UPPSALA
UNIVERSITET

IT Licentiate theses
2016-008

Towards Higher Order Immersed Finite Elements for the Wave Equation

SIMON STICKO

UPPSALA UNIVERSITY
Department of Information Technology

UPPSALA
UNIVERSITET

Towards Higher Order Immersed Finite Elements for the Wave Equation

Simon Sticko
simon.sticko@it.uu.se

September 2016

Division of Scientific Computing
Department of Information Technology
Uppsala University
Box 337
SE-751 05 Uppsala
Sweden

http://www.it.uu.se/

Dissertation for the degree of Licentiate of Technology in Scientific Computing with specialization in Numerical Analysis

ISSN 1404-5117
Printed by the Department of Information Technology, Uppsala University, Sweden

Abstract

We consider solving the scalar wave equation using immersed finite elements. Such a method might be useful, for instance, in scattering problems when the geometry of the domain is not known a priori. For hyperbolic problems, the amount of computational work per dispersion error is generally lower when using higher order methods. This serves as motivation for considering a higher order immersed method.

One problem in immersed methods is how to enforce boundary conditions. In the present work, boundary conditions are enforced weakly using Nitsche's method. This leads to a symmetric weak formulation, which is essential when solving the wave equation. Since the discrete system consists of symmetric matrices, having real eigenvalues, this ensures stability of the semi-discrete problem.

In immersed methods, small intersections between the immersed domain and the elements of the background mesh make the system ill-conditioned. This ill-conditioning becomes increasingly worse when using higher order elements. Here, we consider resolving this issue using additional stabilization terms. These terms consist of jumps in higher order derivatives acting on the internal faces of the elements intersected by the boundary.

Acknowledgments

I would like to thank my advisor Gunilla Kreiss for introducing me to an interesting research area and for her advice, encouragement and enthusiasm. Furthermore, I would like to thank all friends and colleagues at TDB for providing a nice and open working environment.

I am grateful to the Swedish Research Council for providing funding for my salary, and to Sederholms Nordiska, Stiftelsen ÅForsk and Liljewalchs for providing funding for my travels.

"Det får väl bli som det blir, så blir det ju i alla fall."
Min Far. Råd angående deadlines.

List of Papers

This thesis is based on the following papers

I S. Sticko and G. Kreiss. "A stabilized Nitsche cut element method for the wave equation". In: *Computer Methods in Applied Mechanics and Engineering* 309 (Sept. 2016), pp. 364–387. ISSN: 00457825. DOI: 10.1016/j.cma.2016.06.001

II S. Sticko and G. Kreiss. "Higher Order Cut Elements for the Wave Equation". arXiv:1608.03107 [math.NA]. Aug. 2016

Reprints were made with permission from the publisher.

Contents

1 Introduction and Motivation **3**

2 Basic Approach and Boundary Conditions **7**
2.1 Nitsche's Method 9
2.2 Lagrange Multipliers 10
2.3 Strong Enforcement 11

3 Ill-conditioning and Degrees of Freedom **13**
3.1 Preconditioning 14
3.2 Modifying the Finite Element Space 15
3.3 Penalizing the Weak Formulation 16
3.4 Perspectives on FEM-based Immersed Methods 18

4 Implementational Aspects **21**
4.1 Representing Immersed Geometries 21
4.2 Preprocessing and Quadrature 23

Chapter 1

Introduction and Motivation

Waves occur in a variety of forms in nature. As we now, sound is pressure waves that propagate through the air, an earthquake consists of waves that propagate through the ground, and we send information all around the world as electromagnetic waves. Mathematically, these waves can be described by partial differential equations (PDEs), and in a variety of applications it is necessary to solve these in order to predict how the waves propagate. In general, these equations are too difficult to be solved exactly using only pen and paper. The strategy is then to resort to solving them approximately with computers, also known as solving them "numerically". This is the topic of the present thesis. In particular, we are interested in solving these equations using so-called immersed methods.

The setting here is that we have a partial differential equation, posed on some domain, Ω_1, with boundary, $\Gamma = \partial\Omega_1$, such as the one in Figure 1.1. Most numerical methods for solving partial differential equations requires a triangulation, also know as a mesh, in order to solve the problem. The classical approach is to construct a triangulation which conforms to this geometry, such as in Figure 1.2. In an immersed method one instead lets the domain float on top of a background mesh, as illustrated in Figure 1.3. This makes solving the problem more complicated but is sometimes justified. In several applications, the domain under consideration moves during the simulation, for example, if the position and shape of the domain depend on time. Moving the nodes of the triangulation can make the triangles very distorted, such as illustrated in Figure 1.4. The distortion will degrade the performance of the method. This problem can be solved by creating a new triangulation, having the same shape as the deformed geometry, but having triangles with better shapes. This is known as remeshing and is a very time-consuming operation. One reason for considering to use an immersed method is that if the domain is floating on top of a background mesh, the

mesh can never become distorted. Thus, applications where remeshing is best avoided is the main motivation for immersed methods. A second reason is that in many applications it can be very costly to construct a mesh which conforms to the domain. Using an immersed method could in some cases be less time-consuming.

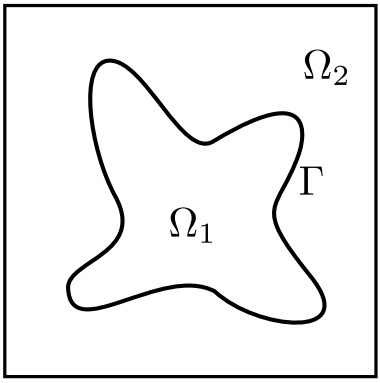

Figure 1.1: A given domain to solve a PDE on.

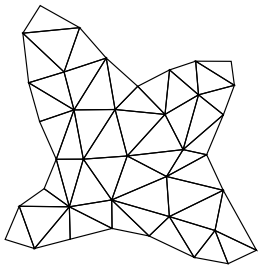

Figure 1.2: A computational mesh conforming to the geometry.

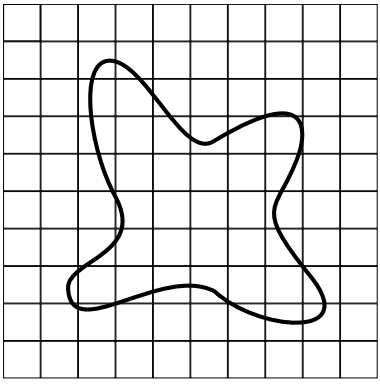

Figure 1.3: A domain immersed in a background mesh.

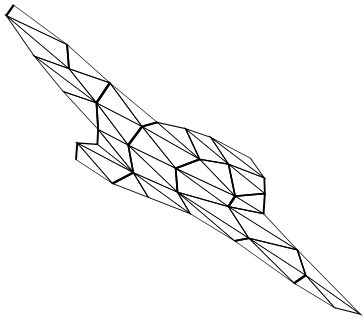

Figure 1.4: A distrorted mesh with triangles of poor quality.

It might seem odd to consider an immersed method in the context of wave propagation. After all, applications involving wave propagation and time-dependent geometries are rather sparse. The main motivation instead comes from scattering problems where some geometry of interest is not known a priori. Consider, for example, the geophysical application illustrated in Fig-

ure 1.5. It is known that deep down in the ground, there exists an interface, Γ, between two different materials. The shape of this interface is unknown, but for some engineering purpose needed. The idea is then that one sends out waves from some point, A, that propagate towards the interface, get reflected from it and finally detected at some other point, B. Given measured data, one would then want to solve an optimization problem to reconstruct the interface geometry. When solving this optimization problem we would need to iterate, and solve the problem repeatedly for a number of different interface geometries. As one iterates over different geometries, remeshing is going to be very expensive. Hence, one might gain a lot by using an immersed method.

In order to solve this optimization problem, it is necessary to be able to first efficiently solve the problem for a single geometry, which is the research direction of the present thesis. In general, in hyperbolic problems work per dispersion error increases slower for higher order methods [3]. This means that for wave propagation problems we gain by using higher order methods. This serves as motivation for investigating the possibility of a higher order immersed method.

Historically, the first immersed method is usually attributed to Peskin [4, 5]. Nowadays, immersed methods have been developed with several different classical numerical methods as base. Since Paper I and II are based on the finite element method, we shall in the following chapters focus on discussing this case.

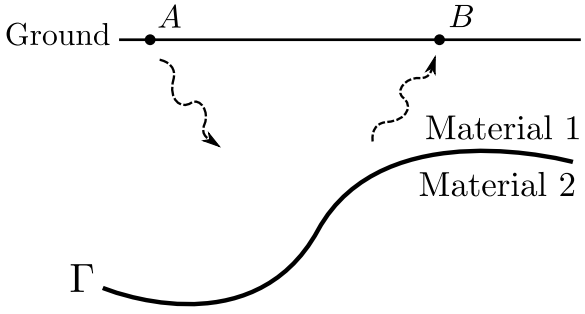

Figure 1.5: An example of a problem where the shape of the material interface, Γ, is unknown.

Chapter 2

Basic Approach and Boundary Conditions

Assume that we want to solve a PDE on the domain Ω_1, shown in Figure 1.1. Perhaps the most common approach in immersed methods based on finite elements is to solve for the degrees of freedom of the elements illustrated in Figure 2.1. That is, the degrees of freedom of the smallest set of elements that cover Ω_1. This is the approach used in Paper I and II.

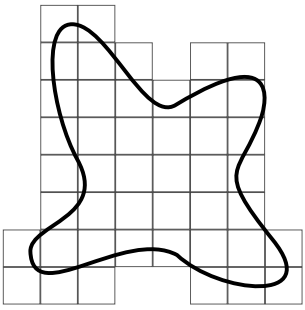

Figure 2.1: Smallest set of elements covering the domain Ω_1.

A finite element method always starts from a weak formulation, consisting of various quantities integrated over the domain, Ω_1, and its boundary, Γ. This integration is typically done element by element. Thus, the basic component of most immersed finite element methods is the following. On the elements intersected by the boundary we are required to intergrate over just a part of the element, as illustrated in Figure 2.2. That is, perform quadrature over the part of Ω_1, and the part of Γ, that falls within the element.

(a) $\int_{K\cap\Omega_1}$ (b) $\int_{K\cap\Gamma}$

Figure 2.2: Quadrature over different parts of an element.

There are a few different approaches for applying boundary conditions using immersed finite elements. In order to have a model problem to discuss, consider the Poisson equation

$$-\nabla^2 u = f, \quad x \in \Omega_1, \tag{2.1}$$

$$u = g_D(x), \quad x \in \Gamma_D, \tag{2.2}$$

$$\frac{\partial u}{\partial n} = g_N(x), \quad x \in \Gamma_N, \tag{2.3}$$

with $\Gamma_D \cup \Gamma_N = \Gamma$ and $\Gamma_D \neq \emptyset$. In the following, let $(\cdot,\cdot)_X$ denote the L_2-scalar product taken over $X \subset \mathbb{R}^d$. Let also, $\langle\cdot,\cdot\rangle_Y$ denote the L_2-scalar product taken over Y, where $Y \subset \mathbb{R}^{d-1}$. Furthermore, let V_h denote a finite dimensional subspace of $H^1(\Omega_1)$. Different choices for constructing V_h will be discussed in Chapter 3.

Multiplying (2.1) with a test-function, $v \in H^1(\Omega_1)$, and integrating (2.1) by parts leads to

$$(\nabla u, \nabla v)_{\Omega_1} - \left\langle \frac{\partial u}{\partial n}, v \right\rangle_{\Gamma_D\cup\Gamma_N} = (f, v)_{\Omega_1}. \tag{2.4}$$

Now the Neumann boundary condition (2.3) is consistent with

$$\left\langle \frac{\partial u}{\partial n}, v \right\rangle_{\Gamma_N} = \langle g_N, v\rangle_{\Gamma_N},$$

so adding this leads to

$$(\nabla u, \nabla v)_{\Omega_1} - \left\langle \frac{\partial u}{\partial n}, v \right\rangle_{\Gamma_D} = (f, v)_{\Omega_1} + \langle g_N, v\rangle_{\Gamma_N}. \tag{2.5}$$

Note that by adding this term the Neumann boundary condition is now enforced weakly by the new term appearing in the right-hand side. In two

dimensions this term is a line integral that cuts through the elements as in Figure 2.2b. Equation (2.5) is the starting point for a number of approaches for enforcing the Dirichlet boundary condition. In the following we discuss a few different ones.

2.1 Nitsche's Method

A popular way of enforcing boundary conditions in immersed finite elements is to use Nitsche's method [6]. Here, one enforces the Dirichlet boundary condition weakly, in the same way as with the Neumann condition in (2.5). Note that the Dirichlet boundary condition is consistent with the two equations

$$\langle u, v \rangle_{\Gamma_D} = \langle g_D, v \rangle_{\Gamma_D}, \tag{2.6}$$

$$\left\langle u, \frac{\partial v}{\partial n} \right\rangle_{\Gamma_D} = \left\langle g_D, \frac{\partial v}{\partial n} \right\rangle_{\Gamma_D}, \tag{2.7}$$

which are also line-integrals in 2D. Let h denote some measure of the grid-size. Adding (2.6) multiplied with a constant, and adding/subtracting (2.7) from (2.5) gives the following weak formulation: find $u_h \in V_h$ such that

$$a^{\pm}(u_h, v_h) = L^{\pm}(v_h), \quad \forall v_h \in V_h, \tag{2.8}$$

where

$$a^{\pm}(u_h, v_h) = (\nabla u_h, \nabla v_h)_{\Omega_1} - \left\langle \frac{\partial u_h}{\partial n}, v_h \right\rangle_{\Gamma_D} \pm \left\langle u_h, \frac{\partial v_h}{\partial n} \right\rangle_{\Gamma_D} + \frac{\alpha}{h} \langle u_h, v_h \rangle_{\Gamma_D}, \tag{2.9}$$

$$L^{\pm}(v_h) = (f, v_h)_{\Omega_1} + \langle g_N, v_h \rangle_{\Gamma_N} + \left\langle g_D, \frac{\alpha}{h} v_h \pm \frac{\partial v_h}{\partial n} \right\rangle_{\Gamma_D}.$$

We are here free to choose the formulation with either sign. Both versions have advantages. In order for a formulation to be well posed we need the bilinear form $a^{\pm}$ to be coercive,[1] or fulfill the so-called inf-sup condition. Coercivity states that there needs to exist a constant $C > 0$ such that

$$C\|v_h\|_h^2 \leq a^{\pm}(v_h, v_h), \quad \forall v_h \in V_h, \tag{2.10}$$

in some discrete norm $\|\cdot\|_h$. Inf-sup stability can be seen as a slightly weaker condition. For the non-immersed case it turns out that the bilinear form $a^{-}(\cdot,\cdot)$ is coercive [8] for any $\alpha > 0$ and inf-sup stable [9] even for $\alpha = 0$.

[1] This is a requirement in the Lax-Milgram lemma, see e.g. [7] page 83.

Thus, choosing the bilinear form a^- results in a parameter-free method. The disadvantage of the bilinear form a^- is that it is not symmetric:

$$\exists u_h, v_h \in V_h \quad \text{s.t.} \quad a^-(u_h, v_h) \neq a^-(v_h, u_h).$$

That the bilinear form discretizes to a symmetric stiffness matrix is a quite attractive property. This is an advantage of using the symmetric bilinear form a^+ and the main reason why it is used in Paper I and II. However, as shown in the original paper by Nitsche [6] a^+ is only coercive given that one chooses the parameter α to be sufficiently large: $\alpha > C_\alpha$. This is a disadvantage since the smallest constant C_α is usually not known. In practice, α is usually chosen by trial and error. Recently there has been some work [10] on how to avoid the free parameter and still maintain the symmetry in the weak formulation.

In a similar approach as the one above, Nitsche's method can be used to enforce interface conditions. See for example [11, 12].

2.2 Lagrange Multipliers

Lagrange multipliers is a general technique for finding the minimum or maximum of a function subject to a constraint. For example, say that we would like to find a minimum (x, y) of a function $j(x, y) : \mathbb{R}^2 \to \mathbb{R}$ along a curve $c(x, y) = 0$. In order to solve this we introduce an additional scalar, λ, (the Lagrange multiplier) and instead find a stationary point $(x, y, \lambda) \in \mathbb{R}^3$ to the function

$$F(x, y, \lambda) = j(x, y) + \lambda c(x, y).$$

The same technique can be used to apply boundary conditions for an immersed problem. This is done by rewriting (2.1) as an optimization problem. If we look for the solution u in the space

$$H^1_{g_D}(\Omega_1) = \{v \in H^1(\Omega_1) : v|_{\Gamma_D} = g_D\}, \tag{2.11}$$

(2.5) reduces to: find $u \in H^1_{g_D}(\Omega_1)$ such that

$$(\nabla u, \nabla v)_{\Omega_1} = (f, v)_{\Omega_1} + \langle g_N, v \rangle_{\Gamma_N}, \quad \forall v \in H^1_0(\Omega_1).$$

This is now equivalent to the minimization problem[2]: find $u \in H^1_{g_D}(\Omega_1)$ such that

$$J(u) = \min_{v \in H^1_{g_D}(\Omega_1)} J(v),$$

$$J(v) := \frac{1}{2}\|v\|^2_{\Omega_1} - (f, v)_{\Omega_1} - \langle g_N, v \rangle_{\Gamma_N}.$$

[2]See for example [13] page 189.

However, if we introduce a multiplier, $\lambda(x)$, for each point, x, on Γ_D, we can enforce the boundary condition as a constraint:

$$C(u, \lambda) = \int_{\Gamma_D} \lambda(x)(u - g_D)d\Gamma = 0.$$

This enables us to look for the solution, u, in the space $H^1(\Omega_1)$ instead of $H^1_{g_D}(\Omega_1)$ if we at the same time solve for the multiplier $\lambda(x) \in H^{-\frac{1}{2}}(\Gamma_D)$. Thus, we end up with the problem of finding a stationary point (u, λ) to the functional

$$F(v, \xi) = J(v, \xi) + C(v, \xi), \tag{2.12}$$

with $(v, \xi) \in H^1(\Omega_1) \oplus H^{-\frac{1}{2}}(\Gamma_D)$. The problem (2.12) can finally be converted to a weak formulation and solved using finite elements.

There are some disadvantages with using this method. First of all, the multiplier $\lambda(x)$ needs be a member of a finite element space covering Γ_D. This is illustrated in Figure 2.3, assuming that $\Gamma_D = \Gamma$. This will increase the degrees of freedom of the problem. Secondly, the technique does not have an obvious generalization to time-dependent problems. Finally, as stated in [14] the formulation does not have very good properties. The resulting system is a saddle point problem. In order to guarantee inf-sup stability, this requires a careful choice of the combination of subspaces, $V^h_U \oplus V^h_\Lambda$, where we look for the discrete solution, $(u_h, \lambda_h) \in V^h_U \oplus V^h_\Lambda$.

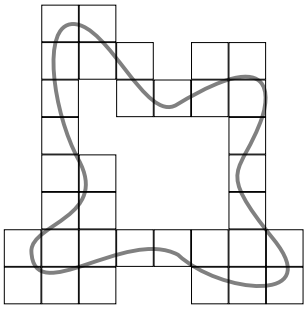

Figure 2.3: Illustration of the finite element space for the Lagrange multiplier.

2.3 Strong Enforcement

Another approach for imposing the boundary condition is to do it strongly. Here, we search for the solution in a space fulfilling the Dirichlet boundary

condition. That is, instead of trying to find the solution in the space $V_h \subset H^1(\Omega)$, we search for it in the space

$$V_h^{g_D} = \{v \in V_h : v|_{\Gamma_D} = g_D\}.$$

The construction of this space is not trivial in an immersed setting. As was discussed in [15] this may lead to too many constraints imposed on V_h close to the boundary, which in turn results in non-optimal convergence rates. In [15], this problem was solved by exchanging the elements intersected by the boundary to discontinuous Galerkin elements, thus introducing more degrees of freedom near the boundary. This resulted in second order convergence in L_2-norm, with piecewise linear basis functions.

Chapter 3

Ill-conditioning and Degrees of Freedom

Even if the choice in Figure 2.1 is perhaps the most common way to distribute degrees of freedom, there is one problem with this approach. Consider the case illustrated in Figure 3.1. Here, the element T has a very small intersection with the domain. This turns out to make the resulting matrices in the system of linear equations very ill-conditioned. Since the domain floats on top of the background mesh, the intersection between an element and the domain can be arbitrarily small. Because of this, the smallest eigenvalues of matrices in the system can not be bounded from below, and thus the matrices can be arbitrarily ill-conditioned.

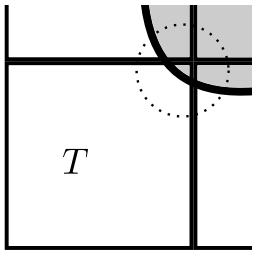

Figure 3.1: An element having a small intersection with the immersed geometry, Ω_1.

The ill-conditioning is perhaps easiest to understand for the case of symmetric positive definite matrices, which are common for many problems discretized with finite elements. Consider for example the mass-matrix, $\mathcal{M} \in \mathbb{R}^N$,

$$\mathcal{M}_{ij} = (\phi_i, \phi_j)_{\Omega_1}.$$

The smallest eigenvalue, $\lambda_{\min}$, can be expressed in terms of the minimum of

the Rayleigh quotient,

$$\lambda_{\min} = \min_{z \in \mathbb{R}^N : z \neq 0} \frac{z^T \mathcal{M} z}{z^T z}. \tag{3.1}$$

By fixing $j \in \{1, \ldots, N\}$ and choosing z as $z_i = \delta_{ij}$ we obtain that each eigenvalue is smaller than each diagonal entry:

$$\lambda_{\min} \leq \mathcal{M}_{jj}, \quad j = 1, \ldots, N. \tag{3.2}$$

Since the cut with the background mesh can be arbitrarily small, some diagonal entries can also be arbitrarily small. Thus, the eigenvalues can be arbitrarily close to zero, which will lead to stability problems unless special measures are taken.

The property that we want an immersed method to have, is that the stability and accuracy of the method should be independent of how the intersection with the mesh occurs. There are several ways of dealing with ill-conditioning. Here, we briefly describe the main approaches for trying to overcome this problem.

3.1 Preconditioning

A fairly common way to try to remedy problems with ill-conditioning is by preconditioning. Advanced preconditioners have been designed, as in for example [16]. Reusken [17] proved for an interface problem that simply scaling the mass matrix, $\mathcal{M}$, by its diagonal, $\mathcal{D}_{\mathcal{M}}$, makes the condition number of the mass matrix independent of the position of the cut and the grid size. In particular, it was shown that there exist constants, c_1 and c_2, independent of how the interface cuts the mesh, such that

$$c_1 z^T \mathcal{D}_{\mathcal{M}} z \leq z^T \mathcal{M} z \leq c_2 z^T \mathcal{D}_{\mathcal{M}} z, \quad \forall z \in \mathbb{R}^N.$$

Diagonal scaling was also used in [18] together with implicit time-stepping for time-dependent problems. One issue when using explicit time-stepping in immersed methods is that one can get a time-step restriction which is not independent of the position of the cut. That is, the time-step τ must be chosen proportional to the smallest cut in the mesh, h_{cut},

$$\tau \leq c h_{\text{cut}},$$

and not proportional to the grid-size, h. One disadvantage with only using diagonal scaling is that it does not overcome this problem. This restricts the use of diagonal scaling to implicit time-stepping, but depending on the problem under consideration this may be acceptable.

3.2 Modifying the Finite Element Space

One strategy to resolve problems with small cuts is to remove or in some way modify basis functions with small support. This can be done in several different ways. Consider, for example, the one dimensional case in Figure 3.2. Here, the domain is $\Omega_1 = [\Gamma_L, \Gamma_R]$, where the two boundary points Γ_L and Γ_R are located in between the grid points. Figure 3.2a corresponds to Figure 2.1.

Perhaps the simplest approach to deal with the ill-conditioning would be to solve for only those degrees of freedom that lie inside the domain, as illustrated in Figure 3.2b. Here, each basis function has a large portion of its support inside the domain. The problem with removing basis functions in this way is that it will have a large effect on the error close to the boundary, which will degrade the convergence of the method. In general, the main difficulty when modifying the basis is how to do it without destroying the approximation properties of the finite element space.

A third option is to include those basis functions which have sufficiently large support. In Figure 3.2c the left-most grid point, x_0, falls close to the left boundary, Γ_L, and its corresponding basis function is included. But the right-most grid point, x_{N+1}, falls far from Γ_R, and its basis function is excluded. This approach has been used in several papers. In for example [17] this strategy was used for an interface problem. Here, basis functions were removed if they fulfilled

$$\frac{\|\phi\|_{H^l(T\cap\Omega)}}{\|\phi\|_{H^l(T)}} \leq c h_T^{\beta},$$

where c, β and l are some user defined parameters.

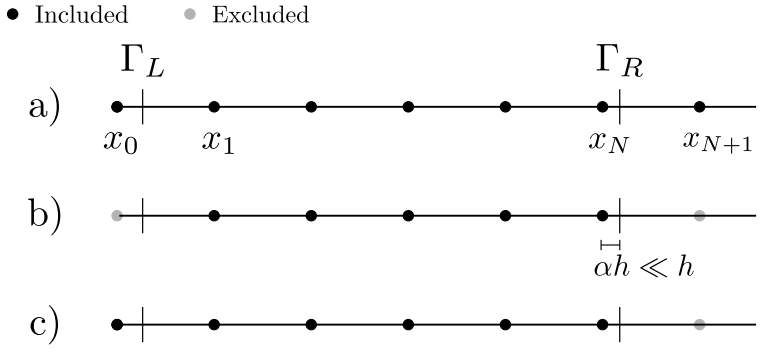

Figure 3.2: Three different options for distributing degrees of freedom.

There are also other approaches for modifying the basis. In [19] the strategy illustrated in Figure 3.3 was used. Those elements which have a

very small intersection with the domain are extended with an element with a large intersection to create a larger element.

One implementational issue with the concept of modifying the finite element space is that we, prior to assembling, need to have some measure of the support of each basis function. This knowledge is needed so that matrices and vectors of appropriate sizes can be allocated.

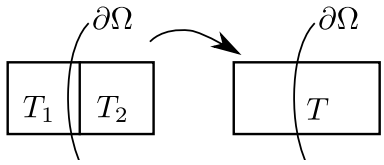

Figure 3.3: An element with a small intersection, T_1, extended with an element with a large intersection, T_2 to form a larger element T.

3.3 Penalizing the Weak Formulation

Another approach to overcome problems with ill-conditioning is to add stabilizing penalty terms to the weak formulation. Say that we would like to L_2-project a given function u to the domain, Ω_1, in Figure 1.1. This problem reads: find $u_h \in V_h$ such that

$$(u_h, v_h)_{\Omega_1} = (u, v_h)_{\Omega_1}, \quad \forall v_h \in V_h,$$

which will result in an ill-conditioned mass matrix $m_{ij} = (\phi_i, \phi_j)_{\Omega_1}$. The idea is now to add a penalty term, j, so that the problem now reads: find $u_h \in V_h$ such that

$$M(u_h, v_h) = (u, v_h)_{\Omega_1}, \quad \forall v_h \in V_h.$$

Here, M is a bilinear form corresponding to the scalar product, but with the added penalty

$$M(u_h, v_h) = (u_h, v_h)_{\Omega_1} + \gamma_M j(u_h, v_h),$$

where $\gamma_M > 0$ is a penalty parameter. The idea is now that with this added penalty one should be able to show that the bilinear form $\mathcal{M}$ is norm-equivalent to the L_2-norm on the background mesh. That is, there exist constants, $C_1, C_2 > 0$, that are independent of how the boundary cuts the mesh such that

$$C_1 \|v_h\|^2_{\Omega_\mathcal{T}} \leq M(v_h, v_h) \leq C_2 \|v_h\|^2_{\Omega_\mathcal{T}}, \quad \forall v_h \in V_h. \tag{3.3}$$

Given this property, the condition number, κ, of the mass matrix corresponding to the bilinear form M will be bounded by the condition number of

the mass matrix over the whole background mesh

$$\kappa(\mathcal{M}_s) \leq \frac{C_2}{C_1}\kappa(\mathcal{M}_\mathcal{T}). \tag{3.4}$$

Here, the matrices are defines as

$$(\mathcal{M}_s)_{ij} = M(\phi_i, \phi_j),$$
$$(\mathcal{M}_\mathcal{T})_{ij} = (\phi_i, \phi_j)_{\Omega_\mathcal{T}},$$

where $\Omega_\mathcal{T}$ denotes the domain that is covered by all elements in the background mesh:

$$\Omega_\mathcal{T} = \bigcup_{T\in\mathcal{T}} T.$$

Note that $\kappa(\mathcal{M})$ is now bounded independently of how the boundary cuts the mesh, since the right hand side in (3.4) does not depend on the cut.

One possible penalty term that can be used for piecewise linear elements is the following

$$j(u_h, v_h) = \sum_{F\in\mathcal{F}_\Gamma} h^3 \left\langle \left[\frac{\partial u_h}{\partial n}\right], \left[\frac{\partial v_h}{\partial n}\right] \right\rangle_F. \tag{3.5}$$

Here, $[v]$ denotes the jump over an element face, F,

$$[v] = v|_{F+} - v|_{F-},$$

and n is the unit normal vector to the face. $\mathcal{F}_\Gamma$ denotes the internal faces of the elements intersected by the boundary, as illustrated in Figure 3.4. In this way, the penalty acts in the zone where the problems with the small cuts occur. The penalty term (3.5) was suggested in [20] for solving the Poisson equation using piecewise linear elements and was proved to give second order convergence. It has also been used for a number of other problems, as in for example [21, 22, 23] and in Paper I. A generalization of this stabilization to higher order elements was suggested in [24, 25], where also jumps in higher order derivatives were included. This stabilization is considered in Paper II for quadratic and cubic elements.

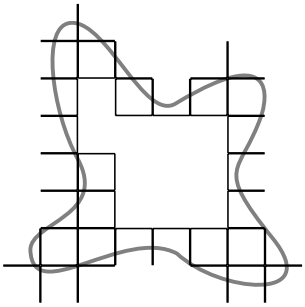

Figure 3.4: Faces, $\mathcal{F}_\Gamma$, that the penalty term act on.

3.4 Perspectives on FEM-based Immersed Methods

Immersed methods based on finite elements come under a variety of names, such as GFEM [26], XFEM [27], CutFEM [28], Finite Cell [29], and PUM [30]. Not all of these methods are specifically immersed methods, but lead to an immersed method by some choice. As pointed out in [27], many of these methods are very similar. Their differences appear to mainly be from what perspective you look at the problem from. Still, what appears to be the most natural thing to do might not be the same in each perspective. For example, distributing degrees of freedom.

Consider XFEM, which is perhaps the most commonly used variant. It appears to be especially popular in the mechanics community. Here, the point of view is that one modifies the finite element basis. The idea in XFEM is to introduce some extra "enriched" degrees of freedom, a_i, into our solution ansatz, which will now look like:

$$u_h = \sum_{i \in I_O} u_i \phi_i(x) + \sum_{i \in I_E} a_i E(x) \phi_i(x). \tag{3.6}$$

Here, $E(x)$ is the so-called enrichment function. This is chosen depending on the application, but in immersed methods it is common to choose the enrichment function to be the Heaviside function. In order to have an example to discuss, assume that we have an interface problem, as illustrated in Figure 3.5. Here, the solution has a discontinuity at the point Γ.

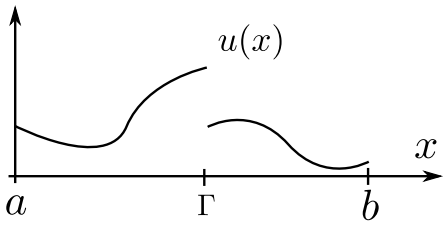

Figure 3.5: A 1D-function with a discontinuity at the point Γ.

If we use the standard piecewise linear basis functions over the whole interval $[a, b]$ (as illustrated in Figure 3.6a), the convergence will be suboptimal due to the discontinuity. One way to try to capture this discontinuity is to keep the standard space from Figure 3.6a and add the basis functions illustrated in Figure 3.6b. Here, the basis functions of the element intersected by Γ have been enriched by the Heaviside function. This creates the following finite element space

$$V_h = \text{span}\{\phi_1, \phi_2, \ldots, \phi_N, H(x-\Gamma)\phi_i(x), H(x-\Gamma)\phi_{i+1}(x)\}.$$

The choice of space in Figure 3.6 is perhaps the most natural in the XFEM-framework and was discussed in [27]. Another option, which is more natural from the CutFEM point of view, is to put each piecewise continuous part of the solution in its own finite element space. As illustrated in Figure 3.7, we could consider each restriction

$$u^L = u|_{x<\Gamma}\,,$$
$$u^R = u|_{\Gamma<x}\,,$$

.

separately, and let each be approximated in its own subspace

$$u_h^L \in V_h^L,$$
$$u_h^R \in V_h^R.$$

Here, V_h^L and V_h^R are the spaces of piecewise linear functions over all elements that have at least some part inside the intervals $[a, \Gamma]$ and $[\Gamma, b]$. This has been used in several papers, for example in [11, 12].

As mentioned in [31], the approach in Figure 3.6 might be preferable in two or three dimensions in problems where the interface does not split the domain in two but ends inside it. This kind of problem appears for example in fracture mechanics, when the interface describes a crack in a solid. In other cases it might be easier to implement the approach in Figure 3.7 and to solve issues with ill-conditioning.

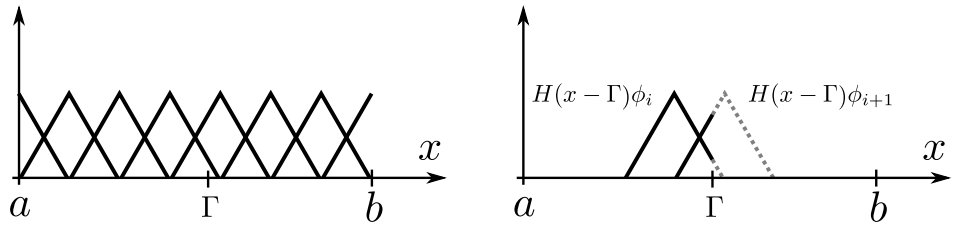

Figure 3.6: (a) Standard piecewise linear finite element space. (b) Added enrichment functions.

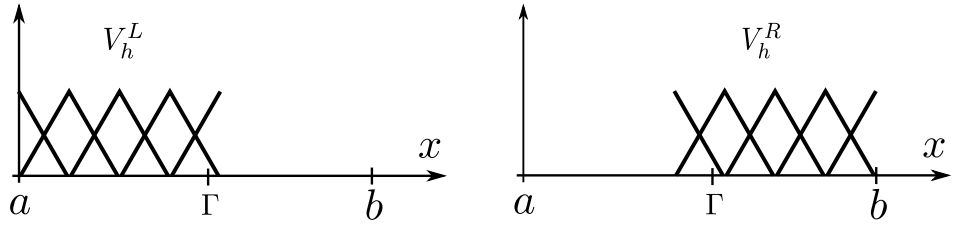

Figure 3.7: Finite element spaces used to capture the left and right part of the solution $u(x)$.

Chapter 4

Implementational Aspects

4.1 Representing Immersed Geometries

Since the geometry of the domain is no longer represented by the mesh, we need some other way to represent it. One approach is to use an additional mesh just for this purpose, as depicted in Figure 4.1. This is usually referred to as overlapping meshes. This can be advantageous if the immersed domain is not deforming, but is only moving by a translation or a rotation. Having a mesh for the immersed domain can also be advantageous in interface problems. With respect to Figure 1.1 this would mean coupling the solution in Ω_1 to the solution in Ω_2, such as in fluid-structure interaction.

Another way of describing the geometry is by a level set function.[1] This is perhaps the most common method and is also used in Paper II. As depicted in Figure 4.2, Γ is here described as the zero-contour of a scalar function, $\psi : \mathbb{R}^d \to \mathbb{R}$. The different sets in Figure 1.1 are identified in terms of the sign of ψ, as

$$\begin{aligned}\Omega_1 &= \{x \in \mathbb{R}^d : \psi(x) < 0\},\\ \Gamma &= \{x \in \mathbb{R}^d : \psi(x) = 0\},\\ \Omega_2 &= \{x \in \mathbb{R}^d : \psi(x) > 0\}.\end{aligned}$$

There are a few benefits of using a level set-function. First of all, the description of the domain is conceptually the same regardless of the spatial dimension. Moreover, normals, n, and curvatures, κ, can easily be computed along Γ, as functions of the gradient and Hessian of the level set function:

$$\begin{aligned}n &= \frac{\nabla\psi}{\|\nabla\psi\|},\\ \kappa &= \kappa(\nabla\psi, \nabla\nabla\psi).\end{aligned}$$

[1] For a comprehensive overview see [32] or [33].

In some applications, an important feature is also that topological changes can be handled quite easily. That is, it is possible to handle events such as when two fluid droplets merge into one, or two biological cells break apart.

There are, of course, disadvantages of using a level set function. The level set function is a member of a finite element space, typically over the whole background mesh. This requires a lot of degrees of freedom to represent the geometry. However, this can be made more efficient by the so-called narrow band level set method, where one only keeps a description of the geometry for the elements in a narrow region around the immersed interface. Furthermore, sharp features can not be represented. Even if the analytic level set function, ψ, has corners, the approximation of the level set function, ψ_h, will be smooth as soon as it is transferred to the finite element space: $\psi \approx \psi_h \in V_h$.

In relation to the level set methods, there are a number of methods, such as the volume of fluid and phase field method, which are conceptually very similar.

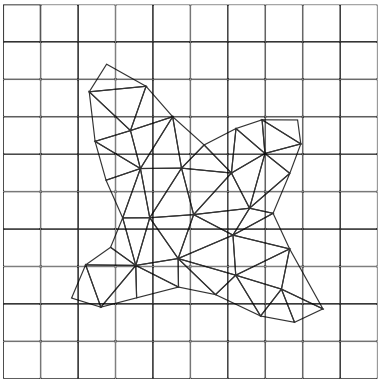

Figure 4.1: Two overlapping meshes.

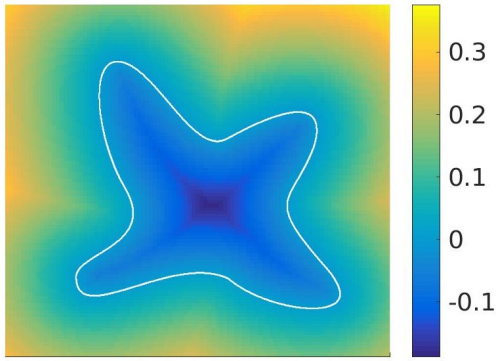

Figure 4.2: A level set function describing the geometry. Zero contour in white.

Another option for keeping track of the geometry is so-called marker points. This is illustrated in Figure 4.3. In 2D the approximation of the boundary, Γ_h, is represented as a sequence of ordered points, $\{x_i\}_{i=1}^N$. One problem with marker points is that the points can start to cluster as they are advected with time.

In many industrial applications, design of various components in CAD software is done using NURBS (Non-Uniform Rational B-Splines). A time-consuming problem is then to go from the CAD-geometry to a mesh that conforms to the geometry. This is the main motivation for the isogeometric analysis method [34]. It can thus be useful also in immersed methods to represent the immersed geometry as a CAD-surface.

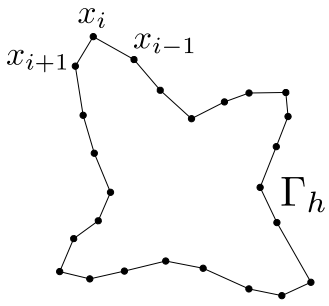

Figure 4.3: Illustration of marker points.

4.2 Preprocessing and Quadrature

Even if the aim of immersed methods is to avoid meshing the domain, there are still procedures that need to be done that relates the domain to the background mesh. Consider again Figure 1.1 and 1.3. As illustrated in Figure 4.4, we need to be able to determine which elements in the mesh lie completely in Ω_1, which are intersected by Γ, and which ones lie completely in Ω_2. This information is required in order to allocate matrices and vectors of appropriate sizes so that we in the next step can assemble the system.

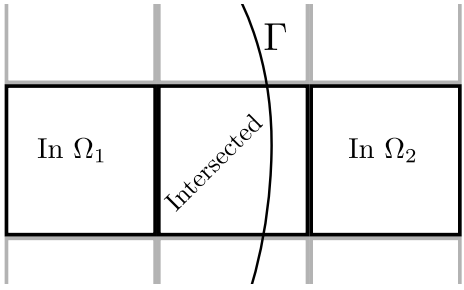

Figure 4.4: Elements divided into categories depending on their relation to the immersed geometry.

During the assembly process, we are in need of some algorithm to perform quadrature over the different parts of an element, as in Figure 2.2. It is not uncommon to actually create a sub-triangulation over a part of the element, as illustrated in Figure 4.5, merely in order to perform the quadrature. This might seem strange when the point of immersed methods is to avoid triangulating the domain. One argument in favor of sub-triangulation is that the triangulation does not need to be of high quality, since it is only

used to distribute quadrature points over $K \cap \Omega_1$ and $K \cap \Gamma$.

One difficulty with higher order immersed finite elements is that the order of the quadrature over the intersected elements needs to be sufficiently high. We could, for example, create a quadrature rule very simply by making a straight line approximation of the boundary, as in Figure 4.6. However, this will not be good enough in order to obtain high convergence order. The quadrature needs to take curved boundaries into account. Performing this higher order quadrature is one of the main difficulties when going to high order immersed methods.

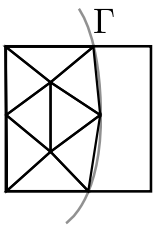

Figure 4.5: A sub-triangulation over a part of an element.

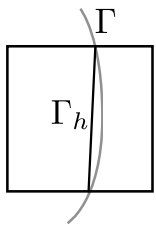

Figure 4.6: A straight line approximation, Γ_h, of Γ.

What algorithm is used in order to perform quadrature does, of course, depend on how the immersed geometry is represented. Recently there has been a number of articles published on how to generate quadrature rules, given that the domain is described by a level set function [35, 36].

In Paper II we use the algorithm from [36], which works on hyperrectangles (rectangles in 2D and bricks in 3D). This algorithm is illustrated in Figure 4.7 and is based on finding height functions. On the element K in Figure 4.7 we can express the x-coordinate on $\Gamma \cap K$ as a height function, $h(y)$, of the y-coordinate:

$$\Gamma \cap K = \{(y, h(y)) : y \in [y_1, y_2]\}.$$

Starting from the y-direction we do root finding along the left and right faces to find the points (here only y_1) where $\psi = 0$. We then place points in the y-direction. From these points we perform root finding on the level set function in the x-direction to find points on the surface Γ. These will be the quadrature points for the quadrature on $\Gamma \cap K$. Quadrature points for the bulk, $\Omega \cap K$, is then placed along each line in the x-direction according to the one-dimensional quadrature rule. Finding a height direction relies on the implicit function theorem, which states that if there is a point $\mathbf{x}_c$ such that

$$\frac{\partial \psi}{\partial x_k}(\mathbf{x}_c) \neq 0, \tag{4.1}$$

we can express x_k as a function of the remaining coordinates in a neighborhood around $\mathbf{x}_c$. The condition (4.1) is checked by calculating the gradient and the Hessian of ψ at the center of the cell and approximating $\frac{\partial \psi}{\partial x_k}$ as a linear function. The same algorithm can be used in three dimensions by recursion over the dimension, going from cell to faces to lines.

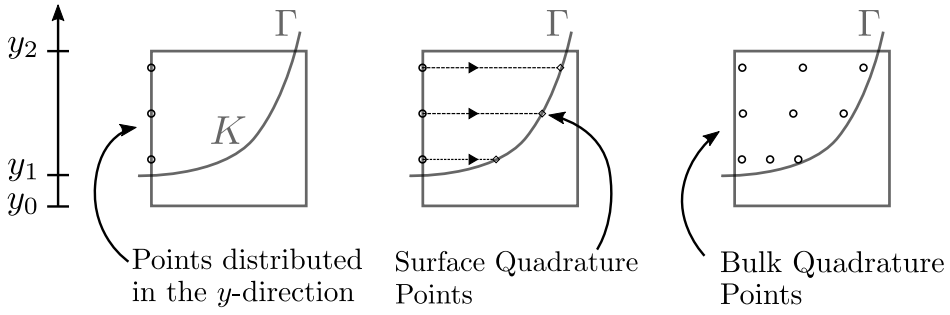

Figure 4.7: Illustration of the algorithm from [36].

Bibliography

[1] S. Sticko and G. Kreiss. "A stabilized Nitsche cut element method for the wave equation". In: *Computer Methods in Applied Mechanics and Engineering* 309 (Sept. 2016), pp. 364–387. ISSN: 00457825. DOI: 10.1016/j.cma.2016.06.001.

[2] S. Sticko and G. Kreiss. "Higher Order Cut Elements for the Wave Equation". arXiv:1608.03107 [math.NA]. Aug. 2016.

[3] H.-O. Kreiss and J. Oliger. "Comparison of accurate methods for the integration of hyperbolic equations". In: *Tellus* 24.3 (1972), pp. 199–215. DOI: 10.1111/j.2153-3490.1972.tb01547.x.

[4] C. S. Peskin. "Numerical analysis of blood flow in the heart". In: *Journal of Computational Physics* 25.3 (Nov. 1977), pp. 220–252. ISSN: 00219991. DOI: 10.1016/0021-9991(77)90100-0.

[5] C. S. Peskin. "The immersed boundary method". In: *Acta Numerica* 11 (Jan. 2002). ISSN: 0962-4929, 1474-0508. DOI: 10.1017/S0962492902000077.

[6] J. 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*. Vol. 36. Springer, 1971, pp. 9–15. DOI: 10.1007/BF02995904.

[7] A. Ern and J.-L. Guermond. *Theory and Practice of Finite Elements*. Ed. by S. S. Antman, J. E. Marsden and L. Sirovich. Vol. 159. Applied Mathematical Sciences. New York, NY: Springer New York, 2004. ISBN: 978-1-4419-1918-2 978-1-4757-4355-5.

[8] J. Freund and R. Stenberg. "On weakly imposed boundary conditions for second order problems". In: *Proceedings of the Ninth Int. Conf. Finite Elements in Fluids* (1995), pp. 327–336.

[9] E. Burman. "A Penalty-Free Nonsymmetric Nitsche-Type Method for the Weak Imposition of Boundary Conditions". In: *SIAM Journal on Numerical Analysis* 50.4 (Jan. 2012), pp. 1959–1981. ISSN: 0036-1429, 1095-7170. DOI: 10.1137/10081784X.

[10] C. Lehrenfeld. "Removing the stabilization parameter in fitted and unfitted symmetric Nitsche formulations". arXiv:1603.00617 [math.NA]. Mar. 2016.

[11] A. Hansbo and P. Hansbo. "An unfitted finite element method, based on Nitsche's method, for elliptic interface problems". In: *Computer methods in applied mechanics and engineering* 191.47 (2002), pp. 5537–5552. DOI: 10.1016/S0045-7825(02)00524-8.

[12] P. Hansbo, M. G. Larson and S. Zahedi. "A cut finite element method for a Stokes interface problem". In: *Applied Numerical Mathematics* 85 (Nov. 2014), pp. 90–114. ISSN: 01689274. DOI: 10.1016/j.apnum.2014.06.009.

[13] M. G. Larson and F. Bengzon. *The Finite Element Method: Theory, Implementation, and Applications*. Vol. 10. Texts in Computational Science and Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. ISBN: 978-3-642-33286-9 978-3-642-33287-6.

[14] S. Fernández-Méndez and A. Huerta. "Imposing essential boundary conditions in mesh-free methods". In: *Computer Methods in Applied Mechanics and Engineering* 193.12-14 (Mar. 2004), pp. 1257–1275. ISSN: 00457825. DOI: 10.1016/j.cma.2003.12.019.

[15] A. J. Lew and G. C. Buscaglia. "A discontinuous-Galerkin-based immersed boundary method". In: *International Journal for Numerical Methods in Engineering* 76.4 (Oct. 2008), pp. 427–454. ISSN: 00295981, 10970207. DOI: 10.1002/nme.2312.

[16] C. Lehrenfeld and A. Reusken. "Optimal preconditioners for Nitsche-XFEM discretizations of interface problems". In: *Numerische Mathematik* (Mar. 2016). ISSN: 0029-599X, 0945-3245. DOI: 10.1007/s00211-016-0801-6.

[17] A. Reusken. "Analysis of an extended pressure finite element space for two-phase incompressible flows". In: *Computing and Visualization in Science* 11.4-6 (Sept. 2008), pp. 293–305. ISSN: 1432-9360, 1433-0369. DOI: 10.1007/s00791-008-0099-8.

[18] E. Burman and M. A. Fernández. "An unfitted Nitsche method for incompressible fluid–structure interaction using overlapping meshes". In: *Computer Methods in Applied Mechanics and Engineering* 279 (Sept. 2014), pp. 497–514. ISSN: 00457825. DOI: 10.1016/j.cma.2014.07.007.

[19] A. Johansson and M. G. Larson. "A high order discontinuous Galerkin Nitsche method for elliptic problems with fictitious boundary". In: *Numerische Mathematik* 123.4 (Apr. 2013), pp. 607–628. ISSN: 0029-599X, 0945-3245. DOI: 10.1007/s00211-012-0497-1.

[20] E. Burman and P. Hansbo. “Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method”. In: *Applied Numerical Mathematics* 62.4 (Apr. 2012), pp. 328–341. ISSN: 01689274. DOI: 10.1016/j.apnum.2011.01.008.

[21] E. Burman, P. Hansbo, M. G. Larson and S. Zahedi. “Cut finite element methods for coupled bulk–surface problems”. In: *Numerische Mathematik* 133.2 (June 2016), pp. 203–231. ISSN: 0029-599X, 0945-3245. DOI: 10.1007/s00211-015-0744-3.

[22] P. Hansbo, M. G. Larson and S. Zahedi. “A cut finite element method for a Stokes interface problem”. In: *Applied Numerical Mathematics* 85 (Nov. 2014), pp. 90–114. ISSN: 01689274. DOI: 10.1016/j.apnum.2014.06.009.

[23] P. Hansbo, M. G. Larson and S. Zahedi. “Characteristic cut finite element methods for convection–diffusion problems on time dependent surfaces”. In: *Computer Methods in Applied Mechanics and Engineering* 293 (Aug. 2015), pp. 431–461. ISSN: 00457825. DOI: 10.1016/j.cma.2015.05.010.

[24] A. Massing, M. G. Larson, A. Logg and M. E. Rognes. “A Stabilized Nitsche Fictitious Domain Method for the Stokes Problem”. In: *Journal of Scientific Computing* 61.3 (Dec. 2014), pp. 604–628. ISSN: 0885-7474, 1573-7691. DOI: 10.1007/s10915-014-9838-9.

[25] E. Burman. “Ghost penalty”. In: *Comptes Rendus Mathematique* 348.21-22 (Nov. 2010), pp. 1217–1220. ISSN: 1631073X. DOI: 10.1016/j.crma.2010.10.006.

[26] T. Strouboulis, K. Copps and I. Babuška. “The generalized finite element method”. In: *Computer Methods in Applied Mechanics and Engineering* 190.32-33 (May 2001), pp. 4081–4193. ISSN: 00457825. DOI: 10.1016/S0045-7825(01)00188-8.

[27] T.-P. Fries and T. Belytschko. “The extended/generalized finite element method: An overview of the method and its applications”. In: *International Journal for Numerical Methods in Engineering* (Aug. 2010). ISSN: 00295981. DOI: 10.1002/nme.2914.

[28] E. Burman, S. Claus, P. Hansbo, M. G. Larson and A. Massing. “CutFEM: Discretizing geometry and partial differential equations”. In: *International Journal for Numerical Methods in Engineering* 104.7 (Nov. 2015), pp. 472–501. ISSN: 00295981. DOI: 10.1002/nme.4823.

[29] M. Dauge, A. Düster and E. Rank. "Theoretical and Numerical Investigation of the Finite Cell Method". In: *Journal of Scientific Computing* 65.3 (Dec. 2015), pp. 1039–1064. ISSN: 0885-7474, 1573-7691. DOI: 10.1007/s10915-015-9997-3.

[30] I. Babuška and J. M. Melenk. "The Partition of Unity Method". In: *International Journal for Numerical Methods in Engineering* 40.4 (Feb. 1997), pp. 727–758. ISSN: 0029-5981, 1097-0207. DOI: 10.1002/(SICI)1097-0207(19970228)40:4<727::AID-NME86>3.0.CO;2-N.

[31] P. M. Areias and T. Belytschko. "A comment on the article "A finite element method for simulation of strong and weak discontinuities in solid mechanics" by A. Hansbo and P. Hansbo [Comput. Methods Appl. Mech. Engrg. 193 (2004) 3523–3540]". In: *Computer Methods in Applied Mechanics and Engineering* 195.9-12 (Feb. 2006), pp. 1275–1276. ISSN: 00457825. DOI: 10.1016/j.cma.2005.03.006.

[32] J. A. Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. 2nd ed. Cambridge monographs on applied and computational mathematics 3. Cambridge, U.K. ; New York: Cambridge University Press, 1999. ISBN: 978-0-521-64204-0 978-0-521-64557-7.

[33] S. Osher and R. P. Fedkiw. *Level set methods and dynamic implicit surfaces*. Applied mathematical sciences v. 153. New York: Springer, 2003. ISBN: 978-0-387-95482-0.

[34] T. J. R. Hughes, J. A. Cottrell and Y. Bazilevs. "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement". In: *Computer Methods in Applied Mechanics and Engineering* 194.39–41 (Oct. 2005), pp. 4135–4195. ISSN: 0045-7825. DOI: 10.1016/j.cma.2004.10.008.

[35] C. Lehrenfeld. "High order unfitted finite element methods on level set domains using isoparametric mappings". In: *Computer Methods in Applied Mechanics and Engineering* 300 (Mar. 2016), pp. 716–733. ISSN: 00457825. DOI: 10.1016/j.cma.2015.12.005.

[36] R. I. Saye. "High-Order Quadrature Methods for Implicitly Defined Surfaces and Volumes in Hyperrectangles". In: *SIAM Journal on Scientific Computing* 37.2 (Jan. 2015), A993–A1019. ISSN: 1064-8275, 1095-7197. DOI: 10.1137/140966290.

Paper I

Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 309 (2016) 364–387
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

A stabilized Nitsche cut element method for the wave equation

Simon Sticko*, Gunilla Kreiss

Division of Scientific Computing, Department of Information Technology, Uppsala University, Sweden

Received 1 March 2016; received in revised form 1 June 2016; accepted 2 June 2016
Available online 21 June 2016

Abstract

We give a weak formulation for solving the wave equation ($\ddot{u} = \nabla^2 u + f$) on a 2-dimensional immersed domain. In the spatial finite element discretization, boundaries do not conform to element boundaries. Dirichlet and Neumann boundary conditions are enforced weakly by Nitsche's method. Additional penalty terms act on the gradient jumps over the *interior faces of the elements cut by the boundary*. These terms ensure a non-stiff temporal system, which makes it possible to perform explicit time stepping. We give optimal a priori error estimates: second order accuracy for $u - u_h$ and $\dot{u} - \dot{u}_h$, and first order accuracy for $\nabla(u - u_h)$ in L_2-norm. Numerical results verify this.

Keywords: Cut elements; Wave equation; Immersed boundary

1. Introduction

Immersed methods are based on using a mesh, which is not aligned with the domain. The boundary is in a sense floating freely on top of the background mesh. Applications appear mainly in interface problems and in other applications where re-meshing is best avoided. A problem with this approach is the difficulty to enforce boundary conditions, since the boundary of the domain will no longer be aligned with element boundaries. There are several ways of resolving this issue (for an overview of different techniques see [1]).

In this article we employ a technique based on modification of the weak formulation, where boundary conditions are enforced via Nitsche's method [2]. We give a symmetric weak formulation for solving the wave equation, using cut elements. The symmetry of the weak formulation is an important property for wave propagation problems. It guaranties that the finite element discretization results in symmetric matrices, thus giving them real eigenvalues. This in turn guaranties that all wave numbers of the semi-discrete system are real, and therefore eliminates artificial growth or decay in time. That is, discrete energy is preserved in the semi-discrete setting.

The work in this paper builds mainly on work by Burman and Hansbo [3], who studied the corresponding Poisson problem. There stabilization was applied to the stiffness matrix to make it well conditioned independently of how the boundary cuts the background mesh. Cut-element techniques have also previously been used for a number of other

* Corresponding author.
E-mail addresses: simon.sticko@it.uu.se (S. Sticko), gunilla.kreiss@it.uu.se (G. Kreiss).

http://dx.doi.org/10.1016/j.cma.2016.06.001
0045-7825/

problems, e.g. for Stokes equations by Massing et al. [4], Burman and Hansbo [5] and Hansbo et al. [6], and related techniques have been used for interface problems by Hansbo and Hansbo [7] and Wadbro et al. [8].

An alternative to the stabilization used in this paper is diagonal scaling of the mass-matrix. This was used by Reusken [9], together with a finite element space where basis functions with small support in the immersed domain had been removed. Moreover, Burman and Fernández [10] used diagonal scaling for time-dependent problems together with implicit time-stepping in order to guarantee a robust system. However, for wave-propagation explicit time-stepping is usually more efficient. Because of this, we here use stabilization of both the mass and the stiffness matrix, which makes both matrices well conditioned. The stabilization is essential in order to allow for use of explicit time stepping, and ensures that the time-step required for stability of the fully discrete method is not unreasonably small.

We focus on describing the technique for piecewise linear finite elements, but the theory can likely be extended to higher order elements, using ideas from Burman [11] and Massing et al. [4]. However, an implementation of a higher order method contains non-trivial elements, such as accurate quadrature rules both for bulk integration and integration along the boundary.

We will discuss the problem only in 2 space dimensions, but the extension of the weak form to 3 dimensions is straightforward. Extending the implementation to 3 dimensions is however non-trivial, and 3D will therefore not be considered in this article. Nevertheless, 3 dimensional problems have been efficiently implemented with similar techniques by Massing et al. [12].

The paper is organized as follows. We will first describe the method in Section 2, followed by a proof of the a priori error estimates in Section 3. We discuss the well conditioning of the mass matrix in Section 4, followed by a discussion on the stability of the time-stepping in Section 5, and finally give numerical examples in Section 6.

2. A semi-discrete approximation of the wave equation

Consider a 2-dimensional bounded domain Ω with boundary $\Gamma = \Gamma_D \cup \Gamma_N$. On this domain we want to solve the wave equation:

$$\begin{cases} \ddot{u} = \nabla^2 u + f(\mathbf{x}, t) \quad \mathbf{x} \in \Omega, \\ u|_{\Gamma_D} = g_D(\mathbf{x}, t), \\ \dfrac{\partial u}{\partial n}|_{\Gamma_N} = g_N(\mathbf{x}, t). \end{cases} \tag{2.1}$$

We assume that g_D, g_N, f and the initial conditions for u and $\dot{u}$ are sufficiently smooth, so that u, $\dot{u}$, $\ddot{u} \in H^2(\Omega)$. If one would make the choice $\Gamma_D = \Gamma$, $f(\mathbf{x}) = 0$, this would (in 2D) physically correspond to a membrane attached in its boundary, vibrating freely up and down. We want to solve this problem while having the domain immersed in a background mesh, $\mathcal{T}_B$, as in Fig. 2.1, where we assume that this mesh covers the domain Ω. In practice we will use the degrees of freedom of the smallest mesh, $\mathcal{T}$, that covers our domain, as illustrated in Fig. 2.2. More precisely we define

$$\mathcal{T} = \{T \in \mathcal{T}_B : T \cap \Omega \neq \emptyset\},$$

where T denotes an element. Furthermore, let $\Omega_{\mathcal{T}}$ denote the domain that these elements cover. That is,

$$\Omega_{\mathcal{T}} = \bigcup_{T \in \mathcal{T}} T.$$

We assume that $\mathcal{T}$ is a member of a quasi-uniform family of meshes. Furthermore, we need to assume that the mesh is sufficiently refined. We have the following two assumptions:

A1 Each face intersected by the boundary is intersected only once.

A2 In each triangle, K, the boundary, Γ, can be parameterized by the straight line from the intersection points, $\mathbf{x_A}$ and $\mathbf{x_B}$, between Γ and ∂K.

This eliminates the situations illustrated in Fig. 2.3.

In the following, let h denote the mesh-size, which we define as the diameter of the largest circle contained in a single triangle. Let $(\cdot, \cdot)_X$ and $\langle \cdot, \cdot \rangle_Y$ denote the 2- and 1-dimensional L_2-inner products taken over the domains X

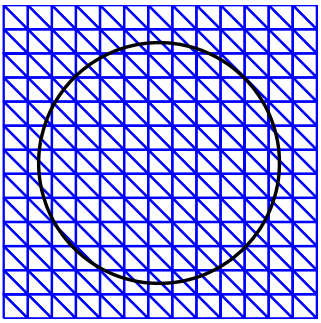

Fig. 2.1. Domain and background mesh, $\mathcal{T}_B$.

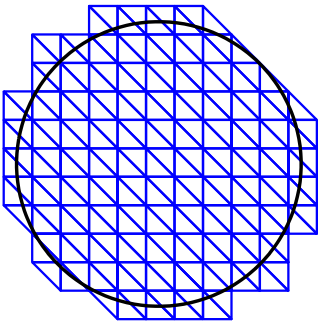

Fig. 2.2. Mesh, $\mathcal{T}$, and fictitious domain $\Omega_{\mathcal{T}}$.

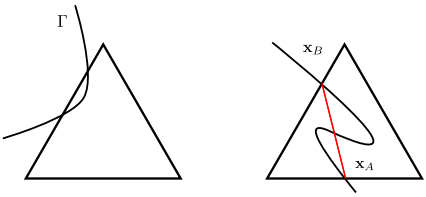

Fig. 2.3. Situations not accounted for.

and Y, respectively. Let also: $\|v\|_X = \sqrt{(v,v)_X}$ or $\|v\|_Y = \sqrt{\langle v,v\rangle_Y}$, where the meaning is clear from the dimension of the subscript. Furthermore, let V_h be the space of piecewise linear polynomials on $\mathcal{T}$, that is:

$$V_h = \{v \in C^0(\bar{\Omega}_{\mathcal{T}}) : v|_K \in P^1(K),\ \forall K \in \mathcal{T}\}.$$

By multiplying (2.1) by a test-function, integrating by parts, and enforcing boundary conditions weakly by Nitsche's method [2], one obtains the following weak formulation: find $u_h \in V_h$ such that

$$(\ddot{u}_h, v)_\Omega + a(u_h, v) = L(v), \quad \forall v \in V_h, \tag{2.2}$$

where

$$a(u,v) = (\nabla u, \nabla v)_\Omega - \left\langle \frac{\partial u}{\partial n}, v \right\rangle_{\Gamma_D} - \left\langle u, \frac{\partial v}{\partial n} \right\rangle_{\Gamma_D} + \frac{\gamma_D}{h} \langle u, v \rangle_{\Gamma_D},$$

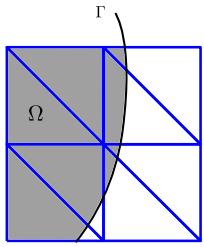

Fig. 2.4. The shaded region denotes the area of integration.

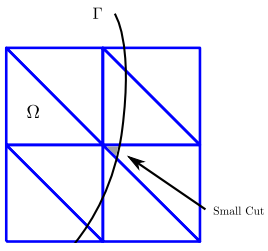

Fig. 2.5. Case when one element has only a small part inside the domain.

and

$$L(v) = (f, v)_\Omega + \left\langle g_D, \frac{\gamma_D}{h} v - \frac{\partial v}{\partial n} \right\rangle_{\Gamma_D} + \langle g_N, v \rangle_{\Gamma_N} . \tag{2.3}$$

Here, γ_D is a scalar penalty parameter.

Embedding the domain in this way is in general not very different from using standard finite elements, the main difference being that one needs to adapt the integration over each element to the part of the element that is inside the domain, as illustrated in Fig. 2.4. However, this may pose a problem when one element is just slightly inside the domain as in Fig. 2.5. In this case the nodes of this element will get a very small contribution from the integration, which will result in ill-conditioned mass and stiffness matrices. As we will see below this will in turn require an unreasonably small time step, or degrade the coercivity of $a(\cdot, \cdot)$. In total this makes the formulation (2.2) a very poor method. Nevertheless, one can remedy these effects by adding a stabilization term, $j(\cdot, \cdot)$, to the parts of the weak form that correspond to the mass matrix and to the stiffness matrix. This stabilization, j, reads:

$$j(u_h, v_h) = \sum_{F \in F_G} \left\langle h[\hat{n} \cdot \nabla u_h], [\hat{n} \cdot \nabla v_h] \right\rangle_F , \tag{2.4}$$

and acts on the jumps in the normal derivative $[\hat{n} \cdot \nabla u_h] = \hat{n} \cdot \nabla u_h\big|_+ - \hat{n} \cdot \nabla u_h\big|_-$ over each face F with normal $\hat{n}$. Here, F_G is the set of faces illustrated in Fig. 2.6. That is, the faces of the elements that are cut by the boundary if one excludes the exterior faces. More precisely, let T_∂ be the elements intersected by the boundary

$$T_\partial = \{T \in \mathcal{T} : T \cap \Gamma \neq 0\},$$

and let F_∂ be the faces of the elements in T_∂. Let now F_E be the faces that make up $\partial \Omega_\mathcal{T}$, then the set F_G is defined as

$$F_G = F_\partial \setminus F_E.$$

Note for future use that j defines a valid semi-norm. Adding the stabilization in (2.4) gives us the following finite element formulation: find $u_h \in V_h$ such that

$$M(\ddot{u}_h, v_h) + A(u_h, v_h) = L(v_h), \quad \forall v_h \in V_h. \tag{2.5}$$

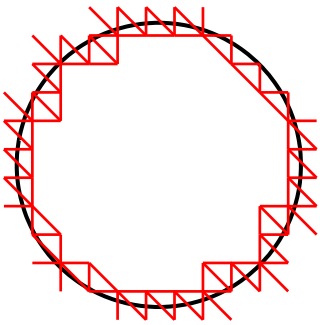

Fig. 2.6. Interior faces of the elements cut by the boundary, F_G.

Here, the bilinear forms M and A are defined by

$$M(\ddot{u}_h, v_h) = (\ddot{u}_h, v_h)_\Omega + \gamma_M h^2 j(\ddot{u}_h, v_h),$$
$$A(u_h, v_h) = a(u_h, v_h) + \gamma_A j(u_h, v_h),$$

which include the scalar parameters γ_A and γ_M. This is now the formulation which we shall be concerned with in the remainder of this paper. Burman and Hansbo [3] used $\gamma_A = 0.5$, $\gamma_D = 5$, which resulted in a well conditioned stiffness matrix. In the numerical experiments we will use these values together with $\gamma_M = 0.25$, which will be justified in Section 6.4.

From the a priori analysis (Lemma 1 in Section 3) one can conclude that the appropriate semi-discrete initial conditions are the Ritz-projections of the continuous initial conditions:

$$\begin{cases} u_h(x,0) = R_h u(x,0), \\ \dot{u}_h(x,0) = R_h \dot{u}(x,0), \end{cases} \tag{2.6}$$

where the Ritz-projection[1] $R_h u \in V_h$ is defined by

$$A(R_h u, v_h) - a(u, v_h) = 0, \quad \forall v_h \in V_h. \tag{2.7}$$

3. A priori analysis

In the following we shall use positive constants C or C_i ($i \in \{\text{a–z, A–Z or } 1, 2, 3 \ldots\}$), which (unless explicitly stated otherwise) may depend on:

- Shape of the triangles in the mesh.
- Shape of Ω.
- How $\partial\Omega$ is partitioned into Γ_N and Γ_D.
- The choice of γ_D, γ_A and γ_M.

In order to bound the a priori error we need to split the error in two parts:

$$\begin{aligned} u_h - u &= \theta + \rho, \\ \theta &:= u_h - R_h u, \\ \rho &:= R_h u - u, \end{aligned} \tag{3.1}$$

where we note that ρ is defined on Ω whereas θ is defined on $\Omega_\mathcal{T}$. The analysis here follows essentially the same path as the analysis of the wave equation with standard finite elements, see for example [13]. We define the "Energy" of

[1] This technically not a projection since u is defined on Ω whereas $R_h u$ is defined on $\Omega_\mathcal{T}$, with $\Omega \subset \Omega_\mathcal{T}$.

theta, E_θ, as

$$E_\theta(t) := \frac{1}{2}(M(\dot{\theta},\dot{\theta}) + A(\theta,\theta)),$$

and utilize that we can bound this from above:

Lemma 1. *The following energy estimate holds*

$$E_\theta(t) \leq \frac{1}{2}\left(\int_0^t \sqrt{\|\ddot{\rho}\|_\Omega^2 + \gamma_M h^2 j(R_h\ddot{u}, R_h\ddot{u})}\right)^2. \tag{3.2}$$

Proof. Using the definition of θ and ρ, and utilizing that the analytical solution $u \in H^2(\Omega)$ satisfies (2.2) now gives us:

$$\begin{aligned} M(\ddot{\theta}, v_h) + A(\theta, v_h) &= \overbrace{M(\ddot{u}_h, v_h) + A(u_h, v_h)}^{L(v_h)} - M(R_h\ddot{u}, v_h) - A(R_h u, v_h) \\ &= \underbrace{(\ddot{u}, v_h)_\Omega + a(u, v_h)}_{L(v_h)} - M(R_h\ddot{u}, v_h) - A(R_h u, v_h) \\ &= (-\ddot{\rho}, v_h)_\Omega + \gamma_M h^2 j(-R_h\ddot{u}, v_h), \end{aligned} \tag{3.3}$$

where we in the last step used the definition of the Ritz-projection in (2.7). Now making the choice $v_h = \dot{\theta}$ in (3.3) and using that j is a valid semi-inner-product gives us

$$\begin{aligned} \frac{dE_\theta}{dt} &\leq \|\ddot{\rho}\|_\Omega \|\dot{\theta}\|_\Omega + \gamma_M h^2 \sqrt{j(R_h\ddot{u}, R_h\ddot{u})}\sqrt{j(\dot{\theta},\dot{\theta})} \\ &\leq \sqrt{\|\ddot{\rho}\|_\Omega^2 + \gamma_M h^2 j(R_h\ddot{u}, R_h\ddot{u})}\sqrt{M_h(\dot{\theta},\dot{\theta})}. \end{aligned}$$

Thus we get

$$\frac{dE_\theta}{dt} \leq \sqrt{\|\ddot{\rho}\|_\Omega^2 + \gamma_M h^2 j(R_h\ddot{u}, R_h\ddot{u})}\sqrt{2E_\theta},$$

which gives us:

$$\frac{d}{dt}(\sqrt{E_\theta}) \leq \frac{1}{\sqrt{2}}\sqrt{\|\ddot{\rho}\|_\Omega^2 + \gamma_M h^2 j(R_h\ddot{u}, R_h\ddot{u})}. \tag{3.4}$$

Note that the initial conditions in (2.6) are equivalent to choosing

$$\begin{cases} \theta|_{t=0} = 0, \\ \dot{\theta}\big|_{t=0} = 0, \end{cases}$$

which gives $E_\theta(0) = 0$. Thus integrating and then squaring (3.4) finally gives us (3.2). □

3.1. Properties of the Ritz projection

Here we collect some results for the Ritz projection in (2.7). It should be noted that Ritz-projection is in fact nothing but the stabilized cut-element solution of the corresponding Poisson problem. That is, if one for a fixed time, t_0, describes the solution $\tilde{u}(\mathbf{x}) = u(\mathbf{x}, t_0)$ by

$$\begin{cases} \nabla^2\tilde{u} = \tilde{f}(\mathbf{x}) \quad \mathbf{x} \in \Omega, \\ \tilde{u}|_{\Gamma_D} = g_D(\mathbf{x}, t_0), \\ \left.\dfrac{\partial\tilde{u}}{\partial n}\right|_{\Gamma_N} = g_N(\mathbf{x}, t_0), \end{cases} \tag{3.5}$$

then this solution will satisfy

$$a(\tilde{u}, v) = \tilde{L}(v), \quad \forall v \in V_h, \tag{3.6}$$

where $\tilde{L}$ is the same as L from (2.3), but using $\tilde{f}$ instead of f:

$$\tilde{L}(v) = (\tilde{f}, v)_\Omega + \left\langle g_D, \frac{\gamma_D}{h} v - \frac{\partial v}{\partial n} \right\rangle_{\Gamma_D} + \langle g_N, v \rangle_{\Gamma_N}.$$

The finite element method to solve (3.5) is now: find $\tilde{u}_h \in V_h$ such that[2]

$$A(\tilde{u}_h, v_h) = \tilde{L}(v_h), \quad \forall v_h \in V_h. \tag{3.7}$$

By subtracting (3.6) from (3.7) and comparing with (2.7) one can now identify $R_h u$ with the solution $\tilde{u}_h$. This problem was studied by Burman and Hansbo [3], which allows us to make use of the properties of the finite element solution derived there.

Some of the results we need are formulated in the following mesh-dependent norms

$$|||v|||_*^2 = \|\nabla v\|_\Omega^2 + h\|\hat{n} \cdot \nabla v\|_\Gamma^2 + h^{-1}\|v\|_{\Gamma_D}^2,$$
$$|||v|||_h^2 = \|\nabla v\|_{\Omega_\mathcal{T}}^2 + h^{-1}\|v\|_{\Gamma_D}^2 + j(v, v).$$

Note that due to the following inverse inequality (proven in [7])

$$h^{1/2}\|\hat{n} \cdot \nabla v\|_{\Gamma \cap T} \leq C\|\nabla v\|_T, \quad v \in V_h, \tag{3.8}$$

where C only depends on the shape of T, we have

$$|||v|||_h^2 \geq C_0 |||v|||_*^2, \quad v \in V_h. \tag{3.9}$$

The inverse inequality (3.8) is a special version of the standard inverse inequalities (see e.g. [14]):

$$h^{1/2}\|\hat{n} \cdot \nabla v\|_F \leq C\|\nabla v\|_T, \quad v \in V_h, \tag{3.10}$$
$$\|\nabla v\|_T \leq Ch^{-1}\|v\|_T, \quad v \in V_h, \tag{3.11}$$

which are also used.

Important properties of the Ritz-projection are Galerkin orthogonality, coercivity and continuity. In the presence of the stabilizing term the Galerkin orthogonality, which follows by subtracting (3.6) from (3.7), takes the form

$$a(R_h u - u, v_h) = -\gamma_A j(R_h u, v_h), \quad \forall v_h \in V_h. \tag{3.12}$$

The appropriate continuity estimate is

$$|a(w, v_h)| \leq C_c |||w|||_* |||v_h|||_*, \tag{3.13}$$

for all $v_h \in V_h$, and for all w such that $w = w_1 + w_2$, with $w_1 \in H^2(\Omega_\mathcal{T})$, $w_2 \in V_h$. Coercivity states that there exists a constant, independent of h and of how the elements are cut by the boundary, such that

$$|||v|||_h^2 \leq C_r A(v_h, v_h), \quad \forall v_h \in V_h. \tag{3.14}$$

For proof of (3.13) and (3.14) we refer to [3]. Note that the norm on the left hand side in (3.14) includes an integral over the domain $\Omega_\mathcal{T}$, while the corresponding integral on the right hand side is over the smaller domain Ω. The stabilizing jump term with coefficient γ_A is essential for the estimate to be valid independently of how the boundary cuts the elements. The constant in (3.14) grows unboundedly if $\gamma_A \to 0$.

Let $\tilde{I}$ be any extension–interpolation operator which extends $u \in H^2(\Omega)$ to $H^2(\Omega_\mathcal{T})$ and then interpolates to V_h. Straightforwardly these properties now imply that $e_h := R_h u - \tilde{I}u$ satisfies

$$\begin{aligned} |||e_h|||_h^2 \leq C_r A(e_h, e_h) &= C_r \left(A(R_h u, e_h) - a(u, e_h) + a(u, e_h) - A(\tilde{I}u, e_h) \right) \\ &= C_r (a(u - \tilde{I}u, e_h) - \gamma_A j(\tilde{I}u, e_h)) \\ &\leq C_r (|||u - \tilde{I}u|||_* |||e_h|||_* + \gamma_A j(\tilde{I}u, \tilde{I}u)^{1/2} j(e_h, e_h)^{1/2}). \end{aligned}$$

[2] If we would have $\Gamma_D = \emptyset$, one could instead solve the problem (2.7) in the space $\tilde{V}_h = \{v \in V_h : \int_\Omega v = C\}$, where $C = \int_\Omega u(x, t_0)$, in order for the problem (3.5) to have a unique solution.

By applying (3.9) to $|||e_h|||_*$ and noting that $j(e_h, e_h)^{1/2} \leq |||e_h|||_h$ we get:

Lemma 2. *For any extension-interpolant, $\tilde{I}$, which extends $u \in H^2(\Omega)$ to $H^2(\Omega_\mathcal{T})$ and interpolates to V_h, we have*

$$|||R_h u - \tilde{I}u|||_h \leq C_I \left(|||u - \tilde{I}u|||_* + j(\tilde{I}u, \tilde{I}u)^{\frac{1}{2}} \right). \tag{3.15}$$

Here (as in [3]) we will use an extension operator $\mathbb{E} : H^2(\Omega) \to H^2(\Omega_\mathcal{T})$ with the property $\|\mathbb{E}u\|_{H^2(\Omega_\mathcal{T})} \leq C_E \|u\|_{H^2(\Omega)}$, and define the interpolant $I^* \coloneqq \mathcal{C}_h \mathbb{E}$, where $\mathcal{C}_h : H^1(\Omega_\mathcal{T}) \to V_h$ is the ordinary Clement interpolant. The Clement interpolant uses a local averaging procedure in order to handle functions where pointwise values are undefined. For the interpolant we have (see e.g. [15])

$$|v - \mathcal{C}_h v|_{r,T} \leq C h^{s-r} |v|_{s,N(T)}, \quad 0 \leq r \leq s \leq 2, \tag{3.16}$$

$$|v - \mathcal{C}_h v|_{r,F} \leq C h^{s-r-\frac{1}{2}} |v|_{s,N(F)}, \quad 0 \leq r \leq s \leq 2. \tag{3.17}$$

Here, the region $N(T)$ ($N(F)$) consists of the region of elements sharing a vertex with T (F). By $|v|_{s,X}$ we denote the $H^s(X)$-semi-norm. Furthermore, we also need:

Lemma 3. *For the interpolant, I^*, we have that for all $u \in H^2(\Omega)$*

$$|||u - I^* u|||_* + j(I^* u, I^* u)^{\frac{1}{2}} \leq C_1 h |u|_{2,\Omega}. \tag{3.18}$$

The proof involves using trace inequalities, the interpolation properties, and the properties of the extension operator. For details see Lemma 5 in [3]. The final error estimate is:

Lemma 4. *For all $u \in H^2(\Omega)$ the solution of (2.7), $R_h u$, satisfies*

$$\|R_h u - u\|_\Omega \leq C_2 h^2 |u|_{2,\Omega}, \tag{3.19}$$

$$\|R_h u - u\|_{H^1(\Omega)} \leq C_3 h |u|_{2,\Omega}. \tag{3.20}$$

Here (3.20) follows directly from Lemma 3, while the proof of (3.19) involves an auxiliary Poisson problem with $R_h u - u$ as right hand side. For details see [3], in particular the proof of Proposition 10.

3.2. Semi-discrete stability

Consider again the semi-discrete wave equation in (2.5). Utilizing Lemmas 2–4, will now make us able to bound the term $j(R_h \ddot{u}, R_h \ddot{u})$ in Lemma 1.

Lemma 5. *For the term $j(R_h u, R_h u)$ we have the estimate*

$$j(R_h u, R_h u) \leq C_4 h^2 |u|^2_{2,\Omega}. \tag{3.21}$$

Proof. Using that $\sqrt{j(v,v)}$ is a valid semi-norm now gives us

$$\sqrt{j(R_h u, R_h u)} \leq \sqrt{j(R_h u - I^* u, R_h u - I^* u)} + \sqrt{j(I^* u, I^* u)}. \tag{3.22}$$

From the definition of $|||\cdot|||_h$ and (3.15) and (3.18) we get

$$j(R_h u - I^* u, R_h u - I^* u)^{\frac{1}{2}} \leq |||R_h u - I^* u|||_h \leq C_I \left(|||u - I^* u|||_* + j(I^* u, I^* u)^{\frac{1}{2}} \right).$$

Since both terms in (3.22) can be bounded by Lemma 3 we finally get (3.21). □

By putting Lemmas 1, 4 and 5 together we obtain the final estimate for the convergence rate.

Proposition 6. *At any given time, t, the solution of the finite element method (2.5) fulfills the following a priori estimates:*

$$\begin{aligned} \|u_h(t) - u(t)\|_\Omega &\leq C_5 h^2, \\ \|\dot{u}_h(t) - \dot{u}(t)\|_\Omega &\leq C_6 h^2, \\ \|\nabla u_h(t) - \nabla u(t)\|_\Omega &\leq C_7 h, \end{aligned} \tag{3.23}$$

where the constants depend on t.

Proof. Using Lemmas 1 and 5 gives

$$E_\theta \leq \frac{1}{2}\left(\int_0^t \sqrt{\|\ddot{u} - R_h\ddot{u}\|_\Omega^2 + \gamma_M C_4 h^4 |\ddot{u}|_{2,\Omega}^2}\, dt\right)^2,$$

and by using Lemma 4 we get

$$E_\theta \leq C(t)h^4.$$

So by the definition of the energy we obtain

$$\|\dot{u}_h(t) - R_h\dot{u}(t)\|_\Omega \leq C(t)h^2, \tag{3.24}$$

$$\|\nabla(u_h(t) - R_h u(t))\|_\Omega \leq C(t)h^2. \tag{3.25}$$

Finally given that the boundary is not moving with time we have

$$2\|\theta\|_\Omega \frac{d}{dt}\|\theta\|_\Omega = \frac{d}{dt}\|\theta\|_\Omega^2 = \frac{d}{dt}\int_\Omega \theta^2 d\Omega = 2\int_\Omega \theta\dot{\theta} d\Omega \leq 2\|\theta\|_\Omega \|\dot{\theta}\|_\Omega.$$

Now using that $\theta|_{t=0} = 0$ we get

$$\|\theta\|_\Omega \leq \int_0^t \|\dot{\theta}\|_\Omega dt,$$

which gives us

$$\|u_h(t) - R_h u(t)\|_\Omega \leq C(t)h^2. \tag{3.26}$$

Finally using (3.24)–(3.26) together with the triangle inequality and Lemma 4 we obtain the estimates in Proposition 6. □

4. Conditioning of the mass matrix

Massing et al. [4] studied the Stokes equations under similar assumptions as A1–A2. In particular the following lemma was formulated in Proposition 5.1 in [4]:

Lemma 7. *For $v_h \in V_h$ and $\gamma_M = 1$ we have the following bound:*

$$\|v_h\|_{\Omega_\mathcal{T}}^2 \leq C_M M(v_h, v_h) \leq C_M \|v_h\|_{\Omega_\mathcal{T}}^2, \tag{4.1}$$

where C_M is a constant depending on the shape regularity of the background mesh, but not on how the elements are cut by the boundary.

Note that the stabilizing term is essential for this estimate. As $\gamma_M \to 0$ the constant C_M would grow unboundedly. Lemma 7 will give us a bound for the condition number of the mass matrix, $\mathcal{M}$, with elements

$$\mathcal{M}_{ij} = M(\phi_i, \phi_j),$$

where $\{\phi_j\}_{j=1}^N$ is our finite-element basis. Let $\mathcal{M}_\mathcal{T}$ denote the "ordinary" mass matrix over the background mesh $\mathcal{T}$, that is: $(\mathcal{M}_\mathcal{T})_{ij} = (\phi_i, \phi_j)_{\Omega_\mathcal{T}}$. By letting $v_h = \sum_{j=1}^N \mathcal{V}_j \phi_j$ and denoting $\mathcal{V} = [\mathcal{V}_1, \ldots, \mathcal{V}_N]^T \in \mathbb{R}^N$ now (4.1) immediately translates to

$$\mathcal{V}^T \mathcal{M}_\mathcal{T} \mathcal{V} \leq C_M \mathcal{V}^T \mathcal{M} \mathcal{V} \leq C_M \mathcal{V}^T \mathcal{M}_\mathcal{T} \mathcal{V}.$$

This shows us that the eigenvalues of $\mathcal{M}$, $\lambda_{\mathcal{M}}$, satisfy

$$\lambda_{\mathcal{M}} \in [\mu_{\min}/C_M,\ \mu_{\max}], \tag{4.2}$$

where $\mu_{\max}$, $\mu_{\min}$ denote the maximum and minimum eigenvalues of $\mathcal{M}_{\mathcal{T}}$, respectively. So this means that the condition numbers, $\kappa(\cdot)$, relate as

$$\kappa(\mathcal{M}) \leq C_M \kappa(\mathcal{M}_{\mathcal{T}}).$$

Now, since we have assumed that the mesh is a member of a quasi-uniform family of meshes, $\kappa(\mathcal{M}_{\mathcal{T}})$ is bounded by a constant, independent of h. Consequently $\kappa(\mathcal{M})$ is bounded independently of h, and of how the elements are cut by the boundary.

5. Time-discrete stability

The space discretization leads to a system of the form

$$\mathcal{M}\ddot{\xi} + \mathcal{A}\xi = \mathcal{L}(t). \tag{5.1}$$

Here ξ is a vector containing the N degrees of freedom, and $\mathcal{A}$ is the stiffness matrix: $\mathcal{A}_{ij} = A(\phi_i, \phi_j)$. By Lemma 7 and since $A(\cdot,\cdot)$ is coercive in the $|||\cdot|||_h$-norm, (see (3.14)), both $\mathcal{M}$ and $\mathcal{A}$ are symmetric and positive definite matrices. This means that in a non forced system ($\mathcal{L}(t) = 0$) a discrete energy of the form

$$E(t) = \frac{1}{2}\left(\dot{\xi}^T \mathcal{M}\dot{\xi} + \xi^T \mathcal{A}\xi\right), \tag{5.2}$$

is conserved:

$$\frac{dE}{dt} = 0.$$

This means that with respect to this energy there is no growth or decay, which thus determines the systems asymptotic behavior in time.

Furthermore, one can rewrite (5.1) on the form

$$\ddot{\eta} + \mathcal{B}\eta = \mathcal{F}(t), \tag{5.3}$$

where

$$\mathcal{B} = \mathcal{M}^{-\frac{1}{2}}\mathcal{A}\mathcal{M}^{-\frac{1}{2}}, \qquad \eta = \mathcal{M}^{\frac{1}{2}}\xi, \qquad \mathcal{F} = \mathcal{M}^{-\frac{1}{2}}\mathcal{L}.$$

This system is stable if the corresponding homogeneous system:

$$\ddot{\eta} + \mathcal{B}\eta = 0$$

is stable. Note that $\mathcal{B}$ is also a symmetric and positive definite matrix. For this kind of equation it is possible to construct a stable higher order time-discretization given that the time-step k fulfills a condition of the form[3]:

$$k^2\|\mathcal{B}\|_N \leq \alpha.$$

Here $\|\cdot\|_N$ is the matrix norm corresponding to the usual Euclidean vector norm, and α is some constant depending on the time-stepping scheme. We do for example obtain $\alpha = 8$, if we rewrite (5.3) as a first order system, and discretize it by the classical fourth order Runge–Kutta method.[4] For future reference it should be noted that $\|\mathcal{B}\|_N$ is equal to the largest eigenvalue of the following generalized eigenvalue problem

$$\mathcal{A}x - \lambda\mathcal{M}x = 0,$$

[3] See for example [16] page 72.

[4] Follows by using the approach in [17] Section I.1 and [18] Section IV.2.

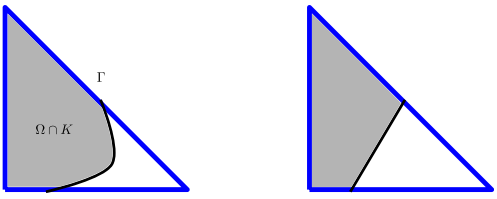

Fig. 6.1. An element, K, together with the boundary, Γ, and the straight-line approximation of the boundary.

which gives us a stability relation of the form

$$k^2\lambda_{\max} \leq \alpha. \tag{5.4}$$

Furthermore, we have

$$\|\mathcal{B}\|_N = \|\mathcal{M}^{-\frac{1}{2}}\mathcal{A}\mathcal{M}^{-\frac{1}{2}}\|_N \leq \|\mathcal{M}^{-\frac{1}{2}}\|_N^2\|\mathcal{A}\|_N.$$

In [3] it was shown that there exists a positive constant such that the stiffness matrix satisfies

$$\|\mathcal{A}\|_N \leq C_{\mathcal{A}}h^{-2}\mu_{\max},$$

where $C_{\mathcal{A}}$ is independent of how the boundary cuts the mesh ($\mu_{\max}$ defined in Section 4). From (4.2) we now obtain that

$$\|\mathcal{M}^{-\frac{1}{2}}\|_N^2 \leq \frac{C_M}{\mu_{\min}},$$

thus giving us

$$\|\mathcal{B}\|_N \leq C_{\mathcal{A}}C_M h^{-2}\kappa(\mathcal{M}_{\mathcal{T}}).$$

This shows us that as sufficient condition for the time-stability of the method is

$$k \leq h\sqrt{\frac{\alpha}{C_{\mathcal{A}}C_M\kappa(\mathcal{M}_{\mathcal{T}})}}.$$

We note that this stability condition is independent of how the boundary cuts the mesh. Without stabilization the mass matrix might become nearly singular, resulting in an exceedingly strict stability condition.

6. Numerical examples

In the following we present five different numerical experiments. First we intend to show that the method converges with the estimated rates, which is done for two different problems, presented in Sections 6.1 and 6.2. Then in Section 6.3 we aim to show that the stabilization is essential, in order for the time-step restriction to be reasonable. In Section 6.4 we aim to show that the choice $\gamma_M = 0.25$ is robust with respect to the geometry of the domain. Finally, the experiment in Section 6.5 intends to show how the added stabilization affects waves with respect to dispersion and wave amplitudes.

In the implementation, a cut with the boundary was approximated by a straight line, as seen in Fig. 6.1. In the experiments time-stepping was performed with an explicit 4th order classical Runge–Kutta, after rewriting the second order system as a first order system.

6.1. A vibrating membrane

Let Ω be a circular domain with radius $R = 1$, and consider (2.1) with $f(\mathbf{x}) = 0$, and constant Dirichlet boundary condition: $u|_\Gamma(\mathbf{x}) = \bar{u} = 1$. Using separation of variables it is possible to show that an analytic solution is

$$u(\mathbf{x}, t) = \bar{u} + J_0\left(\alpha_n\frac{\|\mathbf{x}\|}{R}\right)\cos(\omega_n t), \tag{6.1}$$

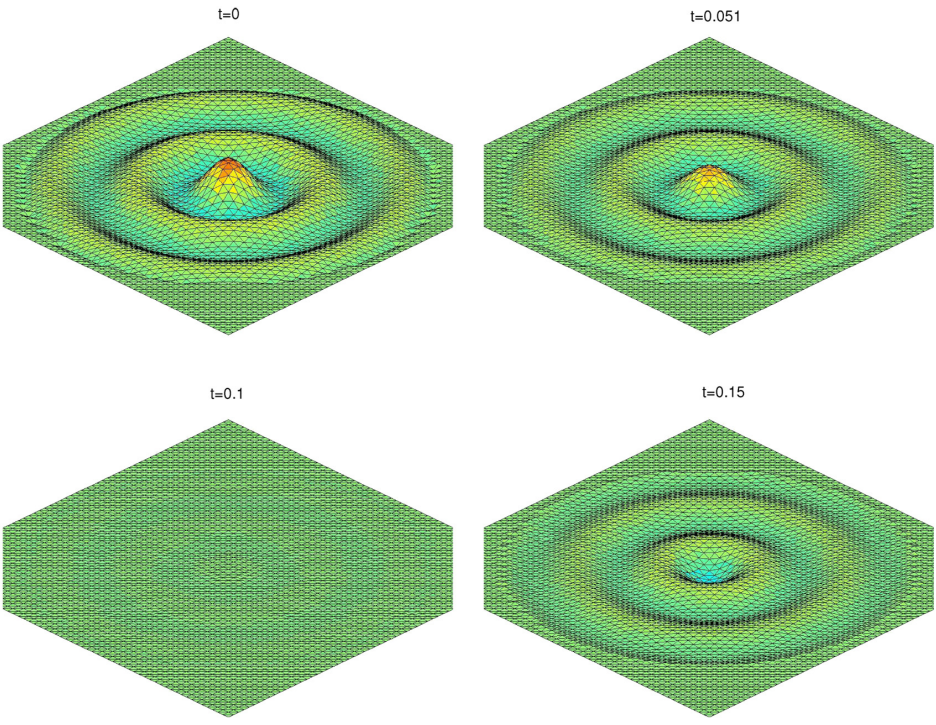

Fig. 6.2. Snapshots of the vibrating membrane.

where J_0 is the zeroth Bessel function of the first kind, α_n the nth zero of J_0 and $\omega_n = \frac{\alpha_n}{R}$. So in order to obtain the analytic solution in (6.1) let us use the initial conditions

$$u(\mathbf{x})|_{t=0} = \bar{u} + J_0\left(\alpha_n \frac{\|\mathbf{x}\|}{R}\right),$$
$$\dot{u}(\mathbf{x})|_{t=0} = 0.$$

Furthermore set $n = 5$. For the computation we use a fictitious domain, $\Omega_{\mathcal{T}}$, as in Fig. 2.2. A few snapshots of the numerical solution are provided in Fig. 6.2.

The L_2-norm of $u - u_h$ as depending on mesh-size is seen in Fig. 6.3. This was obtained after time-stepping until an end-time $t_{\text{end}} = \frac{9}{8}T$ (T being the period-time) with decreasingly smaller grid sizes, while keeping the time-step constant. Included is also a straight line with the expected convergence rate. In the same way the L_2-error of the time derivative is seen in Fig. 6.4, and the L_2-error of the gradient in Fig. 6.5. The results in Figs. 6.3–6.5 are all in agreement with the estimates in Proposition 6. Even though we have not provided an estimate for it, Fig. 6.6 shows the $L_2(\Gamma)$-error along the boundary. We see that the convergence rate is close to 2.

For this problem the condition number of the mass matrix is seen in Fig. 6.7. We see that the condition number is very small and oscillates, but varies very little in magnitude upon refinement.

6.2. An outer problem

Consider a plane wave of the form:

$$u_{\text{in}} = \text{Re}(e^{i\gamma(x-t)}),$$

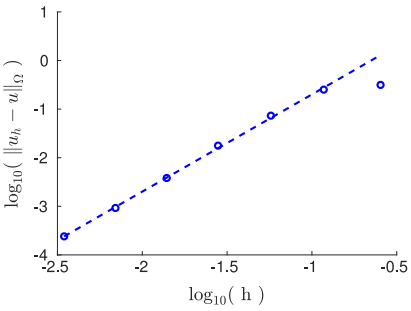

Fig. 6.3. $\|u_h - u\|_\Omega$ against mesh size h.

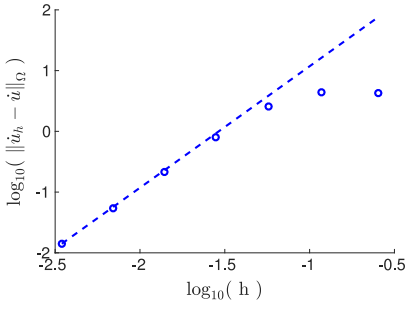

Fig. 6.4. $\|\dot{u}_h - \dot{u}\|_\Omega$ against mesh size h.

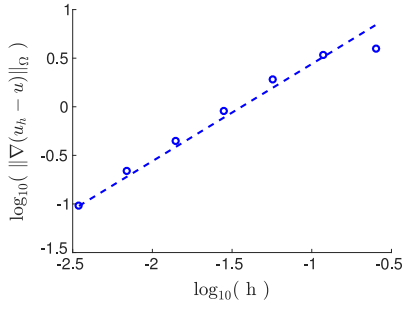

Fig. 6.5. $\|\nabla(u_h - u)\|_\Omega$ against mesh size h.

traveling in a 2-dimensional infinite domain (i.e. all of $\mathbb{R}^2$). This wave travels towards a circular object (as depicted in Fig. 6.8) centered in origo, with radius R, where a homogeneous Neumann boundary condition:

$$\left.\frac{\partial u}{\partial n}\right|_{\Gamma_N} = 0,$$

is enforced along the boundary, Γ_N. For this problem one can show (see [19] pages 395–398) that the sum of the incoming and reflected wave $\tilde{u}$ can be written in cylinder coordinates as

$$\tilde{u}(r,\theta,t) = \mathrm{Re}\left(e^{i\gamma t}\left(R_0(r) + \sum_{n=1}^{\infty} R_n(r)\,\Theta_n(\theta)\right)\right), \tag{6.2}$$

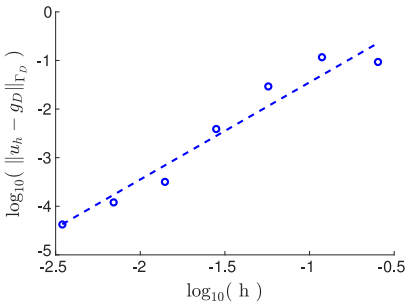

Fig. 6.6. $\|u_h - g_D\|_\Gamma$ against mesh size h.

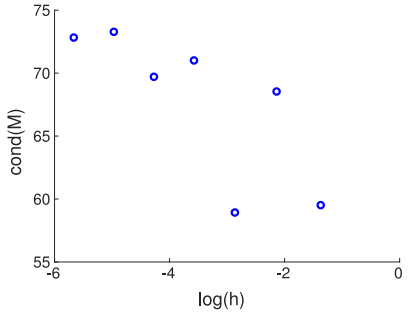

Fig. 6.7. Condition number of the mass matrix as a function of mesh size.

where the θ- and r-dependence is

$$\Theta_n = 2(-i)^n \cos(n\theta),$$
$$R_n = J_n(\gamma r) + a_n H_n^{(2)}(\gamma r),$$

with coefficients

$$a_n = -\frac{\frac{\partial J_n}{\partial r}(\gamma R)}{\frac{\partial H_n^{(2)}}{\partial r}(\gamma R)}.$$

Here J_n is the nth Bessel function of the first kind and $H_n^{(2)}$ is the nth Hankel function of the second kind. To obtain this analytic solution on a finite domain we first let Ω be a square domain with side-length L with the circle taken out, as seen in Fig. 6.8. Then we start from initial conditions:

$$u|_{t=0} = \tilde{u}|_{t=0}, \qquad \left.\frac{\partial u}{\partial t}\right|_{t=0} = \left.\frac{\partial \tilde{u}}{\partial t}\right|_{t=0},$$

and enforce the time dependent Dirichlet boundary conditions:

$$u|_{\Gamma_D}(t) = \tilde{u}|_{\Gamma_D}(t)$$

along the outer boundary Γ_D.

To solve this numerically we use a mesh where the outer boundary Γ_D conforms to the element boundaries but the inner boundary Γ_N does not. To get a test case we choose $\gamma = \pi$. A few snapshots of the numerical solution are provided in Fig. 6.9.

By keeping 50 terms in the series expansion in (6.2), we are able to test our numerical solution. In the same way as in Section 6.1, we solve to an end-time $t_{\text{end}} = 1$ with constant time-step and calculate the L_2-errors for decreasing

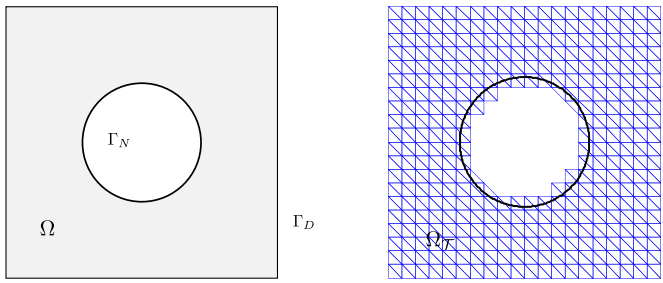

Fig. 6.8. To the left: Domain Ω for the outer problem, with inner boundary, Γ_N, and outer boundary, Γ_D. To the right: the triangulated fictitious domain $\Omega_\mathcal{T}$.

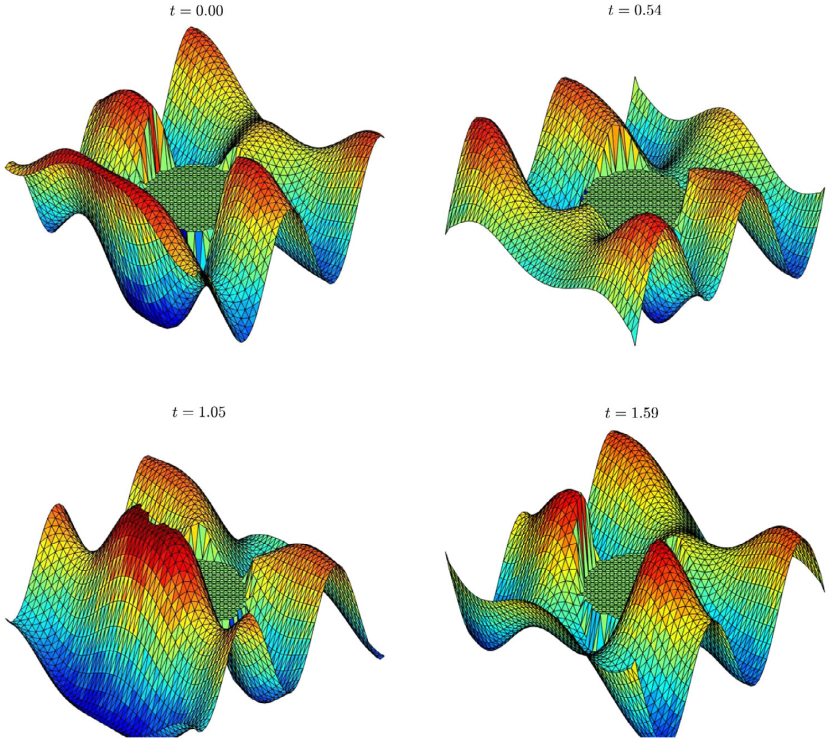

Fig. 6.9. Snapshots of the plane wave test case.

mesh sizes, h. The results are seen in Figs. 6.10–6.13. We see that we get second order convergence for the error in u_h and $\dot{u}_h$, and first order convergence for ∇u_h. All in agreement with the estimates in Proposition 6. The $L_2(\Gamma_N)$-error along the inner boundary is shown in Fig. 6.13. We see that the error converges approximately linearly.

6.3. Time-step restriction

Consider the problem in Fig. 6.14. We have a square background mesh of side-length $L = 3$, and a smaller rectangular domain Ω floating on top of this mesh. The domain is aligned with the mesh at the top, bottom and left side, but at the right side Ω cuts the background mesh with a cut of size ϵH, where H is the horizontal distance

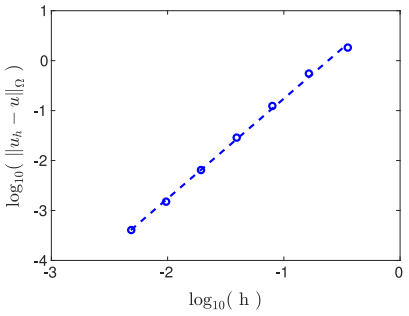

Fig. 6.10. $\|u_h - u\|_\Omega$ against mesh size h.

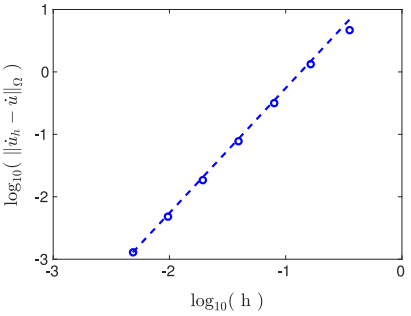

Fig. 6.11. $\|\dot{u}_h - \dot{u}\|_\Omega$ against mesh size h.

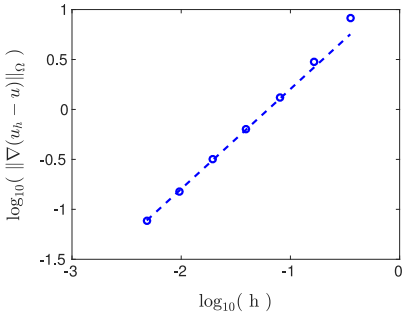

Fig. 6.12. $\|\nabla(u_h - u)\|_\Omega$ against mesh size h.

between mesh nodes. Now, denote as previously

$$\mathcal{M}_{ij} = M(\phi_i, \phi_j),$$
$$\mathcal{A}_{ij} = A(\phi_i, \phi_j),$$

and define $\tilde{\mathcal{M}}$ as the unstabilized version of the mass matrix

$$\tilde{\mathcal{M}}_{ij} = (\phi_i, \phi_j)_\Omega.$$

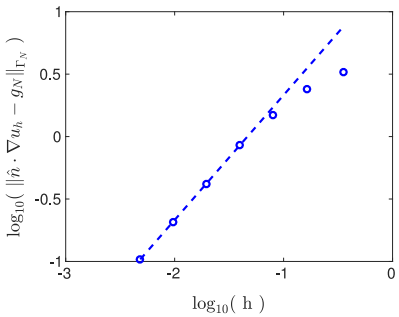

Fig. 6.13. $\|\frac{\partial u_h}{\partial n} - g_N\|_{\Gamma_N}$ against mesh size h.

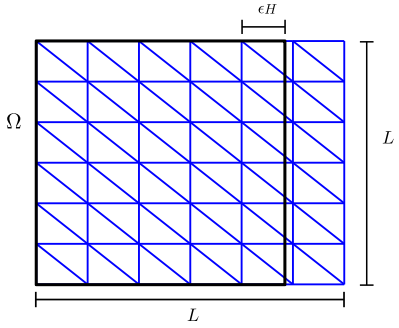

Fig. 6.14. A square domain with a cut of size ϵH.

With Section 5 in mind we would now like to understand how the stabilization effects the stability requirement on the time-step. Thus we are interested in the maximal eigenvalues of the following two generalized eigenvalue problems,

$$\mathcal{A}x = \lambda_u \tilde{\mathcal{M}}x, \tag{6.3}$$

$$\mathcal{A}x = \lambda_s \mathcal{M}x, \tag{6.4}$$

as ϵ approaches zero. Here the subscripts s and u are intended to indicate the "stabilized" and "unstabilized" problems. The maximal eigenvalues for these two problems are shown in Figs. 6.15 and 6.16, against decreasing ϵ and for three different levels of mesh-refinement. Since the eigenvalues at best can be proportional to h^{-2} all eigenvalues have been scaled by h^2, for easier comparison over mesh refinements. In Fig. 6.15 we see that the maximal eigenvalue for the unstabilized case increases very rapidly when decreasing ϵ. The figure indicates that

$$\lambda_u \propto \frac{1}{h^2\epsilon^4}. \tag{6.5}$$

In contrast the largest generalized eigenvalue when the mass matrix is stabilized is shown in Fig. 6.16. By noting the difference in range of the y-axis, we see that the largest eigenvalue is essentially independent of ϵ.

Note that, from Fig. 6.16 we have $\log_{10}(h^2\lambda_{\max}) \approx 1.03$. Using (5.4) this implies the stability relation for this problem is

$$k \leq h\sqrt{\frac{\alpha}{10^{1.03}}}.$$

So, if we would use 4th order classical Runge–Kutta for time-stepping ($\alpha = 8$), we would get

$$k \leq 0.86h.$$

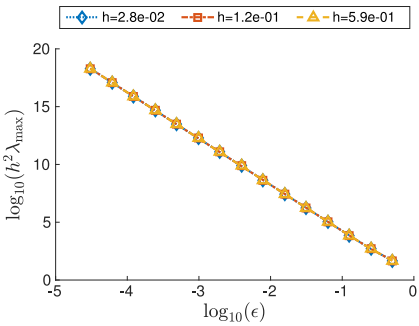

Fig. 6.15. Largest generalized eigenvalues for the unstabilized case (6.3), against decreasing cut-size.

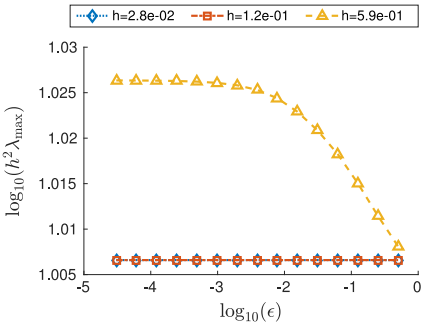

Fig. 6.16. Largest generalized eigenvalues for the stabilized case (6.4), against decreasing cut-size.

6.4. The mass matrix stabilization

The choice of $\gamma_M = 0.25$ in the previous experiments was a bit heuristic. Here we shall try to justify this by considering how the choice of γ_M influences the condition number of the mass matrix. Consider first the same circular domain as in Section 6.1, for a fix mesh size ($h = 0.0138$). The condition number as a function of γ_M is seen in Fig. 6.17. We clearly see that the condition number grows unboundedly as γ_M decreases towards zero. When going away from $\gamma_M = 0$, the condition number decreases and reaches a minimum around $\gamma_M = 0.1$, and then increases slowly and approximately linearly as γ_M increases beyond this point. According to this figure $\gamma_M = 0.25$ seems like a reasonable choice. However, one could argue that the location of the minimum could be different for a different domain. To investigate this, let the domain instead be an ellipse centered in origo, with major and minor axes a and b. Fix $a = 1$ and consider for what value of γ_M the minimum in condition number occurs as b varies. Define $\tilde{\gamma}_M$ as the value of γ_M for which minimum occurs. That is,

$$\tilde{\gamma}_M = \arg\min_{\gamma_M} \kappa(\tilde{\mathcal{M}} + \gamma_M h^2 \mathcal{J}), \tag{6.6}$$

where $\mathcal{J}_{ij} = j(\phi_i, \phi_j)$. The result is shown in Fig. 6.18. As b/a varies between $b/a = 0.1$ and $b/a = 1$ we see that the location of the minimum varies only slightly between 0.1 and 0.2, which indicates that the choice $\gamma_M = 0.25$ is robust.

6.5. Boundary effects from the stabilization

It is possible that the stabilization has a negative effect on the wave propagation. In particular we want to examine how the wave speed and the amplitude of waves are affected by it. In order to examine this, consider the problem in

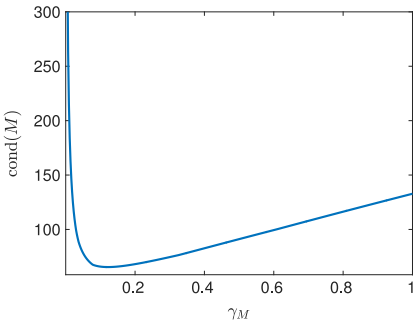

Fig. 6.17. Condition number of the mass matrix $\mathcal{M}$ depending on γ_M. For a disk domain and for $h = 0.0138$.

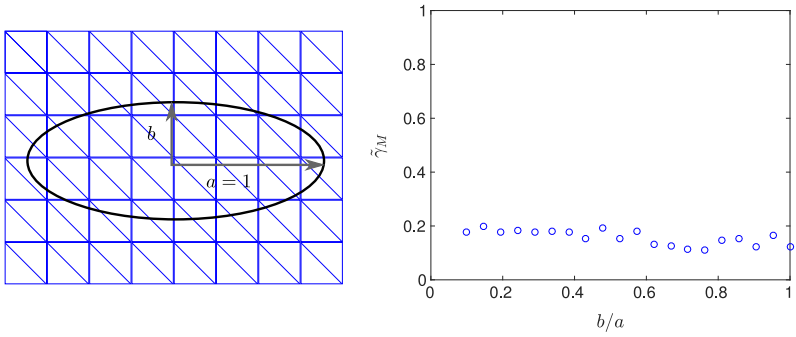

Fig. 6.18. Location of minimizing value of γ_M (see (6.6)), depending on the ratio of the half axes of the ellipse.

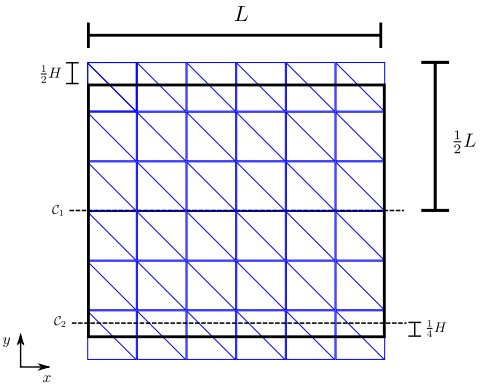

Fig. 6.19. Setup for examining the stabilization effects on wave speed and wave amplitude.

Fig. 6.19. We have a square mesh of side length $L = 3$ with a rectangular domain, Ω on top of it. The left and right domain boundaries are aligned with the boundaries of the mesh. But the top and bottom domain boundaries cut half way through the top and bottom rows of elements. In the x-direction we set periodic boundary conditions, and on the top and bottom boundaries, homogeneous Neumann boundary conditions are enforced.

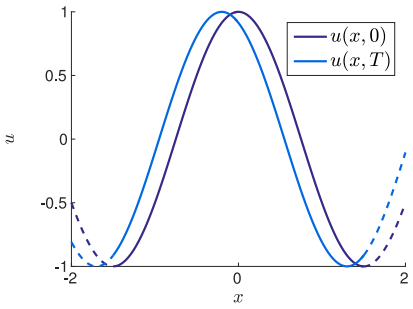

Fig. 6.20. Illustration of the phase shift between initial and final solution along one of the straight lines $\mathcal{C}_i$. Periodic extensions as dashed lines.

On this domain we start off with initial conditions corresponding to a plane wave traveling in the positive x-direction.

$$\begin{cases} u|_{t=0} = \cos(k(x-t)), \\ \left.\dfrac{\partial u}{\partial t}\right|_{t=0} = -k\sin(k(x-t)). \end{cases} \tag{6.7}$$

This wave is propagated until an end time,

$$T = \frac{L}{c},$$

where $c = 1$ is the exact wave speed. At this time the analytical solution coincides with the initial condition.

We consider this problem for a single grid size, $H = 0.1$, but for a number of different wave numbers, k, corresponding to wave lengths that fit into an interval of length L:

$$k = \frac{2\pi}{L}m, \quad m \in \{1, 2, \ldots\}. \tag{6.8}$$

For convenience we shall present the results with respect to the normalized wave number, $\xi = Hk$, which is a problem independent quantity. Less than 10 points per wavelength (poor resolution of waves) corresponds to $\xi > 2\pi/10$, and the highest frequency that could exist on the grid corresponds to $\xi = \pi$.

We shall consider both the cases when the stabilization is present in the weak formulation (2.5), and when it has been removed. Furthermore, effects are evaluated along two different straight lines, $\mathcal{C}_1$ and $\mathcal{C}_2$, as illustrated in Fig. 6.19. The line $\mathcal{C}_2$ is close to the boundary, where we expect that the stabilization has a large effect on the waves, and $\mathcal{C}_1$ goes through the center of the domain, where we expect that the stabilization has a minor effect.

Due to the error in the wave speed the numerical solution has a shift between the initial condition and the solution at time T. This can be seen by evaluating the solution along one of the lines $\mathcal{C}_i$, as illustrated in Fig. 6.20. This shift is used to calculate the numerical wave speed along the lines $\mathcal{C}_i$. The shift itself on $\mathcal{C}_i$ is computed by extending the solution periodically in the x-direction, and comparing the discrete Fourier transform of the numerical solution at time $t = 0$, and at $t = T$.

The numerical wave speeds are shown in Fig. 6.21. We see that as the wave number increases the numerical wave speed, c_ξ, becomes increasingly worse, as expected. The speed decreases in a similar fashion along both lines, both with and without stabilization. As expected the wave speed along $\mathcal{C}_1$ is essentially unaffected by the stabilization. Quite surprisingly the added stabilization seems to have a positive effect on the errors along the line $\mathcal{C}_2$. The speed is even more accurate than along $\mathcal{C}_1$.

A few snapshots of the numerical solution when solving the problem for an extended period of time are shown in Fig. 6.22. We see that at time $t = 0$, the waves have an even amplitude in the y-direction, but at time $t = 7.96$, this is no longer true. The amplitude of the waves has decreased close to the boundary. However, at the later time, $t = 14.96$, the amplitudes are restored to being almost even in the y-direction. Thus, the shape of waves with respect to the y-direction seems to oscillate in time.

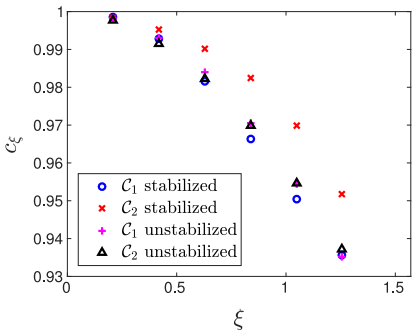

Fig. 6.21. Numerical wave speed, c_ξ, depending on normalized wave number.

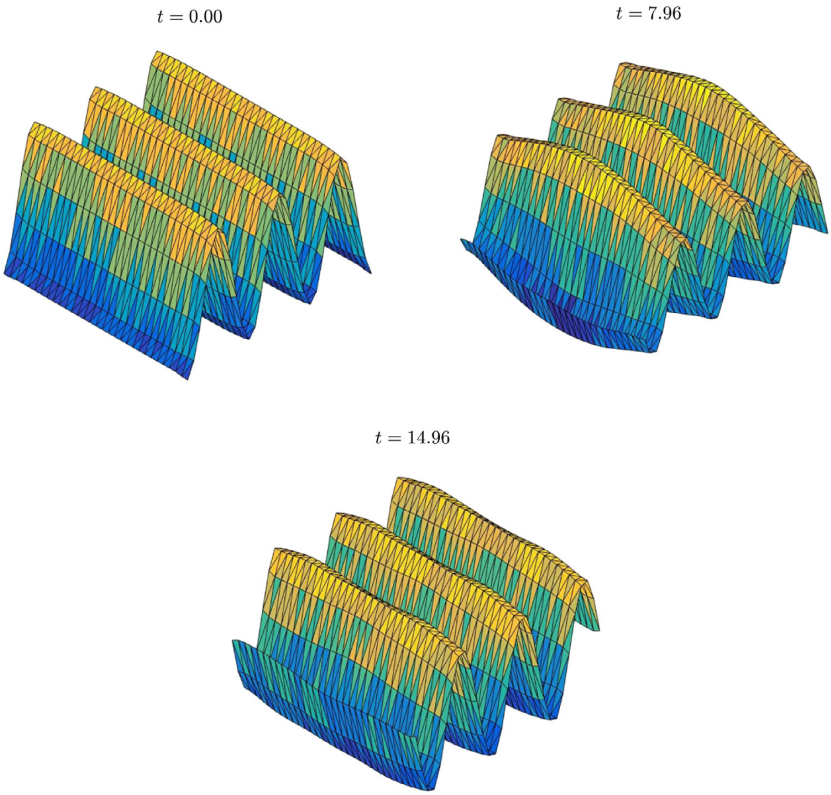

Fig. 6.22. Snapshots of the solution with stabilization, when running for an extended period of time.

To investigate how much this oscillatory behavior is due to the stabilization and how it depends on ξ, we use the root-mean-square

$$A = \sqrt{\int_{\mathcal{C}_i} u^2 dx},$$

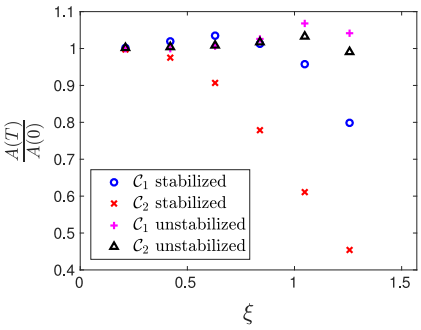

Fig. 6.23. Relative change in root mean square amplitude (5.2), depending on normalized wave number.

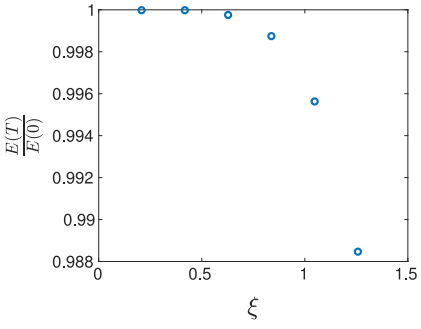

Fig. 6.24. Relative drop in discrete energy, depending on normalized wave number.

as a measure of the amplitudes along the two curves, $\mathcal{C}_i$. The reason is that given the initial conditions (6.7), with wave numbers of the form (6.8), this is analytically constant independently of the wave number:

$$A = \frac{1}{\sqrt{2}}.$$

The relative change in root mean square amplitude between the start and end time is shown in Fig. 6.23. For the unstabilized case we see that the change in amplitude is virtually nonexistent along both curves. However, when using stabilization, the amplitude at time T, has decreased quite severely along the line $\mathcal{C}_2$, and becomes increasingly worse with ξ.

One could think that these changes in amplitude are contradicting the conservation of discrete energy from Section 5. With this in mind, the ratio of discrete energy between the initial solution and the solution at time T is shown in Fig. 6.24, for the stabilized problem. We see that the energy behaves qualitatively in the same way as the dispersion errors. However, as ξ increases the drop in energy is very small. So that the energy conservation is not particularly dependent on the normalized wave number.

7. Discussion and outlook

The results in Sections 6.1 and 6.2 clearly indicate that the method converges according to the estimates given in Section 3. Furthermore, the results in Section 6.3 demonstrate that if explicit time-stepping shall be used, some form of stabilization is necessary. Moreover, Fig. 6.16 shows that adding the given stabilization term to the mass matrix makes explicit time-stepping possible.

One disadvantage with the method is the tunable parameters γ_A, γ_D and γ_M. The choice of parameters γ_A and γ_D was discussed in [3]. Regarding γ_M, Fig. 6.17 suggests that it is possible to choose any value of γ_M which is just on

the right side of the minimum, for example $\gamma_M = 0.25$. Moreover, the experiments in Section 6.4 suggest that the choice of the parameter γ_M is not particularly sensitive to the shape of the domain.

The experiment in Section 6.5 shows that the added stabilization adds some effects close to the boundary. The numerical wave speed is increased compared to other parts of the domain. Even if the discrete energy is essentially conserved, the experiment shows that the stabilization can introduce oscillations in the wave amplitudes. The slight decrease in discrete energy that is seen in Fig. 6.24 is likely due to dissipation introduced by the time-stepping. Energy is after all only conserved in the semi-discrete setting.

Explicit time-stepping is often combined with so called mass-lumping (see for example [20]). In the standard setting this gives a diagonal mass matrix, which significantly improves efficiency. Mass-lumping has not been addressed in the previous sections. The cut-elements and additional stabilization make mass lumping non-trivial near the boundary. However, one can imagine that it is possible to apply mass lumping away from the boundary. This would create a 2-by-2 block system, with one big diagonal block, which would be cheap to solve.

We believe that it would be possible to extend the method to higher order (triangular or quadrilateral) Lagrange elements. An immediate way of extending the stabilization would then be to include jumps also in higher order derivatives in the stabilization term, as suggested in [4,11]. Although this is straightforward theoretically, there may be issues concerning how the conditioning of the mass and stiffness matrices depends on the polynomial order of the elements.

However, for higher orders the straight-line approximation of the boundary (Fig. 6.1) is not sufficient. One needs to take curved boundaries into account. Performing this integration in an efficient way is a practical obstacle for higher order convergence, although there has been some recent work in this area (see e.g. [21] and [22]).

If one would like to extend the method to other problems there are several directions to go in. One direction would be to extend the current formulation to the elastic wave equation, another to consider waves coming in towards a material interface, and a third to consider boundaries or interfaces moving with time.

Acknowledgment

This research was supported by the Swedish Research Council Grant No. 2014-6088.

References

[1] S. Fernández-Méndez, A. Huerta, Imposing essential boundary conditions in mesh-free methods, Comput. Methods Appl. Mech. Engrg. (ISSN: 00457825) 193 (2004) 1257–1275. URL http://dx.doi.org/10.1016/j.cma.2003.12.019.

[2] J. Nitsche, Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind, Abh. Math. Sem. Univ. Hamburg 36 (1) (1971) 9–15. URL http://dx.doi.org/10.1007/BF02995904.

[3] E. Burman, P. Hansbo, Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method, Appl. Numer. Math. (ISSN: 01689274) 62 (4) (2012) 328–341. URL http://dx.doi.org/10.1016/j.apnum.2011.01.008.

[4] A. Massing, M.G. Larson, A. Logg, M.E. Rognes, A stabilized Nitsche fictitious domain method for the Stokes problem, J. Sci. Comput. (ISSN: 08857474) (2014) 1–28. URL http://dx.doi.org/10.1007/s10915-014-9838-9.

[5] E. Burman, P. Hansbo, Fictitious domain methods using cut elements III: A stabilized Nitsche method for the Stokes problem, Mech. Eng. (ISSN: 01689274) 62 (2011) 328–341. URL http://dx.doi.org/10.1016/j.apnum.2011.01.008.

[6] P. Hansbo, M.G. Larson, S. Zahedi, A cut finite element method for a Stokes interface problem, Applied Numerical Mathematics (ISSN: 01689274) 85 (2014) 90–114. URL http://dx.doi.org/10.1016/j.apnum.2014.06.009.

[7] A. Hansbo, P. Hansbo, An unfitted finite element method, based on Nitsche's method, for elliptic interface problems, Comput. Methods Appl. Mech. Engrg. (ISSN: 00457825) 191 (2002) 5537–5552. URL http://dx.doi.org/10.1016/S0045-7825(02)00524-8.

[8] E. Wadbro, S. Zahedi, G. Kreiss, M. Berggren, A uniformly well-conditioned, unfitted Nitsche method for interface problems, BIT (ISSN: 00063835) 53 (2013) 791–820. URL http://dx.doi.org/10.1007/s10543-012-0417-x.

[9] A. Reusken, Analysis of an extended pressure finite element space for two-phase incompressible flows, Comput. Vis. Sci. 11 (4–6) (2008) 293–305. (ISSN: 1432-9360, 1433-0369) URL http://dx.doi.org/10.1007/s00791-008-0099-8.

[10] E. Burman, M.A. Fernández, An unfitted Nitsche method for incompressible fluid–structure interaction using overlapping meshes, Comput. Methods Appl. Mech. Engrg. (ISSN: 00457825) 279 (2014) 497–514. URL http://dx.doi.org/10.1016/j.cma.2014.07.007.

[11] E. Burman, Ghost penalty, C. R. Math. (ISSN: 1631073X) 348 (21–22) (2010) 1217–1220. URL http://dx.doi.org/10.1016/j.crma.2010.10.006.

[12] A. Massing, M.G. Larson, A. Logg, Efficient implementation of finite element methods on nonmatching and overlapping meshes in three dimensions, SIAM J. Sci. Comput. 35 (1). URL http://dx.doi.org/10.1137/11085949X.

[13] V. Larsson, S., Thomee, Partial Differential Equations with Numerical Methods, Springer, ISBN: 9783540017721, 2003.

[14] M.G. Larson, F. Bengzon, The Finite Element Method: Theory, Implementation, and Applications, Sprinter, ISBN: 9783642332869, 2013, URL http://dx.doi.org/10.1007/978-3-642-33287-6.

[15] A. Ern, J.-L. Guermond, Theory and Practice of Finite Elements, vol. 159, Springer, ISBN: 0-387-20574-8, 2004, URL http://dx.doi.org/10.1007/978-1-4757-4355-5.

[16] J.C. Gilbert, P. Joly, Higher order time stepping for second order hyperbolic problems and optimal CFL conditions, in: R. Glowinski, P. Neittaanmäki (Eds.), Partial Differential Equations: Modeling and Numerical Simulation, Springer, Netherlands, Dordrecht, ISBN: 978-1-4020-8758-5, 2008, pp. 67–93. URL http://dx.doi.org/10.1007/978-1-4020-8758-5_4.

[17] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I: Nonstiff Problems, second rev. ed, in: Springer Series in Computational Mathematics, vol. 8, Springer, Heidelberg, London, 2009, ISBN: 978-3-540-56670-0 978-3-642-05163-0 978-3-540-78862-1.

[18] E. Hairer, G. Wanner, Solving ordinary differential equations II, in: Springer Series in Computational Mathematics, vol. 14, Springer, Berlin, Heidelberg, 1996, ISBN: 978-3-642-05220-0 978-3-642-05221-7. URL http://link.springer.com/10.1007/978-3-642-05221-7.

[19] K.F. Graff, Wave Motion in Elastic Solids, in: Dover Books on Engineering Series, Dover Publications, ISBN: 9780486667454, 1975.

[20] T.J.R. Hughes, Finite Element Method — Linear Static and Dynamic Finite Element Analysis, Dover Publications, ISBN: 978-0-486-41181-1, 2000.

[21] R.I. Saye, High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles, SIAM J. Sci. Comput. (ISSN: 1064-8275) 37 (2) (2015) A993–A1019. URL http://dx.doi.org/10.1137/140966290.

[22] C. Lehrenfeld, High order unfitted finite element methods on level set domains using isoparametric mappings, Comput. Methods Appl. Mech. Engrg. (ISSN: 00457825) 300 (2016) 716–733. URL http://dx.doi.org/10.1016/j.cma.2015.12.005.

Paper II

Higher Order Cut Elements for the Wave Equation

Simon Sticko[a,*], Gunilla Kreiss[a]

[a] *Division of Scientific Computing, Department of Information Technology, Uppsala University, Sweden*

Abstract

The scalar wave equation is solved using higher order immersed finite elements. We demonstrate that higher order convergence can be obtained. Small cuts with the background mesh are stabilized by adding penalty terms to the weak formulation. This ensures that the condition numbers of the mass and stiffness matrix are independent of how the boundary cuts the mesh. The penalties consist of jumps in higher order derivatives integrated over the interior faces of the elements cut by the boundary. The dependence on the polynomial degree of the condition number of the stabilized mass matrix is estimated. We conclude that the condition number grows extremely fast when increasing the polynomial degree of the finite element space.

Keywords: Cut Elements, Stabilization, Fictitious, Immersed, XFEM

1. Introduction

Cut elements [1] is an immersed finite element method. For a domain immersed in a background mesh, one solves for the degrees of freedom of the smallest set of elements covering the domain. The inner products in the weak form are taken over the immersed domain. That is, on each element one integrates over the part of the element that is inside the domain. As a result of this, some elements will have a very small intersection with the immersed domain. This will make some eigenvalues of the discrete system very small and in turn, result in poorly conditioned matrices. A suggested way to remedy this is by adding stabilizing terms to the weak formulation. A jump-stabilization was suggested by Burman and Hansbo [2] for the case of piecewise linear elements, where the jump in the normal derivative is integrated over the faces of the elements intersected by the boundary. This form of stabilization has been used with good results in several recent papers, see for example [1, 3, 4, 5], and has also been used for PDEs posed on surfaces in [6, 7].

*Corresponding Author

Email addresses: simon.sticko@it.uu.se (Simon Sticko), gunilla.kreiss@it.uu.se (Gunilla Kreiss)

Thus, a lot of attention has been directed to the use of lower order elements. Higher order cut elements have received less attention so far. These are interesting in wave propagation problems since for this type of problems the amount of work per dispersion error typically increases slower for higher order methods. Massing et al. [4] suggested stabilizing higher order elements by integrating also jumps in higher derivatives over the faces. This is the intuitive higher order generalization of the stabilization first suggested in [2] and was also mentioned as a possibility by Burman [8].

In this paper, we consider solving the scalar wave equation using higher order cut elements. Both the mass and stiffness matrix are stabilized using the higher order jump-stabilization. We present numerical results showing that the method results in higher order convergence. The time-step restriction of the resulting system is computed numerically and is concluded to be of the same size as for standard finite elements with aligned boundaries. Furthermore, we estimate how the condition number of the stabilized mass matrix depend on the polynomial degree of the basis functions. The estimate suggests that the condition number grows extremely fast with respect to the polynomial degree, which is supported by the numerical experiments. As a remedy for this behavior, we consider lowering the order of the elements close to the boundary. This results in a better condition number, but also in at least half an order lower convergence compared to having elements of full order everywhere. All numerical experiments are performed in 2D, but the generalization to three dimensions is immediate.

One reason why the considered stabilization is attractive is because it is quite easy to implement. Integrals over internal faces occur also in discontinuous Galerkin methods, thus making the implementation similar to what is already supported in many existing libraries.

The suggested jump-stabilization is one but not the only possibility for stabilizing an immersed method. Several papers have used preconditioners to try to overcome problems with ill-conditioning, such as [9, 10, 11]. In [12] a higher order discontinuous Galerkin method was suggested, where the problem of ill-conditioning was solved by merging some of the intersected elements with neighboring elements. One approach which was suggested in [8] is to stabilize the method by making L_2-projections onto local patches consisting of a small number of elements. Another possibility is to use additional streamline diffusion stabilization on the elements intersected by the boundary, as was done in [13].

This paper is organized in the following way. Notation and some basic problem setup is explained in Section 2.1, the stabilized weak formulation is described in Section 2.2, and the stability of the method is discussed in Section 2.3. Analysis of how fast the condition number increases when increasing the polynomial degree is presented in Section 3, and numerical experiments is presented in Section 4.

2. Theory

2.1. Notation and Setting

Consider the wave equation

$$\begin{aligned} \ddot{u} &= \nabla^2 u + f(x,t) && x \in \Omega, \quad t \in [0, t_f], \\ u &= g_D(x,t) && x \in \Gamma_D, \quad t \in [0, t_f], \\ \frac{\partial u}{\partial n} &= g_N(x,t) && x \in \Gamma_N, \quad t \in [0, t_f], \\ u &= u_0(x) && x \in \Omega, \quad t = 0, \\ \dot{u} &= v_0(x) && x \in \Omega, \quad t = 0, \end{aligned} \tag{2.1}$$

posed on a given domain Ω, with $\Gamma_D \cup \Gamma_N = \partial\Omega$. Let $\Omega \subset \mathbb{R}^d$ be immersed in a triangulation, $\mathcal{T}$, as in Figure 2.1. We assume that each element $T \in \mathcal{T}$ has some part which is inside Ω, that is: $T \cap \Omega \neq \emptyset$. Furthermore, let $\Omega_{\mathcal{T}}$ be the domain that corresponds to $\mathcal{T}$, that is

$$\Omega_{\mathcal{T}} = \bigcup_{T \in \mathcal{T}} T.$$

Let $\mathcal{T}_\Gamma$ denote the set of elements that are intersected by $\partial\Omega$:

$$\mathcal{T}_\Gamma = \{T \in \mathcal{T} : T \cap \partial\Omega \neq \emptyset\},$$

as in Figure 2.2. Let $\mathcal{F}_\Gamma$ denote the faces seen in Figure 2.3. That is, the faces of the elements in $\mathcal{T}_\Gamma$, excluding the faces that make up $\partial\Omega_{\mathcal{T}}$. To be precise, $\mathcal{F}_\Gamma$ is defined as

$$\mathcal{F}_\Gamma = \{F = T_1 \cap T_2 \ : \ T_1 \in \mathcal{T}_\Gamma \ \text{ or } \ T_2 \in \mathcal{T}_\Gamma, \quad T_1, T_2 \in \mathcal{T}\}.$$

We assume that our background mesh is sufficiently fine, so that the immersed geometry is well resolved by the mesh. Furthermore, we shall restrict ourselves to meshes as the one in Figure 2.1, where we have a mesh consisting of hypercubes and our coordinate axes are aligned with the mesh faces. That is, the face normals have a nonzero component only in one of the coordinate directions. Denote the distance between two grid points by h.

Consider the situation in Figure 2.4, where two neighboring elements, T_1 and T_2, are sharing a common face F. Denote by $\partial_n^k v$ the kth directional derivative in the direction of the face normal. That is, fix $j \in \{1, \ldots, d\}$ and let the normal of the face, n, be such that

$$n_i = \begin{cases} 1 & i = j \\ 0 & i \neq j, \end{cases} \tag{2.2}$$

then define

$$\partial_n^k v = \frac{\partial^k v}{\partial x_j^k}.$$

In the following we denote by $(\cdot,\cdot)_X$ and $\langle\cdot,\cdot\rangle_Y$ the L_2 scalar products taken over the d and $d-1$ dimensional domains $X \subset \mathbb{R}^d$ and $Y \subset \mathbb{R}^{d-1}$. Let $\|\cdot\|_Z$ denote the corresponding norm, and let $|\cdot|_{H^s(Z)}$ denote the H^s-semi-norm. By $[v]$ we shall denote a jump over a face, F, that is: $[v] = v|_{F_+} - v|_{F_-}$.

We shall assume that our basis functions are tensor products of one-dimensional polynomials of order p. In particular, we shall use Lagrange elements with Gauss-Lobatto nodes, in the following referred to as Q_p-elements, $p \in \{1, 2, \ldots\}$. Let V_h^p denote a continuous finite element space, consisting of Q_p-elements on the mesh $\mathcal{T}$:

$$V_h^p = \left\{ v \in C^0(\Omega_{\mathcal{T}}) : v|_T \in Q_p(T) \right\}. \tag{2.3}$$

Define also the following semi-norm

$$|v|_\star^2 = \|\nabla v\|_{\Omega_{\mathcal{T}}}^2 + \frac{1}{h}\|v\|_{\Gamma_D}^2,$$

which is a norm on V_h^p in the case that $\Gamma_D \neq \emptyset$.

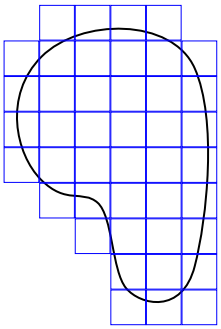

Figure 2.1: Ω immersed in a mesh $\mathcal{T}$, covering $\Omega_{\mathcal{T}}$.

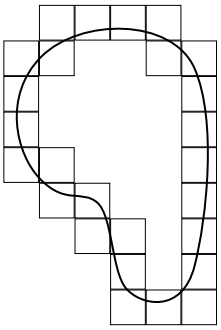

Figure 2.2: Intersected elements $\mathcal{T}_\Gamma$.

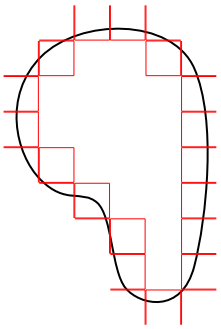

Figure 2.3: Faces $\mathcal{F}_\Gamma$.

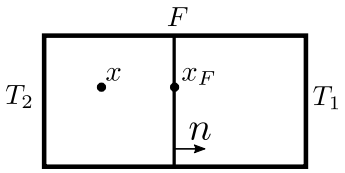

Figure 2.4: Two elements sharing a common face.

2.2. The Stabilized Weak Formulation

Multiplying (2.1) by a test-function, integrating by parts, and applying boundary conditions by Nitsche's method [14] leads to a weak formulation of the following form: find u_h such that for each fix $t \in (0, t_f]$, $u_h \in V_h^p$ and

$$(\ddot{u}_h, v)_\Omega + a(u_h, v) = L(v), \quad \forall v \in V_h^p, \tag{2.4}$$

where

$$a(u_h, v) = (\nabla u_h, \nabla v)_\Omega - \left\langle \frac{\partial u_h}{\partial n}, v \right\rangle_{\Gamma_D} - \left\langle u_h, \frac{\partial v}{\partial n} \right\rangle_{\Gamma_D} + \frac{\gamma_D}{h} \langle u_h, v \rangle_{\Gamma_D},$$

$$L(v) = (f, v)_\Omega + \left\langle g_D, \frac{\gamma_D}{h} v - \frac{\partial v}{\partial n} \right\rangle_{\Gamma_D} + \langle g_N, v \rangle_{\Gamma_N}.$$

What makes this different from standard finite elements is that the integration on each element needs to be adapted to the part of the element that is inside the domain. As illustrated in Figure 2.5, some elements will have a very small intersection with the domain. Consider the mass-matrix in (2.4):

$$\tilde{\mathcal{M}}_{ij} = (\phi_i, \phi_j)_\Omega.$$

Note that its smallest eigenvalue is smaller than each diagonal entry:

$$\lambda_{\min} = \min_{z \in \mathbb{R}^N \,:\, z \neq 0} \frac{z^T \tilde{\mathcal{M}} z}{z^T z} \leq \tilde{\mathcal{M}}_{ii}, \quad i = 1, \ldots, N. \tag{2.5}$$

Depending on the size of the cut with the background mesh some diagonal entries can become arbitrarily close to zero. Thus, both the mass and stiffness matrix can now be arbitrarily ill-conditioned depending on how the cut occurs.

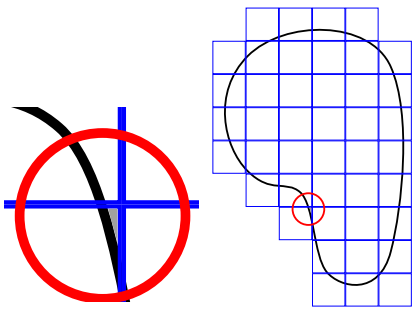

Figure 2.5: An element having a small intersection (in gray) with the domain.

One way to try to remedy this is by adding stabilizing terms, j, to the two bilinear forms

$$\begin{aligned} M(\ddot{u}_h, v) &= (\ddot{u}_h, v)_\Omega + \gamma_M j(\ddot{u}_h, v), \\ A(u_h, v) &= a(u_h, v) + \gamma_A h^{-2} j(u_h, v), \end{aligned} \tag{2.6}$$

where $\gamma_M, \gamma_A > 0$ are penalty parameters. This gives us the following weak formulation: find u_h such that for each fix $t \in (0, t_f]$, $u_h \in V_h^p$ and

$$M(\ddot{u}_h, v) + A(u_h, v) = L(v), \quad \forall v \in V_h^p. \tag{2.7}$$

In [4] a stabilization term of the following form was suggested

$$j(u,v) = \sum_{F\in\mathcal{F}_\Gamma}\sum_{k=1}^{p} h^{2k+1}\left\langle [\partial_n^k u],[\partial_n^k v]\right\rangle_F, \tag{2.8}$$

which in some sense is the intuitive extension of the stabilization which was suggested in [2]. The stabilization in (2.8) was also briefly mentioned as a possibility in [8]. As was discussed in [4] the bilinear form (2.6) can be shown to define a scalar product which is norm equivalent to the L_2-norm on the whole background mesh:

$$C_L\|v\|^2_{\Omega_\mathcal{T}} \leq M(v,v) \leq C_U\|v\|^2_{\Omega_\mathcal{T}}, \quad \forall v\in V_h^p, \tag{2.9}$$

and a corresponding equivalence also holds for the gradient

$$\tilde{C}_L\|\nabla v\|^2_{\Omega_\mathcal{T}} \leq \|\nabla v\|^2_\Omega + \gamma_A h^{-2} j(v,v) \leq \tilde{C}_U\|\nabla v\|^2_{\Omega_\mathcal{T}}, \quad \forall v\in V_h^p. \tag{2.10}$$

The constants C_L and C_U in (2.9) depend on the polynomial degree of our basis functions, but not on how the boundary cuts through the mesh. Let $\mathcal{M}$ denote the mass matrix with respect to the bilinear form M, and $\mathcal{M}_\mathcal{T}$ with respect to the scalar product on the background mesh, that is:

$$\begin{aligned}\mathcal{M}_{ij} &= M(\phi_i,\phi_j),\\ (\mathcal{M}_\mathcal{T})_{ij} &= (\phi_i,\phi_j)_{\Omega_\mathcal{T}}.\end{aligned}$$

Now, (2.9) implies that the condition number, $\kappa(\mathcal{M})$, of $\mathcal{M}$ is bounded by the condition number of $\mathcal{M}_\mathcal{T}$:

$$\kappa(\mathcal{M}) \leq \frac{C_U}{C_L}\kappa(\mathcal{M}_\mathcal{T}). \tag{2.11}$$

Starting from (2.10) one can also show that $A(\cdot,\cdot)$ is coercive in V_h^p with respect to the $|\cdot|_\star$-semi-norm on the background mesh:

$$\exists C_c>0: \quad C_c|v|^2_\star \leq A(v,v), \quad \forall v\in V_h^p. \tag{2.12}$$

This follows by the same procedure as in [4], given the following inverse inequality

$$h^{1/2}\left\|\frac{\partial v}{\partial n}\right\|_{\Gamma\cap T} \leq Cp^2\|\nabla v\|_T, \quad \forall v\in V_h^p, \tag{2.13}$$

which we derive in Lemma 7 in AppendixA.[1]

The stabilization in (2.8) is the basic form of stabilization that we shall consider. However, each time we differentiate we will introduce some dependence on the polynomial degree. It therefore seems reasonable that each term in the

[1] This was also shown for piecewise linear basis functions in the proof of Lemma 4 in [15].

sum should be scaled in some way. Because of this, we consider a stabilization of the following form:

$$j(u,v) = \sum_{F\in\mathcal{F}_\Gamma}\sum_{k=1}^{p} w_k \frac{h^{2k+1}}{(2k+1)(k!)^2}\left\langle [\partial_n^k u], [\partial_n^k v]\right\rangle_F, \tag{2.14}$$

where $w_j \in \mathbb{R}^+$ are some weights, which we are free to choose as we wish. The choice of weights will determine how large our constants C_U, C_L in (2.11) are and in turn influence how well conditioned our system is.

2.3. Stability

The bilinear forms in (2.7) are symmetric, which is a quite important property, since it in the end will guarantee stability of the system. In order to show stability we want a bound on $\|u\|_{\Omega_\mathcal{T}}$. Define an energy, E, of the form

$$E(t) := \frac{1}{2}\left(M(\dot{u}_h,\dot{u}_h) + A(u_h,u_h)\right). \tag{2.15}$$

Since both bilinear forms are at least positive semi-definite, this energy has the property $E \geq 0$. The symmetry now allows us to show that for a homogeneous system,

$$f(x) = 0, \quad g_D(x) = 0, \quad g_N(x) = 0,$$

the energy is conserved:

$$\frac{\mathrm{d}E}{\mathrm{d}t} = M(\ddot{u}_h,\dot{u}_h) + A(u_h,\dot{u}_h) \overset{(2.7)}{=} 0, \tag{2.16}$$

so that

$$E(t) = E(0). \tag{2.17}$$

By the definition of the energy together with (2.9) and (2.12) this immediately implies that $\|\dot{u}_h\|_{\Omega_\mathcal{T}}$ and $\|\nabla u_h\|_{\Omega_\mathcal{T}}$ are both bounded. For the case $\Gamma_D \neq \emptyset$ the semi-norm $|\cdot|_\star$ is a norm for the space V_h^p and (2.12) implies that $\|u_h\|_{\Omega_\mathcal{T}}$ is also bounded. When $\Gamma_D = \emptyset$ we can use that

$$2\|u_h\|_{\Omega_\mathcal{T}}\frac{\mathrm{d}}{\mathrm{d}t}\|u_h\|_{\Omega_\mathcal{T}} = \frac{\mathrm{d}}{\mathrm{d}t}\|u_h\|^2_{\Omega_\mathcal{T}} = 2(u_h,\dot{u}_h)_{\Omega_\mathcal{T}} \leq 2\|u_h\|_{\Omega_\mathcal{T}}\|\dot{u}_h\|_{\Omega_\mathcal{T}},$$

which gives us

$$\frac{\mathrm{d}}{\mathrm{d}t}\|u_h\|_{\Omega_\mathcal{T}} \leq \|\dot{u}_h\|_{\Omega_\mathcal{T}}.$$

By integrating we obtain that $\|u_h\|_{\Omega_\mathcal{T}}$ is bounded since $\|\dot{u}_h\|_{\Omega_\mathcal{T}}$ is bounded:

$$\|u_h(t)\|_{\Omega_\mathcal{T}} \leq \|u_h(0)\|_{\Omega_\mathcal{T}} + \int_0^{t_f}\|\dot{u}_h\|_{\Omega_\mathcal{T}}dt.$$

Thus the system is stable.

In total the system (2.7) discretizes to a system of the form

$$\mathcal{M}\frac{\mathrm{d}^2\xi}{\mathrm{d}t^2} + \mathcal{A}\xi = \mathcal{F}(t), \tag{2.18}$$

with $\mathcal{M}, \mathcal{A} \in \mathbb{R}^{N\times N}$, and $\xi \in \mathbb{R}^N$, and where

$$\mathcal{A}_{ij} = A(\phi_i, \phi_j).$$

When solving this system in time we will have a restriction on the time-step, τ, of the form

$$\tau \leq \alpha C_{FL} h,$$

where α is a constant which depends on the time-stepping algorithm. The constant C_{FL} is determined as

$$C_{FL} = \frac{h^{-1}}{\sqrt{\lambda_{\max}}}, \tag{2.19}$$

where $\lambda_{\max}$ is the largest eigenvalue of the generalized eigenvalue problem: find (x, λ) such that

$$\mathcal{A}x - \lambda\mathcal{M}x = 0, \quad x \in \mathbb{R}^N.$$

Since we would expect that the added stabilization has some effect on the C_{FL}-constant we will investigate it numerically in Section 4. It turns out that it is not worse than for the standard case when the boundaries are aligned with the mesh.

3. Analysis of the Condition Number of the Mass Matrix

We would like to choose the weights in (2.14) in order to minimize the condition number of the mass matrix. This will require us to know how the condition number depends on the weights and the polynomial degree. To determine this, we follow essentially the same path as in [4] and keep track of the weights and the polynomial dependence of the involved inequalities. In the following, we denote by C various constants which do not depend on h and p, unless explicitly stated otherwise. We shall also by w denote the vector $w = (w_1, \ldots, w_p)$, where w_j are the weights in the stabilization term (2.14). We can now derive the following inequality, which is a weighted version of Lemma 5.1 in [4].

Lemma 1. *Given two neighboring elements, T_1 and T_2, sharing a face F (as in Figure 2.4), and $v \in V_h^p$, we have that:*

$$\|v\|_{T_1}^2 \leq L(w)\left(\|v\|_{T_2}^2 + \sum_{k=1}^{p} w_k \frac{h^{2k+1}}{(2k+1)(k!)^2}\|[\partial_n^k v]\|_F^2\right), \tag{3.1}$$

where

$$L(w) = C_1(p) + \sum_{k=1}^{p}\frac{1}{w_k}. \tag{3.2}$$

Proof. Denote by v_i the restriction of v to T_i and then extended by expression to the whole of $T_1 \cup T_2$. For a point $x \in T_1 \cup T_2$ denote by $x_F(x)$ the projection of x onto the face, as in Figure 2.4. We may now Taylor expand from the face:

$$v_i(x) = \sum_{k=0}^{p} \frac{1}{k!} \partial_n^k v_i(x_F(x)) |x - x_F(x)|^k,$$

and consequently

$$v_1(x) = v_2(x) + \sum_{k=1}^{p} \frac{1}{k!} [\partial_n^k v(x_F)] |x - x_F|^k. \tag{3.3}$$

Now introduce the following weighted $l^2(\mathbb{R}^{p+1})$-norm:

$$\|z\|_\alpha^2 := \sum_{k=0}^{p} \alpha_k z^2,$$

where $\alpha_k > 0$ and $z \in \mathbb{R}^{p+1}$. If $\|\cdot\|_1$ denotes the usual $l^1(\mathbb{R}^{p+1})$-norm we have that:

$$\|z\|_1^2 \leq C_\alpha \|z\|_\alpha^2, \tag{3.4}$$

where

$$C_\alpha = \sum_{k=0}^{p} \frac{1}{\alpha_k}. \tag{3.5}$$

Taking the $L_2(T_1)$-norm of (3.3) and using (3.4) now results in:

$$\|v_1\|_{T_1}^2 \leq C_\alpha \left(\alpha_0 \|v_2\|_{T_1}^2 + \sum_{k=1}^{p} \alpha_k \frac{h^{2k+1}}{(2k+1)(k!)^2} \|[\partial_n^k v]\|_F^2 \right). \tag{3.6}$$

Since v_2 lies in a finite dimensional polynomial space on $T_1 \cup T_2$ the norms on T_1 and T_2 are equivalent:

$$\|v_2\|_{T_1}^2 \leq C_1 \|v_2\|_{T_2}^2,$$

where $C_1 = C_1(p)$. Using this in (3.6) and choosing

$$\begin{aligned} \alpha_0 &= 1/C_1, \\ \alpha_k &= w_k, \quad k = 1, \ldots, p \end{aligned}$$

gives us (3.1). □

Lemma 1 will now allow us to bound the bilinear form M from below.

Lemma 2. *A lower bound for $M(v, v)$ is:*

$$\|v\|_{\Omega_\mathcal{T}}^2 \leq C_l L(w)^{N_J} M(v, v), \tag{3.7}$$

where N_J is some sufficiently large integer and C_l is a constant independent of h and p.

Proof. Let $T_0 \in \mathcal{T}_\Gamma$ and let $\{T_i\}_{i=1}^{N-1}$ (with $T_i \in \mathcal{T}_\Gamma$) be a sequence of elements that need to be crossed in order to get to an element $T_N \in \mathcal{T} \setminus \mathcal{T}_\Gamma$ as in Figure 3.1, and denote $F_i = T_{i-1} \cap T_i$. By using (3.1) we get

$$\|v\|_{T_0}^2 \leq L(w)^N \left(\|v\|_{T_N}^2 + \sum_{i=1}^{N} \sum_{k=1}^{p} w_k \frac{h^{2k+1}}{(2k+1)(k!)^2} \|[\partial_n^k v]\|_{F_i}^2 \right),$$

where we have used that $L(w) \geq 1$ (since at least $C_1 \geq 1$). Let now $N_J \geq 1$ denote some upper bound on the maximum number of jumps that needs to be made in the mesh. If our geometry is well resolved by our background mesh N_J is a small integer. This gives us

$$\|v\|_{\Omega_\mathcal{T}}^2 = \sum_{T \in \mathcal{T}_\Gamma} \|v\|_T^2 + \sum_{T \in \mathcal{T} \setminus \mathcal{T}_\Gamma} \|v\|_T^2 \leq C L(w)^{N_J} \left(\sum_{T \in \mathcal{T} \setminus \mathcal{T}_\Gamma} \|v\|_T^2 + j(v,v) \right),$$

from which (3.7) follows. □

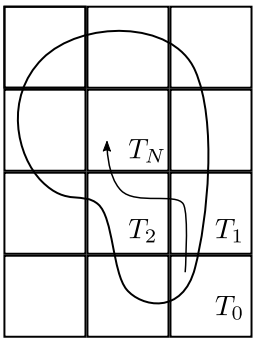

Figure 3.1: A sequence of jumps from a boundary element $T_0 \in \mathcal{T}_\Gamma$ to an inside element T_N.

We proceed by estimating how a bound on the jumps depend on the polynomial degree.

Lemma 3. *For the jumps in the normal derivative we have that:*

$$\|[\partial_n^k v]\|_F^2 \leq C_k \frac{p^{4k+2}}{h^{2k+1}} \left(\|v\|_{T_F^+}^2 + \|v\|_{T_F^-}^2 \right), \quad \text{for} \quad k = 1, 2, \ldots, p \tag{3.8}$$

where T_F^+ and T_F^- denotes the two elements sharing the face F.

Proof. Note first that

$$\|[\partial_n^k v]\|_F^2 \leq 2 \left(\|\partial_n^k v_1\|_F^2 + \|\partial_n^k v_2\|_F^2 \right). \tag{3.9}$$

We shall need the following inequalities:

$$\|v\|_F \leq C \frac{p}{\sqrt{h}} \|v\|_T, \tag{3.10}$$

$$|v|_{H^s(T)} \leq C^s \frac{p^{2s}}{h^s} \|v\|_T, \tag{3.11}$$

which were discussed[2] in [16]. Although (3.11) holds for a whole element we shall use the corresponding inequality applied to a face:

$$|v|_{H^s(F)} \leq C^s \frac{p^{2s}}{h^s} \|v\|_F. \tag{3.12}$$

This is valid since a function v in the tensor product space over T will have a restriction $v|_F$ in the tensor product space over the face F. Note that the constants, C, in (3.11) and (3.12) are not necessarily the same. By combining (3.9), (3.10) and (3.12) we obtain (3.8). □

Using Lemma 3 we can now bound the bilinear form $M(\cdot,\cdot)$ from above.

Lemma 4. *An upper bound for $M(v,v)$ is:*

$$M(v,v) \leq (1 + C_g G(w)) \|v\|^2_{\Omega_\mathcal{T}}, \tag{3.13}$$

where

$$G(w) = \sum_{k=1}^{p} w_k \frac{p^{4k+2}}{(2k+1)(k!)^2}, \tag{3.14}$$

and C_g is a constant independent of h and p.

Proof. Using the definition of $j(\cdot,\cdot)$ and applying Lemma 3 on each order of derivatives in the sum individually we have

$$j(v,v) \leq CG(w) \sum_{F \in \mathcal{F}_\Gamma} \left(\|v\|^2_{T_F^-} + \|v\|^2_{T_F^+} \right).$$

Let N_F denote the number of faces that an element has in $\mathbb{R}^d$. We now have

$$\sum_{F \in \mathcal{F}_\Gamma} \left(\|v\|^2_{T_F^+} + \|v\|^2_{T_F^-} \right) \leq 2N_F \sum_{T \in \mathcal{T}} \|v\|^2_{T_F} \leq 2N_F \|v\|^2_{\Omega_\mathcal{T}}, \tag{3.15}$$

so we finally obtain:

$$j(v,v) \leq C_g G(w) \|v\|^2_{\Omega_\mathcal{T}},$$

which gives us (3.13). □

Using Lemma 2 and 4 we now have the following bound on the condition number.

[2] In particular see (4.6.4) and (4.6.5) in Theorem 4.76, together with the argumentation leading to Corollary 3.94

Lemma 5. *An upper bound for the condition number of the mass matrix is*

$$\kappa(\mathcal{M}) \leq C_M(w)\kappa(\mathcal{M}^*), \tag{3.16}$$

where

$$C_M = C_l L(w)^{N_J}(1 + C_g G(w))\kappa(\mathcal{M}^*).$$

Proof. Let $\lambda(\cdot)$ denote eigenvalues. From Lemma 2 and 4 we obtain

$$\frac{\lambda_{\min}(\mathcal{M}^*)}{C_l L(w)^{N_J}} \leq \lambda_{\min}(\mathcal{M}),$$

$$\lambda_{\max}(\mathcal{M}) \leq (1 + C_g G(w))\lambda_{\max}(\mathcal{M}^*),$$

which gives us (3.16). □

Here, we would like to choose the weights in order to minimize the constant C_M. However, we have the following unsatisfactory result, which shows that no matter how we choose the weights our analysis cannot yield a p-independent bound on the conditioning.

Lemma 6. *The constant $C_M(w)$ in Lemma 5 fulfills $C_M(w) \geq C_0 P(p)$, where C_0 does not depend on p. Here $P(p)$ is the function*

$$P(p) = \sum_{k=1}^{p} \frac{p^{4k+2}}{(k!)^2(2k+1)}, \tag{3.17}$$

which is independent of the choice of weights w.

Proof. First note that

$$C_l L(w)^{N_J}(1 + C_g G(w)) \geq C_l C_g G(w) L(w)^{N_J} \geq C_l C_g L(w) G(w).$$

Now we have

$$L(w)G(w) \geq \sum_{k=1}^{p} \left(w_k \frac{p^{4k+2}}{(2k+1)(k!)^2} \right) \sum_{k=1}^{p} \left(\frac{1}{w_k} \right) \geq$$
$$\sqrt{\sum_{k=1}^{p} \left(w_k^2 \left(\frac{p^{4k+2}}{(2k+1)(k!)^2} \right)^2 \right)} \sqrt{\sum_{k=1}^{p} \left(\frac{1}{w_k} \right)^2} \geq P(p),$$

where we first used that the $l^1(\mathbb{R}^p)$-norm is greater than the $l^2(\mathbb{R}^p)$-norm and finally Cauchy-Schwartz. From this the result follows. □

The function $P(p)$ increases incredibly fast when increasing the polynomial degree. This result could reflect either:

1. The analysis leading to Lemma 5 is not sharp. The bound C_M is too generous, and a better bound exists.

2. The bound in Lemma 5 is not unnecessarily generous, so that the constant C_M is in some sense "tight". This means that the condition number of the stabilized mass matrix (2.6) will grow faster than the function $P(p)$, regardless of the choice of weights.

Alternative 2 is rather devastating from a time-stepping perspective, since in order to time-step (2.18) an inverse of the mass matrix needs to be available in each time-step. If this inversion is done with an iterative method the number of required iterations until convergence is going to be large.

A combination of these two alternatives is, of course, possible. The estimate in Lemma 5 could be too pessimistic, but even the optimal bound increases incredibly fast. Given the results in Section 4 this appears to be the most plausible alternative.

Lowering the Order at the Boundary

As a remedy to the expected poor behavior of the condition number of the mass matrix, we shall consider lowering the order of the elements close to the boundary. This will be done in the way illustrated in Figure 3.2. This idea is based on two observations:

- In finite difference methods it is possible to lower the order close to the boundary and still get full convergence [17, 18].
- By using lower order elements close to the boundary we only need to stabilize jumps in derivatives up to order $p-1$.

Let $N_F(T)$ denote the neighboring element of the element T sharing the face F with T. We now construct a new finite element space $\tilde{V}_h^p$ in the following way. Elements which are intersected or have an intersected neighbor are lowered one order compared to the interior of the domain. More precisely:

$$\tilde{V}_h^p = \left\{ v \in C^0(\Omega_{\mathcal{T}}) : \begin{cases} v|_T \in Q_{p-1}(T), & T \in \mathcal{T}_\Gamma \text{ or } \exists F : N_F(T) \in \mathcal{T}_\Gamma \\ v|_T \in Q_p(T), & \text{Otherwise} \end{cases} \right\}. \tag{3.18}$$

This form of space will introduce hanging nodes between elements of different orders. This can be solved in several ways, but in the experiments in Section 4 we treat this by adding constraints that enforce continuity at the hanging nodes.

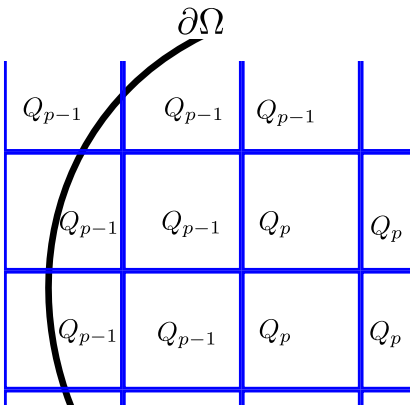

Figure 3.2: Space, $\tilde{V}_h^p$, constructed by lowering the order of the elements close to the boundary.

3.1. Choosing Weights in the Jump-Stabilization

In order to do a computation, we are forced to make some choice of the weights w_i. The essence of Lemma 6 is that we can bound $L(w)G(w)$ from below. So in order to choose weights let us assume that:

$$\kappa(\mathcal{M}) \propto L(w)G(w).$$

From Lemma 4 it is seen that choosing $w_i \gg 1$ makes $G(w)$ very large. In the same way, Lemma 2 tells us that choosing $w_i \ll 1$ for some i makes $L(w)$ very large. From this observation it seems reasonable to try to enforce both bounds to be of about the same magnitude. In this way, we minimize $L(w)G(w)$ with respect to w and enforce $G(w) = L(w)$. This leaves us with

$$\nabla_w L + \nabla_w G = 0,$$

where ∇_w denotes the gradient with respect to w. This now gives us the following choice of weights:

$$w_k = k! \frac{\sqrt{2k+1}}{p^{2k+1}}. \tag{3.19}$$

There is no reason why this argumentation should lead to the optimal choice of weights, but it seems reasonable that this is *not a particularly bad choice.*

4. Numerical Experiments

In the following, we shall consider a problem both with the finite element space V_h^p from (2.3) and with the finite element spaces $\tilde{V}_h^p$ from (3.18). The weights from (3.19) are used, with p determined by the order of the polynomials at the boundary. In addition, the following parameters are used

$$\begin{aligned}\gamma_M &= 0.25/\sqrt{3},\\ \gamma_A &= 0.5/\sqrt{3},\\ \gamma_D &= 2.5p(p+1).\end{aligned}$$

The errors are computed in norms which are some quantities integrated over the domain Ω. It is worth noting that the geometry of Ω is represented by a level set function, ψ_h. Both for the case when $u \in V_h^p$ and when $u \in \tilde{V}_h^p$ the level set function is an element in the space

$$W_h^p = \{v \in C_0(\Omega_{\mathcal{T}_B}) : v|_T = Q_p(T)\},$$

where

$$\Omega_{\mathcal{T}_B} = \bigcup_{T \in \mathcal{T}_B} T,$$

and where $\mathcal{T}_B$ is a larger background mesh from which $\mathcal{T}$ was created. In order to perform the quadratures over the elements intersected by the boundary we have used an algorithm by Saye [19], which generates the quadrature rules on the intersected elements with respect to ψ_h. Thus, also the errors of the solution are calculated with respect to this approximation of the geometry. That is, the L_2-norms are approximated as

$$\begin{aligned} \|\cdot\|_\Omega &\approx \|\cdot\|_{\psi_h<0}, \\ \|\cdot\|_{\partial\Omega} &\approx \|\cdot\|_{\psi_h=0}, \end{aligned} \quad \psi_h \in V_h^p,$$

where ψ_h is initialized by L_2-projecting the analytic level set function onto the space W_h^p. Convergence-rates are estimated as

$$\frac{\log(e_i/e_{i+1})}{\log(h_i/h_{i+1})},$$

where e_i denotes an error corresponding to mesh size h_i.

Time-stepping is performed with a classical fourth order explicit Runge-Kutta, after rewriting the system (2.18) as a first order system in time. A time step, τ, of size

$$\tau = \frac{0.9}{p^2} h$$

is used. Implementation was done in the library `deal.II` [20].

4.1. Standard Reference Problem with Aligned Boundary

It is relevant to compare some of the properties of the mass and stiffness matrix with standard (non-immersed) finite elements. For this purpose Table 1 shows the C_{FL} number, the minimal and maximal eigenvalues of the mass matrix, together with the condition number of the mass matrix, for the non-immersed case. The table also shows how these change when changing the order of the elements in the space. The values were computed on a rectangular grid with size $[-1.5, 1.5] \times [-1.5, 1.5]$, with Neumann boundary conditions, for a single grid size $h = 0.036$. As for the immersed case, quadrilateral Lagrange elements with Gauss-Lobatto nodes were used. The C_{FL}-constant was computed according to (2.19). Since all eigenvalues should be proportional to h^2, the eigenvalues have been scaled by h^{-2} for easier comparison.

Table 1: Quantities computed with (non-immersed) finite elements aligned with the boundary.

p	C_{FL}	$h^{-2}\lambda_{\min}(\tilde{\mathcal{M}}))$	$h^{-2}\lambda_{\max}(\tilde{\mathcal{M}}))$	$\kappa(\tilde{\mathcal{M}}))$
1	0.204	6.25e-02	9.99e-01	16.0
2	0.091	1.15e-02	3.36e-01	29.2
3	0.054	3.67e-03	1.74e-01	47.3

4.2. An Inner Problem

Let Ω be a disk domain, centered at origo, with radius $R = 1$, and enforce homogeneous Dirichlet boundary condition along the boundary

$$u|_{\partial\Omega} = 0.$$

Let J_0 denote the 0:th order Bessel-function and let α_n denote its n:th zero. By starting from initial conditions:

$$\begin{aligned} u|_{t=0} &= J_0(\alpha_n \frac{\|x\|}{R}), \\ \left.\frac{\partial u}{\partial t}\right|_{t=0} &= 0, \end{aligned}$$

we can calculate the error in our numerical solution with respect to the analytic solution:

$$u(x,t) = J_0(\alpha_n \frac{\|x\|}{R}) \cos(\omega_n t), \quad \omega_n = \frac{\alpha_n}{R}.$$

A few snapshots of the numerical solution are shown in Figure 4.1. The problem was solved with the given method until an end-time, t_f, corresponding to a three periods:

$$t_f = 3T_p, \quad T_p = \frac{2\pi}{\omega_n}.$$

At this end-time the errors were computed.

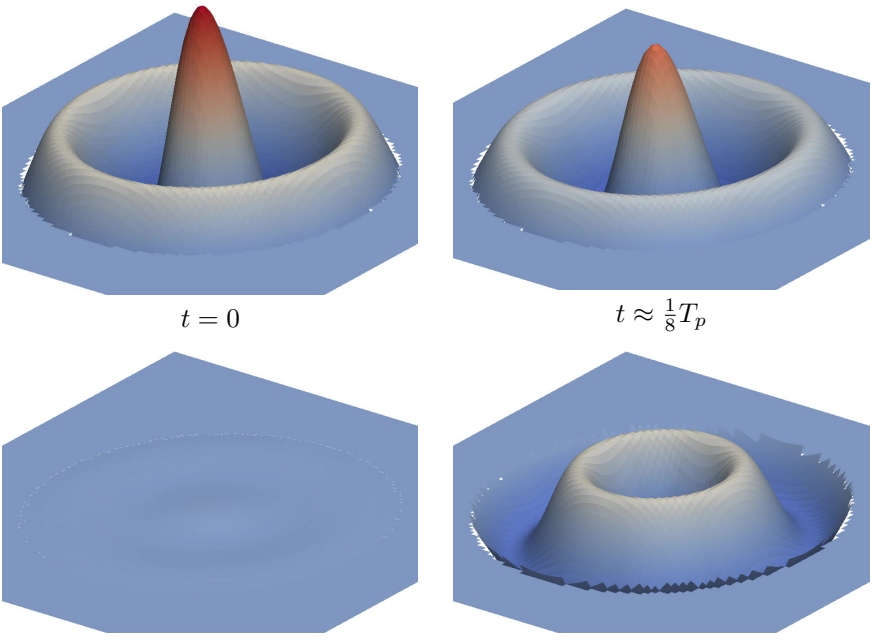

Figure 4.1: Snapshots of the vibrating membrane problem.

Results Using the Space V_h^p

The calculated errors and estimated convergence rates for the different element orders are shown in Tables 2 to 4. One would expect that the $L_2(\Omega)$- and $L_2(\partial\Omega)$-errors are proportional to h^{p+1} and that the $H^1(\Omega)$-error is proportional to h^p. The rates of the $L_2(\partial\Omega)$-error is slightly low when using Q_2-elements. Otherwise, the rates are higher than expected.

The computed C_{FL}-constant for the different orders of elements are shown in Table 5. Tabulated are the value after calculating the CFL-number according to (2.19) for each grid-level and then taking the mean over all grid sizes. The C_{FL}-constant is not worse than for the non-immersed case in Table 1.

How the condition number of the mass matrix depend on the grid size is shown in Figure 4.2, for the different orders of p. We see that the condition numbers are essentially constant when refining h, in agreement with (2.9). We also see that the condition numbers increase extremely rapidly when increasing the polynomial degree, as predicted by Lemma 6. In particular when compared to the condition numbers of the non-immersed case in Table 1. The dashed lines in Figure 4.2 denote the function $CP(p)$, where P is the function from (3.17) and C is a constant determined by fitting it to the mean (with respect to h) of the condition numbers for V_h^1. The estimate from Lemma 6 does not agree particularly well with the numerical results.

The minimal and maximal eigenvalues for the different polynomial orders and refinements are seen in Figure 4.3. As can be seen, the scaled eigenvalues are

essentially constant with respect to h. Thus the dependence on h is as expected. We see that the minimal eigenvalues decrease quite fast when increasing the polynomial degree, and that they are substantially smaller than in the non-immersed case in Table 1. At the same time, the maximal eigenvalues grow and are larger than in Table 1.

The spectra of the mass matrix corresponding to the coarsest refinement level are shown in Figure 4.4. As can be seen, there is a "kink" in the spectrum. This appears at the eigenvalue λ_i, corresponding to $i/n \approx 0.45$ (n being the number of degrees of freedom for this grid size) for V_h^3. The same behavior can be seen in the spectrum for V_h^2, but at the eigenvalue corresponding to $i/n \approx 0.8$.

Table 2: Errors when using the space V_h^1.

h	$L_2(\Omega)$		$H^1(\Omega))$		$L_2(\partial\Omega)$	
1.000e-01	7.017e-01	-	6.108e+00	-	1.383e-02	-
5.000e-02	2.288e-01	1.62	2.012e+00	1.60	4.641e-03	1.58
2.500e-02	2.503e-02	3.19	3.007e-01	2.74	7.876e-04	2.56
1.250e-02	4.641e-03	2.43	1.172e-01	1.36	8.217e-05	3.26

Table 3: Errors when using the space V_h^2.

h	$L_2(\Omega)$		$H^1(\Omega))$		$L_2(\partial\Omega)$	
1.000e-01	1.107e-02	-	2.036e-01	-	5.871e-03	-
5.000e-02	1.302e-03	3.09	3.944e-02	2.37	9.376e-04	2.65
2.500e-02	1.029e-04	3.66	9.699e-03	2.02	1.480e-04	2.66

Table 4: Errors when using the space V_h^3.

h	$L_2(\Omega)$		$H^1(\Omega))$		$L_2(\partial\Omega)$	
1.000e-01	6.104e-04	-	2.534e-02	-	4.403e-04	-
5.000e-02	2.153e-05	4.83	2.130e-03	3.57	1.725e-05	4.67
2.500e-02	4.346e-07	5.63	1.275e-04	4.06	5.547e-07	4.96

Table 5: Computed CFL-numbers.

	V_h^1	V_h^2	V_h^3
C_{FL}	0.37	0.13	0.07

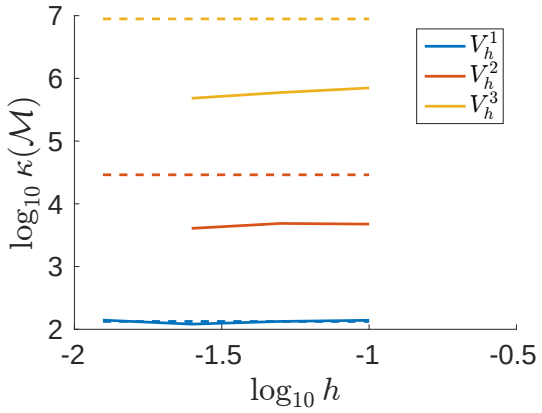

Figure 4.2: Condition number of the mass matrix when using the space V_h^p, for different h and p. The dashed lines denotes estimates according to the function $P(p)$.

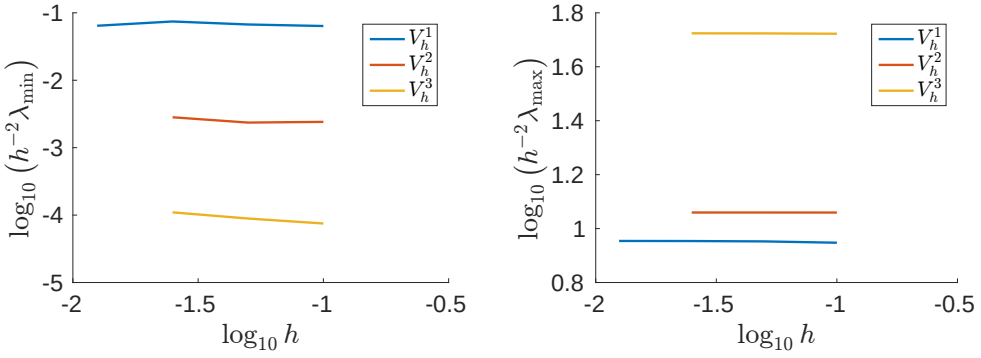

Figure 4.3: Maximal/minimal eigenvalues (scaled by h^{-2}) of the mass matrix, when using the space V_h^p, for different h and p.

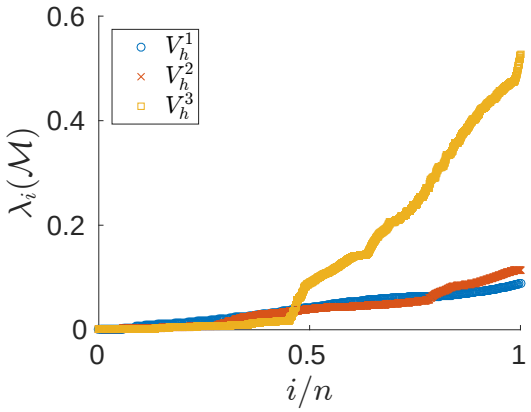

Figure 4.4: Spectra of the mass matrix when using the spaces V_h^1-V_h^3, for $h = 0.1$.

Results Using the Space $\tilde{V}_h^p$

The errors and convergence rates when using the space $\tilde{V}_h^2$ are shown in Table 6. For the errors in $L_2(\Omega)$- and $H^1(\Omega)$-norm it seems that we obtain the rate corresponding to the highest element (Q_2) in the space. Not unexpectedly we seem to get the lower order convergence for the $L_2(\Omega)$-error along the boundary. However, when using the space $\tilde{V}_h^3$ the situation appears to be different. Here, it seems that one looses at least half an order for the rates of the $L_2(\Omega)$- and $H^1(\Omega)$-errors.

How the condition number of the mass-matrix depends on h for the two spaces $\tilde{V}_h^2$ and $\tilde{V}_h^3$ are shown in Figure 4.5. By comparing to Figure 4.2 we see that the condition number of the space V_h^p is essentially the same as for the corresponding space with the lowest order element everywhere. That is:

$$\kappa\left(\mathcal{M}_{\tilde{V}_h^p}\right) \approx \kappa\left(\mathcal{M}_{V_h^{p-1}}\right),$$

which is not surprising since we expect that all ill-conditioning is due to the added penalty term, $j(\cdot,\cdot)$. The minimal and maximal eigenvalues of the mass matrix and the CFL-numbers for the space $\tilde{V}_h^p$ look essentially the same as for the space V_h^{p-1}. The mass matrix spectra for these spaces are shown in Figure 4.6, for the coarsest grid level. The spectra look similar to the spaces V_h^{p-1}.

Table 6: Errors when using the space $\tilde{V}_h^2$.

h	$L_2(\Omega)$		$H^1(\Omega))$		$L_2(\partial\Omega)$	
1.000e-01	2.528e-02	-	5.101e-01	-	6.211e-03	-
5.000e-02	5.526e-03	2.19	1.670e-01	1.61	1.467e-03	2.08
2.500e-02	4.055e-04	3.77	2.274e-02	2.88	1.953e-04	2.91
1.250e-02	4.156e-05	3.29	4.086e-03	2.48	4.144e-05	2.24

Table 7: Errors when using the space $\tilde{V}_h^3$.

h	$L_2(\Omega)$		$H^1(\Omega))$		$L_2(\partial\Omega)$	
1.000e-01	3.290e-03	-	1.283e-01	-	2.935e-03	-
5.000e-02	3.925e-04	3.07	2.570e-02	2.32	4.460e-04	2.72
2.500e-02	3.547e-05	3.47	4.766e-03	2.43	4.705e-05	3.24

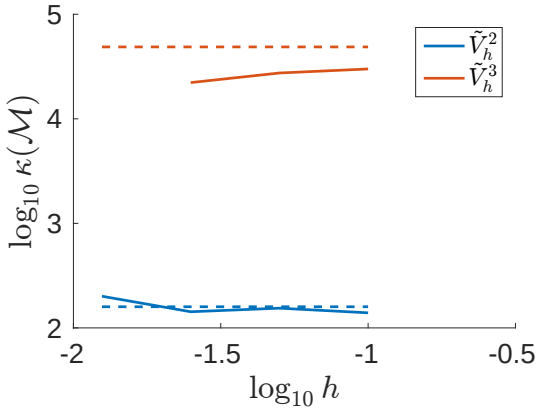

Figure 4.5: Condition number of the mass matrix when using the space $\tilde{V}_h^p$, for different h and p. The dashed lines denotes estimates according to the function $P(p)$.

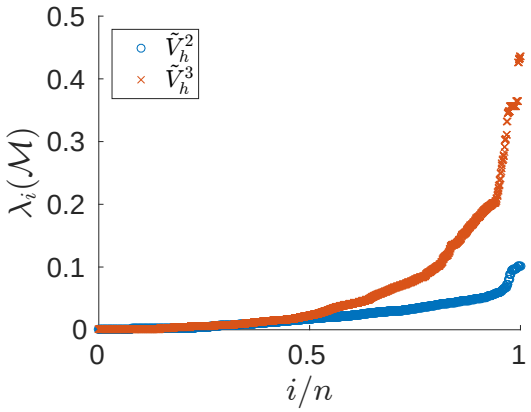

Figure 4.6: Spectra of the mass matrix when using the spaces $\tilde{V}_h^2$ and $\tilde{V}_h^3$, for $h = 0.1$.

5. Discussion

The results in Section 4.2 show that it is possible to solve the problem and obtain up to 4th order convergence. In particular, it is also promising that the CFL-condition is not stricter than for the non-immersed case. However, both the theoretical results in Lemma 6 and the results in Section 4.2 show that there are problems with the conditioning of the mass matrix. It should be emphasized that even if the added stabilization creates some new problems it is by far better than using no stabilization at all. With the added stabilization the method can be proved to be stable, which is essential.

It would, of course, be advantageous if one would be able to create a stabilization which does not lead to conditioning problems. However, the prospects for creating a good preconditioner for the mass matrix is rather good, since the stabilization maintains the symmetry of the mass matrix and since one obtains bounds on its spectrum from the analysis.

The choice of the weights in (3.19) were based on hand-waving arguments and can, therefore, be criticized. We have tried other choices of weights but have not presented the results here. This is mainly because they give similar results and we have no reason to believe that there exists a choice which makes the condition number significantly better.

The idea of lowering the order of the elements close to the boundary worked quite well for the space $\tilde{V}_h^2$. We obtained the convergence corresponding to the higher elements in the space, but the condition number corresponding to the lower order elements. Sadly this was not the case when increasing the element order further and going to the space $\tilde{V}_h^3$. Thus, the procedure does not appear to be a plausible solution for going to higher orders.

AppendixA. Derivation of (2.13)

Lemma 7. *Given that the boundary is sufficiently resolved by the mesh the following inequality holds for $v|_T \in Q_p(T)$ with $T \in \mathcal{T}_\Gamma$,*

$$h^{1/2} \left\| \frac{\partial v}{\partial n} \right\|_{\Gamma \cap T} \leq C p^2 \|\nabla v\|_T, \tag{2.13}$$

where C is a constant independent of h and p.

Proof. Let $\Gamma_T = T \cap \Gamma$. Note that

$$\left\| \frac{\partial v}{\partial n} \right\|_{\Gamma_T} \leq \|\nabla v\|_{\Gamma_T} \leq \sqrt{\int_{\Gamma_T} \nabla v \cdot \nabla v \, dV} \leq |\Gamma_T|^{1/2} \|\nabla v\|_{L^\infty(T)}. \tag{A.1}$$

Furthermore, using the following inequality from (4.6.1) in Theorem 4.76 in [16]

$$\|v\|_{L^\infty(T)} \leq C \frac{p^2}{|T|^{1/2}} \|v\|_T$$

gives us

$$\|\nabla v\|^2_{L^\infty(T)} \leq \sum_{i=1}^{d} \left\| \frac{\partial v}{\partial x_i} \right\|^2_{L^\infty(T)} \leq C \frac{p^4}{|T|} \sum_{i=1}^{d} \left\| \frac{\partial v}{\partial x_i} \right\|^2_T \leq C \frac{p^4}{|T|} \|\nabla v\|^2_T. \tag{A.2}$$

Assuming that the boundary is sufficiently resolved by the mesh there must exist a constant, C, such that

$$|\Gamma_T| \leq C h^{d-1}. \tag{A.3}$$

Given that our mesh is non-degenerate we also have $|T| \geq C h^d$. Combining this with (A.1), (A.2) and (A.3) gives us (2.13). □

Acknowledgements

This research was supported by the Swedish Research Council Grant No. 2014-6088.

[1] E. Burman, S. Claus, P. Hansbo, M. G. Larson, A. Massing, CutFEM: Discretizing geometry and partial differential equations, International Journal for Numerical Methods in Engineering 104 (7) (2015) 472–501, ISSN 00295981, doi:10.1002/nme.4823.

[2] E. Burman, P. Hansbo, Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method, Applied Numerical Mathematics 62 (4) (2012) 328–341, ISSN 0168-9274, doi:10.1016/j.apnum.2011.01.008.

[3] P. Hansbo, M. Larson, S. Zahedi, A cut finite element method for a Stokes interface problem, Applied Numerical Mathematics 85 (2014) 90–114, ISSN 0168-9274, doi:10.1016/j.apnum.2014.06.009.

[4] A. Massing, M. G. Larson, A. Logg, M. E. Rognes, A Stabilized Nitsche Fictitious Domain Method for the Stokes Problem, Journal of Scientific Computing 61 (3) (2014) 604–628, ISSN 0885-7474, 1573-7691, doi:10.1007/s10915-014-9838-9.

[5] S. Sticko, G. Kreiss, A stabilized Nitsche cut element method for the wave equation, Computer Methods in Applied Mechanics and Engineering 309 (2016) 364–387, ISSN 00457825, doi:10.1016/j.cma.2016.06.001.

[6] P. Hansbo, M. G. Larson, S. Zahedi, Characteristic cut finite element methods for convection–diffusion problems on time dependent surfaces, Computer Methods in Applied Mechanics and Engineering 293 (2015) 431–461, ISSN 00457825, doi:10.1016/j.cma.2015.05.010.

[7] E. Burman, P. Hansbo, M. G. Larson, S. Zahedi, Cut finite element methods for coupled bulk–surface problems, Numerische Mathematik 133 (2) (2016) 203–231, ISSN 0029-599X, 0945-3245, doi:10.1007/s00211-015-0744-3.

[8] E. Burman, Ghost penalty, Comptes Rendus Math. 348 (21-22) (2010) 1217–1220, ISSN 1631073X, doi:10.1016/j.crma.2010.10.006.

[9] A. Reusken, Analysis of an extended pressure finite element space for two-phase incompressible flows, Computing and Visualization in Science 11 (4-6) (2008) 293–305, ISSN 1432-9360, 1433-0369, doi:10.1007/s00791-008-0099-8.

[10] C. Lehrenfeld, A. Reusken, Optimal preconditioners for Nitsche-XFEM discretizations of interface problems, Numerische Mathematik (2016) 1–20ISSN 0945-3245, doi:10.1007/s00211-016-0801-6.

[11] E. Burman, M. A. Fernández, An unfitted Nitsche method for incompressible fluid–structure interaction using overlapping meshes, Computer Methods in Applied Mechanics and Engineering 279 (2014) 497–514, ISSN 00457825, doi:10.1016/j.cma.2014.07.007.

[12] A. Johansson, M. G. Larson, A high order discontinuous Galerkin Nitsche method for elliptic problems with fictitious boundary, Numerische Mathematik 123 (4) (2013) 607–628, ISSN 0029-599X, 0945-3245, doi:10.1007/s00211-012-0497-1.

[13] A. Johansson, M. G. Larson, A. Logg, High order cut finite element methods for the Stokes problem, Advanced Modeling and Simulation in Engineering Sciences 2 (1) (2015) 24, ISSN 2213-7467, doi:10.1186/s40323-015-0043-7.

[14] J. 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, vol. 36, Springer, 9–15, 1971.

[15] A. Hansbo, P. Hansbo, An unfitted finite element method, based on Nitsche's method, for elliptic interface problems, Computer methods in applied mechanics and engineering 191 (47) (2002) 5537–5552.

[16] C. Schwab, p- and hp- Finite Element Methods, Oxford University Press, ISBN 0-19-850390-3, 1998.

[17] B. Gustafsson, The Convergence Rate for Difference Approximations to Mixed Initial Boundary Value Problems, Mathematics of Computation 29 (130) (1975) 396, ISSN 00255718, doi:10.2307/2005559.

[18] S. Wang, G. Kreiss, Convergence of summation-by-parts finite difference methods for the wave equation, arXiv:1503.08926 [math] ArXiv: 1503.08926.

[19] R. I. Saye, High-Order Quadrature Methods for Implicitly Defined Surfaces and Volumes in Hyperrectangles, SIAM Journal on Scientific Computing 37 (2) (2015) A993–A1019, ISSN 1064-8275, doi:10.1137/140966290.

[20] W. Bangerth, D. Davydov, T. Heister, L. Heltai, G. Kanschat, M. Kronbichler, M. Maier, B. Turcksin, D. Wells, The deal.ii library, version 8.4, Journal of Numerical Mathematics 0 (0), ISSN 1570-2820, doi: 10.1515/jnma-2016-1045.

Recent licentiate theses from the Department of Information Technology

2016-007 Volkan Cambazoglou: *Protocol, Mobility and Adversary Models for the Verification of Security*

2016-006 Anton Axelsson: *Context: The Abstract Term for the Concrete*

2016-005 Ida Bodin: *Cognitive Work Analysis in Practice: Adaptation to Project Scope and Industrial Context*

2016-004 Kasun Hewage: *Towards a Secure Synchronous Communication Architecture for Low-power Wireless Networks*

2016-003 Sven-Erik Ekström: *A Vertex-Centered Discontinuous Galerkin Method for Flow Problems*

2016-002 Rubén Cubo: *Mathematical Modeling for Optimization of Deep Brain Stimulation*

2016-001 Victor Shcherbakov: *Radial Basis Function Methods for Pricing Multi-Asset Options*

2015-006 Hanna Holmgren: *Towards Accurate Modeling of Moving Contact Lines*

2015-005 Siyang Wang: *Analysis of Boundary and Interface Closures for Finite Difference Methods for the Wave Equation*

2015-004 Pavol Bauer: *Parallelism and Efficiency in Discrete-Event Simulation*

2015-003 Fredrik Hellman: *Multiscale and Multilevel Methods for Porous Media Flow Problems*

2015-002 Ali Dorostkar: *Developments in preconditioned iterative methods with application to glacial isostatic adjustment models*

UPPSALA
UNIVERSITET

Department of Information Technology, Uppsala University, Sweden