure 6.4: Background mesh for the heaving rectangle.

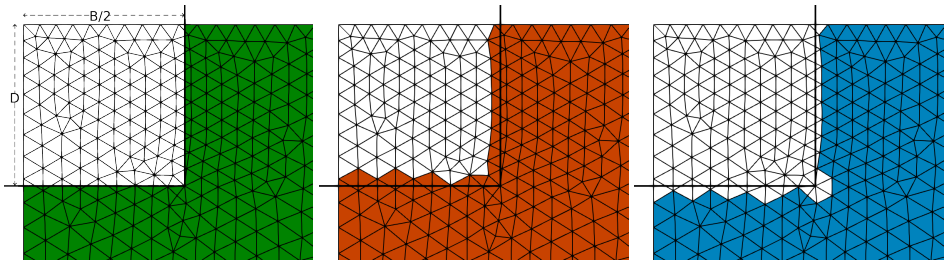


Figure 6.5: Close up view of the heaving rectange triangulations: physical triangulation, i.e. integration domain for AgFEM and CutFEM (left), active triangulation used by AgFEM and CutFEM (center) and surrogate triangulation used for the SBM (right).

left shows the integration domain for AgFEM and CutFEM, the middle displays the active domain for finite element space generation, and the right presents the surrogate domain for integration and finite element space generation for the SBM method.

Verification study

The verification study is conducted for both linear and quadratic element types. For linear elements, the mesh consists of 48105 elements, while for quadratic elements, the mesh consists of 8217 elements. The results are depicted in Figure 6.6 and Figure 6.7 for linear and quadratic element types.

All unfitted FE methods follow the general trend dictated by the reference methods. For the added damping, the results agree well; the differences are more noticeable for the added mass coefficient. One general observation is that the SBM is a visible outlier in this scenario. For SBM with order $p_e = 1$, we anticipate a difference due to the missing gradient recovery approach. For SBM $p_e = 2$, we see that the results match more accurately, indicating the correct shifting of the gradient. However, for SBM $p_e = 2$, the added damping coefficient is more underestimated than for SBM $p_e = 1$, although the differences with the other methods are small. For AgFEM $p_e = 1$ and CutFEM $p_e = 1$, the added mass appears to be slightly underestimated; however, it becomes indistinguishable for $p_e = 2$.

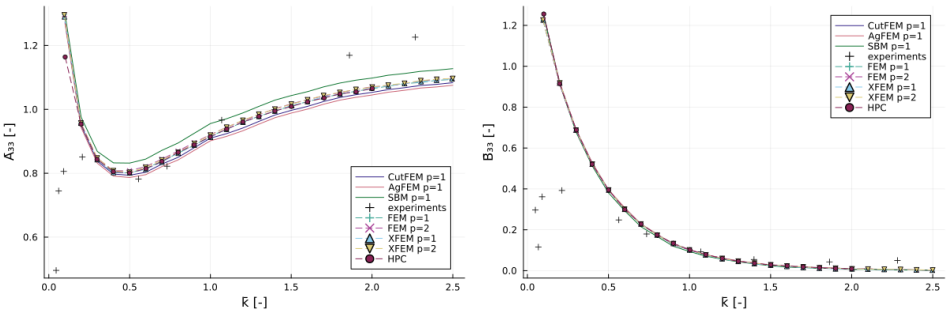


Figure 6.6: Non-dimensional wave number $\bar{k}$ versus non-dimensional added mass $\bar{A}_{33}$ and added damping $\bar{B}_{33}$ for elements of order $p_e = 1$ with reference data from [86] [172] [173].

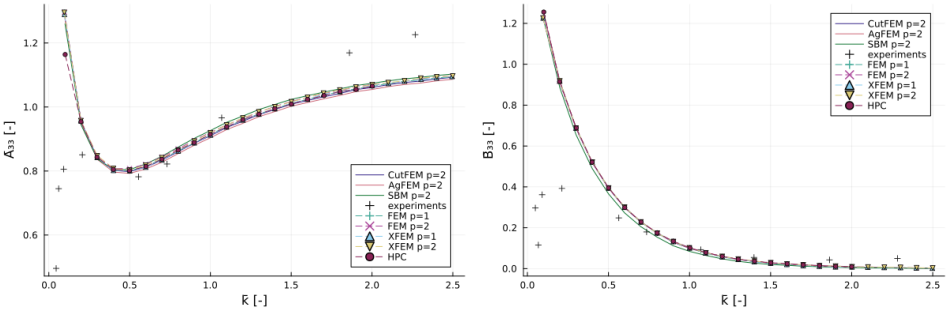


Figure 6.7: Non-dimensional wave number $\bar{k}$ versus non-dimensional added mass $\bar{A}_{33}$ and added damping $\bar{B}_{33}$ for elements of order $p_e = 2$ with reference data from [86] [172] [173].

6.2.2 Horizontal Cylinder

We consider a two-dimensional surface-piercing horizontal cylinder of radius R. The domain dimensions are: depth $d = 20R$ and length $L_x = \lambda_{max}$. We determine λ_{max} using the non-dimensional wave number $\bar{k} = 0.15$. The horizontal cylinder is partially submerged to the submergence distance H using the following ratios of submergence depth over radius: $H/R = 0.0, 0.342, 0.643, 0.809$ and 0.906. The submerged surface area $\forall$ is set as the full surface area of the cylinder in our computational domain, and thus $\forall = \pi R^2/2$. An immediate advantage of the unfitted methods to take note of is that we only require a single background mesh for all submerged cases. In Figure 6.8, we show the mesh used for the study of the partially submerged cylinder. As in the previous section, the mesh is refined towards the body, increasing accuracy without having to generate a body-fitted grid. A close-up view of the background mesh, together with the physical, active, and surrogate triangulations, is depicted in Figure 6.9.

In Figure 6.10, four close-up views of the meshes are depicted for varying depths of submergence for the horizontal cylinder, highlighting the surrogate domain used in the SBM method. Note that in this figure, the mesh is used for illustration purposes; a higher resolution mesh is used for the calculations in this section. For $p_e = 1$, the mesh has 449485 elements, and for $p_e = 2$, the mesh has 162203 elements.

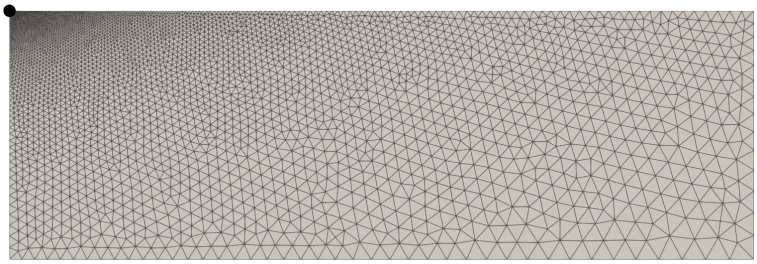


Figure 6.8: Background mesh for the submerged cylinder.

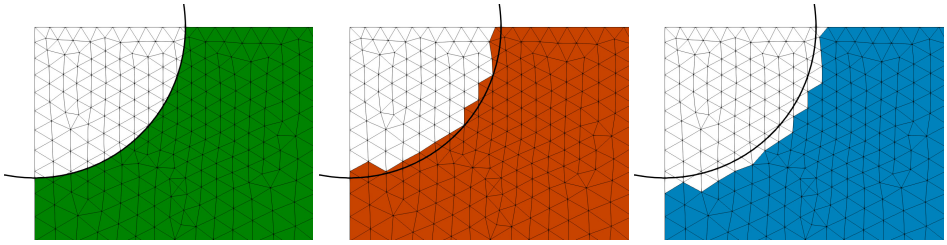


Figure 6.9: Close up view of the submerged cylinder triangulations for the case $H/R = 0.0$: physical triangulation, i.e. integration domain for AgFEM and CutFEM (left), active triangulation used by AgFEM and CutFEM (center) and surrogate triangulation used for the SBM (right).

Verification study

The reference data used for validation consists of two complementary analytical formulations, both detailed in [174]. The first is the Relative Motion Hypothesis (RMH), which estimates added mass and added damping by treating the partially submerged body as though it undergoes motion relative to an otherwise undisturbed fluid. Combined with the Haskind relation — which links the excitation force on a body to the far-field radiation potential — and the asymptotic form of the Kramers-Kronig relation, which enforces causality constraints between added mass and damping, this approach yields closed-form estimates for both coefficients. The underlying assumption is that the waterplane length remains equal to that of the surface-piercing case ($H/R = 0.0$) regardless of submergence depth. This assumption is reasonable for small submergence depths, but causes the method to lose accuracy at larger depths and higher wave numbers [174].

The second formulation, the Equivalent Wavemaker Theory (EWT), is used specifically for added damping. It replaces the oscillating structure with an equivalent wavemaker whose velocity field matches that generated by the body at high frequencies. This approach is in principle valid over a wider range of wave numbers; however, at intermediate wave numbers and for larger submergence depths, it too loses accuracy, providing reliable estimates only in the low wave number limit and asymptotically as the wave number approaches infinity [174].

These analytical methods were selected as reference data because, to the authors' knowledge, they represent a complete analytical treatment available in the literature for the specific problem of a partially submerged horizontal cylinder at varying submergence depths. The RMH and EWT formulations of [174] are specifically derived for this configuration and therefore constitute as an appropriate available benchmark, notwithstanding their known limitations at larger submergence depths. These limitations are accounted for in the interpretation of the results, where quantitative comparisons are restricted to the range of wave numbers for which the reference data is considered reliable.

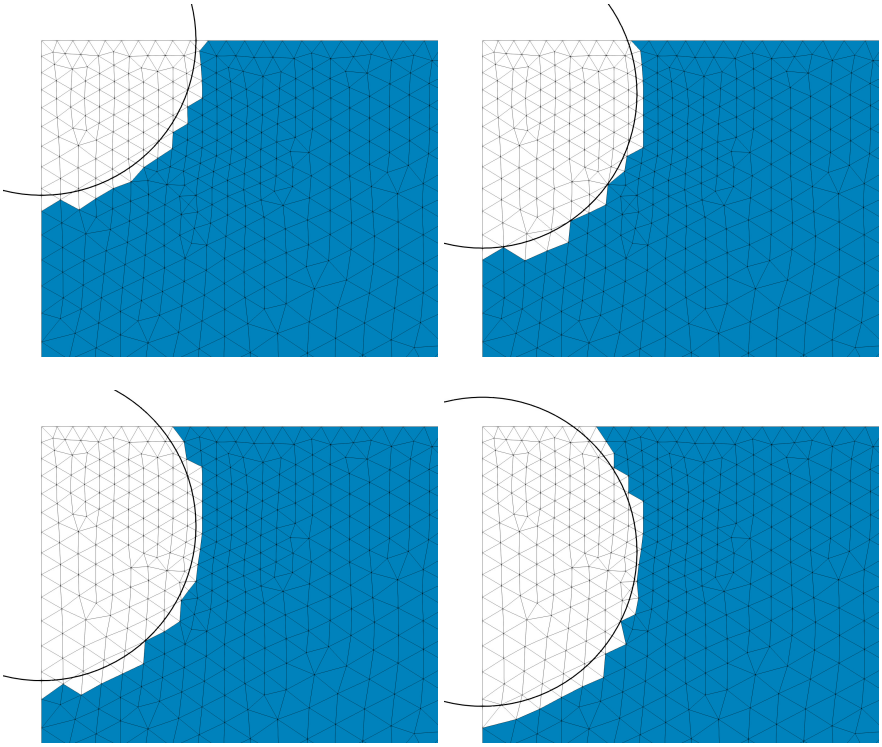


Figure 6.10: Close up view of the submerged cylinder triangulations for the cases $H/R = 0.0$ (top left), $H/R = 0.342$ (top right), $H/R = 0.643$ (bottom left) and $H/R = 0.809$ (bottom right).

The results of this case study are depicted in Figure 6.11-Figure 6.15 for Lagrangian type elements of order $p_e = 1$, with each plot depicting non-dimensional wave number versus the added mass and added damping at increasing submergence depths. For elements of order $p_e = 2$, the results are depicted in Figure 6.16-Figure 6.20. All unfitted finite element methods capture the qualitative trends present in the reference data, though a quantitative assessment is complicated by the known limitations of the analytical reference at larger submergence depths. Especially for small depths of submergence, the results align with the reference; however, for larger submergence depths, it appears that the unfitted methods differ more from the added mass and added damping coefficients in comparison

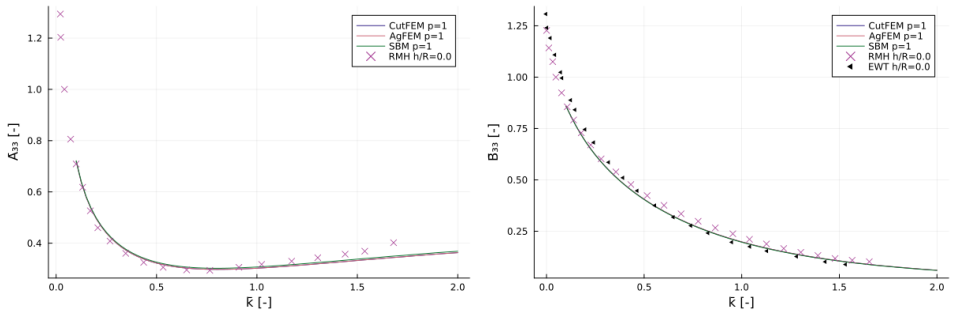


Figure 6.11: Non-dimensional wave number $\bar{k}$ versus non-dimensional added mass $\bar{A}_{33}$ and added damping $\bar{B}_{33}$ for submergence distance $H/R = 0.0$ with reference data from [174].

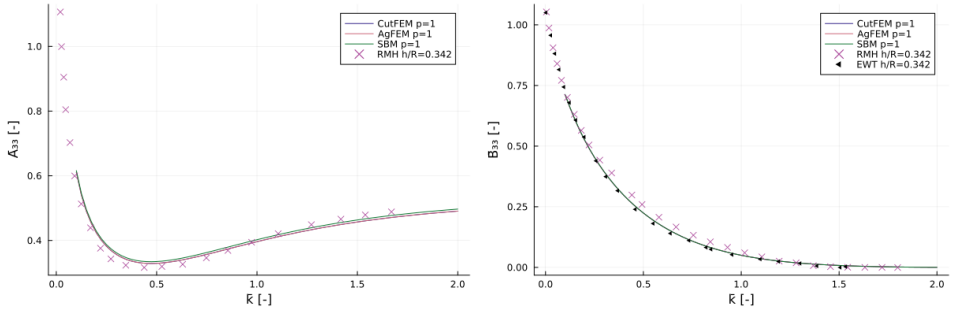


Figure 6.12: Non-dimensional wave number $\bar{k}$ versus non-dimensional added mass $\bar{A}_{33}$ and added damping $\bar{B}_{33}$ for submergence distance $H/R = 0.342$ with reference data from [174].

with the reference. This is likely due to the definition of the reference data, which is no longer valid for these submergence depths. The AgFEM and CutFEM results appear to be nearly identical for the cases considered in this study. Given the spread between the numerical results and the analytical reference — itself subject to modeling assumptions that break down at higher submergence depths — no definitive conclusion regarding the relative accuracy of the unfitted methods can be drawn from this particular benchmark alone. AgFEM and CutFEM display results that are close to identical for both first and second order elements. Compared to the analytical reference, the SBM method tends to underestimate or overestimate the hydrodynamic coefficients — an observation that should be interpreted with caution, given that the reference itself loses validity at larger submergence depths and intermediate wave numbers. This is more prominent in the added mass coefficient. This can be caused by the lack of resolution of the volume around the object and the loss of accuracy due to the expansion of the gradient. It is especially prevalent in the higher order case, which has fewer elements than the first order setup. Possible measures to mitigate these errors include the use of gradient recovery techniques as proposed in [140] or the weighted SBM proposed in [129, 130].

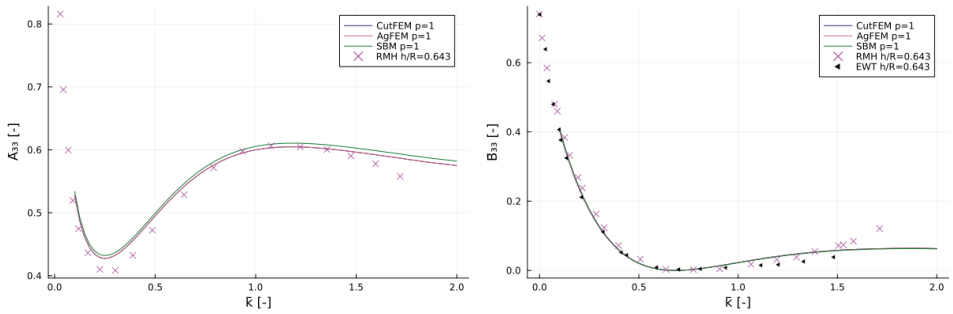


Figure 6.13: Non-dimensional wave number $\bar{k}$ versus non-dimensional added mass $\bar{A}_{33}$ and added damping $\bar{B}_{33}$ for submergence distance $H/R = 0.643$ with reference data from [174].

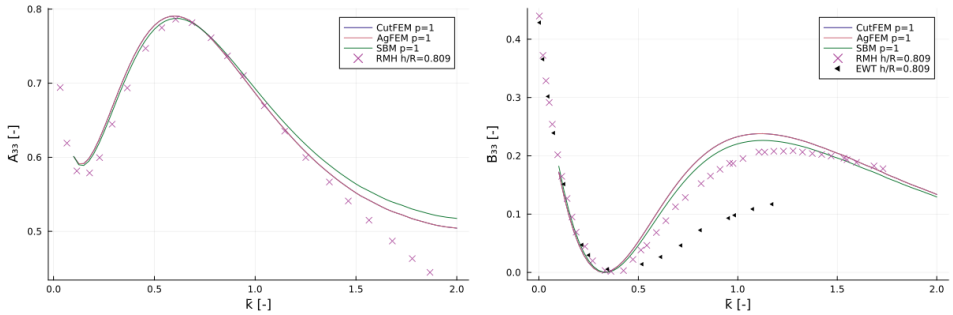


Figure 6.14: Non-dimensional wave number $\bar{k}$ versus non-dimensional added mass $\bar{A}_{33}$ and added damping $\bar{B}_{33}$ for submergence distance $H/R = 0.809$ with reference data from [174].

Matrix conditioning

In this section, we address the efficiency of each method by discussing the condition number of the block matrix $\mathbb{A}_{w\phi}$. This is the only matrix component that has to be inverted.

In Figure 6.21, the condition number κ is plotted against the number of elements n for the rectangle on the left and the cylinder case study on the right. Generally, AgFEM always appears to have the largest condition number. For CutFEM, the condition number is slightly smaller than that for SBM, but for a certain number of elements n, there is a peak for the cases where there are small cut cells. Especially in Figure 6.22, this becomes apparent. It depicts the condition number for varying depths of submergence at $n = 128$, where, in some cases, the CutFEM condition number exceeds that of the AgFEM condition number. The SBM shows a constant condition number because of the absence of cut cells, by definition.

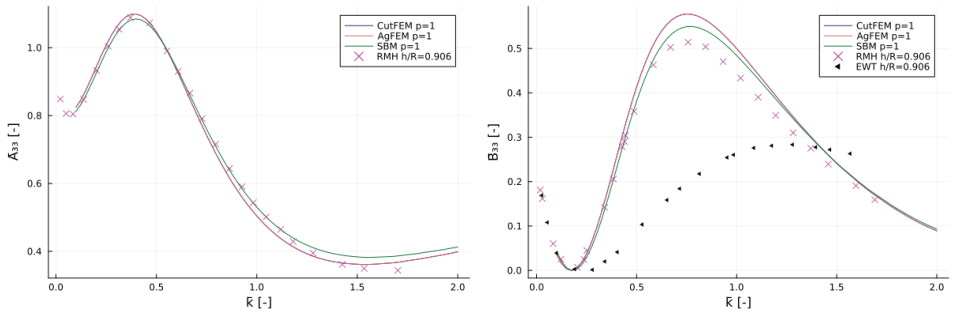


Figure 6.15: Non-dimensional wave number $\bar{k}$ versus non-dimensional added mass $\bar{A}_{33}$ and added damping $\bar{B}_{33}$ for submergence distance $h/R = 0.906$ with reference data from [174].

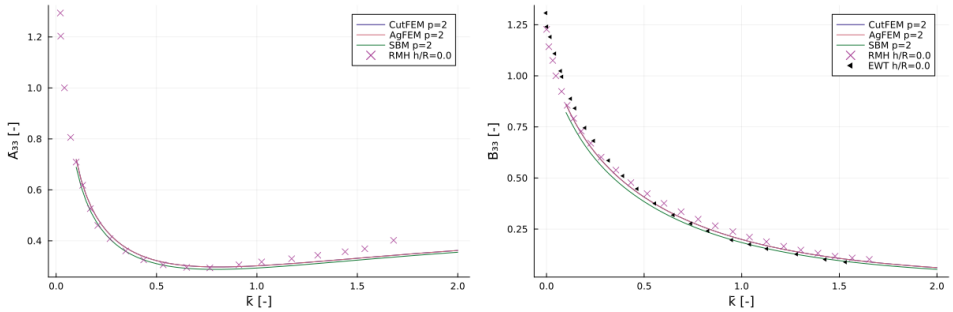


Figure 6.16: Non-dimensional wave number $\bar{k}$ versus non-dimensional added mass $\bar{A}_{33}$ and added damping $\bar{B}_{33}$ for submergence distance $H/R = 0.0$ with reference data from [174].

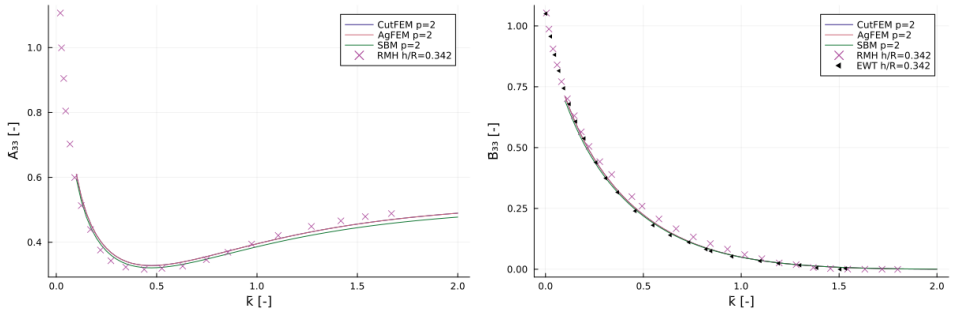


Figure 6.17: Non-dimensional wave number $\bar{k}$ versus non-dimensional added mass $\bar{A}_{33}$ and added damping $\bar{B}_{33}$ for submergence distance $H/R = 0.342$ with reference data from [174].

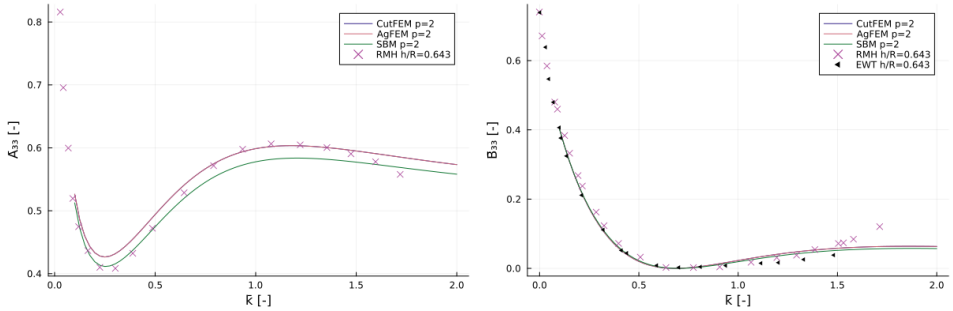


Figure 6.18: Non-dimensional wave number $\bar{k}$ versus non-dimensional added mass $\bar{A}_{33}$ and added damping $\bar{B}_{33}$ for submergence distance $H/R = 0.643$ with reference data from [174].

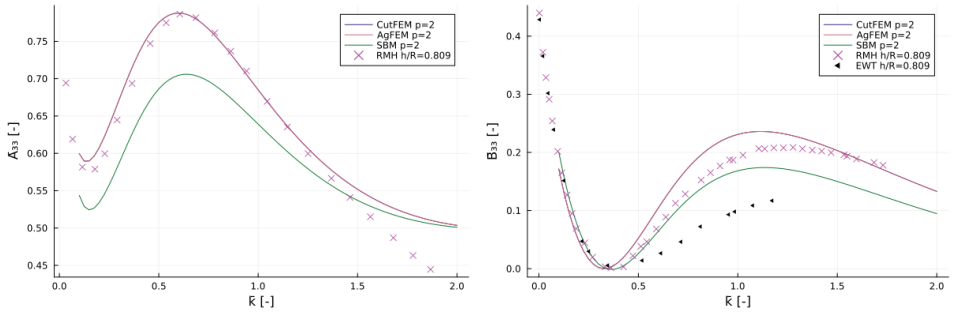


Figure 6.19: Non-dimensional wave number $\bar{k}$ versus non-dimensional added mass $\bar{A}_{33}$ and added damping $\bar{B}_{33}$ for submergence distance $H/R = 0.809$ with reference data from [174].

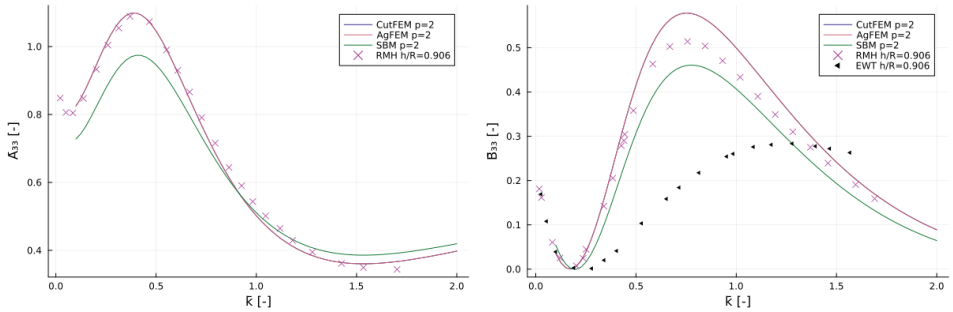


Figure 6.20: Non-dimensional wave number $\bar{k}$ versus non-dimensional added mass $\bar{A}_{33}$ and added damping $\bar{B}_{33}$ for submergence distance $h/R = 0.906$ with reference data from [174].

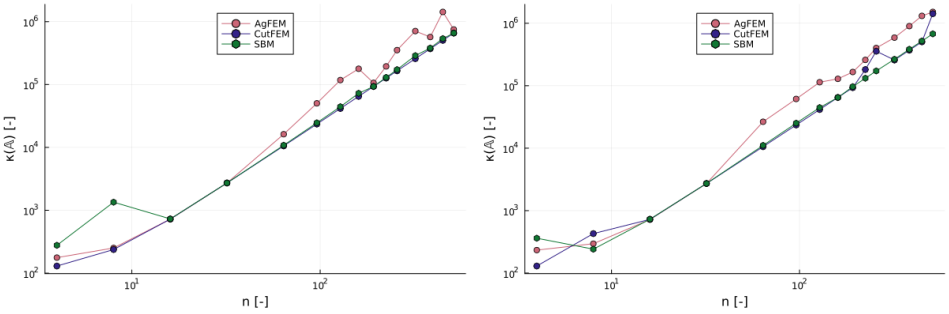


Figure 6.21: Condition number $\kappa(\mathbb{A})$ for varying number of elements n for the rectangle (left) and cylinder (right) case studies.

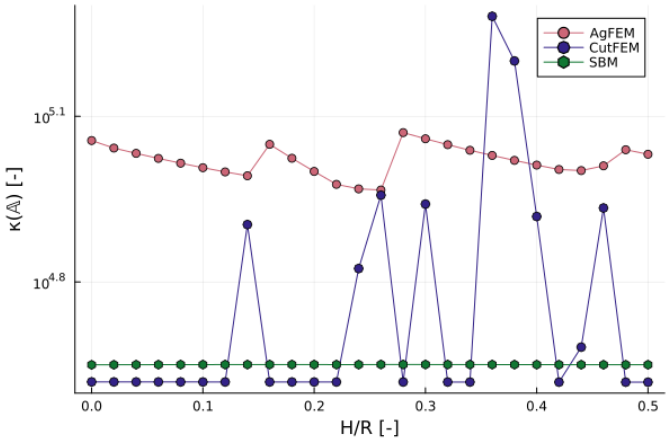


Figure 6.22: Condition number $\kappa(\mathbb{A})$ for varying submergence depth H for the cylinder case study with radius R.

6.3 Single Body - Realistic Geometry

We verify the code by comparing the added mass and added damping with the reference added mass and added damping of the OC3 Phase IV Hywind spar floating offshore wind turbine [175] and of the OC4 Phase II DeepCWind semisubmersible FOWT [169]. The OC3 and OC4 projects are international benchmarking efforts aimed at verifying and validating numerical models for offshore wind turbines through code-to-code comparisons. These projects established standardized validation methodologies that provide a reliable basis for assessing the accuracy of offshore wind turbine simulations. The reference data from the aforementioned works were generated using the WAMIT software package. WAMIT utilizes a 3D BEM in the frequency domain to solve the linearized potential flow equations that govern hydrodynamic radiation for wave interactions with offshore structures. Although a comparison between the meshing process of a BEM and a FEM can never be fair, we would like to highlight that the meshing process for the simulation of the reference data requires manual refinement in the computational mesh to mitigate irregular frequency effects [169, 175]. In contrast, an AgFEM approach requires only the generation of a background mesh.

Relevant Degrees of Freedom

The added mass and added damping matrices considered in this work are of dimension 6x6. It accounts for six degrees of freedom of the structure: three translational and three rotational. Motions are defined as surge, sway, heave, roll, pitch, and yaw for structural degrees of freedom one to six, respectively. The six diagonal terms of the added mass matrix $\boldsymbol{A}$ represent the fluid-induced inertial force that the body experiences in a direction as a result of the acceleration of the surrounding fluid in that same direction. The off-diagonal terms represent the fluid-induced inertial forces that the body experiences in one direction of motion due to the acceleration of the surrounding fluid in a different direction.

The added damping matrix $\boldsymbol{B}$ represents the dissipation of energy due to radiation damping, which arises when the body oscillates and generates waves. This wave radiation carries energy away from the body, leading to a resistive force proportional to velocity. The matrix components are the damping forces experienced by the structure in a certain direction of motion, associated with the velocity in a particular direction of motion, similarly to the added mass terms.

Force - motion direction	$\boldsymbol{A}$	Unit	$\boldsymbol{B}$	Unit
surge-surge	A_{11}	kg	B_{11}	kg/s
sway-sway	A_{22}	kg	B_{22}	kg/s
heave-heave	A_{33}	kg	B_{33}	kg/s
roll-roll	A_{44}	kg·m^2	B_{44}	kg·m^2/s
pitch-pitch	A_{55}	kg·m^2	B_{55}	kg·m^2/s
yaw-yaw	A_{66}	kg·m^2	B_{66}	kg·m^2/s
surge-pitch	A_{15}	kg·m	B_{15}	kg·m/s
sway-roll	A_{24}	kg·m	B_{24}	kg·m/s

Table 6.1: Added mass and added damping matrix components.

Due to the geometrical symmetry of both OC3 and OC4 structures, certain components of the added mass and added damping matrices are trivially zero. The layout of the added mass matrix is given in Equation 6.16; it should be noted that for the added damping matrix, the same terms are neglected as for the added mass matrix.

$$\boldsymbol{A} = \begin{pmatrix} A_{11} & 0 & 0 & 0 & A_{15} & 0 \\ 0 & A_{22} & 0 & A_{24} & 0 & 0 \\ 0 & 0 & A_{33} & 0 & 0 & 0 \\ 0 & A_{42} & 0 & A_{44} & 0 & 0 \\ A_{51} & 0 & 0 & 0 & A_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & A_{66} \end{pmatrix} \tag{6.16}$$

In Table 6.1, an overview of the non-neglected components of the added mass and the added damping matrix is presented. It includes the direction of the force that the structure experiences due to acceleration or velocity in a certain direction of motion for added mass or added damping, respectively.

Geometrical Representation of Hywind and DeepCWind

The OC3 Phase IV spar has a relatively simple geometry, as it consists of two cylinders of different diameters connected by a tapered cylinder. The geometrical representation of the OC4 Phase II DeepCWind semisubmersible consists of four vertical cylindrical columns connected by horizontal and diagonal braces. The three outer columns have wider base columns attached at the bottom. In Figure 6.23, the geometry of the OC3 Hywind spar is depicted on the left, and on the right, the geometry of the OC4 DeepCWind semisubmersible is depicted.

Computational Domain

The computational domain is set up using the data in Table 6.2, in which an overview of the relevant domain parameters is shown for each case study. The water depths are selected from both respective reference papers. The characteristic length is chosen to be equal to the largest diameter of the largest component for each of the structures. For OC3, this is the bottom cylindrical section, and for OC4, this corresponds to the diameter of one of the bottom cylindrical sections of the three outer cylinders. From the work by [86], we consider the longest wave $\lambda_{\max}$ in this work to be restricted by the non-dimensional wave number $\omega^2 D_c/g = 0.1$ with a characteristic length D_c. The minimum wavelength $\lambda_{\min}$ is derived from the maximum angular frequency considered in the reference data.

The background mesh should be able to capture the smallest wavelengths at the free surface, whilst the domain radius is limited by the maximum wavelength. Additionally, in the reference data, a panel size of 2 m is used for the structure, which is also utilised in this work. For AgFEM, this might be insufficient, as the reference data applied enhancements near the logarithmic singularities [169, 175], allowing for larger panel sizes close to sharp features in the geometry. Given these requirements, a background mesh is created with a minimum characteristic element length $h_{e,\min} = 0.37$ m and a maximum characteristic element length $h_{e,\max} = 59.43$ m. The background mesh for the OC3 case study is shown in Figure 6.24. For the OC4 case study, a similar cylindrical computational domain and background mesh are constructed given the required horizontal domain radius. Note

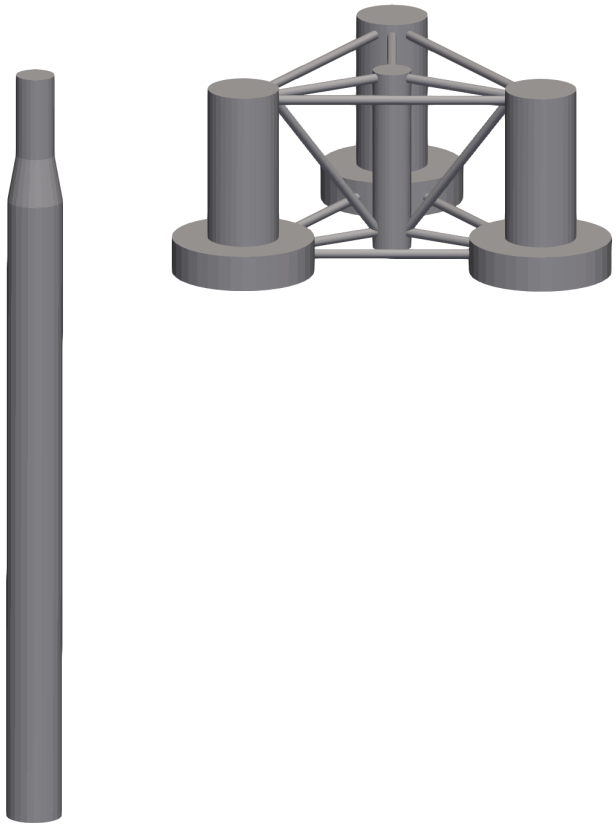


Figure 6.23: Geometrical representation of the Hywind spar (left) and the DeepCWind semisubmersible (right).

Study:	OC3	OC4
Water depth: d [m]	320.0	200.0
Characteristic length: D_c [m]	9.4	24.0
Minimum wavelength: λ_{min} [m]	2.5	2.5
Maximum wavelength: λ_{max} [m]	590.6	1507.9
Horizontal domain radius: R_x [m]	600.0	1500.0

Table 6.2: Dimensions and relevant parameters for the case study

that the requirement on the horizontal domain radius could be relaxed, but this would negatively impact the accuracy of the results for the lower frequency range.

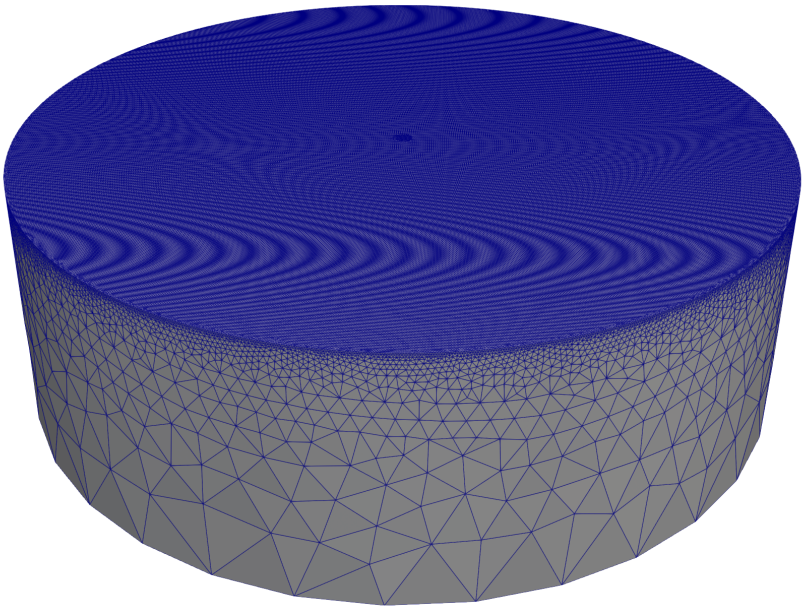


Figure 6.24: Background mesh for the OC3 case study.

6.3.1 OC3 Phase IV Hywind Spar

The OC3 Hywind is a spar-type floater, which means that it is expected that the surge and sway forces, as well as the rolling and pitching moments, dominate in relation to the heave and yaw. The added mass and added damping terms in and around the horizontal plane, i.e., those coefficients denoted by subscripts 11 and 22, and 44 and 55, are identical due to the geometrical symmetry of the structure.

In Figure 6.25, the added mass and added damping of the OC3 spar are depicted. It appears that the added mass slightly underestimates the reported values from [175]. The added damping matches well, and for higher frequency values, we report oscillations for

surge-surge B_{11} and sway-sway B_{22}. These oscillations might be due to elements not fully capturing the wavelength at those higher frequencies.

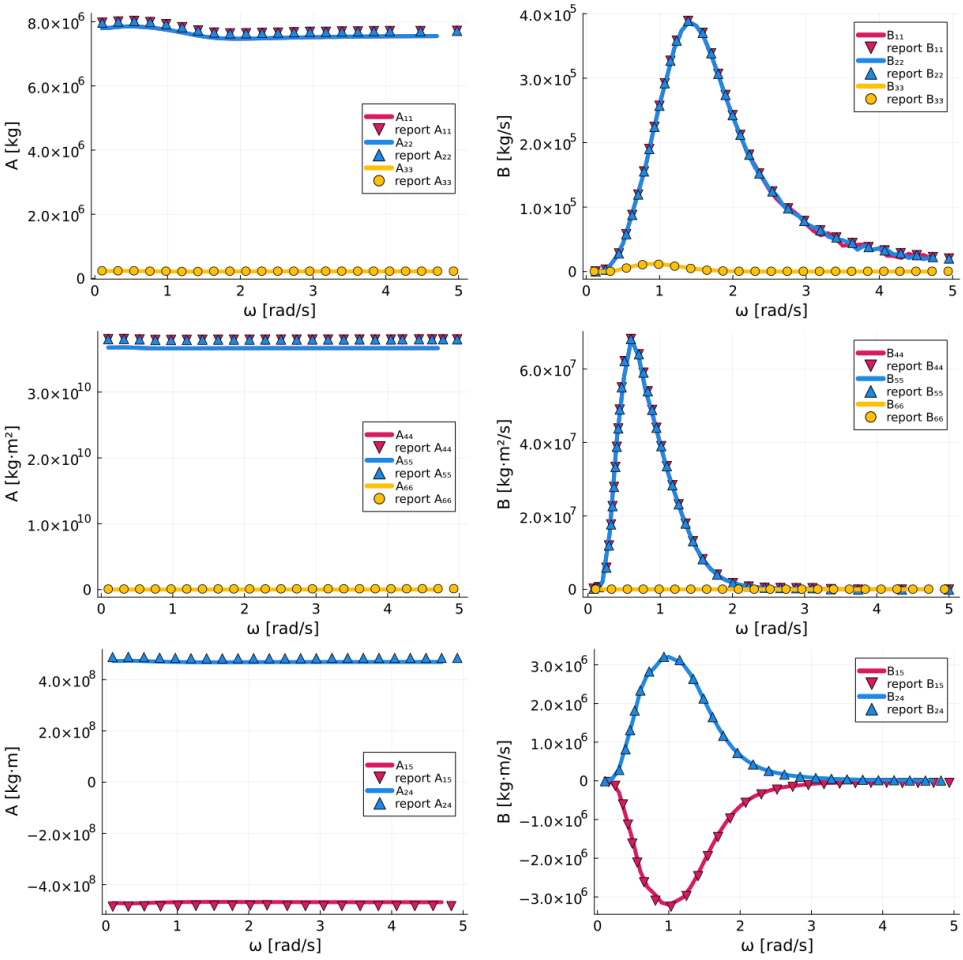


Figure 6.25: Added mass and added damping matrix terms of OC3 for the six diagonal terms A_{11} to A_{66} and the off-diagonal A_{15} and A_{24}. Similar terms for matrix B.

6.3.2 OC4 Phase II DeepCWind Semisubmersible

The OC4 DeepCwind is a semisubmersible type floater, meaning the heave force and yaw moment are significantly larger than the forces and moments around the horizontal. In Figure 6.26, the added mass over the angular frequency is plotted for the diagonal terms of the added mass matrix and for the off-diagonal terms: surge-pitch A_{15} and sway-roll A_{24}. The reported values from Robertson et al. (2014) are depicted using marker symbols. It is clear that the current method slightly underestimates the added mass moments of inertia, especially for heave-heave A_{33}; roll-roll A_{44}; and pitch-pitch A_{55}, where the discrepancy is

significant. However, it correctly captures the behavior over the frequency, as the peaks and troughs match the reference data. Indicating that the equations have been solved correctly. Furthermore, due to geometrical symmetries, the terms surge-surge A_{11} and sway-sway A_{22}, as well as roll-roll A_{44} and pitch-pitch A_{55}, are expected to be identical, as confirmed by the AgFEM model. The differences between the AgFEM model and the reference values might be a matter of mesh resolution. Especially close to the lower base columns, a highly refined mesh is required.

In Figure 6.26, the added damping is depicted in the same terms on the right. These results match well with the values reported by [169], matching both the amplitude and phase. However, for the higher frequency range, there appear to be more oscillations in the AgFEM model. This is likely due to the mesh resolution being too coarse for the small wavelengths located in this frequency range.

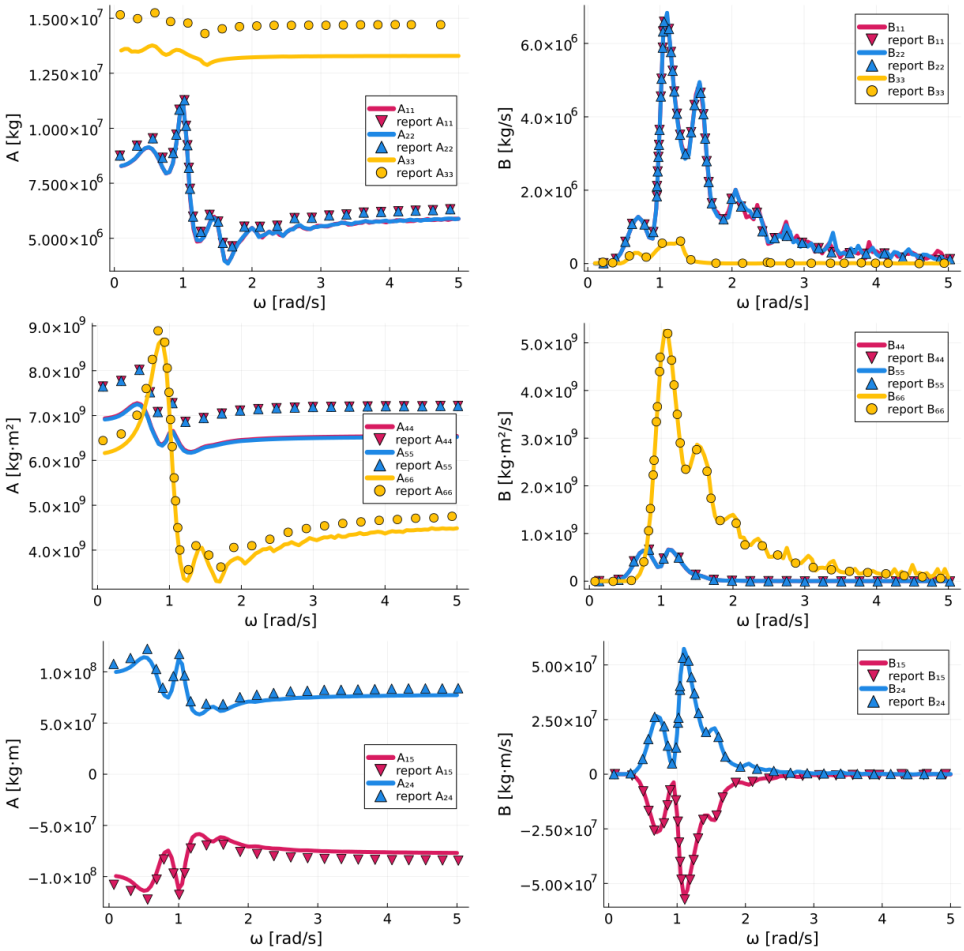


Figure 6.26: Added mass matrix terms of OC4 for the six diagonal terms A_{11} to A_{66} and the off-diagonal A_{15} and A_{24}. Similar terms for matrix B.

6.4 Conclusions

In this chapter, we applied CutFEM, AgFEM, and SBM as accurate tools to solve 2D problems involving linear potential flow around rigid objects for several geometries in the frequency domain. We demonstrate a novel application of CutFEM, AgFEM, and SBM using linear potential flow in the frequency domain and how to estimate the added mass and added damping whilst incorporating these methods. For CutFEM, a sensitivity study is conducted for the ghost penalty parameter γ, and the reference value from the literature is deemed sufficient. We highlight efficiency by allowing a single background mesh to simulate several geometrical scenarios using different unfitted finite element methods.

Regarding conditioning, the three methods perform comparably, remaining within the same order of magnitude across all cases. CutFEM yields a marginally lower condition number than the other methods in the surface-piercing configuration (Figure 6.21); however, this advantage is not consistent across all submergence ratios, and the relative ordering of methods varies with H/R (Figure 6.22). Notably, the largest condition number outliers occur for CutFEM, while SBM produces no such outliers.

The ghost penalty stabilisation parameter is studied and included in the case studies to reduce the condition number for CutFEM. We highlight the accuracy of the unfitted methods by comparing them with other numerical methods, as well as experimental and analytical reference data. Identifying possible bottlenecks and knowledge gaps for future research. Our results show that the difference in the estimations of added mass and added damping is small. For SBM, a gradient recovery technique using a mixed formulation should be applied to study the accuracy of SBM for linear order elements. Additionally, our SBM formulation allows for further research on the correct shifting of the hydrodynamic and hydrostatic pressure terms, as these are to be integrated on the true boundary. Therefore, we should shift these Jacobian integral contributions from the true boundary to the surrogate boundary. This contribution should become visible in the matrix term $\mathbb{A}_{v\phi}$ and requires further investigation. A follow-up study investigating the application of a weighted SBM should also be conducted.

We also presented a novel application of the AgFEM, which solves for a linear potential flow frequency domain representation to estimate added mass and added damping for floating structures of arbitrary geometry. The proposed method does not require manual intervention in the meshing procedure after a non-conforming background mesh has been generated. The results appear promising, as the general trends of added mass and added damping are recovered for both case studies.

However, more conclusive research should be conducted on the required dimensions of the domain and the minimum mesh refinement necessary to make this tool more competitive. Domain sizes can become large when the maximum wavelength is taken into account. In addition, the minimum wavelength indicates the element size at the free surface, and this combination requires a large number of elements in the mesh. This allows for further research and applications using a symmetry condition, numerical damping zones, or the addition of a Perfectly Matched Layer to reduce the domain size. Arbitrary mesh refinement should be coupled with the existing model using a forest of octrees based on cut elements for optimal refinement.

7

Conclusions

In this dissertation, we have developed a toolbox for modeling fluid-structure interaction using unfitted finite element methods in both the time and frequency domains. We present a primary research objective with five sub-objectives. This chapter assesses these objectives and suggests directions for future research.

7.1 Conclusions

The main goal of this dissertation is defined as follows:

> **Thesis objective:** Develop a numerical wave tank based on unfitted finite element methods to efficiently simulate large-scale offshore wave–structure interactions while eliminating the meshing bottleneck.

In chapter 2, we present the model formulation using linear potential flow coupled with a rigid structure. Their weak formulations are presented for both the time and frequency domains. The choice is made for linear potential flow, as this is a simple model to implement, a low fidelity model that still sees use within the offshore industry, and serves as a foundation for higher fidelity models. Similarly, the rigid structure is chosen for its simplicity, but it could be replaced by an elastic formulation.

After which, in chapter 3, we identified the Cut Finite Element Method (CutFEM) [124], the Aggregated unfitted Finite Element Method (AgFEM)[77], the Shifted Boundary Method (SBM) [125], and the Weighted Shifted Boundary Method (WSBM) [129] as unfitted methods that could be applicable for numerical wave tanks. We explained the workings of each unfitted method and introduced the discrete weak formulations for each unfitted finite element method using the reduced linear potential flow formulation.

In chapter 4, we perform a benchmarking initiative to better understand and compare the aforementioned unfitted finite element methods. We conduct a convergence test for both two-dimensional and three-dimensional problems. Additionally, we use implicit and explicit geometrical representations and compare their convergence rates. Additionally, the condition numbers are compared as indicators of numerical conditioning and efficiency. For the two-dimensional study, a performance comparison is conducted, focusing on

computational time and allocations. Lastly, an implementation comparison is drawn. This benchmarking study demonstrates that CutFEM and AgFEM achieve the expected rates of convergence. SBM and WSBM have one order lower rate of convergence compared to CutFEM and AgFEM, as we have not implemented a gradient recovery technique. The lowest condition numbers are for SBM, but for polynomial order $p_e = 1$, AgFEM achieves comparable condition numbers. For polynomial order $p_e = 2$, AgFEM has condition numbers comparable to those of CutFEM. In this study, WSBM has the largest condition numbers. Performance-wise, both computational time and the number of allocations required exhibit the same trends across unfitted methods. SBM, being both the fastest and requiring the least allocations, is followed by CutFEM, then WSBM, and finally AgFEM. The largest bottleneck for AgFEM occurs during the initialization of the finite element space, indicating that it requires significant time and allocations for the aggregation algorithm. Implementation-wise, AgFEM is deemed the most complex. CutFEM and WSBM both have their challenges, but due to the dependency on a tessellation algorithm for integration, we consider CutFEM to be more complex. SBM has the simplest implementation; however, the discrete weak formulation becomes more involved.

Thereafter, in chapter 5, we demonstrate the current capabilities of the numerical framework in the time domain. We show that we can solve both two- and three-dimensional problems for simplistic and realistic floating structures, extending even to multi-body simulation. We demonstrate that all four unfitted finite element methods match the results of a body-fitted finite element method and that we can reproduce experimental results for heave free decay tests of simple and realistic geometries, given a sufficient number of elements and polynomial order. The initial development used Cartesian grids; however, for three-dimensional problems, it becomes inefficient to use this as a background mesh. Therefore, we propose an adaptive mesh refinement strategy using a forest of octrees from the `p4est` library [167], which has not been fully developed for all four unfitted methods. For both CutFEM and AgFEM, the adaptive mesh refinement strategy is complete. For SBM and WSBM, we still need to define the Hessian operator on the reference element for the adaptively refined grid, and the current implementation neglects this term of the shifting operator. For the unfitted methods, adequate mesh refinement and polynomial degree yield consistent outcomes. CutFEM and AgFEM exhibit less sensitivity to coarse discretizations, whereas SBM and WSBM may benefit from gradient recovery strategies to achieve improved accuracy with fewer elements. In the case of WSBM, non-adaptively refined meshes can produce spurious oscillations and even divergence, likely related to element size, which warrants additional study. Mesh adaptivity therefore plays a key role, and defining the Hessian operator on the reference element is necessary to enable a thorough comparison with fully implemented SBM and WSBM approaches.

Additionally, in chapter 6, we demonstrate the capabilities of our numerical framework for frequency domain problems. CutFEM, AgFEM, and the SBM are applied to simple geometries in two-dimensional case studies to estimate the added mass and added damping coefficients. For AgFEM, this work has been extended to three-dimensional realistic floating structures using unstructured refined meshes generated by the GMSH software [168]. From the results, we note that this approach can reproduce the added mass and added damping coefficients for simple geometries across methods and for AgFEM with realistic geometries. The quality of the solutions for the added mass could be more dependent on the mesh

resolution close to the structure than on the added damping. One challenge remains: to optimally balance the domain size and local mesh refinement for each wavelength instead of using a single background mesh.

Reflecting on the main objective of this dissertation, we have developed a numerical framework based on several unfitted finite element methods that can simulate offshore wave-structure interactions without being constrained by the meshing bottleneck. The missing keywords here are: numerical wave tank, efficiently, and large-scale. The argument for the first missing keyword is rooted in semantics, as the proposed framework addresses an interaction problem between a fluid and a structure. However, we lack wave generation and absorption zones. It is possible to use simplified wave generation and damping approaches within the framework, but these have not been validated; therefore, we refrain from calling it a numerical wave tank specifically. The keywords efficiently and large-scale, in part, go together, as it becomes non-feasible to have one without the other. With efficiency, we indicate a more optimal approach to using computational resources, i.e., model fidelity, mesh adaptivity, and parallel implementation, such that the scalability of finite element formulations can be exploited. The results in this dissertation show the milestones achieved for a linear potential flow fluid coupled to rigid structures, which have been successful. We show the milestones achieved for mesh adaptivity and indicate which parts are missing. Similarly, no parallel implementation is demonstrated; therefore, a more efficient framework is achievable. Another requirement for efficiency could be a comparison to other frameworks or tools to assess performance, which has not been conducted. The last keyword, large-scale, is not included, as the case studies show only validation for single bodies. No validation case studies for hydrodynamics at multi-body array or farm-scale levels have been presented; however, we propose a framework in which such validation studies can be conducted.

7.2 Future Recommendations

Looking forward, there are several avenues for continuing this research.

Starting with the parts of this dissertation that will improve completeness: for SBM and WSBM, we require the implementation of a gradient recovery technique; see Atallah et al. [140], such that the rate of convergence matches that of CutFEM and AgFEM. Additionally, when coupled with the adaptively refined mesh method, we require the Hessian to be defined in the reference element for SBM and WSBM, especially when applying higher-order SBM types. For the benchmarking study, this would mean redoing some of the test cases to confirm grid convergence and to study the impact of gradient recovery on computational performance. For the frequency domain problems, a case by case optimal mesh algorithm can be designed and developed with the adaptive mesh refinement from chapter 5. Finding a balance in domain size and mesh refinement level for each wavelength, along with the automated generation of the background grid. At the same time, case studies could also be conducted for three-dimensional domains with realistic geometries using CutFEM and SBM. The WSBM was originally developed for moving free surfaces, and based on the results in this dissertation and the static nature of frequency domain problems, we advise against further pursuing frequency domain problems with the WSBM specifically due to the numerical instabilities and challenges encountered during this project. That said, according to our results from the time domain simulations, the WSBM with first polynomial

order could be applied and used to potentially obtain results with higher accuracy than the traditional SBM.

An extension to parallel solvers is already under development, and this is a necessary addition for large-scale multi-structure simulations. The largest bottleneck to date in `Gridap` is the implementation of the adaptive AgFEM in a parallel setting, as the existence of a root cell on a different processor than the aggregated cell requires careful treatment. Additionally, there is a more involved constraining scheme on the degrees-of-freedom; see Badia et al. [176] for a possible implementation. For CutFEM, SBM, and WSBM, a parallel implementation should be possible.

For the completeness of naming the proposed numerical framework, a numerical wave tank, it is imperative that validated modules for wave generation and wave damping are added so that the simulation time can be increased with a limited domain size.

Regarding large-scale simulations, one of the first steps should be to validate the results of multi-body simulations. In chapter 6, we propose a non-validated multi-body formulation in the frequency domain. This formulation should be validated against the work by Raghavan et al. [177], which presents a novel frequency domain code for the evaluation of added mass and added damping for multiple structures, focusing on wave energy converter type floating structures. It also compares computational resource usage between different frameworks, tying into the efficiency keyword that is missing from our main objective. For the time domain problems, similar large-scale validation studies should be identified and conducted.

Reflecting on model fidelity: the linear potential flow approximation is sufficient for moderate sea states; however, consideration of nonlinearities is imperative for critically large waves, for wave resonances between floating bodies, and for viscous drag modeling. Higher fidelity fluid models could be incorporated to deal with the latter, such as fully nonlinear potential flow or Navier-Stokes, for cases where viscous effects cannot be neglected, such as the OC4 DeepCWind semisubmersible [169]. Although conducting large-scale simulations using the Navier-Stokes equations throughout the entire domain is likely not cost-effective, hybrid model fidelity could be explored, as described, for example, by Saincher and Sriram [178]. Another avenue, albeit unexplored in this dissertation, is to use the unfitted methods described in this work with a Navier-Stokes fluid model without a free surface to study the flow dynamics on mooring lines that have accumulated marine growth. See, for example, the experimental works by Marty et al. [179, 180] in which blocks of varied, artificial, uniform marine growth of different types are studied. The explicit geometrical representations via STL files could be incorporated to conduct similar studies with non-artificial marine growth based on images of mooring lines with natural marine growth.

An initial next step would be to incorporate the mooring system via linear stiffness matrices [169, 175] or more advanced mooring line models [170, 181]. In doing so, the hydrostatic restoring moments can be validated using the current formulations. Introducing elasticity for the structure is another milestone, as, at least for the design of floating wind turbine support structures, the rigid body assumption is not always sufficient [182].

To accurately model the phenomena associated with nonlinear surface waves, an increase in the fidelity of the fluid model to a fully nonlinear potential flow is required. Then, for this model, the use of an unfitted method to capture the free surface needs

to be explored. This application sees significant development for WSBM with higher fidelity models, i.e., for Stokes' flow [130], incompressible Navier-Stokes [129, 183], and the full Navier-Stokes [151]. Similarly, the SBM, CutFEM, and AgFEM could be extended for this application. Taking into account that a space-time approach [128] might be more interesting for AgFEM, as it would eliminate the need to perform new cell aggregation at each time step. Note that these moving free surface studies do not incorporate an additional moving structure; perhaps it is simpler to introduce a moving mesh for the structure whilst still maintaining the small amplitude free surface assumption.

Lastly, adjacent to this dissertation and based on the challenges encountered during this project, the generalized SBM (GSBM) [154] has been proposed. This strategy yields a finite element formulation that is independent of the underlying geometry and does not require redefining finite element spaces, integration schemes, or geometric entities. The GSBM formulation supports differentiation with respect to the degrees of freedom, even in the presence of moving domains or changing geometries, including topological variations. Using the same GSBM framework, one can retrieve the original SBM formulation, the SBM formulation with optimal surrogate boundaries, or the WSBM formulation. The trade-off between computational efficiency and implementation complexity should be studied using the approach defined in chapter 4. Additionally, the GSBM should be explored for time domain simulations, as defined in chapter 5, and the future steps outlined in the previous paragraph should be extended to moving domains.

Bibliography

References

[1] First Edition, Johan M J Journée, and W W Massie. Offshore hydromechanics. *Delft University of Technology*, 523, 2001.

[2] Bernard Molin. *Offshore structure hydrodynamics*. Cambridge University Press, 2023.

[3] European Commission COM(2020)741. Explanatory memorandum to com(2020)741 - eu strategy to harness the potential of offshore renewable energy for a climate neutral future, 2020.

[4] European Commission EC. Blue energy action needed to deliver on the potential of ocean energy in european seas and oceans by 2020 and beyond, 2014.

[5] Vegard Aksnes, Hagbart Alsos, Erin Bachynski-Polić, Petter Andreas Berthelsen, Virgile Delhaye, Birgitte R Furevik, Hans Petter Jostad, Trygve Kristiansen, and Babak Ommani. On common research needs for the next generation of floating support structures. In *International Conference on Offshore Mechanics and Arctic Engineering*, volume 85888, page V004T05A027. American Society of Mechanical Engineers, 2022.

[6] Maria Xylia, Marlon Vieira Passos, Tommaso Piseddu, and Karina Barquet. Exploring multi-use platforms: A literature review of marine, multifunctional, modular, and mobile applications (m4s). *Heliyon*, 9(6), 2023.

[7] Jian Dai, Øyvind Hellan, Arnstein Watn, and Kok Keng Ang. Modular multi-purpose floating structures for space creation. In *WCFS2020*, pages 257–271. Springer, 2022.

[8] Walid M Nassar, Olimpo Anaya-Lara, Khaled H Ahmed, David Campos-Gaona, and Mohamed Elgenedy. Assessment of multi-use offshore platforms: structure classification and design challenges. *Sustainability*, 12(5):1860, 2020.

[9] Sander W K Van Den Burg, Maximilian Felix Schupp, Daniel Depellegrin, Andrea Barbanti, and Sandy Kerr. Development of multi-use platforms at sea: Barriers to realising blue growth. *Ocean Engineering*, 217:107983, 2020.

[10] Maarten Flikkema and Olaf Waals. Space@ sea the floating solution. *Frontiers in Marine Science*, 6:553, 2019.

[11] William Otto and Clara Hüsken. Results from demonstration at wave tank d10.4. Technical report, Space@Sea, 2020. Grant Agreement No. 774253.

[12] Xiangyuan Zheng, Huadong Zheng, Yu Lei, Yi Li, and Wei Li. An offshore floating wind–solar–aquaculture system: Concept design and extreme response in survival conditions. *Energies*, 13(3):604, 2020.

[13] Luofeng Huang, Yuzhu Li, Daniela Benites-Munoz, Christian Windt, Anna Feichtner, Sasan Tavakoli, Josh Davidson, Ruben Paredes, Tadea Quintuna, Edward Ransley, et al. A review on the modelling of wave-structure interactions based on openfoam. *OpenFOAM® Journal*, 2:116–142, 2022.

[14] John Grue and Morten Huseby. Higher-harmonic wave forces and ringing of vertical cylinders. *Applied ocean research*, 24(4):203–214, 2002.

[15] Sanne van Essen, Jule Scharnke, Tim Bunnik, Bülent Düz, Henry Bandringa, Rink Hallmann, and Joop Helder. Linking experimental and numerical wave modelling. *Journal of Marine Science and Engineering*, 8(3):198, 2020.

[16] Momchil Terziev, Tahsin Tezdogan, and Atilla Incecik. Scale effects and full-scale ship hydrodynamics: A review. *Ocean Engineering*, 245:110496, 2022.

[17] Christophe Maisondieu, Gerry Sutton, Frances Judges, and Stéphanie Ordonez. Deliverable d4.3: Science plan final report. Technical report, MARINERGi: Marine Renewable Research Infrastructure Network for Energy Research, 2019. Grant Agreement No. 739550.

[18] Mohd Atif Siddiqui, Finn-Christian Wickmann Hanssen, Marilena Greco, and Eirik Anda. Comparing the utility of coupled aero-hydrodynamic analysis using a cfd solver versus a potential flow solver for floating offshore wind turbines. *Energies*, 16(23):7833, 2023.

[19] Rizwan Haider, Xin Li, Wei Shi, Zaibin Lin, Qing Xiao, and Haisheng Zhao. Review of computational fluid dynamics in the design of floating offshore wind turbines. *Energies*, 17(17):4269, 2024.

[20] Edward J Ransley, Scott A Brown, Martyn Hann, Deborah M Greaves, Christian Windt, John Ringwood, Josh Davidson, Pal Schmitt, Shiqiang Yan, Junxian X Wang, et al. Focused wave interactions with floating structures: A blind comparative study. *Proceedings of the Institution of Civil Engineers-Engineering and Computational Mechanics*, 174(1):46–61, 2021.

[21] Shimin Yu, Edward Ransley, Ling Qian, Yang Zhou, Scott Brown, Deborah Greaves, Martyn Hann, Anna Holcombe, Emma Edwards, Tom Tosdevin, et al. Modelling the hydrodynamic response of a floating offshore wind turbine–a comparative study. *Applied Ocean Research*, 155:104441, 2025.

[22] Jonas Bjerg Thomsen, Amélie Têtu, and Henrik Stiesdal. A comparative investigation of prevalent hydrodynamic modelling approaches for floating offshore wind turbine foundations: a tetraspar case study. *Journal of Marine Science and Engineering*, 9(7):683, 2021.

[23] Christopher Allen, Anthony Viscelli, Habib Dagher, Andrew Goupee, Evan Gaertner, Nikhar Abbas, Matthew Hall, and Garrett Barter. Definition of the umaine volturnus-s reference platform developed for the iea wind 15-megawatt offshore reference wind turbine. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2020.

[24] Michael Borg, Anthony Viselli, Christopher K Allen, Matthew Fowler, Christoffer Sigshøj, Andrea Grech La Rosa, Morten Thøtt Andersen, and Henrik Stiesdal. Physical model testing of the tetraspar demo floating wind turbine prototype. In *International Conference on Offshore Mechanics and Arctic Engineering*, volume 59353, page V001T01A021. American Society of Mechanical Engineers, 2019.

[25] Aditya Nair, Luofeng Huang, and Patrick G Verdin. Computational fluid dynamics and potential flow modelling techniques for floating photovoltaic systems: A systematic review. *Symmetry*, 17(9):1508, 2025.

[26] Matthew Leary, Curtis Rusch, Zhe Zhang, and Bryson Robertson. Comparison and validation of hydrodynamic theories for wave energy converter modelling. *Energies*, 14(13):3959, 2021.

[27] Jason M Jonkman. Loads analysis of a floating offshore wind turbine using fully coupled simulation. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2007.

[28] Sijia Deng, Yingyi Liu, and Dezhi Ning. Fully coupled aero-hydrodynamic modelling of floating offshore wind turbines in nonlinear waves using a direct time-domain approach. *Renewable Energy*, 216:119016, 2023.

[29] John N Newman. *Marine hydrodynamics*. The MIT press, 2018.

[30] Stephan T Grilli, Jesper Skourup, and Ib A Svendsen. An efficient boundary element method for nonlinear water waves. *Engineering Analysis with Boundary Elements*, 6(2):97–107, 1989.

[31] Stéphan T Grilli and Juan Horrillo. Numerical generation and absorption of fully nonlinear periodic waves. *Journal of engineering mechanics*, 123(10):1060–1069, 1997.

[32] Hans Bihs, Weizhi Wang, Csaba Pakozdi, and Arun Kamath. Reef3d:: Fnpf—a flexible fully nonlinear potential flow solver. *Journal of Offshore Mechanics and Arctic Engineering*, 142(4):041902, 2020.

[33] Brian E Launder and Dudley B Spalding. The numerical computation of turbulent flows. In *Numerical prediction of flow, heat transfer, turbulence and combustion*, pages 96–116. Elsevier, 1983.

[34] Joseph Smagorinsky. General circulation experiments with the primitive equations: I. the basic experiment. *Monthly weather review*, 91(3):99–164, 1963.

[35] Philippe R Spalart. Comments on the feasibility of les for wings and on the hybrid rans/les approach. In *Proceedings of the First AFOSR International Conference on DNS/LES, 1997*, pages 137–147, 1997.

[36] Muk Chen Ong. Cfd applications in offshore engineering. In *EPJ Web of Conferences*, volume 143, page 01002. EDP Sciences, 2017.

[37] Didier Bresch and Benoît Desjardins. Existence of global weak solutions for a 2d viscous shallow water equations and convergence to the quasi-geostrophic model. *Communications in mathematical physics*, 238(1):211–223, 2003.

[38] Didier Bresch. Shallow-water equations and related topics. *Handbook of differential equations: evolutionary equations*, 5:1–104, 2009.

[39] Joseph Boussinesq. Théorie de l'intumescence liquide appelée onde solitaire ou de translation se propageant dans un canal rectangulaire. *CR Acad. Sci. Paris*, 72(755-759):1871, 1871.

[40] Ge Wei, James T Kirby, Stephan T Grilli, and Ravishankar Subramanya. A fully nonlinear boussinesq model for surface waves. part 1. highly nonlinear unsteady waves. *Journal of fluid mechanics*, 294:71–92, 1995.

[41] Paul Landesman, Jeffrey C Harris, Christophe Peyrard, and Michel Benoit. Wave–structure interaction by a two–way coupling between a fully nonlinear potential flow model and a navier–stokes solver. *Ocean Engineering*, 308:118209, 2024.

[42] Fabien Robaux and Michel Benoit. Assessment of one-way coupling methods from a potential to a viscous flow solver based on domain-and functional-decomposition for fixed submerged bodies in nonlinear waves. *European Journal of Mechanics-B/Fluids*, 95:315–334, 2022.

[43] Gordon D Smith. *Numerical solution of partial differential equations: finite difference methods*. Oxford university press, 1985.

[44] Henk K Versteeg. *An introduction to computational fluid dynamics the finite volume method, 2/E*. Pearson Education India, 2007.

[45] Claes Johnson. *Numerical solution of partial differential equations by the finite element method*. Courier Corporation, 2009.

[46] Joe J Monaghan. Smoothed particle hydrodynamics. *Reports on progress in physics*, 68(8):1703, 2005.

[47] Seiichi Koshizuka and Yoshiaki Oka. Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear science and engineering*, 123(3):421–434, 1996.

[48] Jacek Pozorski and Michał Olejnik. Smoothed particle hydrodynamics modelling of multiphase flows: an overview. *Acta Mechanica*, 235(4):1685–1714, 2024.

[49] Giacomo Rizzieri, Liberato Ferrara, and Massimiliano Cremonesi. Simulation of viscoelastic free-surface flows with the particle finite element method. *Computational Particle Mechanics*, 11(5):2043–2067, 2024.

[50] Carlos A Brebbia. The boundary element method for engineers. *(No Title)*, 1978.

[51] John N Newman and Chang-Ho Lee. Boundary-element methods in offshore structure analysis. *J. Offshore Mech. Arct. Eng.*, 124(2):81–89, 2002.

[52] Yanlin Shao and Odd M Faltinsen. A harmonic polynomial cell (hpc) method for 3d laplace equation with application in marine hydrodynamics. *Journal of Computational Physics*, 274:312–332, 2014.

[53] Anthony T Patera. A spectral element method for fluid dynamics: laminar flow in a channel expansion. *Journal of computational Physics*, 54(3):468–488, 1984.

[54] Hrvoje Jasak. Openfoam: Open source cfd in research and industry. *International journal of naval architecture and ocean engineering*, 1(2):89–94, 2009.

[55] Niels G Jacobsen, David R Fuhrman, and Jørgen Fredsøe. A wave generation toolbox for the open-source cfd library: Openfoam®. *International Journal for numerical methods in fluids*, 70(9):1073–1088, 2012.

[56] Pablo Higuera, Javier L Lara, and Inigo J Losada. Three-dimensional interaction of waves and porous coastal structures using openfoam®. part ii: Application. *Coastal Engineering*, 83:259–270, 2014.

[57] Siemens Digital Industries Software. Simcenter STAR-CCM+ User Guide v. 2021.1, Siemens 2021.

[58] ANSYS. Ansys fluent - cfd software | ansys, 2025.

[59] Alejandro J C Crespo, José M Domínguez, Benedict D Rogers, Moncho Gómez-Gesteira, Stephen Longshaw, Ricardo J F B Canelas, Renato Vacondio, Anxo Barreiro, and Orlando García-Feal. Dualsphysics: Open-source parallel cfd solver based on smoothed particle hydrodynamics (sph). *Computer Physics Communications*, 187:204–216, 2015.

[60] L Orcina. Orcaflex user manual version 11.0 b. *Orcina: Ulverston, UK*, 2016.

[61] COMSOL Inc. Comsol multiphysics®, 2023.

[62] Ian R Warren and Hanne K Bach. Mike 21: a modelling system for estuaries, coastal waters and seas. *Environmental software*, 7(4):229–240, 1992.

[63] Hans Bihs, Arun Kamath, Mayilvahanan Alagan Chella, Ankit Aggarwal, and Øivind A Arntsen. A new level set numerical wave tank with improved density interpolation for complex wave hydrodynamics. *Computers & Fluids*, 140:191–208, 2016.

[64] Allan P Engsig-Karup, Harry B Bingham, and Ole Lindberg. An efficient flexible-order model for 3d nonlinear water waves. *Journal of computational physics*, 228(6):2100–2118, 2009.

[65] Fengyan Shi, James T Kirby, Jeffrey C Harris, Joseph D Geiman, and Stephan T Grilli. A high-order adaptive time-stepping tvd solver for boussinesq modeling of breaking waves and coastal inundation. *Ocean Modelling*, 43:36–51, 2012.

[66] Stéphane Popinet. A quadtree-adaptive multigrid solver for the serre–green–naghdi equations. *Journal of Computational Physics*, 302:336–358, 2015.

[67] Ruddy Kurnia and Guillaume Ducrozet. Nemoh: Open-source boundary element solver for computation of first-and second-order hydrodynamic loads in the frequency domain. *Computer Physics Communications*, 292:108885, 2023.

[68] Marcel Zijlema, Guus Stelling, and Pieter Smit. Swash: An operational public domain code for simulating wave fields and rapidly varied flows in coastal waters. *Coastal Engineering*, 58(10):992–1012, 2011.

[69] Félicien Bonnefoy, David Le Touzé, and Pierre Ferrant. A fully-spectral 3d time-domain model for second-order simulation of wavetank experiments. part a: Formulation, implementation and numerical properties. *Applied Ocean Research*, 28(1):33–43, 2006.

[70] Patrick Knupp. Remarks on mesh quality. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2007.

[71] Jonathan R Shewchuk. What is a good linear element? interpolation, conditioning, and quality measures. In *11th International Meshing Roundtable,{IMR} 2002*, 2002.

[72] Jeffrey P Slotnick, Abdollah Khodadoust, Juan Alonso, David Darmofal, William Gropp, Elizabeth Lurie, and Dimitri J Mavriplis. Cfd vision 2030 study: a path to revolutionary computational aerosciences. Technical report, NASA: Washington, DC, USA, 2014.

[73] Thomas J R Hughes, John A Cottrell, and Yuri Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering*, 194(39-41):4135–4195, 2005.

[74] Odd M Faltinsen. Hydrodynamics of marine and offshore structures. *Journal of Hydrodynamics, Ser. B*, 26(6):835–847, 2015.

[75] Matthieu Ancellin and Frédéric Dias. Capytaine: a python-based linear potential flow solver. *Journal of Open Source Software*, 4(36):1341, 2019.

[76] Dominik Schillinger and Martin Ruess. The finite cell method: a review in the context of higher-order structural analysis of cad and image-based geometric models. *Archives of Computational Methods in Engineering*, 22(3):391–455, 2015.

[77] Santiago Badia, Francesc Verdugo, and Alberto F Martín. The aggregated unfitted finite element method for elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 336:533–553, 2018.

[78] Pere A Martorell and Santiago Badia. High order unfitted finite element discretizations for explicit boundary representations. *Journal of Computational Physics*, 511:113127, 2024.

[79] Santiago Badia, Eric Neiva, and Francesc Verdugo. Robust high-order unfitted finite elements by interpolation-based discrete extension. *Computers & Mathematics with Applications*, 127:105–126, 2022.

[80] Frits de Prenter, Clemens V Verhoosel, E Harald van Brummelen, Mats G Larson, and Santiago Badia. Stability and conditioning of immersed finite element methods: analysis and remedies. *Archives of Computational Methods in Engineering*, 30(6):3617–3656, 2023.

[81] Stavros Kontos, Harry B Bingham, Ole Lindberg, and Allan P Engsig-Karup. On nonlinear wave-structure interaction using an immersed boundary method in 2d. In *31th International Workshop on Water Waves and Floating Bodies (IWWWFB 2016)*, 2016.

[82] Stavros Kontos. *Robust numerical methods for nonlinear wave-structure interaction in a moving frame of reference*. Technical University of Denmark, 2016.

[83] Yan Xu, Harry B Bingham, and Yanlin Shao. Finite difference solutions for nonlinear water waves using an immersed boundary method. *International Journal for Numerical Methods in Fluids*, 93(4):1143–1162, 2021.

[84] Chao Tong, Yanlin Shao, Harry B Bingham, and Finn-Christian W Hanssen. An adaptive harmonic polynomial cell method with immersed boundaries: Accuracy, stability, and applications. *International Journal for Numerical Methods in Engineering*, 122(12):2945–2980, 2021.

[85] Chao Tong, Yanlin Shao, Harry B Bingham, and Finn-Christian W Hanssen. An adaptive harmonic polynomial cell method for three-dimensional fully nonlinear wave-structure interaction with immersed boundaries. *Physics of Fluids*, 36(3), 2024.

[86] Ying Wang, Yanlin. Shao, Jikang Chen, and Hui Liang. Accurate and efficient hydrodynamic analysis of structures with sharp edges by the extended finite element method (xfem): 2d studies. *Applied Ocean Research*, 117:102893, 2021.

[87] Ahmet Soydan, Widar W Wang, and Hans Bihs. An improved direct forcing immersed boundary method with integrated mooring algorithm for floating offshore wind turbines. *Journal of Offshore Mechanics and Arctic Engineering*, 147(4):042101, 2025.

[88] Chiang C Mei, Michael A Stiassnie, and Dick K-P Yue. *Theory and applications of ocean surface waves: Part 1: linear aspects*. World Scientific, 2005.

[89] Oriol Colomes, Francesc Verdugo, and Ido Akkerman. A monolithic finite element formulation for the hydroelastic analysis of very large floating structures. *International Journal for Numerical Methods in Engineering*, 124(3):714–751, 2023.

[90] Min W Kim, Weoncheol Koo, and Sa Y Hong. Numerical analysis of various artificial damping schemes in a three-dimensional numerical wave tank. *Ocean engineering*, 75:165–173, 2014.

[91] James Lighthill. Fundamentals concerning wave loading on offshore structures. *Journal of Fluid Mechanics*, 173:667–681, 1986.

[92] Ido Akkerman, J H A Meijer, and Marco F P ten Eikelder. Isogeometric analysis of linear free-surface potential flow. *Ocean Engineering*, 201:107114, 2020.

[93] Shagun Agarwal, Oriol Colomes, and Andrei V. Metrikine. Dynamic analysis of viscoelastic floating membranes using monolithic finite element method. *Journal of Fluids and Structures*, 129:104167, 2024.

[94] Thomas J R Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2003.

[95] Olgierd C Zienkiewicz and Robert L Taylor. *The finite element method set*. Elsevier, 2005.

[96] Dietrich Braess. *Finite elements: Theory, fast solvers, and applications in solid mechanics*. Cambridge University Press, 2001.

[97] Joachim Nitsche. Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind. In *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*, volume 36, pages 9–15. Springer, 1971.

[98] V Saulev. A method for automatization of the solution of boundary value problems on high performance computers. *Dokl. Akad. Nauk. SSSR v144*, pages 497–500, 1962.

[99] Roland Glowinski, Tsorng-Whay Pan, and Jacques Periaux. A fictitious domain method for dirichlet problem and applications. *Computer Methods in Applied Mechanics and Engineering*, 111(3-4):283–303, 1994.

[100] Alexander Düster, Jamshid Parvizian, Zhengxiong Yang, and Ernst Rank. The finite cell method for three-dimensional problems of solid mechanics. *Computer methods in applied mechanics and engineering*, 197(45-48):3768–3782, 2008.

[101] William F Noh. Cel: A time-dependent, two-space-dimensional, coupled eulerian-lagrange code. Technical report, Lawrence Radiation Lab., Univ. of California, Livermore, 1963.

[102] Charles S Peskin. The immersed boundary method. *Acta numerica*, 11:479–517, 2002.

[103] Charles S Peskin. Numerical analysis of blood flow in the heart. *Journal of computational physics*, 25(3):220–252, 1977.

[104] Rajat Mittal and Gianluca Iaccarino. Annual review of fluid mechanics. *Annual Review of Fluid Mechanics*, 37:239–261, 2005.

[105] Anita Hansbo and Peter Hansbo. An unfitted finite element method, based on nitsche's method, for elliptic interface problems. *Computer methods in applied mechanics and engineering*, 191(47-48):5537–5552, 2002.

[106] John W Barrett and Charles M Elliott. A finite-element method for solving elliptic equations with neumann data on a curved boundary using unfitted meshes. *IMA Journal of Numerical Analysis*, 4(3):309–325, 1984.

[107] John W Barrett and Charles M Elliott. Fixed mesh finite element approximations to a free boundary problem for an elliptic equation with an oblique derivative boundary condition. *Computers & mathematics with applications*, 11(4):335–345, 1985.

[108] John W Barrett and Charles M Elliott. Fitted and unfitted finite-element methods for elliptic equations with smooth interfaces. *IMA journal of numerical analysis*, 7(3):283–300, 1987.

[109] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.

[110] Stéphane P A Bordas, Erik Burman, Mats G Larson, and Maxim A Olshanskii. *Geometrically unfitted finite element methods and applications: Proceedings of the UCL workshop 2016*, volume 121. Springer, 2018.

[111] Thomas-Peter Fries and Ted Belytschko. The extended/generalized finite element method: an overview of the method and its applications. *International journal for numerical methods in engineering*, 84(3):253–304, 2010.

[112] Ronald P Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of computational physics*, 152(2):457–492, 1999.

[113] Cheng Tu and Charles S Peskin. Stability and instability in the computation of flows with moving immersed boundaries: a comparison of three methods. *SIAM Journal on Scientific and Statistical Computing*, 13(6):1361–1376, 1992.

[114] Jens M Melenk and Ivo Babuvska. The partition of unity finite element method: basic theory and applications. *Computer methods in applied mechanics and engineering*, 139(1-4):289–314, 1996.

[115] Theofanis Strouboulis, Ivo Babuvska, and Kevin Copps. The design and analysis of the generalized finite element method. *Computer methods in applied mechanics and engineering*, 181(1-3):43–69, 2000.

[116] Theofanis Strouboulis, Kevin Copps, and Ivo Babuvska. The generalized finite element method: an example of its implementation and illustration of its performance. *International journal for numerical methods in engineering*, 47(8):1401–1417, 2000.

[117] Theofanis Strouboulis, Kevin Copps, and Ivo Babuvska. The generalized finite element method. *Computer methods in applied mechanics and engineering*, 190(32-33):4081–4193, 2001.

[118] Ted Belytschko and Tom Black. Elastic crack growth in finite elements with minimal remeshing. *International journal for numerical methods in engineering*, 45(5):601–620, 1999.

[119] Nicolas Moës, John Dolbow, and Ted Belytschko. A finite element method for crack growth without remeshing. *International journal for numerical methods in engineering*, 46(1):131–150, 1999.

[120] Jamshid Parvizian, Alexander Düster, and Ernst Rank. Finite cell method: h-and p-extension for embedded domain problems in solid mechanics. *Computational Mechanics*, 41(1):121–133, 2007.

[121] Maxim A Olshanskii, Arnold Reusken, and Jörg Grande. A finite element method for elliptic equations on surfaces. *SIAM journal on numerical analysis*, 47(5):3339–3358, 2009.

[122] Erik Burman, Susanne Claus, Peter Hansbo, Mats G Larson, and André Massing. Cutfem: discretizing geometry and partial differential equations. *International Journal for Numerical Methods in Engineering*, 104(7):472–501, 2015.

[123] Erik Burman. Ghost penalty. *Comptes Rendus. Mathématique*, 348(21-22):1217–1220, 2010.

[124] Erik Burman, Peter Hansbo, Mats G Larson, and Sara Zahedi. Cut finite element methods. *Acta Numerica*, 34:1–121, 2025.

[125] Alex Main and Guglielmo Scovazzi. The shifted boundary method for embedded domain computations. part i: Poisson and stokes problems. *Journal of Computational Physics*, 372:972–995, 2018.

[126] Erik Burman and Peter Hansbo. Fictitious domain finite element methods using cut elements: I. a stabilized lagrange multiplier method. *Computer Methods in Applied Mechanics and Engineering*, 199(41-44):2680–2686, 2010.

[127] Dominik Schillinger, Martin Ruess, Nils Zander, Yuri Bazilevs, Alexander Düster, and Ernst Rank. Small and large deformation analysis with the p-and b-spline versions of the finite cell method. *Computational Mechanics*, 50(4):445–478, 2012.

[128] Santiago Badia, Hridya Dilip, and Francesc Verdugo. Space-time unfitted finite element methods for time-dependent problems on moving domains. *Computers & Mathematics with Applications*, 135:60–76, 2023.

[129] Oriol Colomes, Alex Main, Léo Nouveau, and Guglielmo Scovazzi. A weighted shifted boundary method for free surface flow problems. *Journal of Computational Physics*, 424:109837, 2021.

[130] Danjie Xu, Oriol Colomes, Alex Main, Kangan Li, Nabil M Atallah, Nabil Abboud, and Guglielmo Scovazzi. A weighted shifted boundary method for immersed moving boundary simulations of stokes' flow. *Journal of Computational Physics*, 510:113095, 2024.

[131] Kwang J Bai. A localized finite-element method for two-dimensional steady potential flows with a free surface. *Journal of Ship Research*, 22(04):216–230, 1978.

[132] Guoxiong Wu and Rodney Eatock Taylor. Finite element analysis of two-dimensional non-linear transient water waves. *Applied Ocean Research*, 16(6):363–372, 1994.

[133] Guoxiong Wu and Rodney Eatock Taylor. Time stepping solutions of the two-dimensional nonlinear wave radiation problem. *Ocean Engineering*, 22(8):785–798, 1995.

[134] Günther F Clauss and Ulrich Steinhagen. Numerical simulation of nonlinear transient waves and its validation by laboratory data. In *ISOPE International Ocean and Polar Engineering Conference*, pages ISOPE–I. ISOPE, 1999.

[135] Guoxiong Wu, Qingwei Ma, and Rodney Eatock Taylor. Numerical simulation of sloshing waves in a 3d tank based on a finite element method. *Applied ocean research*, 20(6):337–355, 1998.

[136] Iain Robertson and Spencer Sherwin. Free-surface flow simulation using hp/spectral elements. *Journal of Computational Physics*, 155(1):26–53, 1999.

[137] Dick K P Yue, Hsuan S Chen, and Chiang C Mei. A hybrid element method for diffraction of water waves by three-dimensional bodies. *International Journal for Numerical Methods in Engineering*, 12(2):245–266, 1978.

[138] Qingwei Ma, Guoxiong Wu, and Rodney Eatock Taylor. Finite element simulation of fully non-linear interaction between vertical cylinders and steep waves. part 1: methodology and numerical procedure. *International Journal for Numerical Methods in Fluids*, 36(3):265–285, 2001.

[139] Qingwei Ma, Guoxiong Wu, and Rodney Eatock Taylor. Finite element simulations of fully non-linear interaction between vertical cylinders and steep waves. part 2: numerical results and validation. *International Journal for Numerical Methods in Fluids*, 36(3):287–308, 2001.

[140] Nabil M Atallah, Claudio Canuto, and Guglielmo Scovazzi. The shifted boundary method for solid mechanics. *International Journal for Numerical Methods in Engineering*, 122(20):5935–5970, 2021.

[141] Ting Song, Alex Main, Guglielmo Scovazzi, and Mario Ricchiuto. The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows. *Journal of Computational Physics*, 369:45–79, 2018.

[142] Erik Burman and Miguel A Fernández. An unfitted nitsche method for incompressible fluid–structure interaction using overlapping meshes. *Computer Methods in Applied Mechanics and Engineering*, 279:497–514, 2014.

[143] André Massing, Mats Larson, Anders Logg, and Marie Rognes. A nitsche-based cut finite element method for a fluid-structure interaction problem. *Communications in Applied Mathematics and Computational Science*, 10(2):97–120, 2015.

[144] Benedikt Schott, Christoph Ager, and Wolfgang A Wall. Monolithic cut finite element–based approaches for fluid-structure interaction. *International Journal for Numerical Methods in Engineering*, 119(8):757–796, 2019.

[145] Christoph Ager, Benedikt Schott, Magnus Winter, and Wolfgang A Wall. A nitsche-based cut finite element method for the coupling of incompressible fluid flow with poroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 351:253–280, 2019.

[146] Kyle Dunn, Roger Lui, and Marcus Sarkis. An unconditionally stable semi-implicit cutfem for an interaction problem between an elastic membrane and an incompressible fluid. *arXiv preprint arXiv:1801.05794*, 2018.

[147] Josefin Ahlkrona and Daniel Elfverson. A cut finite element method for non-newtonian free surface flows in 2d-application to glacier modelling. *Journal of Computational Physics: X*, 11:100090, 2021.

[148] Jens Visbech, Allen P Engsig-Karup, Harry B Bingham, Mostafa Amini-Afshar, and Mario Ricchiuto. A high-order shifted boundary method for water waves and floating bodies. In *39th International Workshop on Water Waves and Floating Bodies*, 2024.

[149] Christopher E Kees, J Haydel Collins, and Alvin Zhang. Simple, accurate, and efficient embedded finite element methods for fluid–solid interaction. *Computer Methods in Applied Mechanics and Engineering*, 389:114404, 2022.

[150] Rubén Zorrilla and Alessandro Franci. Cut-pfem: a particle finite element method using unfitted boundary meshes. *Engineering with Computers*, 40(5):2739–2760, 2024.

[151] Danjie Xu. The weighted shifted boundary method for moving-boundary flow simulations. *PhD Thesis*, 2025.

[152] Santiago Badia, Pere A Martorell, and Francesc Verdugo. Space–time unfitted finite elements on moving explicit geometry representations. *Computer Methods in Applied Mechanics and Engineering*, 428:117091, 2024.

[153] Eric Neiva and Hervé Turlier. Unfitted finite element modelling of surface-bulk viscous flows in animal cells. *arXiv preprint arXiv:2505.05723*, 2025.

[154] Oriol Colomes, Jan Modderman, and Guglielmo Scovazzi. The generalized shifted boundary method for geometry-parametric pdes and time-dependent domains. *Computer Methods in Applied Mechanics and Engineering*, 452:118748, 2026.

[155] Santiago Badia, Pere A Martorell, and Francesc Verdugo. Geometrical discretisations for unfitted finite elements on explicit boundary representations. *Journal of Computational Physics*, 460:111162, 2022.

[156] Chennakesava Kadapa, Xinyu Wang, and Yue Mei. A comprehensive assessment of accuracy of adaptive integration of cut cells for laminar fluid-structure interaction problems. *Computers & Mathematics with Applications*, 122:1–18, 2022.

[157] Peter Hansbo, Mats G Larson, and Karl Larsson. Cut finite element methods for linear elasticity problems. In *Geometrically Unfitted Finite Element Methods and Applications: Proceedings of the UCL Workshop 2016*, pages 25–63. Springer, 2017.

[158] Ramon Codina, Joan Baiges, Inocencio Castañar, Ignacio Martínez-Suárez, Laura Moreno, and Samuel Parada. An embedded strategy for large scale incompressible flow simulations in moving domains. *Journal of Computational Physics*, 488:112181, 2023.

[159] Santiago Badia, Eric Neiva, and Francesc Verdugo. Linking ghost penalty and aggregated unfitted methods. *Computer Methods in Applied Mechanics and Engineering*, 388:114232, 2022.

[160] Morten Bech Kramer, Jacob Andersen, Sarah Thomas, Flemming Buus Bendixen, Harry Bingham, Robert Read, Nikolaj Holk, Edward Ransley, Scott Brown, Yi-Hsiang Yu, et al. Highly accurate experimental heave decay tests with a floating sphere: A public benchmark dataset for model validation of fluid–structure interaction. *Energies*, 14(2):269, 2021.

[161] Jintai Chung and GM1223971 Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. *Journal of applied mechanics*, 60(2):371–375, 1993.

[162] Nathan M Newmark. A method of computation for structural dynamics. *Journal of the engineering mechanics division*, 85(3):67–94, 1959.

[163] Soichi Ito. Study of the transient heave oscillation of a floating cylinder, massachusetts institute of technology, 1977.

[164] Autodesk Inc. Autodesk Meshmixer, 2017.

[165] Pere A Martorell and Santiago Badia. Stlcutters. jl: A scalable geometrical framework library for unfitted finite element discretisations. *Computer Physics Communications*, 309:109479, 2025.

[166] Cheng-Hau Yang, Kumar Saurabh, Guglielmo Scovazzi, Claudio Canuto, Adarsh Krishnamurthy, and Baskar Ganapathysubramanian. Optimal surrogate boundary selection and scalability studies for the shifted boundary method on octree meshes. *Computer Methods in Applied Mechanics and Engineering*, 419:116686, 2024.

[167] Carsten Burstedde, Lucas C Wilcox, and Omar Ghattas. p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing*, 33(3):1103–1133, 2011.

[168] Christophe Geuzaine and Jean-Francois Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331, 2009.

[169] Amy Robertson, Jason Jonkman, Marco Masciola, Huimin Song, Andrew Goupee, Alexander Coulling, and C Luan. Definition of the semisubmersible floating system for phase ii of oc4. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2014.

[170] Lu Wang, Amy Robertson, Jason Jonkman, Jang Kim, Zhi-Rong Shen, Arjen Koop, Adrià Borràs Nadal, Wei Shi, Xinmeng Zeng, Edward Ransley, et al. Oc6 phase ia: Cfd simulations of the free-decay motion of the deepcwind semisubmersible. *Energies*, 15(1):389, 2022.

[171] Irene Rivera-Arreba, Niek Bruinsma, Erin E Bachynski, Axelle Viré, Bo T Paulsen, and Niels G Jacobsen. Modeling of a semisubmersible floating offshore wind platform in severe waves. *Journal of offshore mechanics and Arctic engineering*, 141(6):061905, 2019.

[172] Hui Liang, Odd M Faltinsen, and Yanlin Shao. Application of a 2d harmonic polynomial cell (hpc) method to singular flows and lifting problems. *Applied Ocean Research*, 53:75–90, 2015.

[173] J H Vugts. The hydrodynamic coefficients for swaying, heaving and rolling cylinders in a free surface. *International Shipbuilding Progress*, 15(167):251–276, 1968.

[174] Martin Greenhow and S I Ahn. Added mass and damping of horizontal circular cylinder sections. *Ocean Engineering*, 15(5):495–504, 1988.

[175] Jason Jonkman. Definition of the floating system for phase iv of oc3. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2010.

[176] Santiago Badia, Alberto F Martín, Eric Neiva, and Francesc Verdugo. The aggregated unfitted finite element method on parallel tree-based adaptive meshes. *SIAM Journal on Scientific Computing*, 43(3):C203–C234, 2021.

[177] Vaibhav Raghavan, Eva Loukogeorgaki, Nikos Mantadakis, Andrei V Metrikine, and George Lavidas. Hams-mrel, a new open source multiple body solver for marine renewable energies: Model description, application and validation. *Renewable Energy*, 237:121577, 2024.

[178] Shaswat Saincher and Venkatachalam Sriram. A three dimensional hybrid fully nonlinear potential flow and navier stokes model for wave structure interactions. *Ocean Engineering*, 266:112770, 2022.

[179] Antoine Marty, Christian Berhault, Guillaume Damblans, Jean V Facq, Benoit Gaurier, Gregory Germain, T Soulard, and Franck Schoefs. Experimental study of hard marine growth effect on the hydrodynamical behaviour of a submarine cable. *Applied Ocean Research*, 114:102810, 2021.

[180] Antoine Marty, Franck Schoefs, Guillaume Damblans, Jean V Facq, Benoît Gaurier, and Gregory Germain. Experimental study of two kinds of hard marine growth effects on the hydrodynamic behavior of a cylinder submitted to wave and current loading. *Ocean Engineering*, 263:112194, 2022.

[181] Shagun Agarwal, Sergio Sánchez Gómez, and Oriol Colomes. Dynamic finite-strain analysis of mooring lines based on tangential differential calculus. *Available at SSRN 5271454*, 2025.

[182] Jason Jonkman, Emmanuel Branlard, Matthew Hall, Greg Hayman, Andrew Platt, and Amy Robertson. Implementation of substructure flexibility and member-level load capabilities for floating offshore wind turbines in openfast. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2020.

[183] Guglielmo Scovazzi, Danjie Xu, Oriol Colomes, Alex Main, Kangan Li, Nabil M Atallah, and Nabil Abboud. A weighted shifted boundary method for the navier-stokes equations with immersed moving boundaries. *Available at SSRN 5251653*, 2025.

[184] Santiago Badia and Francesc Verdugo. Gridap: An extensible finite element toolbox in julia. *Journal of Open Source Software*, 5(52):2520, 2020.

[185] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.

A

Derivation of Discrete Weak Forms for Benchmark Study

In this appendix, we show the derivation of the weak formulations for the benchmark study described in chapter 4 of this dissertation. Recalling that the conformal weak form is given as:

find $\phi_h \in \mathcal{W}_h$ such that

$$a_h(\phi_h, w_h) = b_h(w_h) \qquad \forall w_h \in \mathcal{W}_h, \tag{A.1}$$

with

$$a_h(\phi_h, w_h) := (\nabla w_h, \nabla \phi_h)_{\Omega_h}, \tag{A.2}$$

and

$$b_h(w_h) := (w_h, f)_{\Omega_h} + (w_h, \nabla \phi^* \cdot \boldsymbol{n}_1)_{\Gamma_{1,h}} + (w_h, \nabla \phi^* \cdot \boldsymbol{n}_2)_{\Gamma_{2,h}}. \tag{A.3}$$

For AgFEM, we know the weak form remains the same, and only the integration domains and discrete space definition are modified as described in chapter 3. Similarly, for CutFEM, the discrete space is defined on the active domain; additionally, we add the ghost penalty stabilization enforced on the ghost skeleton to the weak form. As these derivations are straightforward, the reader is referred to their definitions in Equation 4.8 and Equation 4.11 for CutFEM and AgFEM, respectively.

For SBM and WSBM, the derivations are more involved. Consider the shifted Neumann-type boundary condition as defined Equation 3.26. We now enforce that the shifted boundary condition has the prescribed value, given by $\nabla \phi^*$. We substitute $\nabla \phi^*$ for $\frac{\partial \mathbf{u}_h}{\partial t}$ in the derivation process started in Equation 3.26, and since this term is known, it is moved to the linear term of the weak form. Starting with the following assumption:

$$\nabla \phi^* \cdot \boldsymbol{n}_1 = (\nabla \phi_h \cdot \boldsymbol{n}_1)|_{\Gamma_{1,h}} \approx (\nabla \phi_h \cdot \boldsymbol{n}_1)|_{\tilde{\Gamma}_{1,h}} + ((\nabla \nabla \phi_h \cdot \boldsymbol{n}_1) \cdot \mathbf{d})|_{\tilde{\Gamma}_{1,h}}. \tag{A.4}$$

We then apply the same substitution as conducted in chapter 3 to include this approximation in the weak formulation and reformulate the equation such that there are no tangential vector contributions.

A

$$(w_h, \nabla\phi^* \cdot (\tilde{\boldsymbol{n}}_1 \cdot \boldsymbol{n}_1)\boldsymbol{n}_1)_{\tilde{\Gamma}_{1,h}} + (w_h, \nabla\phi_h \cdot \tilde{\boldsymbol{n}}_1 - (\nabla(\nabla\phi_h)\cdot \boldsymbol{d} + \nabla\phi_h)\cdot(\tilde{\boldsymbol{n}}_1 \cdot \boldsymbol{n}_1)\boldsymbol{n}_1))_{\tilde{\Gamma}_{1,h}}. \tag{A.5}$$

We can then reformulate the entire weak form using the shifted contribution defined in Equation A.5, taking into account the modified finite element space $\mathcal{W}_h^{\text{in}}$:

find $\phi_h \in \mathcal{W}_h^{\text{in}}$ such that

$$a_h^{\text{sbm}}(\phi_h, w_h) = b_h^{\text{sbm}}(w_h) \qquad \forall w_h \in \mathcal{W}_h^{\text{in}}, \tag{A.6}$$

with

$$\begin{aligned} a_h^{\text{sbm}}(\phi_h, w_h) := &(\nabla w_h, \nabla\phi_h)_{\tilde{\Omega}_h} + \\ &(w_h, \nabla\phi_h \cdot \tilde{\boldsymbol{n}}_1 - (\nabla(\nabla\phi_h)\cdot \boldsymbol{d} + \nabla\phi_h)\cdot(\tilde{\boldsymbol{n}}_1 \cdot \boldsymbol{n}_1)\boldsymbol{n}_1))_{\tilde{\Gamma}_{1,h}}, \end{aligned} \tag{A.7}$$

and

$$b_h^{\text{sbm}}(w_h) := (w_h, f)_{\tilde{\Omega}_h} + (w_h, \nabla\phi^* \cdot (\tilde{\boldsymbol{n}}_1 \cdot \boldsymbol{n}_1)\boldsymbol{n}_1)_{\tilde{\Gamma}_{1,h}} + (w_h, \nabla\phi^* \cdot \boldsymbol{n}_2)_{\Gamma_{2,h}}. \tag{A.8}$$

Similarly, for the WSBM, we obtain the same general weak form, but now we have to add all contributions to the ghost skeleton triangulation as well. Lastly, the ghost penalty stabilization terms should also be applied. We start by building on the weak form defined for the SBM, see Equation A.9. For the WSBM, we need to introduce the weighted test functions $w_{h,\alpha}$, accompanied by the Petrov-Galerkin discretization, see Equation 3.40. Also note that we enforce continuity such that $[\![\nabla\phi]\!] = 0$. Rewriting the bilinear form Equation A.7 term by term as:

$$\begin{aligned} (\nabla w_h, \nabla\phi_h)_{\tilde{\Omega}_h} &\to (\nabla w_h, \nabla\phi_h)_{\tilde{\Omega}_h} + (\nabla w_{h,\alpha}, \nabla\phi_h)_{\Omega_{\text{c},h}} \\ (w_h, \nabla\phi_h \cdot \tilde{\boldsymbol{n}}_1)_{\tilde{\Gamma}_{1,h}} &\to \left(w_{h,\alpha}, \nabla\phi_h \cdot \hat{\boldsymbol{n}}_1\right)_{\hat{\Gamma}_{1,h}} \\ (w_h, (\nabla(\nabla\phi_h)\cdot \boldsymbol{d} + \nabla\phi_h)\cdot(\tilde{\boldsymbol{n}}_1 \cdot \boldsymbol{n}_1)\boldsymbol{n}_1))_{\tilde{\Gamma}_{1,h}} &\to \left(w_{h,\alpha}, (\nabla(\nabla\phi_h)\cdot \boldsymbol{d} + \nabla\phi_h)\cdot(\hat{\boldsymbol{n}}_1 \cdot \boldsymbol{n}_1)\boldsymbol{n}_1)\right)_{\hat{\Gamma}_{1,h}} \end{aligned}$$

here, we have changed the contributions from the inner to the outer surrogate boundary, and we have applied the weighted test functions.

We introduce the terms related to ghost skeleton triangulation. We reiterate that these consist of the results from the application of the shifting operators and the ghost penalty stabilization.

$$\begin{aligned} \left(w_{h,\alpha}, \nabla\phi_h \cdot \tilde{\boldsymbol{n}}\varepsilon_0\right)_{\mathcal{E}_h^0} &\to \langle [\![w_{h,\alpha}]\!], \{\nabla\phi_h\}\rangle_{\mathcal{E}_h^0} + \cancelto{0}{\langle \{w_{h,\alpha}\}, [\![\nabla\phi_h]\!]\rangle_{\mathcal{E}_h^0}} \\ \left(w_{h,\alpha}, \nabla(\nabla\phi_h)\cdot \mathbf{d} + \nabla\phi_h \cdot \boldsymbol{n}\varepsilon_0\right)_{\mathcal{E}_h^0} &\to \langle \widetilde{[\![w_{h,\alpha}]\!]}, \{\nabla(\nabla\phi_h)\cdot \mathbf{d} + \nabla\phi_h\}\rangle_{\mathcal{E}_h^0} + \\ &\quad \langle \{w_{h,\alpha}\}, [\![\widetilde{\nabla(\nabla\phi_h)\cdot \mathbf{d}} + \nabla\phi_h]\!]\rangle_{\mathcal{E}_h^0} \\ \text{ghost penalty} &\to \sum_{i=1}^{p} \langle [\![\nabla^{(i)} w_h]\!], \gamma_g h_e^{(2i+1)} [\![\nabla^{(i)}\phi_h]\!]\rangle_{\mathcal{E}_h^0} \end{aligned}$$

Similarly, for the linear form of Equation A.9, we add weighted test functions, rewrite from the inner surrogate to the outer surrogate boundary, and include the ghost skeleton terms:

$$
\begin{aligned}
(w_h, f)_{\tilde{\Omega}_h} &\rightarrow (w_h, f)_{\tilde{\Omega}_h} + \left(w_{h,\alpha}, f\right)_{\Omega_{c,h}} \\
(w_h, \nabla\phi^* \cdot (\tilde{\boldsymbol{n}}_1 \cdot \boldsymbol{n}_1)\boldsymbol{n}_1)_{\tilde{\Gamma}_{1,h}} &\rightarrow \left(w_{h,\alpha}, \nabla\phi^* \cdot (\boldsymbol{n}_1 \cdot \hat{\boldsymbol{n}}_1)\boldsymbol{n}_1\right)_{\hat{\Gamma}_{1,h}} \\
\left(w_h, \nabla\phi^* \cdot (\tilde{\boldsymbol{n}}_{\mathcal{E}_0} \cdot \boldsymbol{n}_{\mathcal{E}_0})\boldsymbol{n}_{\mathcal{E}_0}\right)_{\mathcal{E}_h^0} &\rightarrow \langle \widetilde{[\![w_{h,\alpha}]\!]}, \{\nabla\phi^*\}\rangle_{\mathcal{E}_h^0} + \langle \{w_{h,\alpha}\}, \widetilde{[\![\nabla\phi^*]\!]}\rangle_{\mathcal{E}_h^0}
\end{aligned}
$$

In doing so, we end up with the complete weak form for the WSBM, which is written as:

find $\phi_h \in \mathcal{W}_h^{\text{act}}$ such that

$$
a_h^{\text{wsbm}}(\phi_h, w_h) = b_h^{\text{wsbm}}(w_h) \qquad \forall w_h \in \mathcal{W}_h^{\text{act}}, \tag{A.9}
$$

with

$$
\begin{aligned}
a_h^{\text{wsbm}}(\phi_h, w_h) := &(\nabla w_h, \nabla\phi_h)_{\tilde{\Omega}_h} + \left(\nabla w_{h,\alpha}, \nabla\phi_h\right)_{\Omega_{c,h}} - \left(w_{h,\alpha}, \nabla\phi_h \cdot \hat{\boldsymbol{n}}_1\right)_{\hat{\Gamma}_{1,h}} + \\
&\left(w_{h,\alpha}, (\boldsymbol{n}_1 \cdot \hat{\boldsymbol{n}}_1)\boldsymbol{n}_1 \cdot [\nabla(\nabla\phi_h) \cdot \mathbf{d} + \nabla\phi_h]\right)_{\hat{\Gamma}_{1,h}} - \\
&\langle [\![w_{h,\alpha}]\!], \{\nabla\phi_h\}\rangle_{\mathcal{E}_h^0} + \\
&\langle \widetilde{[\![w_{h,\alpha}]\!]}, \{\nabla(\nabla\phi_h) \cdot \mathbf{d} + \nabla\phi_h\}\rangle_{\mathcal{E}_h^0} + \\
&\langle \{w_{h,\alpha}\}, [\![\nabla(\widetilde{\nabla\phi_h) \cdot \mathbf{d}} + \nabla\phi_h]\!]\rangle_{\mathcal{E}_h^0} + \\
&\sum_{i=1}^{p} \langle [\![\nabla^{(i)} w_h]\!], \gamma_g h_e^{(2i+1)} [\![\nabla^{(i)} \phi_h]\!]\rangle_{\mathcal{E}_h^0},
\end{aligned} \tag{A.10}
$$

and

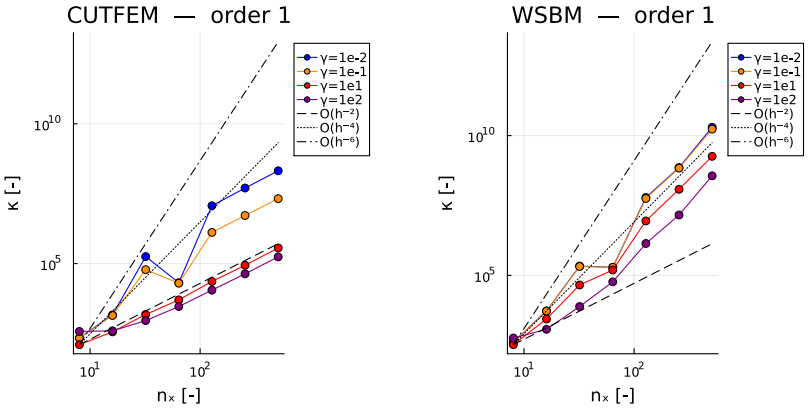
$$
\begin{aligned}
b_h^{\text{wsbm}}(w_h) := &(w_h, f)_{\tilde{\Omega}_h} + \left(w_{h,\alpha}, f\right)_{\Omega_{c,h}} + \left(w_{h,\alpha}, \nabla\phi^* \cdot \boldsymbol{n}_2\right)_{\hat{\Gamma}_{2,h}} + \\
&\left(w_{h,\alpha}, \nabla\phi^* \cdot (\boldsymbol{n}_1 \cdot \hat{\boldsymbol{n}}_1)\boldsymbol{n}_1\right)_{\hat{\Gamma}_{1,h}} + \\
&\langle \widetilde{[\![w_{h,\alpha}]\!]}, \{\nabla\phi^*\}\rangle_{\mathcal{E}_h^0} + \langle \{w_{h,\alpha}\}, \widetilde{[\![\nabla\phi^*]\!]}\rangle_{\mathcal{E}_h^0}.
\end{aligned} \tag{A.11}
$$

B

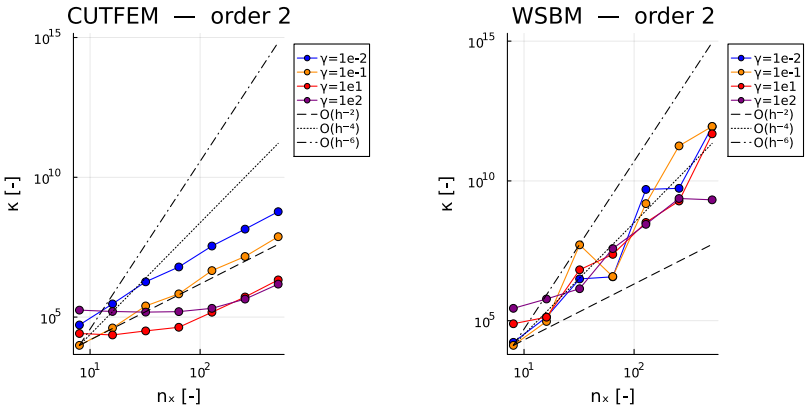
Condition Number Sensitivity Analysis for Benchmark Study

In this appendix, we present the results of the sensitivity analysis for the ghost penalty parameter γ_g for the benchmark study described in Chapter 4. The sensitivity analysis considers $\gamma_g \in 0.01, 0.1, 1.0, 10.0$ and evaluates both the condition number scaling with mesh size h and the convergence rate of the error. For all CutFEM simulations, $\gamma_g = 0.1$ is selected, as this value recovers the theoretically expected condition number scaling of $\mathcal{O}(h^{-2})$ while preserving optimal convergence rates for both $p_e = 1$ and $p_e = 2$. For WSBM in 2D, the sensitivity analysis reveals that no tested value of γ_g recovers the expected $\mathcal{O}(h^{-2})$ condition number scaling. Furthermore, the convergence rates and absolute errors are not improved by the inclusion of ghost penalty stabilization for any tested value, suggesting that the ghost penalty term as formulated does not effectively stabilize the surrogate boundary strip in the 2D case. We therefore choose to run the 2D WSBM simulations without ghost penalty stabilization $\gamma_g = 0$, noting that this is a limitation of the current formulation and that the stability properties of the 2D WSBM without ghost penalty are assessed directly through the observed convergence behavior. For WSBM in 3D, $\gamma_g = 10$ is selected on the basis of two criteria: it yields condition number scaling closest to the theoretically expected $\mathcal{O}(h^{-2})$ rate among all tested values, and it produces, generally, the lowest absolute condition numbers at comparable mesh refinement levels. Smaller values of γ_g result in steeper condition number growth, indicating insufficient stabilization of the strip between the true and surrogate boundaries, while larger values do not improve conditioning or convergence rates.

B

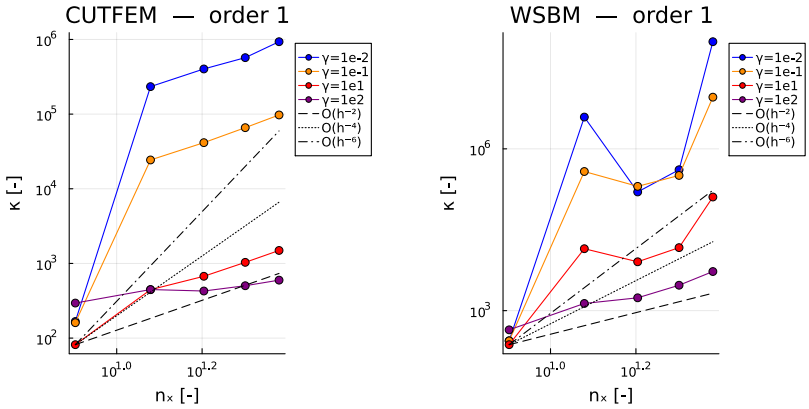


(a) Order $p_e = 1$

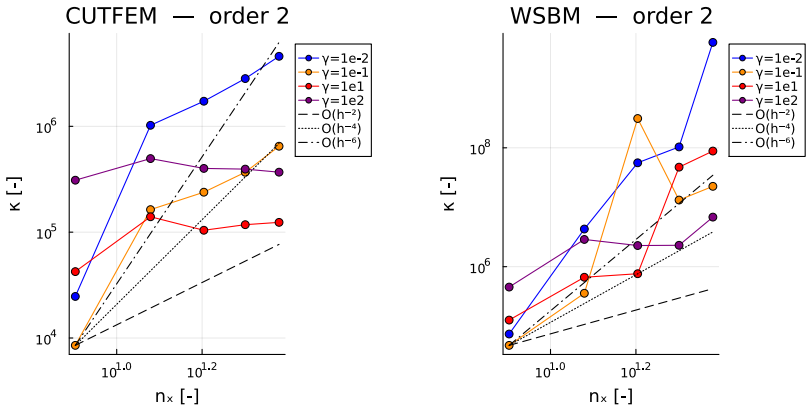


(b) Order $p_e = 2$

Figure B.1: Condition number sensitivity study for varying grid resolutions via method of manufactured solutions for CutFEM and WSBM with varying orders for the 2D cylinder implicit geometry.



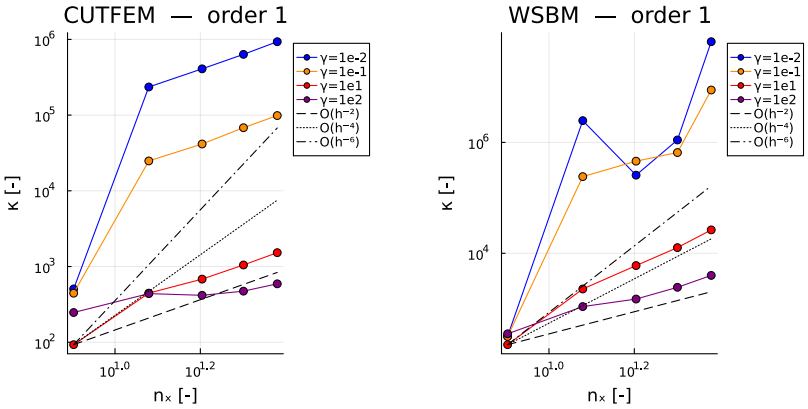
(a) Order $p_e = 1$



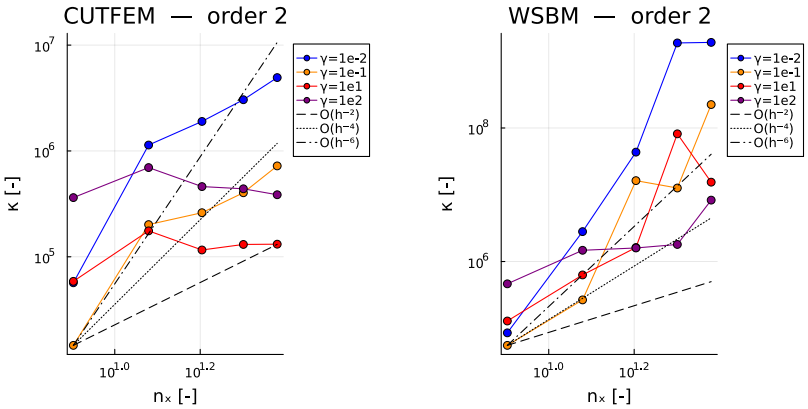
(b) Order $p_e = 2$

Figure B.2: Condition number sensitivity study for varying grid resolutions via method of manufactured solutions for CutFEM and WSBM with varying orders for the 3D sphere implicit geometry.

B

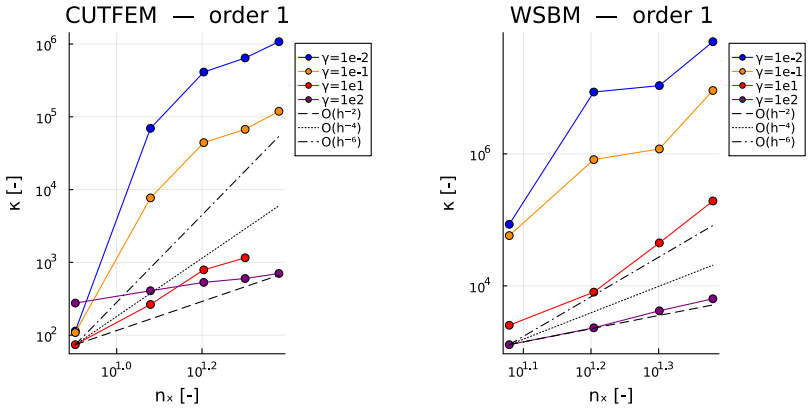


(a) Order $p_e = 1$

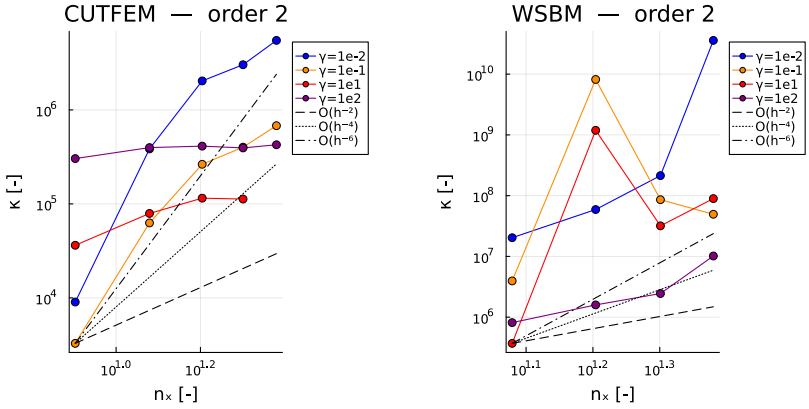


(b) Order $p_e = 2$

Figure B.3: Condition number sensitivity study for varying grid resolutions via method of manufactured solutions for CutFEM and WSBM with varying orders for the 3D sphere explicit geometry.



(a) Order $p_e = 1$



(b) Order $p_e = 2$

Figure B.4: Condition number sensitivity study for varying grid resolutions via method of manufactured solutions for CutFEM and WSBM with varying orders for the 3D Stanford bunny explicit geometry.

C

Numerical Implementation and Reproducibility

In this appendix, we provide a brief introduction to `Gridap.jl` [184] software package and some of its sub-packages. There are several tutorials available online, located at `https://gridap.github.io/Tutorials/dev/`, and this chapter serves as a brief introduction to readers unfamiliar with Gridap; it provides them with a concise description of how the software operates and how it has been used within this work. It explains how the work in this dissertation is generated, ensuring that the results are reproducible and the process of obtaining these results is transparent.

Code overview and philosophy

`Gridap.jl` is an open-source finite element framework implemented in the Julia [185] programming language. Its design follows the abstraction layers of the finite element method:

- **Discrete model:** geometric and topological description of the computational domain, including cells, faces, quadrature rules, and boundary tags.
- **Finite element spaces:** conforming or nonconforming function spaces constructed algebraically from the discrete model.
- **Variational forms:** user-specified weak formulations expressed in a symbolic, Julia-like syntax.
- **Assembly:** automatic generation and numerical evaluation of element contributions to form global matrix contributions.
- **Solvers:** direct or iterative methods from the Julia ecosystem, with support for Newton iterations, preconditioners, and external (sub-)packages.

Additionally, the main dependencies required for this dissertation are the sub-packages `GridapEmbedded.jl` and `STLCutters.jl`. The former package holds the infrastructure to define embedded boundaries using level-set functions and stores this data as a geometry data structure. It also provides the data structures for the tessellation of the integration space, as well as the backend for the cell aggregation algorithm and skeleton triangulation for ghost penalty stabilization. The latter package introduces the inclusion of explicitly defined geometrical representations, which are compatible with the data structures defined in the former package.

Software Environment and Dependencies

Accompanying this dissertation, there are three repositories available on my personal Github page: `github.com/janmodderman`. An overview of the chapter in this dissertation and the accompanying repository is presented in Table C.1.

Chapter 4	`github.com/janmodderman/EmbeddedBenchmark.jl`
Chapter 5	`github.com/janmodderman/EmbeddedTransientPF.jl`
Chapter 6	`github.com/janmodderman/EmbeddedCoefficients.jl`

Table C.1: Overview of available Github repositories for each chapter in this dissertation.

These repositories can be cloned, and all required dependencies are listed in the `Project.toml` file for each repository. The Julia version used to generate the results is listed in the `Manifest.toml` for each repository. For postprocessing in Paraview, we have used both Paraview 5.9.1 and 5.12.0.

Reproducibility Statement

All code used to generate the results in this dissertation is available in the aforementioned repositories. The mesh files from GMSH for the 2D and 3D simulations of chapter 6 are not available on `GitHub`, as these files are too large. Therefore, the .geo files are added instead so that the mesh files can be generated locally using GMSH. The GMSH version used is 4.13.1.

Code Organization and Execution Workflow

All repositories follow the same structure regarding code organization. Each repository consists of three directories, namely `data`, `scripts`, and `src`. In `data`, we store the mesh files required for running with GMSH meshes, the STL files for the geometries when using an explicit representation, and lastly, the results from the simulations are stored there. In `scripts`, the execution scripts for the code are stored, specifically bash scripts for code execution on a cluster. In `src`, the majority of the code is stored. Generally, a separation is made between the unfitted methods for each file, and there will be an accompanying helper file in which all required definitions are stored. Each unfitted method file will then call upon this helper file to construct the case study, and an execution script will call upon each of these unfitted method files to conduct the full case study. There is an option for the simulation files to output the spatial data for each time step, and this is written to the

VTK/VTU file type. for static problems, this means only one time step, and for transient problems, a PVD file is created. Additionally, there are option flags to write the data of interest to a CSV file and an output text file, which has the option to include compute times for each time step, as well as compute times for all the work before the solve phase starts. It is important to note that, as of the writing of this dissertation, the work to incorporate both the embedded methods and adaptive mesh refinement using `p4est` is still ongoing. For this reason, the implementation that is available online is functional but quite crude and non-optimal. Hopefully, a more elegant implementation will be available within the coming year of 2026.

Minimal Working Example

For a working example and guidance on how to read through the code, we select the transient simulation of the OC4 semisubmersible with a Cartesian mesh using AgFEM. This case is available online as `oc4_agfem.jl`. This piece of code is a simpler version of the results available in chapter 5.

In Figure C.1, we start the problem by introducing the geometry and generating the background model data structure. The domain size is given as a tuple of coordinates, and the second tuple contains the number of elements defined in each direction. We add labels to the background model so that we can identify the relevant boundaries, i.e., the free surface in this case.

```
function main()
    folder = "/home/jmodderman1/jmodderman1/research/benchmarking/data/sims/oc4_agfem_cart"


    geo = STLGeometry("/home/jmodderman1/jmodderman1/research/benchmarking/data/geometries/oc4_nobraces_coarse.stl")



    bgmodel = CartesianDiscreteModel((-100,100,-100,100,-200,0),(200,200,100))
    labels_Γ = get_face_labeling(bgmodel)  # get the face labeling of model_Γ
    add_tag_from_tags!(labels_Γ, "surface", [22])
```

Figure C.1: Definition of the geometry, background model and boundary tags.

Then in Figure C.2, we call upon the cut functions: `cut` and `cut_facets`. These return the data structures related to the IN, OUT, and CUT flags we need to define the active and physical triangulations. Within these functions, an intersection algorithm is called. Note that the `cut_facets` function in this case is dependent on the variable `cutgeo`. This is only the case for `STLCutters.jl`, as both data structures are returned at the same time. In contrast, in `GridapEmbedded.jl`, these are generated separately.

The physical domain, which is the triangulation that is algorithmically tessellated, is created using `cutgeo`, where we explicitly define the flag `PHYSICAL_OUT`. The standard flag is `PHYSICAL`, which corresponds to `PHYSICAL_IN`. The embedded boundary, which is algorithmically tessellated as well, is generated using the `EmbeddedBoundary` function. When one of the domain boundaries intersects with the embedded boundary, we need to obtain the physical domain of this intersected boundary as well. To generate the physical free surface boundary, we require the aforementioned `cutfacets`, again with the flag defining the region of interest, as well as the boundary tags.

Then, we define the normal vectors, now requiring the minus sign as our domain of interest is flagged OUT. Afterward, we briefly introduce the polynomial order, define the degree of the quadrature, and generate the required integration space using the `Measure` function.

```
cutgeo = cut(bgmodel,geo)
cutfacets = cut_facets(cutgeo)

Ω_bg = Triangulation(bgmodel)
Ω = Triangulation(cutgeo,PHYSICAL_OUT)
Γ = EmbeddedBoundary(cutgeo)
Γf = BoundaryTriangulation(cutfacets,PHYSICAL_OUT;tags=["surface"])

# Setup normal vectors
n_Γ = -get_normal_vector(Γ)

order = 1
degree = 2*order
dΩ = Measure(Ω,degree)
dΓ = Measure(Γ,degree)
dΓf = Measure(Γf,degree)
```

Figure C.2: Creation of data structures of geometries cut with the background model, definition of the triangulations, normal vectors, and integration spaces.

The finite element spaces should then be constructed; see Figure C.3. As this is the AgFEM, there is a need for an additional step with respect to the other unfitted methods. Initially, we define the active triangulation, now using the flag `ACTIVE_OUT`. Then, a standard finite element space using Lagrangian reference elements is defined on this active triangulation. For AgFEM, we then require identifying the aggregates using the cell aggregation algorithm, after which it is used to restrict the degrees of freedom in the finite element space `Wstd` by calling the function `AgFEMSpace`. This modified finite element space is a finite element space with linear constraints.

The displacement of the rigid object is defined using the `ConstantFESpace`; this is a space that is defined over the entire domain but has a single degree of freedom. Note that more degrees of freedom can be added if we are interested in multidirectional motion. To facilitate time stepping, the trial finite element spaces are defined, and all spaces are stored into multifield spaces.

We introduce a number of variables required for the weak formulation in Figure C.4. The weak formulation is constructed using four functions: `a` corresponds to the terms of the weak form that hold second order derivatives in time, `b` corresponds to first order derivatives in time, and `c` corresponds to zeroth order derivatives in time. In `rhs` the righthand side is defined and set to zero.

```
# Setup FESpace
Ω_act = Triangulation(cutgeo,ACTIVE_OUT)
Wstd = FESpace(Ω_act,ReferenceFE(lagrangian,Float64,order),conformity=:H1)
aggregates = aggregate(AggregateAllCutCells(),cutgeo,geo,OUT)
W = AgFEMSpace(Wstd,aggregates)

Φ = TransientTrialFESpace(W)
V = ConstantFESpace(bgmodel)

U = TransientTrialFESpace(V)

X = TransientMultiFieldFESpace([Φ,U])
Y = MultiFieldFESpace([W,V])
```

Figure C.3: Introducing the active triangulation and defining the finite element spaces for both unknowns.

```
nz = VectorValue(0.0,0.0,1.0)
g=9.81
m_ρ = 2.4212094663404615#(1.3989e8/g)/1025/sum(∫(1.0)dΓ)  #0.000005    # [kg/m] mass 13437500
γg=0.1
h=200.0/5.0

a(t, (ϕ, u), (w, v)) = ∫( v * (u * m_ρ ) )dΓ + ∫(w * (ϕ/g ))dΓf

b(t, (ϕ, u), (w, v)) = ∫( v * (ϕ * (nz⋅n_Γ)) )dΓ -
                       ∫( w * u * (nz⋅n_Γ) )dΓ

c(t, (ϕ, u), (w, v)) = ∫( ∇(ϕ)⋅∇(w) )dΩ +
                       ∫( v * ( g * u * (nz⋅n_Γ) ))dΓ

rhs(t, (w, v)) = ∫(w * 0.0)dΩ
```

Figure C.4: Defining variables required for the weak formulation and introducing the weak formulation itself.

In Figure C.5, the time integrator variables are set up. We construct the operator to solve for our presented weak formulation. Additionally, we provide the optional keyword for constant forms, as the weak form does not change over time. The spatial solver is defined, as well as the transient solver. Then, the initial conditions must be presented, which are created using new functions that, in this case, return scalar values. Using the `interpolate_everywhere` function, these initial conditions are applied throughout our domain. Lastly, we can construct the iterator for solving this problem.

Lastly, in Figure C.6, we show how the solve iterator can be used to solve our transient problem and output all relevant data to the desired output formats using the `DataFrames.jl` and `CSV.jl` packages.

C

```
# time integrator variables
ρ∞ = 0.0                    # time integrator constant corresponding to Newmark settings

op = TransientLinearFEOperator((c,b,a),rhs, X, Y;constant_forms=(true,true,true))

Δt = 1.0#10e-6#1/200                          # [s]: time step
t₀ = 0.0                                # [s]: zero time
Tf = 40*Δt #0.66 #3.0                             # [s]: final time

solver = LUSolver()
ode_solver = GeneralizedAlpha2(solver, Δt, ρ∞)

uᵢ(x,t) = 4.0#2             # [m] heave displacement
uᵢ(t::Real) = x -> uᵢ(x,t)
ϕᵢ(x,t) = 0.0             # [m²/s] velocity potential
ϕᵢ(t::Real) = x -> ϕᵢ(x,t)
ηᵢ(x,t) = 0.0              # [m] wave height
ηᵢ(t) = x -> ηᵢ(x,t)

x₀ = interpolate_everywhere([ϕᵢ(0.0), uᵢ(0.0)], X(0.0))
v₀ = interpolate_everywhere([ϕᵢ(0.0), ηᵢ(0.0)], X(0.0))

ϕhₜ = solve(ode_solver, op, t₀, Tf, (x₀,v₀,v₀))
```

Figure C.5: Introducing the variables required for time discretization, as well as the solvers and setting the initial conditions.

```
ϕhₜ = solve(ode_solver, op, t₀, Tf, (x₀,v₀,v₀))

Δtout = Δt
global tout = 0
plot_u = Float64[]
plot_t = Float64[]
push!(plot_u,4.0)
push!(plot_t,0.0)
touch("$folder/heave.csv")
createpvd("$(folder)/results_ex1") do pvd
    createvtk(Ω,"$(folder)/results_ex1_0.vtu",cellfields=["phih"=>x₀[1],"grad_phih"=>v₀[1],"uh"=>x₀[2]])
    for (t,(ϕh,uh)) in ϕhₜ
        if t>=tout
            pvd[t] = createvtk(Ω,"$(folder)/results_ex1_$t.vtu",cellfields=["phih"=>ϕh,"grad_phih"=>∇(ϕh),"uh"=>uh])
            tout=t+Δtout
            push!(plot_u,uh.free_values[1])
            push!(plot_t,t)
        end # if
    end # for
end # do
df = DataFrame(time=plot_t,heave=plot_u,heave_norm=plot_u./4.0)
CSV.write("$(folder)/heave.csv", df, writeheader=false)
end # function
```

Figure C.6: Given the solve constructor, we can iterate over this and obtain the solution at each time step, which is then transformed to VTK output file and to CSV.

Acknowledgments

A PhD is often portrayed as a solitary journey, but in reality it is never walked alone. This work is the result of the guidance, insight, and encouragement of many people whom I have had the privilege to meet along the way. I am deeply grateful to all those who contributed, in ways both large and small, to the completion of this dissertation.

I would like to express my deepest gratitude to my daily supervisor and copromotor, Oriol. He introduced me to the joy of research through a computer screen during the COVID-19 lockdowns of my master's thesis, and it was he who encouraged me to apply for this PhD position. Throughout the PhD, Oriol taught me how to do research in the fullest sense of the word: how to explore ideas, how to navigate the vast landscape of academic literature, and how to remain curious while still staying grounded. He gave me the freedom to follow rabbit holes of inquiry while at the same time providing the boundaries that kept the work focused and meaningful. The countless long meetings in which we discussed everything from numerical methods to broader scientific questions, the hours spent debugging code together, and the shared pursuit of a deeper understanding of the models and methods we encountered have been a defining part of this PhD. Our daily status updates over coffee—especially during the early days when the research group was still small—created a uniquely close and stimulating working environment. I am deeply grateful for his patience, for creating a safe atmosphere in which no question or idea was ever out of bounds, for rewarding curiosity with enthusiasm, for the witty remarks woven into even the most technical discussions, and for showing me the many facets of what it means to be an excellent researcher. I would also like to thank him for introducing me to the finite element method, which has shaped both this thesis and my development as a computational scientist.

I would also like to thank my promotor, Andrei. While his role was more formal and less involved in the day-to-day technical details of this work, I am grateful to him for teaching me how to look at research from a broader scientific perspective. His guidance and oversight helped place this dissertation within a wider academic context, and I sincerely appreciate his support throughout my PhD.

I am grateful to the developers of Gridap and its associated ecosystem, in particular Eric, Jordi, and Alberto, for the tools that made much of this research possible and for their discussions, insights, and support.

I would like to thank the members of the OE and DSS groups for the many lunches, coffees, seminars, and stimulating conversations. The *bright minds* and open atmosphere in these groups made the department a genuinely inspiring place to work.

I am especially thankful to my full-time office mates, Rens, Julie, Avni, and Shagun, and to the visiting PhDs, Nacho and Manuel, for the everyday companionship that made the PhD far more than just research. Thank you for being an outlet for frustrations, for the support, the travels, the memes, the food truck lunches, the beers, and the board games.

Our office was not only a place of work but also a place of friendship, laughter, and shared experiences; I will always look back on it with warmth.

Outside of academia, I want to thank the people who kept me grounded. To the guys from Caldera, TeamBij, and Daniel: thank you for being there, for listening, for distracting me when I needed it, and for reminding me that there is a rich and meaningful world beyond papers, deadlines, and code.

I am deeply grateful to my parents, Netty and Ewoud, for their boundless support. Your belief in me, your encouragement, and your unwavering presence throughout this long journey have meant more to me than I can express. Lastly, I want to thank my partner, Fiona. Thank you for your patience, your love, and your support during the many ups and downs of this PhD. You were my anchor when things were difficult and my joy when things went well.

Jan
Rotterdam, January 2026

Curriculum Vitæ

Jan Modderman

1996-05-21 Born in Woerden, The Netherlands

Education

2008-2014 Voorbereidend Wetenschappelijk Onderwijs (VWO)
Kalsbeek College Schilderspark

2015-2019 Bachelor of Science in Applied Physics
Technische Universiteit Delft

2019-2021 Master of Science in Offshore and Dredging Engineering
Technische Universiteit Delft
Thesis: Exploratory research on application multi-level multi-fidelity Monte Carlo in fluid dynamics topics: study on flow past a porous cylinder
Supervisors: Dr. J.O. Colomes Gene
Prof. dr. A.C. Viré
Prof. dr. A.V. Metrikine
Ir. M. Vergassola

2021-2025 PhD. from Offshore Engineering Research Group
Technische Universiteit Delft
Thesis: Unfitted Finite Element Methods for Floating Structure Hydrodynamics
Promotor: Prof. dr. A.V. Metrikine
Copromotor: Dr. J.O. Colomes Gene

Conference Contributions

2025 **The Thirty-fifth International Ocean and Polar Engineering Conference**
Presentation and Conference Paper
Estimation of Hydrodynamic Coefficients of Floating Offshore Structures using the Aggregated Unfitted Finite Element Method

2024 **16th World Congress on Computational Mechanics**
Presentation
Assessing Modeling of Wave-Structure Interactions with Unfitted Finite Element Methods

2023 **10th International Congress on Industrial and Applied Mathematics**
Presentation
The Aggregated Finite Element Method (AgFEM) for Fluid-Structure Interaction problems

2023 **XII International Conference on Structural Dynamics**
Presentation
The Potential of Monolithic Fluid-Structure Interaction for Floating Objects using Unfitted Finite Element Methods

2023 **22nd Computational Fluids Conference**
Presentation
Monolithic Fluid-Structure Interaction for Floating Objects using AgFEM

List of Publications

Journal publications

1. **J. Modderman**, J.O. Colomes, "Application of Unfitted Finite Element methods for estimating added mass and added damping in floating structures," *Advances in Computational Science and Engineering*, vol 4, p. 142 167, 2025

2. J.O. Colomes, **J. Modderman**, G. Scovazzi, "The Generalized Shifted Boundary Method for Geometry-Parametric PDEs and Time-Dependent Domains," *Computer Methods in Applied Mechanics and Engineering,*, vol 452, p. 118748, 2026

Conference publications

1. **J. Modderman**, J.O. Colomes, "Estimation of Hydrodynamic Coefficients of Floating Offshore Structures Using the Aggregated Unfitted Finite Element Method," *ISOPE International Ocean and Polar Engineering Conference*, 2025

Included in this thesis.

List of Open Source Contributions

As part of this thesis, the following contributions to open-source software libraries have been made. This list consists of issues I have been involved in.

Gridap.jl

Gridap is a collection of tools for grid-based numerical approximation of partial differential equations (PDEs) implemented in the Julia programming language. The library currently handles both linear and nonlinear PDE systems involving scalar and vector fields, supports single- and multi-field formulations, and accommodates conforming as well as nonconforming finite element discretizations on structured and unstructured meshes composed of simplices and n-cubes. In addition, it includes techniques for time integration. Gridap is designed to be modular and easily extensible [184]. Contributions to the sub-packages of `Gridap.jl` have also been made. Briefly describing each package, embedded has the tools to allow for unfitted finite element methods, specifically AgFEM and CutFEM, but also other methods. Distributed allows for parallel computational domain decomposition. Gmsh couples the GMSH [168] software library to `Gridap.jl`. The `STLCutters.jl` package allows for the use of explicit geometrical representations for `GridapEmbedded.jl`. Pull requests are:

869 `Gridap.jl`: Propose fix for bug in multifield automatic differentiation

870 `Gridap.jl`: Reproducer and test function for issue 869

872 `Gridap.jl`: Export and minor bug fixes for ConstantFESpace

986 `Gridap.jl`: ode_start() in Generalized_Alpha results into singular system

1062 `Gridap.jl`: Add get_dof_value_type for FESpaceWithLinearConstraints

105 `GridapEmbedded.jl`: Distributed BoundaryTriangulation and typo on compute_redistribute_weights

111 `GridapEmbedded.jl`: Implement AgFEM on Octree Distributed Meshes

153 `GridapDistributed.jl`: Distributed ConstantFESpace

82 `GridapGmsh.jl` added keyword for positive=true from simplexify() for GMSH mesh, required for coupling with STLCutters.jl

27 `STLCutters.jl` 27: Adding cut_facets feature