



Department of Civil, Geo and Environmental Engineering
Chair for Computation in Engineering
Prof. Dr.-Ing. André Borrmann

A Review of Mass Lumping Schemes for the Spectral Cell Method

Máté Kelemen

Master's thesis

for the Master of Science program Computational Mechanics

Author: Máté Kelemen

Matriculation number:

Supervisor: PD Dr.-Ing. habil. Stefan Kollmannsberger

Advisor: Philipp Kopp, M.Sc.

Date of issue: 01. September 2020

Date of submission: 31. March 2021



Involved Organisations



Chair of Computational Modeling and Simulation
Department of Civil, Geo and Environmental Engineering
Technische Universität München
Arcisstraße 21
D-80333 München

Declaration

With this statement I declare, that I have independently completed this Master's thesis. The thoughts taken directly or indirectly from external sources are properly marked as such. This thesis was not previously submitted to another academic institution and has also not yet been published.

München, January 20, 2023

Máté Kelemen

Máté Kelemen
Freisinger Landstraße 17
D-80939 Munich
e-Mail: ge49pil@mytum.de

Contents

| | | |
|----------|---------------------------------------|-----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Applications | 2 |
| 1.3 | Outline | 3 |
| 2 | Fundamental Concepts | 5 |
| 2.1 | The Wave Equation | 5 |
| 2.1.1 | Spatial Discretization | 6 |
| 2.1.2 | Temporal Discretization | 6 |
| 2.2 | The Finite Cell Method | 8 |
| 2.2.1 | Numerical Quadrature | 9 |
| 2.2.2 | Adaptive Quadrature | 10 |
| 2.2.3 | Moment Fitting | 10 |
| 2.3 | The Spectral Element Method | 12 |
| 2.4 | Error Calculation | 13 |
| 3 | The Spectral Cell Method | 19 |
| 3.1 | Overview | 19 |
| 3.2 | Cut Cells | 20 |
| 3.3 | Moment Fitting | 20 |
| 3.4 | Mass Lumping Schemes | 21 |
| 3.4.1 | Row-Sum Lumping | 21 |
| 3.4.2 | Density Scaling | 22 |
| 3.4.3 | HRZ Lumping | 22 |
| 3.5 | Results | 24 |
| 3.5.1 | Axis-aligned Bar | 24 |
| 3.5.2 | Rotated Bar | 32 |
| 3.5.3 | Complex Geometry | 33 |
| 4 | Conclusion | 37 |

Chapter 1

Introduction

1.1 Motivation

The Finite Element Method (FEM) is a well established approach to obtaining approximate solutions to various problems in physics. Suitable for the discretization of a wide range of Partial Differential Equations (PDEs) and capable of dealing with a variety of geometries, it is often a convenient choice for getting detailed, accurate solution fields. However, it is not without drawbacks, some of which render it unsuitable for specific types of analyses. The most important disadvantages of the standard finite element method include its suboptimal convergence rate [BS94], lengthy and problematic mesh generation, and immense demand for computational resources. These problems might not significantly affect moderately complex models, but become more important with increased accuracy requirements, complicated geometries, heterogeneous materials, or stringent resources. Numerous generalizations and modifications of the method have been explored that address one or several of the mentioned issues, but the ever-expanding number of applications still demand specialized improvements over the standard FEM.

Perhaps the most troublesome step of any FE analysis is mesh generation. While robust algorithms for creating acceptable 2D meshes on almost any geometry exist, the same cannot be said for 3D models in general. Meshes on complex 3D geometries must often be adjusted by hand, a lengthy process that usually constitutes the bulk of the time spent performing the analysis. Additionally, mesh quality greatly influences the results, rendering consistent solutions for slight geometric modifications difficult to achieve. In some structured cases, macro elements that capture the geometry and material distribution inside can be used, but the lack of a general definition for such elements restricts their applicability. Furthermore, mesh generation is infeasible for some types of geometric representations, such as point clouds. Embedded domain methods, and the Finite Cell Method (FCM) in particular, are shown to successfully tackle geometries that are otherwise difficult or impossible to mesh [DPYR08]. Due to shifting the computational burden to numerical integration, they are easily automatable, completely eliminating the need for human supervision during the solution process. In addition, the method is well suited for h-p refinement, offering optimal convergence rates. However, these benefits come at the expense of efficiency for dynamic analyses, since the method inherently leads to dense mass matrices.

To solve dynamic PDEs, FE-based methods primarily rely on finite difference schemes that always involve inverting the mass matrix. The performance of any finite difference approach, but especially explicit schemes, greatly benefit from diagonal mass matrices, cutting their complexity down to a fraction of the original. Heuristic mass lumping procedures have a long and successful history with the linear FEM, but usually fail when applied to higher order methods. In contrast, the Spectral Element Method (SEM) features inherently diagonal consistent mass matrices for any order, offering both efficient time integration and high convergence rates at the expense of some restrictions.

The goal of this thesis is to explore approaches combining the advantages of the finite cell and spectral element methods, in pursuit of an efficient, accurate, and highly automatable means to solving dynamic PDEs on complex domains.

1.2 Applications

Applications benefiting from such an approach address linear dynamic problems with high resolution requirements in time on complex domains, advanced geometric representation, or procedurally modified geometries.

High frequency wave propagation analysis in open cell foams, porous structures, or heterogeneous materials such as sandwich structures or other composites [JDGD14] are all prime examples. Meshing such models is either impossible or immensely time consuming, resulting in unnecessarily numerous degrees of freedom that lead to high computational demand in both memory and processing power. Thus, a fictitious domain approach is evidently more appropriate. Furthermore, a high frequency analysis fundamentally restricts the size of time steps in order to correctly resolve the propagating waves, increasing the total number of time steps that constitute the bulk of the process' computational load. As a result, an efficient time integration scheme greatly reduces the total CPU time demanded by such an analysis.

Inverse problems are another class of candidates benefiting from efficient time integration and the lack of mesh generation. Adjoint-based dynamic inverse problems require repeated solutions on slightly modified models as part of an optimization loop. Since the optimization parameters are typically sensitive to changes in the solution, a consistently accurate method is paramount to achieving a convergent iteration, a task that automatic mesh generation is unsuitable for. A practical example is the optimal thermal load control of structures with heat limitations [A79].

Another set of inverse problems are structural health monitoring methods, and full waveform inversion in particular. The goal of such processes is to reveal the internal geometry, often defects, of a model based on measurements on its surface. The objective of the optimization in this case is minimizing the difference between the computed solution and the measurements at all sample points by varying the internal geometry. The measurements correspond to the structure's response to high frequency excitation. The higher the frequency, the more detail internal defects can be detected in, but the smaller time steps must be in the simulation. Additionally, an appropriate representation of the structure's surface is not always available. For example, the geometry of heavily degraded buildings awaiting renovation is most conveniently captured using lidars, providing a point cloud as a surface representation [KKAR19].

All mentioned applications can already be carried out using the FCM for instance, but their immense demand for computational resources could be reduced by successfully combining it with the SEM. The options of doing so are examined in the rest of this thesis, focusing on the acoustic wave equation as a test problem.

1.3 Outline

The next chapter covers the relevant theoretical background of the physics and applied discretization methods. This includes the spatial finite element discretization of the acoustic wave equation followed by finite differences in time. The core ideas of the FCM and SEM are explained afterwards, including their advantages and disadvantages. Lastly, the chapter features various approaches to evaluating the numerical solution to a dynamic system.

The details on combining the FCM and SEM are covered in the chapter on the Spectral Cell Method (SCM), summarizing the problems that arise from doing so and exploring possible approaches to addressing them.

Afterwards, results of models with varying complexity obtained using different methods are presented. The three featured setups involve a bar aligned with a Cartesian mesh, a bar rotated relative to the mesh, and an ellipsoid, all in three dimensions.

Finally, conclusions are drawn from the results and possible directions for further research are recommended.

Chapter 2

Fundamental Concepts

2.1 The Wave Equation

Although the Finite Element Method (FEM) and its derivatives can be applied to a variety of Partial Differential Equations (PDEs), the linear wave equation is chosen in this thesis as a simple model to demonstrate the discussed methods. Its relation to full waveform inversion [ZL13] and history with the Spectral Element Method (SEM) [MQ94] makes it relevant for the topic, while its simplicity allows for focusing on the important aspects of the Spectral Cell Method (SCM).

The material is assumed to be homogeneous and isotropic, filling a geometry $\Omega \subset \mathbb{R}^3$ with perfectly reflective boundaries $\Gamma = \partial\Omega$ modeled as homogeneous Neumann conditions. The time-dependent load $f(\mathbf{x}, t) = f_x(\mathbf{x})f_t(t)$ is assumed to be separable and applied as a volumetric source.

$$\begin{aligned}
 \rho \frac{\partial^2 u(\mathbf{x}, t)}{\partial t^2} - E \Delta u(\mathbf{x}, t) &= f_x(\mathbf{x})f_t(t) & \mathbf{x} \in \Omega \\
 \partial_n u(\mathbf{x}, t) &= 0 & \mathbf{x} \in \Gamma \\
 u(\mathbf{x}, 0) &= u_0(\mathbf{x}) & \mathbf{x} \in \Omega \\
 \frac{\partial u(\mathbf{x}, 0)}{\partial t} &= v_0(\mathbf{x}) & \mathbf{x} \in \Omega
 \end{aligned} \tag{2.1}$$

$$\rho, E \in \mathbb{R}^+ \tag{2.2}$$

where the displacement u varies in space $\mathbf{x} \in \Omega$ and time $t \in [0, t_{max}]$, the material is modeled with a constant density ρ and Young's modulus E , and ∂_n representing the derivative normal to the geometry's boundary Γ . The initial state of the system at $t = 0$ is defined by its displacement $u_0(\mathbf{x})$ and velocity $v_0(\mathbf{x})$ field. The PDE 2.1 is discretized by finite elements in space, and finite differences in time. Since the theory and details of both discretizations have extensive literature, only a brief overview relevant to the covered topics is provided here. A more detailed derivation can be found in [LB13] for example.

2.1.1 Spatial Discretization

The first step to obtaining a weak form is to multiply 2.1 with a test function $v(\mathbf{x})$ from an appropriate ansatz space V , and integrate over the spatial domain.

$$\int_{\Omega} v(\mathbf{x}) \rho \frac{\partial^2 u(\mathbf{x}, t)}{\partial t^2} d\mathbf{x} - \int_{\Omega} v(\mathbf{x}) E \Delta u(\mathbf{x}, t) d\mathbf{x} = f_t(t) \int_{\Omega} v(\mathbf{x}) f_x(\mathbf{x}) d\mathbf{x} \quad (2.3)$$

Integrating by parts and substituting the homogeneous Neumann boundary condition yields the weak form of the wave equation.

$$\int_{\Omega} v \rho \frac{\partial^2 u}{\partial t^2} d\mathbf{x} + \int_{\Omega} (\nabla v)^T E \nabla u d\mathbf{x} = f_t \int_{\Omega} v f_x d\mathbf{x} \quad (2.4)$$

The spatial discretization is carried out by approximating the solution field and test function with linear combinations of ansatz functions $N_i(\mathbf{x}) \in V_h \subset V$.

$$\begin{aligned} u(\mathbf{x}, t) &\cong \sum_i N_i(\mathbf{x}) \hat{u}_i(t) \\ v(\mathbf{x}) &\cong \sum_i N_i(\mathbf{x}) \hat{v}_i \end{aligned} \quad (2.5)$$

$$\int_{\Omega} \sum_i N_i \hat{v}_i \rho \sum_j N_j \frac{\partial^2 \hat{u}_j}{\partial t^2} d\mathbf{x} + \int_{\Omega} \sum_i (\nabla N_i \hat{v}_i)^T E \sum_j \nabla N_j \hat{u}_j d\mathbf{x} = f_t \int_{\Omega} f_x \sum_i N_i \hat{v}_i d\mathbf{x} \quad (2.6)$$

Since the weak form 2.4 must hold for any $v \in V_h$, equation 2.6 has to hold for all combinations of \hat{v}_i as well, leading to the spatially discretized form of the wave equation:

$$\int_{\Omega} N_i \rho N_j d\mathbf{x} \frac{\partial^2 \hat{u}_j}{\partial t^2} + \int_{\Omega} (\nabla N_i)^T E \nabla N_j d\mathbf{x} \hat{u}_j = f_t \int_{\Omega} f_x N_i d\mathbf{x} \quad (2.7)$$

The mass matrix \mathbf{M} , stiffness matrix \mathbf{K} and load vector \mathbf{f} appear in equation 2.7.

$$M_{ij} \frac{\partial^2 \hat{u}_j}{\partial t^2} + K_{ij} \hat{u}_j = f_t f_i \quad (2.8)$$

$$\begin{aligned} M_{ij} &= \int_{\Omega} N_i \rho N_j d\mathbf{x} \\ K_{ij} &= \int_{\Omega} (\nabla N_i)^T E \nabla N_j d\mathbf{x} \\ f_i &= \int_{\Omega} f_x N_i d\mathbf{x} \end{aligned} \quad (2.9)$$

2.1.2 Temporal Discretization

The result of the spatially discretized wave equation 2.8 is a second order system of Ordinary Differential Equations (ODEs), identical in form for many linear undamped models in structural mechanics and other fields. Although variational time discretization methods with

appealing properties have been explored [ZW14] and applied to the wave equation [KB14], finite difference schemes dominate the literature due to their simplicity, long history, and efficiency in both computational load and memory requirements.

Two main categories of finite difference schemes exist with distinct properties. To allow a more general discussion on them, 2.8 is transformed into a first order system of the form $\dot{\mathbf{y}}(t) = \mathbf{g}(t, \mathbf{y}(t))$

$$\dot{\mathbf{y}}(t) := \begin{bmatrix} \dot{\hat{\mathbf{u}}}(t) \\ \ddot{\hat{\mathbf{u}}}(t) \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & 0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}}(t) \\ \dot{\hat{\mathbf{u}}}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ f_t\mathbf{M}^{-1}\mathbf{f} \end{bmatrix} = \mathbf{g}(t, \mathbf{y}(t)) \quad (2.10)$$

where $\dot{\hat{\mathbf{u}}} = \frac{d\hat{\mathbf{u}}}{dt}$ denotes time differentiation and the extended state is defined as $\mathbf{y} = [\hat{\mathbf{u}} \ \dot{\hat{\mathbf{u}}}]^T$. A generic finite difference scheme marches at discrete points in time t_k to approximate the system's state $\mathbf{y}_k = \mathbf{y}(t_k)$ using 2.10.

$$\mathbf{y}_{k+1} \approx \mathbf{h}^i(\mathbf{y}_{k+1}) + \mathbf{h}^e(\mathbf{y}_k, \mathbf{y}_{k-1}, \dots, \mathbf{y}_0) \quad (2.11)$$

For implicit methods, $\mathbf{h}^i \neq 0$ meaning that they include the yet unknown state \mathbf{y}_{k+1} , requiring the solution of an algebraic equation system at each time step. These methods are unconditionally stable and may prove to be a suitable choice for problems where large time steps are acceptable.

On the other hand, explicit methods only use previously computed states ($\mathbf{h}^i = 0$ and $\mathbf{h}^e \neq 0$), requiring the evaluation of the right hand side in 2.10 at earlier time points, and provides an explicit expression for the unknown state. If the mass matrix can be inverted cheaply, explicit time steps can be performed far more efficiently than implicit ones, at the cost of stability. To be convergent, explicit finite difference schemes must satisfy the Courant-Friedrichs-Lewy (CFL) condition [CFL67], which defines an upper bound for the time step size relative to the spatial discretization. This requirement can defeat the purpose of using explicit methods in some cases as the computational load per step size may exceed that of implicit ones. As mentioned in 1 however, the assumption in this thesis is that the physics of the problem already demand small time steps, justifying the use of explicit finite differences.

A popular method for the temporal discretization of 2.10 is the Newmark-beta scheme [N59] that assigns tunable weights to the accelerations separately for the approximation of the velocity and displacement. Depending on the choice of parameters β and γ , it can either be implicit or explicit.

$$\begin{aligned} \hat{\mathbf{u}}_{k+1} &= \hat{\mathbf{u}}_k + \Delta t \dot{\hat{\mathbf{u}}}_k + \Delta t^2 \frac{(1-2\beta)\ddot{\hat{\mathbf{u}}}_k + 2\beta\ddot{\hat{\mathbf{u}}}_{k+1}}{2} \\ \dot{\hat{\mathbf{u}}}_{k+1} &= \dot{\hat{\mathbf{u}}}_k + \Delta t \left((1-\gamma)\ddot{\hat{\mathbf{u}}}_k + \gamma\ddot{\hat{\mathbf{u}}}_{k+1} \right) \end{aligned} \quad (2.12)$$

The second order Central Difference Method (CDM) is obtained by setting the parameters $\beta = 0$ and $\gamma = \frac{1}{2}$.

$$\begin{aligned}\hat{\mathbf{u}}_{k+1} &= \hat{\mathbf{u}}_k + \Delta t \dot{\hat{\mathbf{u}}}_k + \frac{\Delta t^2}{2} \ddot{\hat{\mathbf{u}}}_k \\ \dot{\hat{\mathbf{u}}}_{k+1} &= \dot{\hat{\mathbf{u}}}_k + \frac{\Delta t}{2} (\ddot{\hat{\mathbf{u}}}_k + \ddot{\hat{\mathbf{u}}}_{k+1})\end{aligned}\tag{2.13}$$

To show that 2.13 is explicit, the first derivatives can be eliminated:

$$\hat{\mathbf{u}}_{k+1} = 2\hat{\mathbf{u}}_k - \hat{\mathbf{u}}_{k-1} + \Delta t^2 \ddot{\hat{\mathbf{u}}}_k\tag{2.14}$$

Substituting 2.10 into 2.14 leads to a fully discretized form of the wave equation.

$$\hat{\mathbf{u}}_{k+1} = 2\hat{\mathbf{u}}_k - \hat{\mathbf{u}}_{k-1} + \Delta t^2 \mathbf{M}^{-1} (f_t \mathbf{f}_k - \mathbf{K} \hat{\mathbf{u}}_k)\tag{2.15}$$

Note that 2.15 requires the inversion of the mass matrix \mathbf{M} , which is a common property of all explicit methods. Most often, \mathbf{M} is factorized before the initial time step instead of directly computing its inverse. However, the ideal case would be if the mass matrix were diagonal, eliminating the need of forward/backward substitutions of the factorized \mathbf{M} and reducing the number of matrix-vector operations. Exploring different approaches to achieve this in combination with the finite cell method is the main goal of this thesis.

2.2 The Finite Cell Method

Generating a boundary-conforming mesh can be costly or in some cases impossible, especially when the model has a complex geometry, must be manipulated procedurally throughout the lifetime of the analysis, or is represented by a geometric model that is not suitable for standard mesh generation, like point clouds. Such is the case with inverse analyses and optimal control problems, where either a set of boundary conditions, or the geometry itself is updated in each iteration of an optimization process. In such cases, generating and tweaking a mesh by hand is infeasible not only because the vast amount of time it would require, but also because the quality of the mesh has a great impact on the results' accuracy and may prohibit the optimization from converging. A number of approaches exist that aim to simplify or avoid the issues of mesh generation such as Generalized Finite Element Methods (GFEM) [BBO04] or meshless methods [CLE06]. Fictitious domain methods address the problem by extending the domain of computation such that it can be easily meshed, and introduce non-uniform material properties that separate the physical and fictitious domains. This approach shifts the focus from the meshing process to numerical integration, which is not without its own challenges, but is more suited for automation.

The Finite Cell Method (FCM) is a fictitious domain method that works with a non-boundary-conforming simple (often Cartesian) mesh over an embedding domain Ω , and introduces an indicator function $\alpha(\mathbf{x})$ that defines the physical domain $\Omega_p \subset \Omega$ (also called the embedded domain) in space.

$$\alpha(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \Omega_p \\ 10^{-\beta} & \mathbf{x} \in \Omega_f \end{cases}\tag{2.16}$$

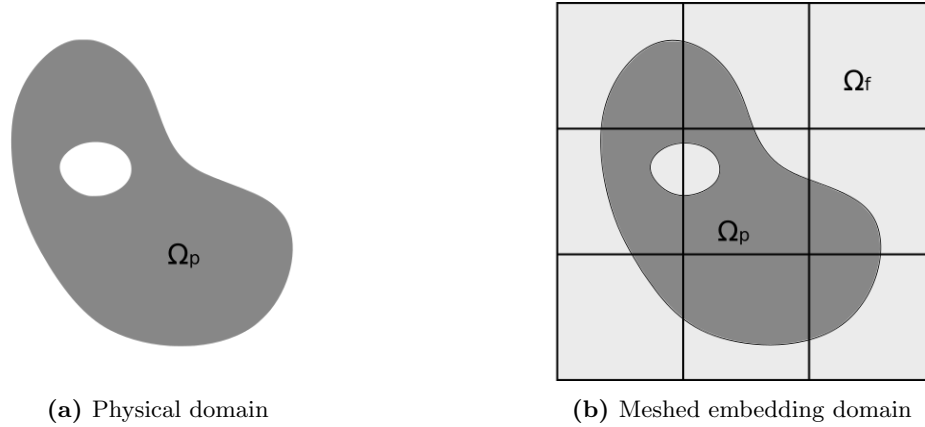


Figure 2.1: Example of an FCM mesh embedding the physical domain

The indicator function $\alpha(\mathbf{x})$ is used as a coefficient for material properties, effectively "softening" the material in the fictitious domain such that the influence of its solution field on that of the physical domain becomes negligible. Ideally, $\beta \rightarrow \infty$ leading to $\alpha(\mathbf{x}) \rightarrow 0$ in the fictitious domain, but is usually limited to $\beta \in [3, 10]$ in order to avoid ill-conditioned structural matrices [PDR07].

$$\begin{aligned}\tilde{\rho}(\mathbf{x}) &= \alpha(\mathbf{x})\rho \\ \tilde{E}(\mathbf{x}) &= \alpha(\mathbf{x})E\end{aligned}\tag{2.17}$$

This modification creates discontinuous material properties in elements that are intersected by the boundary of the physical domain (cut cells), which in turn makes standard gaussian quadrature infeasible for the integration of structural components (mass matrix, stiffness matrix, etc.). Methods that address this issue are presented later in this section.

The FCM primarily uses p- and hp-refinement, requiring an appropriate family of basis functions. The most common choice are integrated Legendre polynomials [DPYR08] not only because of their hierarchical properties, but the d dimensional tensor product space constructed from them can be truncated while preserving its completeness. This greatly reduces the number of degrees of freedom at higher polynomial orders, especially in 3D.

Considering that the shape of the physical domain must be captured during integration, and the high polynomial order of the ansatz space, an appropriate choice of integration scheme for cut cells is key to obtaining accurate results. Several methods exist to capture discontinuities in numerical integration but two adaptive schemes are covered here in detail, both of which rely on cartesian space partitioning trees: a standard quadtree/octree partitioned quadrature and a quadrature based on moment fitting.

2.2.1 Numerical Quadrature

Each considered integration scheme is based on numerical quadrature, that approximates an integral with a weighted sum of samples of the integrand at specific locations [A88].

$$\int_{\Xi} g(\boldsymbol{\xi}) d\boldsymbol{\xi} \approx \sum_{k=1}^m g(\boldsymbol{\xi}_k) w_k \quad (2.18)$$

where the integrand $g(\boldsymbol{\xi})$ is evaluated over a normalized integration domain $\Xi = [-1, 1]^n$ of n dimensions at m sample points $\boldsymbol{\xi}_k$, each multiplied with its corresponding weight w_k . Integrating over a different domain can be carried out after a change of variables.

For fundamental quadrature schemes such as the Gauss-Legendre or Gauss-Lobatto rules, the number of sample points m defines the order of integration. These methods are typically most suited for polynomial or smooth integrands, that can be well approximated by polynomials. Each scheme is capable of exactly integrating polynomials up to a maximum order p_m which scales linearly with the integration order m .

The preferred method of computing the structural components of uncut cells in the FCM is the Gauss-Legendre quadrature, as it offers the highest maximum order p_m^{Leg} of all fundamental quadrature rules.

$$p_m^{Leg} = 2m - 1 \quad (2.19)$$

2.2.2 Adaptive Quadrature

Since the modified material properties break the smoothness of the integrands in 2.9 for cut cells, standard quadrature rules can not be efficiently applied. A simple solution is to separate the domain of integration along the discontinuities and perform the quadrature on them individually, then sum up their contributions.

$$\int_{\Omega} \alpha(\mathbf{x}) g(\mathbf{x}) d\mathbf{x} = \int_{\Omega_p} 1 \cdot g(\mathbf{x}) d\mathbf{x} + \int_{\Omega_f} 10^{-\beta} g(\mathbf{x}) d\mathbf{x} \quad (2.20)$$

The standard approach in FCM is to recursively generate subcells using a quadtree (2D) or octree (3D) and carry out the integration once a desired depth or error has been reached. Figure 2.2 shows a 2D example of a quadtree of depth two. Note, that this is a refinement of the integration mesh, not the finite element mesh. Hence, no additional degrees of freedom are introduced.

The integration order of the underlying standard quadrature scheme is retained, and can be fully exploited if the domain boundaries are captured exactly. While this algorithm is capable of computing the desired integral up to arbitrary precision, it involves evaluating the integrands in 2.9, a series of matrix-matrix and matrix-vector multiplications, at all integration points of each subcell, thus imposing a major computational burden.

2.2.3 Moment Fitting

Some of the computational load introduced by adaptive integration can be reduced with moment fitting [JHD15]. While the process' complexity remains exponential in terms of

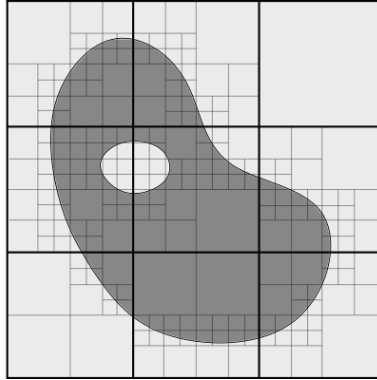


Figure 2.2: Embedding mesh subdivided with a quadtree of depth 2.

refinement depth, the tasks performed at each integration point can be simplified. Moment fitting is a method for generating quadrature rules on arbitrary domains [MKO13]. In the context of this thesis, it is used to preserve the number and location of integration points of the original quadrature scheme used for uncut cells, but adjust their weights by taking the integrand's discontinuities into account.

In general, a new quadrature rule defined by its nodes $\boldsymbol{\xi}_k$ (integration points) and weights w_k is generated by solving the moment fitting equations

$$\sum_{k=1}^m g_j(\boldsymbol{\xi}_k) w_k = \int_{\Xi_a} g_j(\boldsymbol{\xi}) d\boldsymbol{\xi} \quad (2.21)$$

where g_j is a set of n chosen basis functions. The integrals over the arbitrary domain Ξ_a on the right hand side are called moments, hence the name of the method. Without any constraints, solving 2.21 for the nodes and weights is not trivial. However, fixing the location of the integration points $\boldsymbol{\xi}_k$ transforms it into a linear system. Furthermore, choosing the basis functions g_j to be Lagrange polynomials over $\boldsymbol{\xi}_k$ further simplifies the process by diagonalizing the coefficient matrix, leading to equation 2.22. Note that this process of simplification is identical to the premise of the Spectral Element Method in 2.3.

$$\begin{bmatrix} g_1(\boldsymbol{\xi}_1) & & \\ & \ddots & \\ & & g_m(\boldsymbol{\xi}_m) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix} = \begin{bmatrix} \int_{\Xi_a} g_1(\boldsymbol{\xi}) d\boldsymbol{\xi} \\ \vdots \\ \int_{\Xi_a} g_m(\boldsymbol{\xi}) d\boldsymbol{\xi} \end{bmatrix} \quad (2.22)$$

The integrals on the right hand side of 2.22 can be computed with an adaptive quadrature described before, with the difference that in this case, the integrands are much cheaper to evaluate. However, the resulting quadrature rule can only integrate polynomials up to order $p_m^{MF} = m + 1$ exactly, in contrast to $p_m^{Leg} = 2m - 1$ for the Gauss-Legendre scheme. This is due to the fact that the positions are fixed and only the weights are considered as unknowns in the moment fitting equations. Furthermore, this assumes that the integrals on the right hand side of 2.22 are computed exactly, which in turn implies that the geometry of the physical domain can be captured exactly by the applied space partitioning tree. This is rarely the case, but considering the fact that adaptive integration suffers from this issue as well, it is

not regarded as a disadvantage of moment fitting. The reduced accuracy of the quadrature rule can be addressed by increasing the integration order of the underlying scheme.

Working on a non-boundary conforming mesh introduces numerous other challenges, such as the enforcement of boundary conditions, but since a detailed literature is available on them [DPYR08, PDR07] and are handled identically for the SCM, they are not covered in this thesis.

2.3 The Spectral Element Method

As pointed out in section 2.1, a diagonal mass matrix \mathbf{M} is essential to fully exploit the efficiency of explicit time integration schemes. A well-established approach that yields an inherently diagonal consistent mass matrix, is the Spectral Element Method (SEM) [MQ94]. Based on the p-version of the FEM, the SEM relies on a specific choice of ansatz space and quadrature rule for its appealing properties.

Using the same line of thought as for the simplification of the moment fitting equations described in section 2.2, Lagrange polynomials defined on a set of Gauss-Lobatto points are used as basis functions. Accordingly, the quadrature rule for integrating the mass matrix is the Gauss-Lobatto scheme of matching order, which means that the integration points coincide with those used for the definition of the basis functions. As a result, the product of mixed ansatz functions vanish at all integration points, leading to a diagonal mass matrix.

$$N_i(\boldsymbol{\xi}_k)N_j(\boldsymbol{\xi}_k) = 0 \quad \forall i \neq j \quad \forall k \in \{1..m\} \quad (2.23)$$

$$M_{ij} = \int_{\Omega} N_i \rho N_j d\mathbf{x} \approx \det(\mathbf{J}) \sum_{k=1}^m N_i(\boldsymbol{\xi}_k) \rho N_j(\boldsymbol{\xi}_k) w_k = \begin{bmatrix} M_{11} & & 0 \\ & \ddots & \\ 0 & & M_{nn} \end{bmatrix} \quad (2.24)$$

where \mathbf{J} is the Jacobian facilitating the change of variables from \mathbf{x} to $\boldsymbol{\xi}$.

Gauss-Lobatto points are chosen in particular because they always include the limits of the integration domain -1 and 1 , ensuring that all except one ansatz functions vanish on each border, as shown in figure 2.3. This property is essential for FE-based methods to ensure the continuity between elements, allow the blending of specific ansatz functions to exactly capture boundaries, and strongly enforce boundary conditions.

Using a Lagrange basis instead of integrated Legendre polynomials has numerous additional consequences, most of them undesirable. Firstly, the hierarchic nature of the ansatz space is lost, requiring the re-evaluation of the entire stiffness matrix as well when performing p-refinement. Moreover, the conditioning of the stiffness matrix is worse [Z96], and degrades more rapidly with increasing p . Lastly, no complete trunk space of Lagrange polynomials exist, which means that the ansatz space must be constructed from the tensor product of the basis functions, leading to an increased number of degrees of freedom and requiring more computational resources. Lastly, constructing ansatz spaces with inhomogeneous polynomial orders is more complicated than it is for hierarchic basis functions. On the other hand,

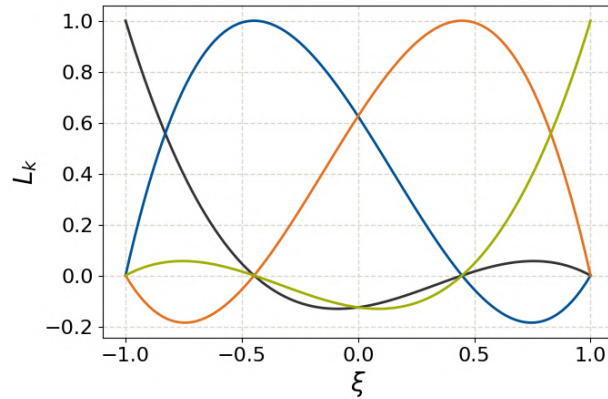


Figure 2.3: Lagrange basis functions on a set of fourth order Gauss-Lobatto points.

Lagrange basis functions are nodal and have a partition of unity $\sum_{k=1}^m L_k(\xi) = 1$.

To ensure the diagonality of the mass matrix, the basis functions' polynomial order p enforces the quadrature scheme to be the tensor product of the $m = p + 1$ order Gauss-Lobatto rule, leading to further undesirable effects. The most obvious one of course is the lack of choice when it comes to the selection of integration schemes. This ties the integration order to the order of the basis functions regardless of the order of the integrands in equation 2.9, meaning that integration errors increase with the order of material parameters. The highest polynomial order a tensor product quadrature scheme is able to exactly integrate is identical to that of the underlying 1D rule for each variable [K96]. In the case of SEM, this translates to $2m - 3 = 2p - 1$. Considering 2.9, the highest order of the mass matrix's integrand is $2p$ per direction, while the stiffness matrix's is $2p - 1$ in 1D and $2p$ in higher dimensions. Consequently, neither the mass nor the stiffness matrix can be integrated exactly (apart from 1D models), even for constant material parameters.

Since no benefit is gained by using the Gauss-Lobatto rule for the stiffness matrix and load vector, a different integration scheme can be applied. Unfortunately, no solution exists to this problem in the mass matrix's case, meaning that it is always underintegrated in the SEM. It is important to note that under certain circumstances, especially in wave propagation problems, this drawback partly balances other errors originating from the discretization [AW10], leading to more accurate results when compared to the classical FEM. In general however, it is an additional source of errors that increase with the order of the density function.

2.4 Error Calculation

Quantifying the quality of a numerical solution to a dynamic system is not a straightforward task, mostly because different types of errors have varying importance depending on the goal of the analysis. In linear wave propagation problems, the main error types include dispersion and spurious oscillations, both of which have an accumulative nature in time. To streamline the analysis of these errors, a simple 1D wave propagation problem is considered, shown in figure 2.4.

A distributed source $f(x, t)$ is applied on a 1D bar of length $L = 0.5$ with homogeneous Neumann conditions on both ends, resulting in a single half-wave of period T traveling in positive x -direction. In order to minimize disturbances originating from boundaries and the source, displacements are recorded at two sample points P_0 and P_1 that divide the bar into 3 equidistant segments. The material of the bar is linear with a constant Young's modulus $E = 1$ and density $\rho = 1$, resulting in a unit wave speed $c = \sqrt{E/\rho} = 1$. To avoid reflections on the right hand side boundary, the observation is terminated at $t_{max} = L/c$.

$$f(x, t) = \begin{cases} e^{-10^4 x^2} \sin\left(\frac{2\pi}{T}t\right) & t \in \left[0, \frac{T}{2}\right] \\ 0 & \text{otherwise} \end{cases} \quad (2.25)$$

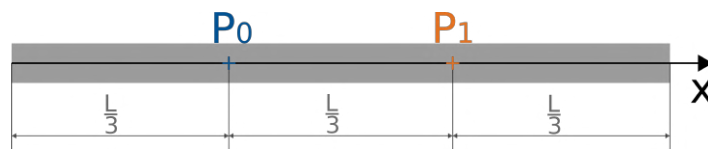
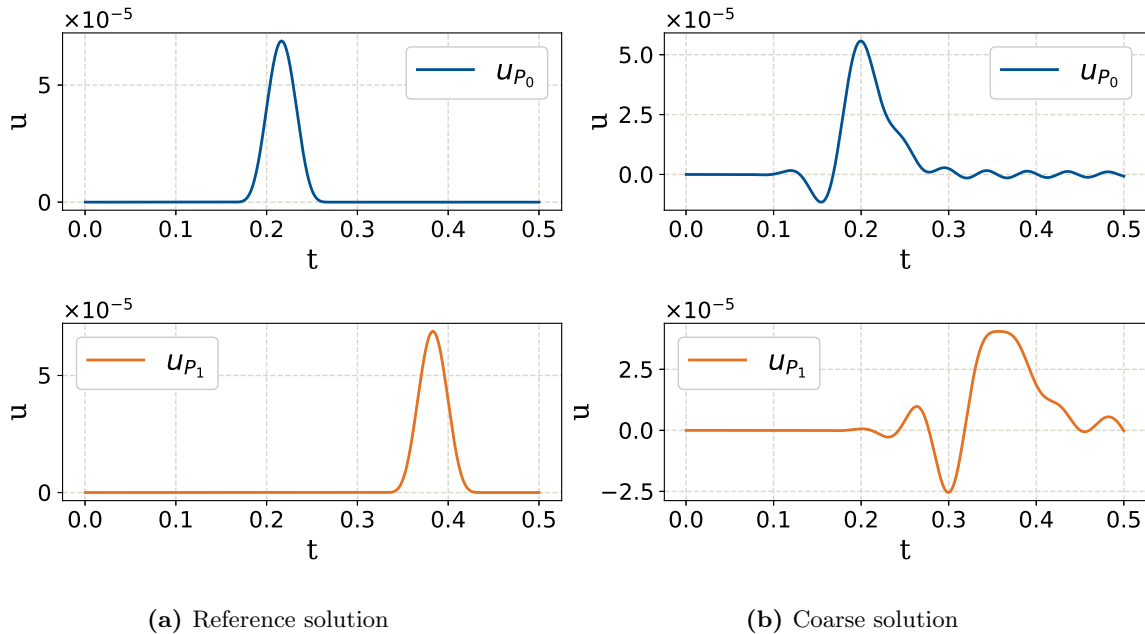


Figure 2.4: Simple 1D bar with two sample points.

Figure 2.5 shows the displacement history at the two sample points for an "overkill" fine mesh and a coarse one. The fine mesh is used as reference, and consists of 50 equidistant elements with an integrated Legendre basis of order $p = 3$. In contrast, the coarse mesh features 20 elements with linear basis functions. The time step size $\Delta t = 2 \cdot 10^{-4}$ is identical for both models.



(a) Reference solution

(b) Coarse solution

Figure 2.5: Displacement history at the two sample points for a reference mesh on the left 2.5a and a coarse one on the right 2.5b.

Directly comparing solution fields at each time step to a reference is often a poor choice, as even small time shifts can be interpreted as huge errors. Conversely, defining the error in the frequency domain's magnitude focuses on the behaviour of the system but completely

neglects phase shifts. An alternative approach is to define the error in terms of the wavelet's time of flight between the two sample points. The difficulty in this case is finding the time of arrival t^a of a wavelet at a given location. Two such methods are explored in this section: the first finds the centroid of the displacement history while the second its peak.

The envelope $e_u(t)$ of a function $u(t)$ does not have a rigorous definition but most often refers to the instantaneous amplitude, which is the magnitude of the function's analytic signal $a_u(t)$ [HSL98]:

$$a_u(t) = u(t) + i\mathcal{H}(u(t)) \quad (2.26)$$

$$e_u(t) = |a_u(t)| = \sqrt{u(t)^2 + \mathcal{H}(u(t))^2} \quad (2.27)$$

where \mathcal{H} denotes the Hilbert transform. The time of arrival t_u^a is then approximated with the envelope's centroid [DJDG14]:

$$t_u^a \approx \frac{\int_0^{t_{max}} t e_u(t) dt}{\int_0^{t_{max}} e_u(t) dt} \quad (2.28)$$

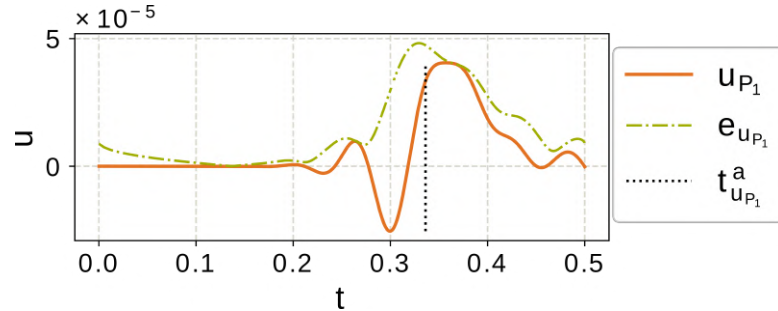


Figure 2.6: Time of arrival based on the envelope of the displacement at P_1 of the coarse solution.

An undesirable consequence of computing the centroid on the entire time domain is that the time of arrival tends to offset towards $\frac{t_{max}}{2}$ if the time of observation is much greater than the wavelet's period $t_{max} \gg T$.

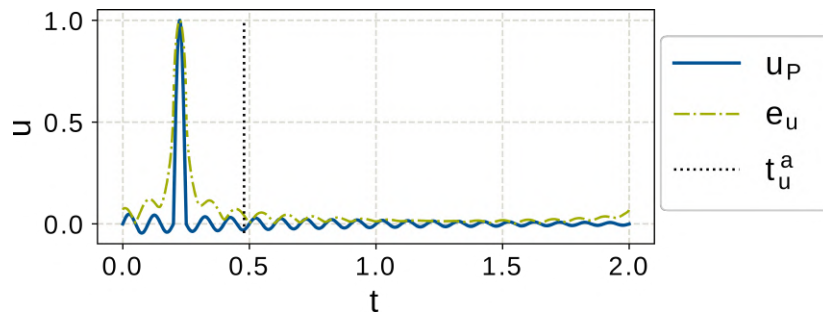


Figure 2.7: Example time of arrival offset as a result of a long time of observation and spurious oscillations.

Assuming the magnitude of the errors is smaller than the wave, this effect can be compensated

by raising the envelope to some power $q > 1$ and calculating its centroid instead. As shown in figure 2.8, the influence of both the spurious oscillations and the time of observation can be suppressed arbitrarily.

$$t_u^a \approx \frac{\int_0^{t_{max}} t e_u^q(t) dt}{\int_0^{t_{max}} e_u^q(t) dt} \quad (2.29)$$

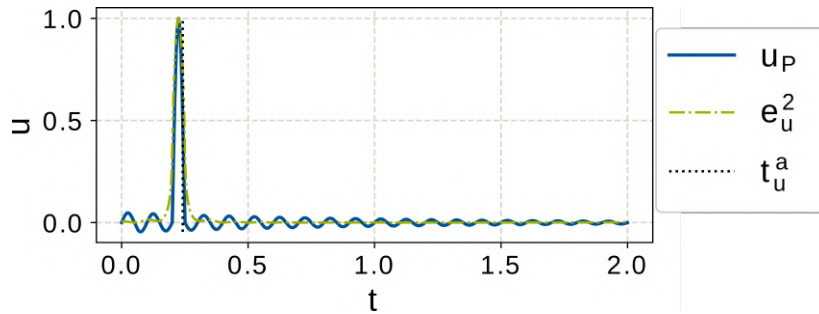


Figure 2.8: Time of arrival computed from a squared envelope.

As the exponent q is increased t_u^a approaches $\text{argmax}(|u(t)|)$, effectively neglecting all oscillations and solely depending on the absolute maximum of the displacement history. This property is generally not desired but can be useful for models where the relation between the discretization and spurious oscillations is chaotic. To avoid round-off errors and increase performance, this approximation is computed using an alternative method instead. Since the displacement history is only available at discrete time points t_k , the accuracy of merely choosing the time at which the sampled displacement has an absolute maximum would be greatly influenced by the size of the time steps Δt . Thus, the extremum of a cubic interpolating spline $s(t)$ is used instead.

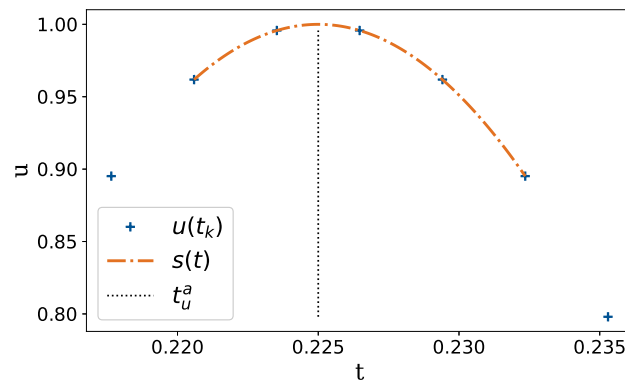


Figure 2.9: Time of arrival based on a cubic interpolating spline.

Once the time of arrival is determined for both sample points, the relative error of the time of flight ϵ can be calculated as follows:

$$\epsilon = 1 - \frac{|t_{u_{P_1}}^a - t_{u_{P_0}}^a|}{\frac{|P_0 P_1|}{c}} \quad (2.30)$$

where $c = \sqrt{\frac{E}{\rho}}$ is the analytical wave speed.

Chapter 3

The Spectral Cell Method

3.1 Overview

The Spectral Cell Method (SCM) combines the main features of the finite cell and spectral element methods with the goal of creating an efficient but at the same time highly automatable algorithm for solving dynamic problems. Should this goal be achieved, the benefit would be the ability to fully exploit the efficiency of explicit time integration schemes, even when working on arbitrarily complex geometries or heterogeneous materials.

However, one has to keep in mind that the drawbacks and limitations of both approaches apply as well. Firstly, the types of PDEs are best suited for linear ones because of the SEM's integration accuracy, mentioned in 2.3. Even if that was not an issue, non-constant material parameters require the re-integration of structural matrices 2.9 at every discrete time point, a task that far outweighs the complexity of a single time step if the model mostly consists of cut cells. Furthermore, generating a boundary-conforming mesh should be difficult or the analysis must justify a fictitious domain approach, as the standard SEM would be more efficient otherwise. Lastly, the size of time steps Δt should be small, restricted by either the problem's physics or accuracy criteria because otherwise larger implicit time steps may prove to be more efficient than small explicit ones.

At its core, the embedded domain concept of the FCM is used as a framework with Lagrange basis functions and Lobatto quadrature borrowed from the SEM. The stiffness matrix \mathbf{K} and load vector \mathbf{f} can be computed using other, more accurate integration schemes such as the Gauss-Legendre quadrature, for the same reasons detailed in 2.3. As a result, the element mass matrices \mathbf{M}^e of uncut cells are underintegrated, but inherently diagonal.

Since a Lagrange basis is used instead of an integrated Legendre one, the stiffness matrix must also be recomputed entirely when performing a p-refinement. However, this has little impact on the efficiency of the method because it only has to be reintegrated at the initial time step, the mass matrix and load vector must be recomputed either way, and the bulk of the computational load is expected to originate from time integration. A potentially more important issue is that element stiffness matrices have higher condition numbers.

3.2 Cut Cells

Cells that are inside the physical domain in their entirety are treated identically as in the SEM, while others located completely in the fictitious domain can be discarded. However, the main issue of the SCM arises when dealing with cut cells that have points in both domains.

While adaptive integration schemes can be used to accurately compute the element stiffness matrices \mathbf{K}^e and load vectors \mathbf{f}^e , mass matrices \mathbf{M}^e do not have this option. Adaptive integration introduces new quadrature points that do not coincide with the Gauss-Lobatto points, leading to non-zero off-diagonal entries in the mass matrix. However, standard Gauss-Lobatto quadrature is unsuitable for integrating discontinuous functions. The two possible approaches to solving this problem are:

- finding an integration scheme capable of dealing with discontinuities while preserving the location of quadrature points
- diagonalizing mass matrices after integration

A candidate for the former is moment fitting, while the latter approach is covered by mass lumping schemes that have extensive literature.

3.3 Moment Fitting

As described in 2.2.3, moment fitting can be used to generate a quadrature scheme capable of integrating discontinuous functions while preserving a fixed set of integration points. At first glance, this makes it an excellent choice for handling cut cells in the SCM, however its reduced accuracy renders it infeasible. To preserve the diagonality of the mass matrix, the polynomial order of the basis functions dictate the integration order of the Gauss-Lobatto scheme: $m = p + 1$. Combined with the fact that the integrand of the mass matrix is of order $p_M = 2p$, and that a quadrature scheme generated by moment fitting can only exactly integrate functions up to order $p_m^{MF} = m + 1 = p + 2$ per direction, the integration error becomes too significant for this method to be viable.

In fact, this approach can lead to negative diagonal entries in the mass matrix even in simple cases, leading to a divergent solution in time. For example, a single square element cut half in figure 3.1, with a basis of order 2 already yields such a mass matrix. With $\rho = 1$ and $\beta = 5$, the diagonal reads:

$$[M_{ii}^e] = [1.4^{-1} \quad 2.2^{-1} \quad -2.8^{-2} \quad 5.6^{-6} \quad 8.9^{-6} \quad 1.4^{-6} \quad 2.2^{-6} \quad -2.8^{-7}] \quad (3.1)$$

Consequently, moment fitting cannot be used for the diagonalization of cut cells' mass matrices.

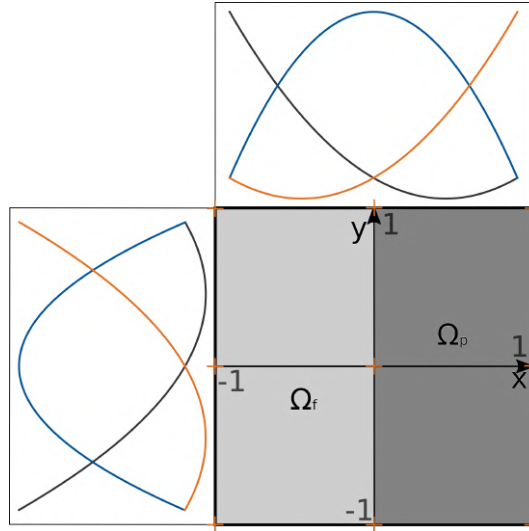


Figure 3.1: Square cell cut in half along the y -axis with a Lagrange basis of order 2, and the corresponding Gauss-Lobatto points.

3.4 Mass Lumping Schemes

Mass lumping schemes are another group of candidates for providing diagonal mass matrices for cut cells. They are a well established means of diagonalization in the standard FEM, usually operating on the element or global matrix level and eliminating off-diagonal components while lumping their contributions onto the main diagonal. A number of such methods exist with the only universal requirement being the preservation each element's total mass. The downside of mass lumping techniques is that they are generally heuristic approaches and as such, their accuracy may not improve with p -refinement in some cases. Moreover, their applicability is usually restricted to certain types of elements or basis functions, and may provide negative masses when these criteria are not met, leading to divergent solutions [DG19]. Three considered methods are presented in the following sections.

As the SCM's viability hinges on the quality of diagonalized element mass matrices of cut cells, choosing an appropriate lumping scheme is essential. To prevent underintegration, Gauss-Legendre-based adaptive quadrature is used for cut cells, since diagonalization through a choice of integration points is no longer possible.

3.4.1 Row-Sum Lumping

Row-sum lumping is one of the simplest approaches to diagonalization. It is often used in the standard FEM and can even have some beneficial effects on the quality of the solution for specific cases, such as adding artificial viscous behaviour to the system [H00]. The diagonal entries of an element mass matrix $M_{ii}^{e,RS}$ using the row-sum technique are computed as follows:

$$\begin{aligned} M_{ii}^{e,RS} &= \sum_k \int_{\Omega^e} N_i \rho N_k d\Omega \\ &= \sum_k M_{ik}^e \end{aligned} \tag{3.2}$$

Since the computation takes place on rows independently, the lumping can be executed on the global mass matrix and can be trivially parallelized. However, note that 3.2 holds only if the shape functions N_i have a partition of unity. Furthermore, this method may produce non-positive masses for high order ($p > 2$) basis functions [D14], which rules out its use in the SCM.

3.4.2 Density Scaling

Density scaling is described in [JDGD14] and is unique to fictitious domain methods, as the lumped matrix $M_{ij}^{e,DS}$ is given by the diagonal of the uncut cell's consistent mass matrix $M_{ij}^{u,e}$, scaled to preserve the cut cell's mass m_e .

$$M_{ii}^{e,DS} = \frac{m_e}{\sum_k M_{kk}^{u,e}} M_{ii}^{u,e} \quad (3.3)$$

For the SCM, the resulting matrix in 3.3 is equivalent to that of an uncut cell, whose density is scaled by the considered cell's mass ratio.

$$\rho_s = \frac{\int_{\Omega^e} \alpha \rho d\Omega}{\int_{\Omega^e} \rho d\Omega} \rho \quad (3.4)$$

As a result, the geometric information on the boundary between the physical and fictitious domains is lost in the process. Moreover, this technique involves extrapolating the density of the physical domain to the fictitious regions of cut cells, a trivial task for constant densities but problematic otherwise.

Compared to row-summing, this scheme imposes the additional overhead of having to compute an extra mass matrix of an uncut cell, and it cannot be performed on the assembled mass matrix. What it provides in return however, is physical interpretability and a guaranteed positive-definite mass matrix.

3.4.3 HRZ Lumping

HRZ lumping, also called diagonal scaling, discards off-diagonal components in the element mass matrix and scales the diagonal such that the total mass is preserved [HRZ76].

$$M_{ii}^{e,HRZ} = \frac{m_e}{\sum_k M_{kk}^e} M_{ii}^e \quad (3.5)$$

Although this method guarantees the positive definiteness of the mass matrix, it likely leads to reduced convergence rates [DG19]. Despite this shortcoming, HRZ lumping is used in this thesis to lump the mass matrices of cut cells because it provides positive diagonal components as opposed to row summing, it takes the material distribution in the cell into account more accurately than density scaling, and it does not impose significant overhead in the used software.

Total mass of a cell

In order to compute the coefficient for scaling the diagonal components, the total mass of the cell must be calculated first.

$$m^e = \int_{\Omega^e} \rho d\Omega \quad (3.6)$$

Using basis functions that have partition of unity, 3.6 can be computed from the sum of all components in the element mass matrix.

$$m^e = \sum_i \sum_j M_{ij}^e \quad (3.7)$$

As shown below, functions in an ansatz space constructed from the tensor product of basis functions retain a partition of unity:

$$\sum_k L_k(x) \equiv 1 \quad x \in [a, b] \quad a, b \in \mathbb{R}$$

$$\begin{aligned} N_i &= N_{3^2r+3s+t} = L_r L_s L_t \\ \sum_i N_i &= \sum_r \sum_s \sum_t L_r L_s L_t \\ &= \sum_r (L_r \sum_s (L_s \sum_t L_t)) \\ &\equiv 1 \end{aligned} \quad (3.8)$$

As a result, the components of a cell's mass matrix add up to its total mass.

$$\begin{aligned} \sum_i \sum_j M_{ij}^e &= \sum_i \sum_j \int_{\Omega^e} \rho N_i N_j d\Omega \\ &= \int_{\Omega^e} \rho \sum_i \sum_j N_i N_j d\Omega \\ &= \int_{\Omega^e} \rho \sum_i \left(N_i \sum_j N_j \right) d\Omega \\ &= \int_{\Omega^e} \rho d\Omega \\ &= m^e \end{aligned} \quad (3.9)$$

This proves useful in the SCM, since lumping can be performed using only the consistent element mass matrix obtained from adaptive integration, without having to separately calculate the total mass.

$$M_{ii}^{e,HRZ} = \frac{\sum_k \sum_l M_{kl}^e}{\sum_n M_{nn}^e} M_{ii}^e \quad (3.10)$$

3.5 Results

In this section, the spectral cell method is applied on a simple problem described in 2.4 with different discretizations and embeddings, and then to an example with a more complex geometry that has a cell distribution more resembling a practical application. For the simple case, approaches without lumping, with density scaling, and with HRZ lumping are compared. All computations are carried out using a modified version of AdhoC++, an in-house high order finite element method developed at the chair of Computational Modeling and Simulation at the Technical University of Munich.

3.5.1 Axis-aligned Bar

A simple straight bar model introduced in 2.4 is considered. To allow a fictitious domain extension and analyze the behaviour of the SCM, the problem is modeled in 3D instead of one.

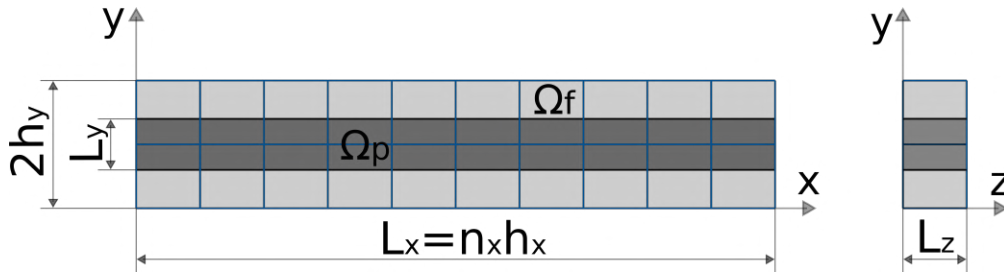


Figure 3.2: Embedded axis-aligned bar and its discretization.

A bar of length L_x with a rectangular cross-section $L_y \times L_z$ is symmetrically embedded into a Cartesian domain of size $L_x \times 2h_y \times L_z$ that does not conform to the bar in y -direction. The domain is discretized by an $n_x \times 2 \times 1$ Cartesian grid of identical cells.

The distributed source defined in equation 2.25 is extended to 3D but only in the physical domain:

$$f(\mathbf{x}, t) = \begin{cases} e^{-10^4 x^2} \sin\left(\frac{2\pi}{T} t\right) & t \in [0, \frac{T}{2}], \mathbf{x} \in \Omega_p \\ 0 & \text{otherwise} \end{cases} \quad (3.11)$$

| geometry | | | material | | source | time | |
|----------|-------|-------|----------|-----|--------|-------------------|-----------|
| L_x | h_y | L_z | ρ | E | T | Δt | t_{max} |
| 0.5 | 0.01 | 0.01 | 1 | 1 | 0.1 | $2 \cdot 10^{-4}$ | 0.5 |

Table 3.1: Constant parameters for the bar example.

Time integration is performed using central differences described in 2.1.2 for every case. In the following subsections, the behaviour of solutions obtained by different approaches is studied, depending on various quantities such as the:

- number of elements in x -direction n_x
- order of basis functions p
- fill ratio of cells η
- fictitious exponent β

Fill Ratio

Since the mass matrices of uncut cells are inherently diagonal, lumping schemes have no effect on them. However, as the proportion of the physical domain in a cell decreases, the role of lumping becomes more dominant. The fill ratio η shows the proportion of the physical domain relative to the total volume of a cell. In the bar example, this is controlled by varying the height L_y of the bar while leaving the mesh untouched to avoid errors originating from element skewness. Hence, the fill ratio can be computed as follows:

$$\eta = \frac{L_y}{2h_y} \quad (3.12)$$

This value is identical for every cell in the model. To make sure that no errors originate from the numerical integration's adaptive nature, the bar's height is always set such that an octree of depth r can perfectly capture its boundary. More precisely, the bar is shrunk sequentially by powers of two: $L_y = 2^{-r} 2h_y$

| | | | | | |
|--------|--------|---------------|-----------------|-----------------|-----------------|
| r | 0 | 1 | 2 | 3 | 4 |
| L_y | $2h_y$ | h_y | $\frac{h_y}{2}$ | $\frac{h_y}{4}$ | $\frac{h_y}{8}$ |
| η | 1 | $\frac{1}{2}$ | $\frac{1}{4}$ | $\frac{1}{8}$ | $\frac{1}{16}$ |

Table 3.2: Cell fill ratios for different bar heights.

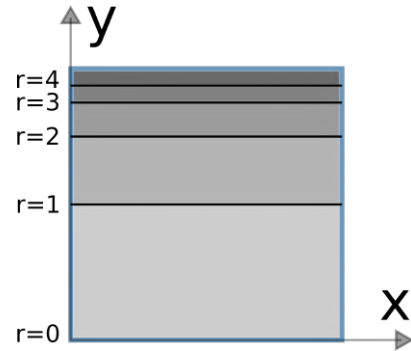


Figure 3.3: Example bottom left cell for different bar heights.

For all following cases, the number of elements in x -direction $n_x = 30$ is fixed and p-refinement is performed $p = 1, \dots, 4$. The indicator function $\alpha(\mathbf{x})$ is characterized by the fictitious exponent $\beta = 5$. Displacements are evaluated at two sample points defined in section 2.4, and the relative error defined in equation 2.30 is computed using wavelet peaks.

Note, that the mesh conforms to the boundary for $r = 0$, which means that no adaptive integration or lumping is performed, and the solution is identical to that of the standard SEM. This case can be used as reference.

Without lumping, the solutions shown in figure 3.4 converge as expected. Spurious oscillations decay as p increases and the effective wave speed approaches the analytical one. The fill

ratio has little impact on the results of embedded setups, but the boundary-conforming case visibly differs from them. This is due to the fact that the boundary-conforming case consists of uncut cells whose mass matrices are integrated using Lobatto quadrature, and are therefore underintegrated. On the other hand, cut cells in the non-conforming cases are exactly integrated using the adaptive scheme based on Gauss-Legendre quadrature. Since no lumping is performed, the mass matrix has non-zero off-diagonal entries from the adaptive integration, thus leading to less efficient time integration.

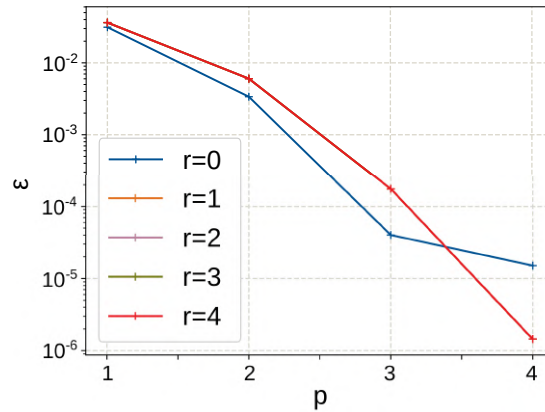


Figure 3.4: Relative time-of-flight error for p-refinement on different fill ratios without lumping. The embedded cases $r \neq 0$ overlap within line width.

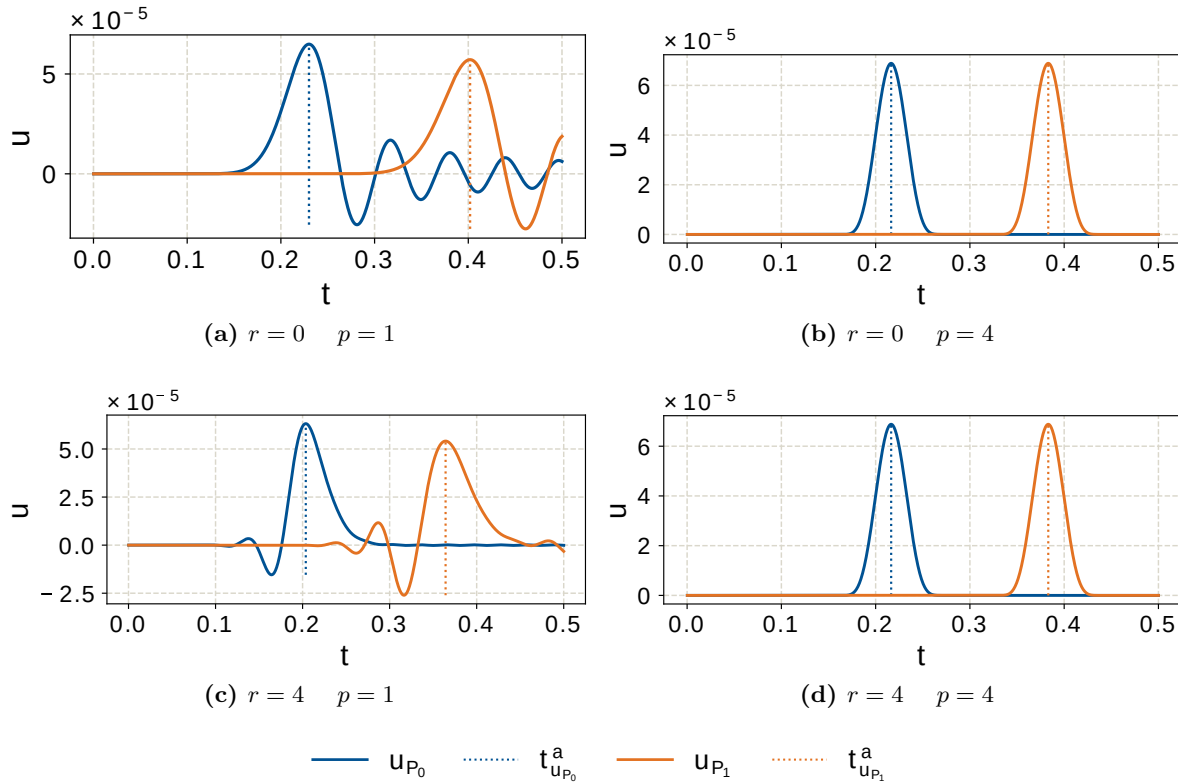


Figure 3.5: Displacement history at the two points for $r = \{0, 4\}$, $p = \{1, 4\}$ and no lumping.

Next, HRZ lumping is applied to the same scenarios. As seen in figure 3.6, this method is

unable to consistently benefit from p-refinement and does not converge in general. To gain a better understanding of why this is the case, the displacements at the two sample points and then on the entire model are studied.

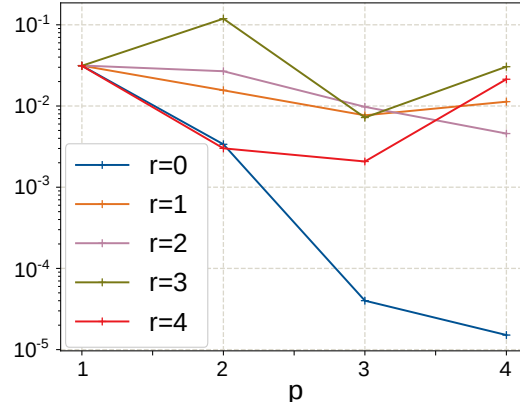


Figure 3.6: Relative time-of-flight error for p-refinement on different fill ratios with HRZ lumping.

Figure 3.7 shows the displacement histories at the two sample points for a fill ratio of $\eta = \frac{1}{2}$ with p ranging from 1 to 4. At first, increasing the order of the basis seem to attenuate the magnitude of spurious oscillations, but at higher orders this trend is clearly disproven. An interesting observation is that the oscillations occur exclusively after the main wavelet, and resemble decaying harmonic functions. Their source can be better understood by studying how the displacement field changes over time.

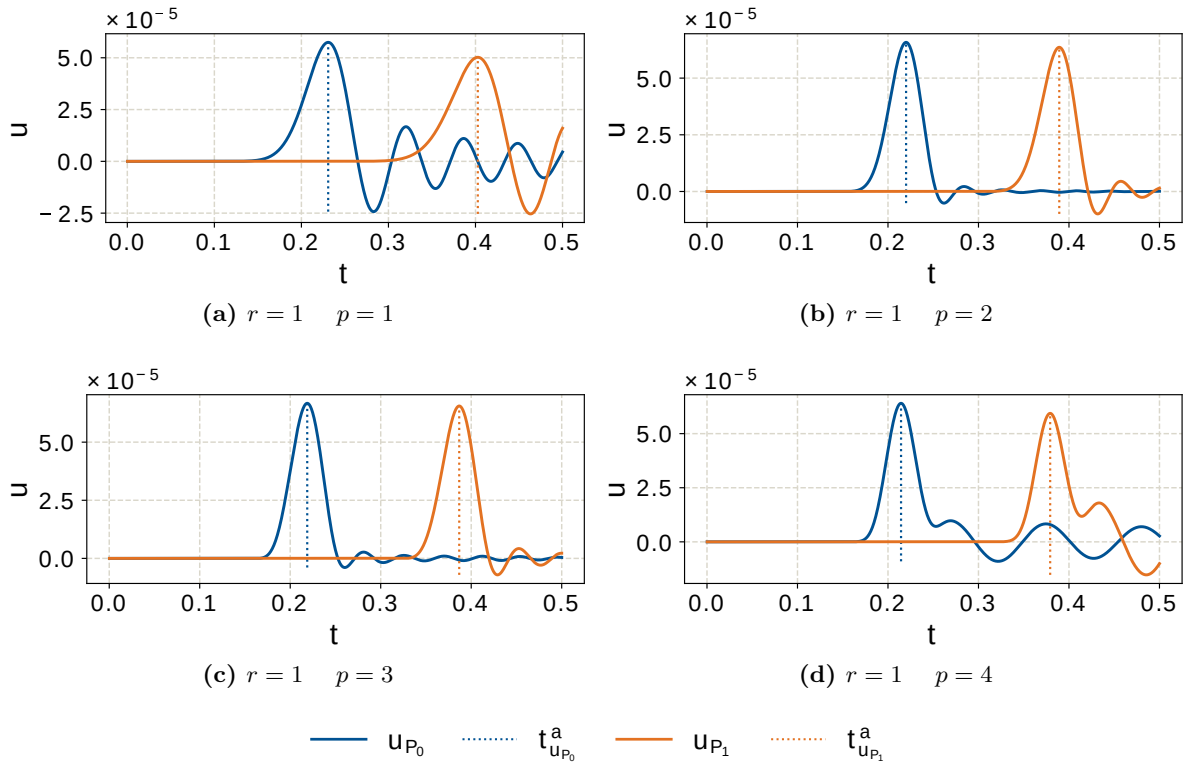


Figure 3.7: Displacement history at the two points for $r = 1$, $p \in \{1, \dots, 4\}$ and HRZ lumping.

Figure 3.8 compares the displacement field with HRZ lumping to the solution without lumping at different time points, for the setup $r = 1$ and $p = 4$. The solution without lumping can be regarded as reference, and as expected, the displacement is constant in y -direction, even in the fictitious domain. The wavelet propagates in the x -direction without skewing, and no reflections occur on the boundary between the physical and fictitious domains. In contrast, HRZ lumping introduces a variance in the y -direction. It is important to note that for better visibility, the displacement color range is adjusted for magnitudes in the physical domain $\pm 7 \cdot 10^{-5}$, but this range is almost an order of magnitude greater in the fictitious domain $\pm 2.5 \cdot 10^{-4}$. Furthermore, the wave speed appears to differ in Ω_f which leads to a wave front propagating in y -direction, endlessly reflected between the boundary of the domains and the mesh. In the FCM, the displacement magnitudes in Ω_f must be at least in the order of 10^β greater than the solution of the physical domain to have notable effect on it. Due to the artificial coupling between the physical and fictitious domains introduced by HRZ lumping however, the waves originating from Ω_f propagate into the physical domain as well, albeit with reduced amplitude. The resulting spurious oscillations can be clearly observed in the displacement history of the sample points as well, shown in figure 3.7d. Their magnitude, frequency, and phase relative to wavelet depend on the size of the fictitious domain and the material properties inside it, defined by β .

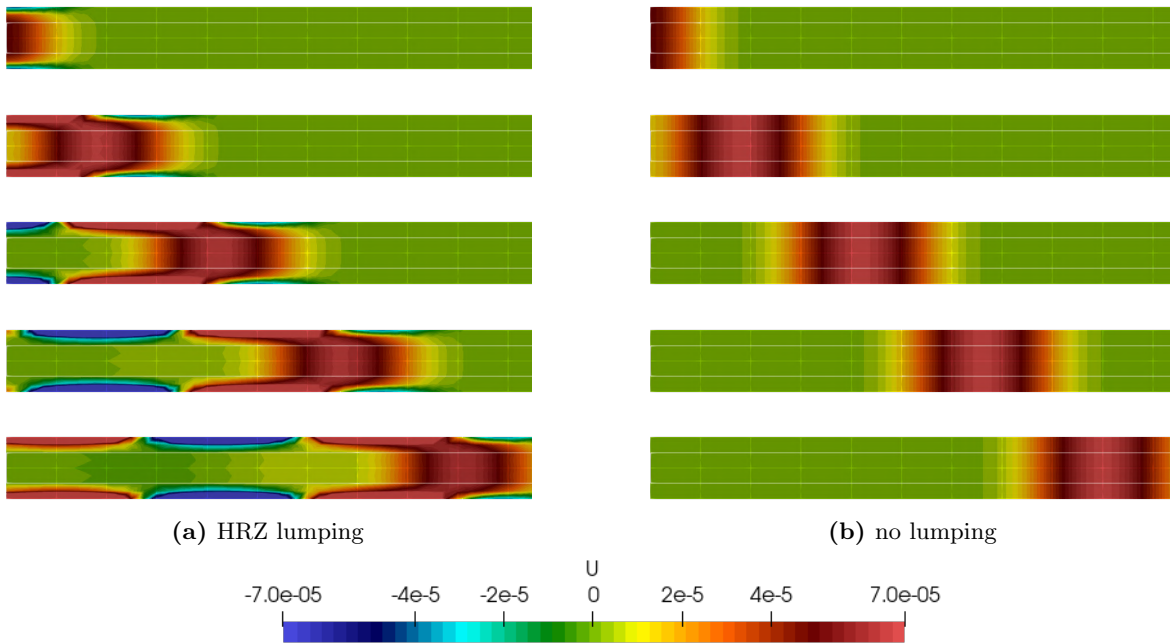


Figure 3.8: Cropped top view of the displacement field at $t = \{0.04, 0.08, 0.12, 0.16, 0.2\}$ with HRZ lumping (left) and without lumping (right). The fill ratio $\eta = \frac{1}{2}$ and basis order $p = 4$ is identical for both cases.

In an attempt to dispose of the solution's variance in y -direction, a model with the source term f extended to the fictitious domain and multiplied by the indicator function α was studied as well, but yielded similar results.

Finally, density scaling is studied for the same setup. Considering that it is similar to HRZ lumping but with more simplifications, the fact that it performs worse is not surprising. However, an interesting property can be observed in the displacement histories that may help better understand why neither lumping schemes are viable. Originally, both the density

ρ and Young's modulus E are multiplied by $10^{-\beta}$ in the fictitious domain, which means that the wave speed remains constant over the entirety of the model. As mentioned in 3.4.2 however, density scaling imposes a uniform mass distribution on cut cells. Apart from introducing an artificial coupling between domains similar to HRZ lumping, this means that the effective wave speed becomes higher in the physical domain while lower in the fictitious one. Furthermore, this speed difference varies from cell to cell, depending on their individual fill ratios. Though combined with other errors, this behaviour can be seen in figure 3.10. As the fill ratio decreases, the propagating wavelet becomes faster in the physical domain. For fill ratios of $\eta = \frac{1}{8}$ and $\eta = \frac{1}{16}$, the wave is reflected from the right boundary and even reaches P_2 on its way back before t_{max} .

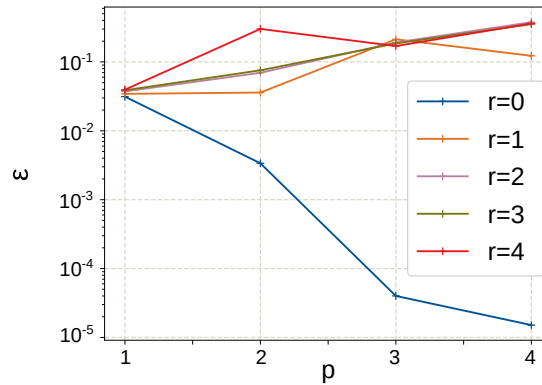


Figure 3.9: Relative time-of-flight error for p-refinement on different fill ratios with density scaling.

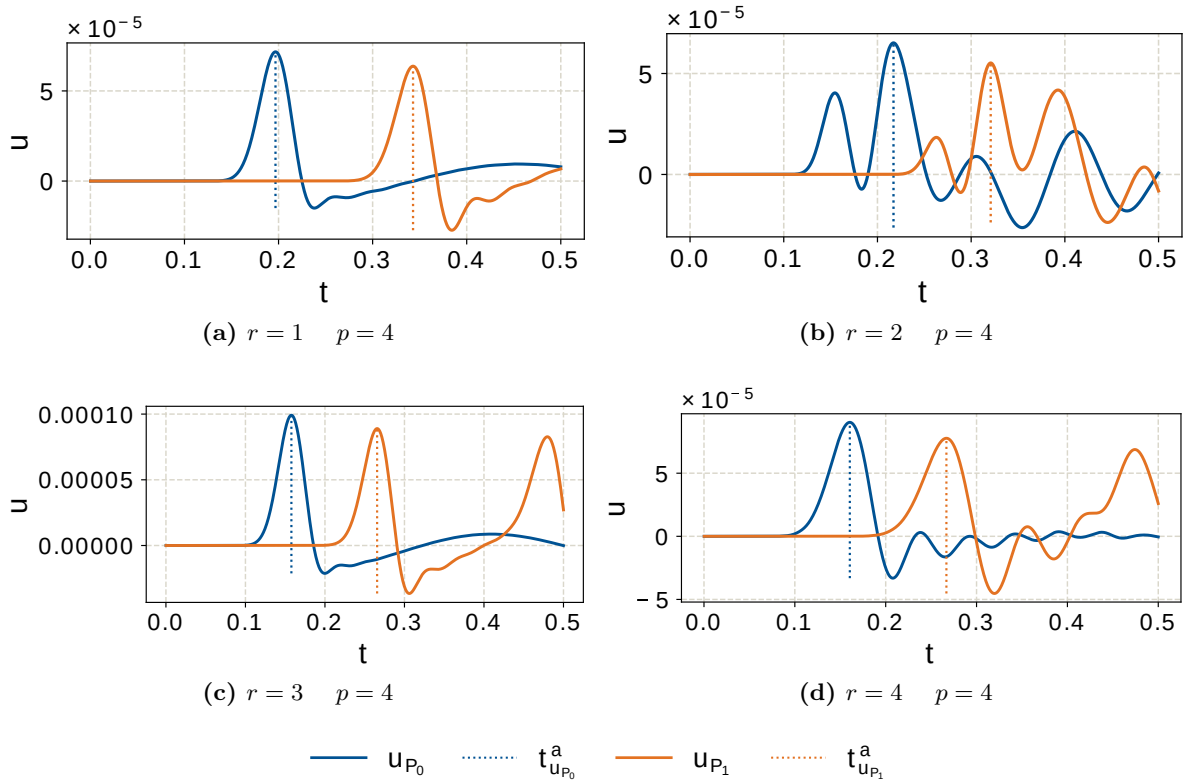


Figure 3.10: Displacement history at the two points for $r \in \{1, \dots, 4\}$, $p = 4$ and density scaling.

In conclusion, neither lumping schemes are viable. An interesting property of both methods

is that due to the artificially coupled domains, the magnitude, frequency, and phase of the spurious oscillations vary with η and p . To show how important and non-trivial the impact the displacement in the fictitious domain has on the physical one, the solution's dependence on the fictitious exponent β is analyzed in the next section. As density scaling is similar to HRZ lumping but clearly inferior, it is not studied in the rest of this thesis.

Fictitious Exponent

Introduced in equation 2.16, the fictitious exponent β defines the relation between the material parameters in the physical and fictitious domains. Its original purpose in the FCM is to avoid ill conditioned structural matrices for badly cut cells. Small values lead to more stable numerics but less accurate physics, thus one has to find an acceptable compromise between the two types of errors. That said, there is no detailed study on its effect, and its value is often arbitrarily chosen between 3 and 10 [PDR07]. Due to lumping however, the solution in the fictitious domain has a greater impact, making β more important in the SCM.

In the following examples, the fill ratio $\eta = \frac{1}{4}$ and basis order $p = 4$ remain constant while the β varies between 0 and 10. Since the focus is on the residual oscillations in this case, the time of arrival is computed based on the envelope of the displacement history. The specifics of this process and reason for doing so are discussed in 2.4.

Figure 3.11 shows how the resulting error changes with the fictitious exponent. At $\beta = 0$, the material in the two domains are identical, thus the stiffness and mass matrices are consistent. Lumping has minimal effect, and no oscillations occur in this case. In the range $\beta \in [2, 6]$ however, oscillations appear and vary in magnitude, frequency, and phase.

The source of these oscillations in the fictitious domain and their varying wavelengths can be observed in figure 3.13, which compares the displacement fields progress over time for $\beta = 3.5$ and $\beta = 4.5$. Note, that unlike in figure 3.8, the color scale covers the entirety of the displacement magnitude in this case, in order to focus on the fictitious domain.

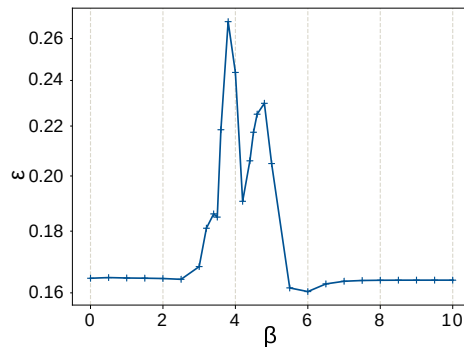


Figure 3.11: Relative time-of-flight error for varying fictitious exponents β .

Although for this specific set of configurations, oscillations occur in a range of β but disappear completely for $\beta > 6$. The same does not hold true in general, and spurious oscillations persist for large fictitious exponents as well. The only general rule is that no such oscillations can be observed for uniform material parameters ($\beta = 0$) and that the behaviour of the solution greatly depends on β .

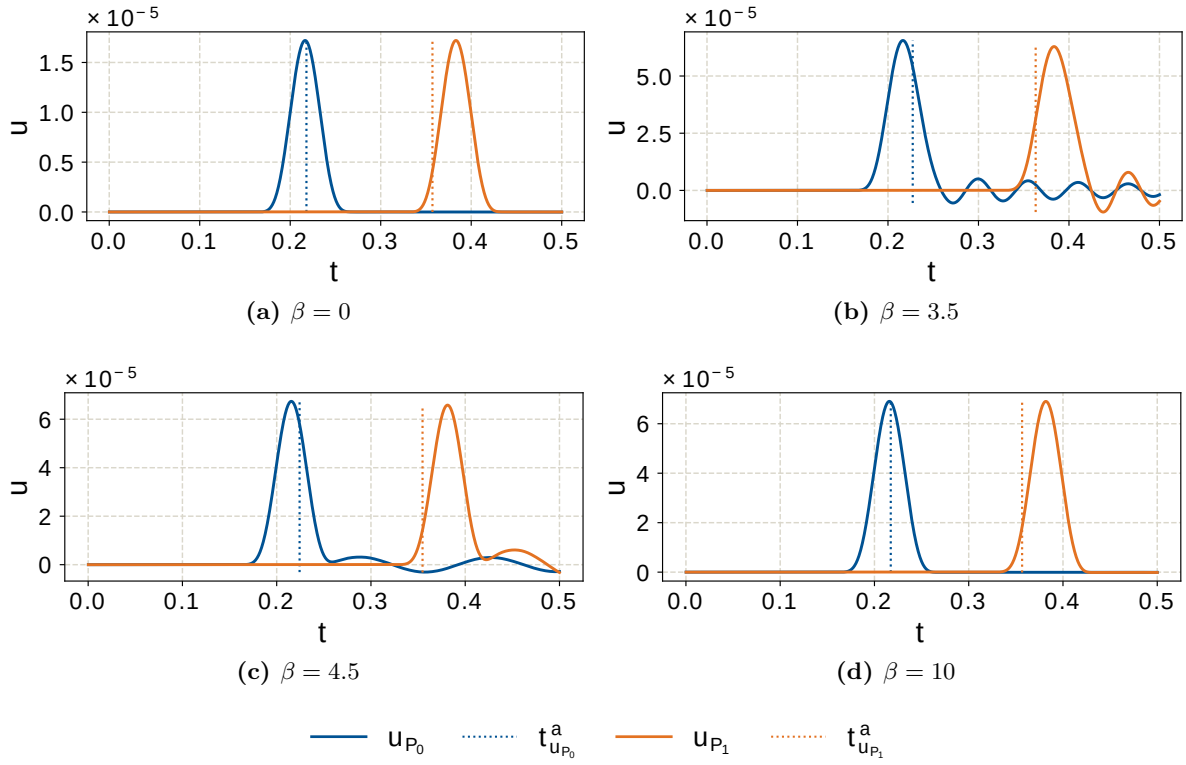


Figure 3.12: Displacement history at the two points for $\eta = \frac{1}{4}$, $p = 4$ and HRZ lumping.

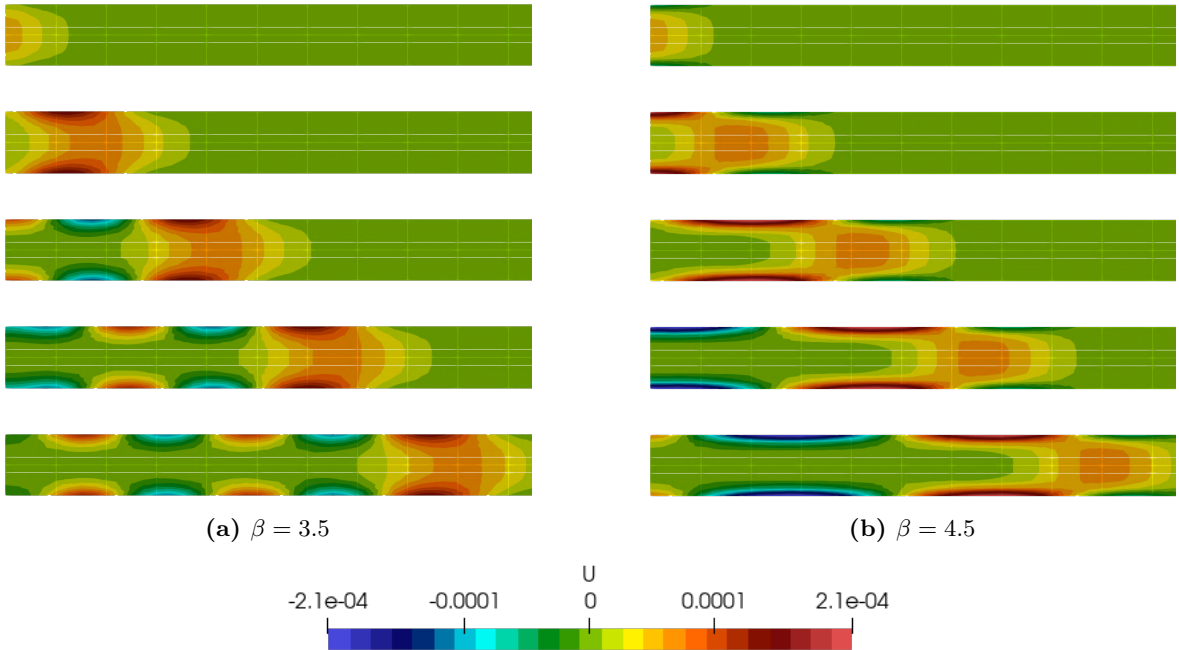


Figure 3.13: Cropped top view of the displacement field at $t = \{0.04, 0.08, 0.12, 0.16, 0.2\}$, $\beta = \{3.5, 4.5\}$ with HRZ lumping. The fill ratio $\eta = \frac{1}{4}$ and basis order $p = 4$ is identical for both cases.

3.5.2 Rotated Bar

In this section, the bar is rotated by 45 degrees relative to the mesh and embedded into a Cartesian domain of size $\frac{L_x+L_y}{\sqrt{2}} \times \frac{L_x+L_y}{\sqrt{2}} \times L_z$, discretized by $30 \times 30 \times 1$ identical cells with a basis order of $p = 4$. The most important difference between the rotated and axis-aligned cases is that the geometry cannot be exactly captured by a standard octree, leading to integration errors. By increasing the octree's depth, this error can be arbitrarily reduced at the expense of an exponential increase in complexity. As a compromise between accuracy and performance, the maximum depth is set to $r = p + 1 = 5$. All other parameters are identical to the ones used in previous examples.

In practical implementations, cells entirely in the fictitious domain are filtered out as they are not part of the original problem and add superfluous degrees of freedom, increasing the computational load. In this example, the solution in the fictitious domain is also of interest, hence such cells are present in the model as well. The resulting mesh contains uncut cells fully in one of the domains, cells cut roughly in half, and badly cut cells that have a small region of physical domain around one of their corners.

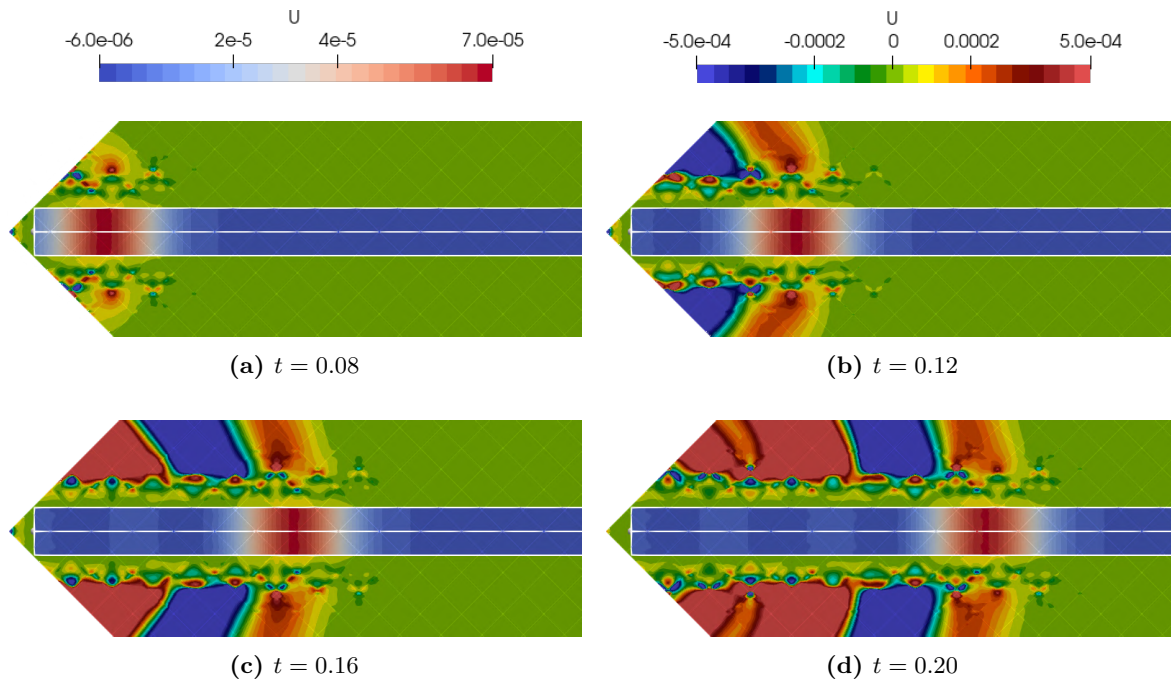


Figure 3.14: Displacement fields of the rotated bar example at different time points. Note, that the physical and fictitious domains have separate ranges and color maps on account of large differences in magnitude.

Similarly to the axis-aligned cases, spurious oscillations are observed in the physical domain, and the range of displacements in the fictitious domain is roughly an order of magnitude greater. As shown in figure 3.16, the displacement of cut cells sharply increases in magnitude in regions opposite the physical domain. The solution in these areas vary greatly across adjacent cells and oscillate with little decay after the wave passes, disturbing the solution in the physical domain.

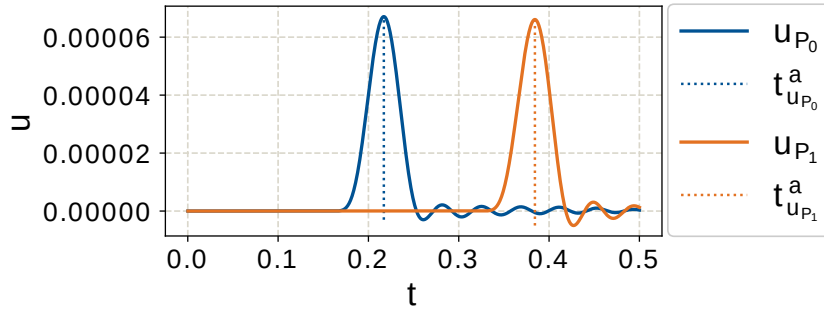


Figure 3.15: Displacement history at the two sample points of the rotated bar.

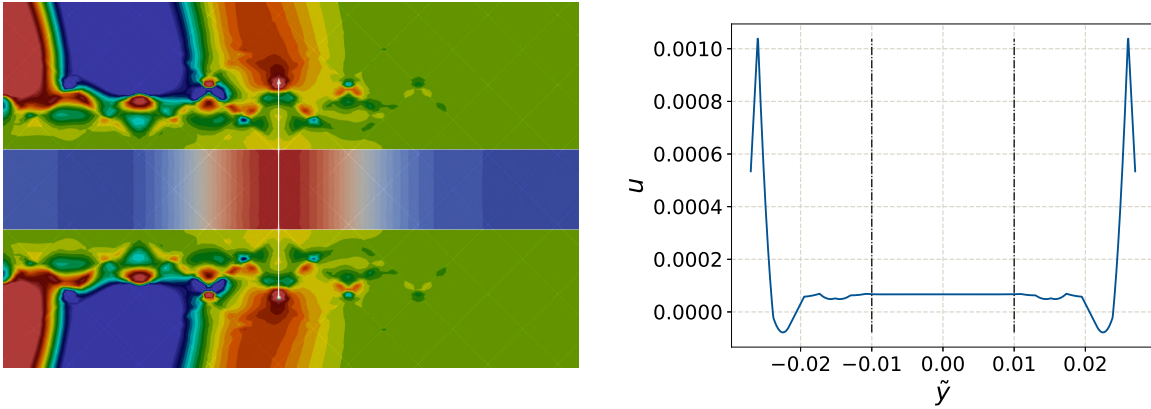


Figure 3.16: Displacement along an extended cross-section of the bar, cutting elements along their diagonals.

3.5.3 Complex Geometry

The examples so far offered little variety in cell boundary shapes and fill ratios, which may bias the conclusions drawn from their results. To see how the method with HRZ lumping performs in a more practical setting, a geometrically more complex model is studied that provides a variety of fill ratios. The model is similar to the ones before, but instead of a bar, the wavelet is sent down the longest axis of an ellipsoid cut in half, defined as:

$$\Omega_p = \left\{ \mathbf{x} \mid x \leq 1 \wedge \left(\frac{x-1}{1} \right)^2 + \left(\frac{y-0.2}{0.2} \right)^2 + \left(\frac{z-0.1}{0.1} \right)^2 \leq 1 \right\} \quad (3.13)$$

In order to compensate for the sharp changes in curvature at the tip of the ellipsoid, the source's shape f_x is slightly modified to decay less abruptly:

$$f(\mathbf{x}, t) = \begin{cases} e^{-10^3 x^2} \sin\left(\frac{2\pi}{T} t\right) & t \in \left[0, \frac{T}{2}\right], \mathbf{x} \in \Omega_p \\ 0 & \text{otherwise} \end{cases} \quad (3.14)$$

The Cartesian embedding domain $\Omega = [0, 1] \times [0, 0.4] \times [0, 0.2]$ is discretized by a $60 \times 24 \times 12$ mesh of cells with a Lagrange basis of order $p = 4$. Cells fully in the fictitious domain are

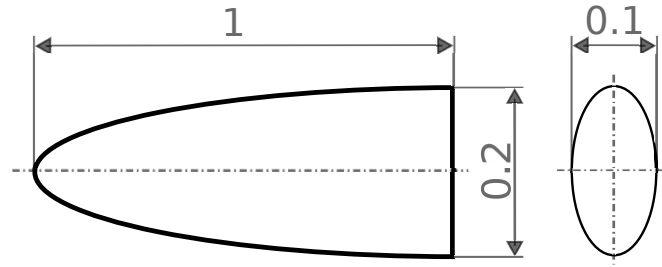


Figure 3.17: Geometry of the physical domain.

neglected. The distribution of cells' fill ratios is shown in figure 3.18, indicating that most cells are uncut and the rest have diverse fill ratios. The mass matrices of uncut cells are integrated with a fifth order Gauss-Lobatto quadrature, while an adaptive scheme with Gauss-Legendre quadrature of identical order is applied to cut cells, then lumped using the HRZ procedure. A reference solution without lumping is computed as well.

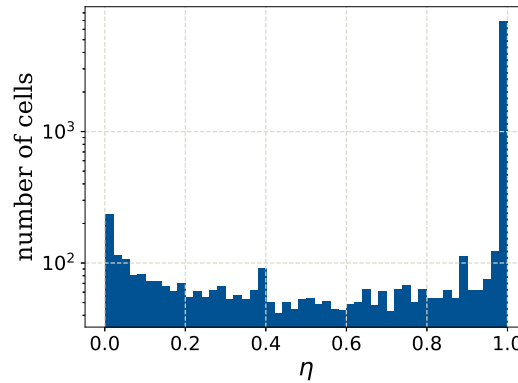


Figure 3.18: Histogram of the cells' fill ratios.

Similarly to the bar example, the displacement history is sampled at two points P_0 and P_1 that trisect the major axis of the half-ellipsoid. Furthermore, a marching cube algorithm is used to recover the geometry's surface and interpolate the solution field on it.

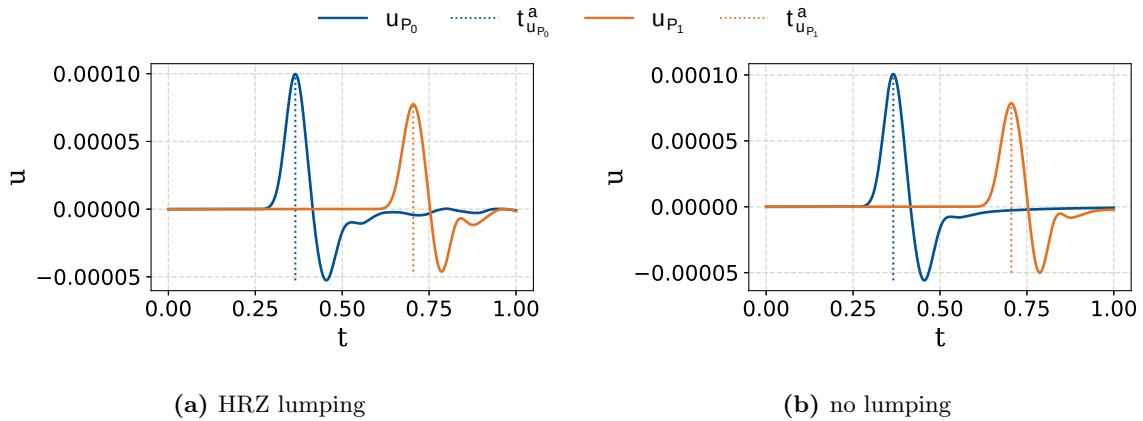


Figure 3.19: Displacement histories at the two sample points on the ellipsoid's major axis with HRZ lumping (left) and without lumping (right).

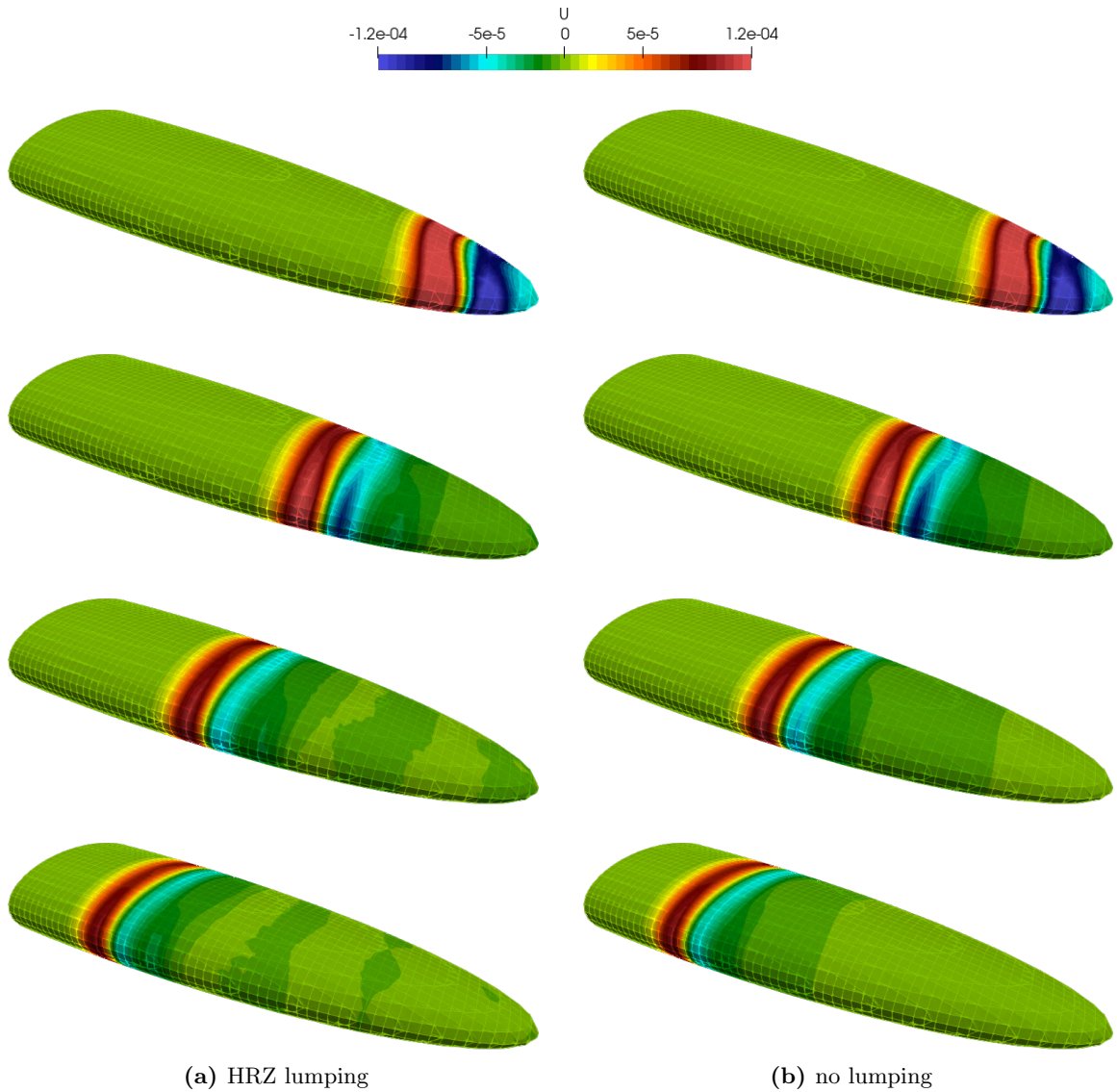


Figure 3.20: Displacement fields on the surface of the ellipsoid at $t = \{0.2, 0.4, 0.6, 0.8\}$, computed with HRZ lumping (left) and no lumping as reference (right).

As figure 3.19 shows, the spurious oscillations originating from cut cells are similar to the ones in the bar examples, and are superposed on the reference solution. However, their influence is more modest due to relatively smaller element sizes, and fewer cut cells compared to uncut ones. Figure 3.21 shows the displacement history on the cross-section at P_0 between $t \in [0.67, 0.82]$, well after the wavelet passed. The solution in the fictitious domain is seen to persistently oscillate, even without notable excitation coming from the physical domain.

In conclusion, the errors introduced by HRZ lumping are similar to earlier results and are still unacceptable even though the mesh mainly consists of uncut cells.

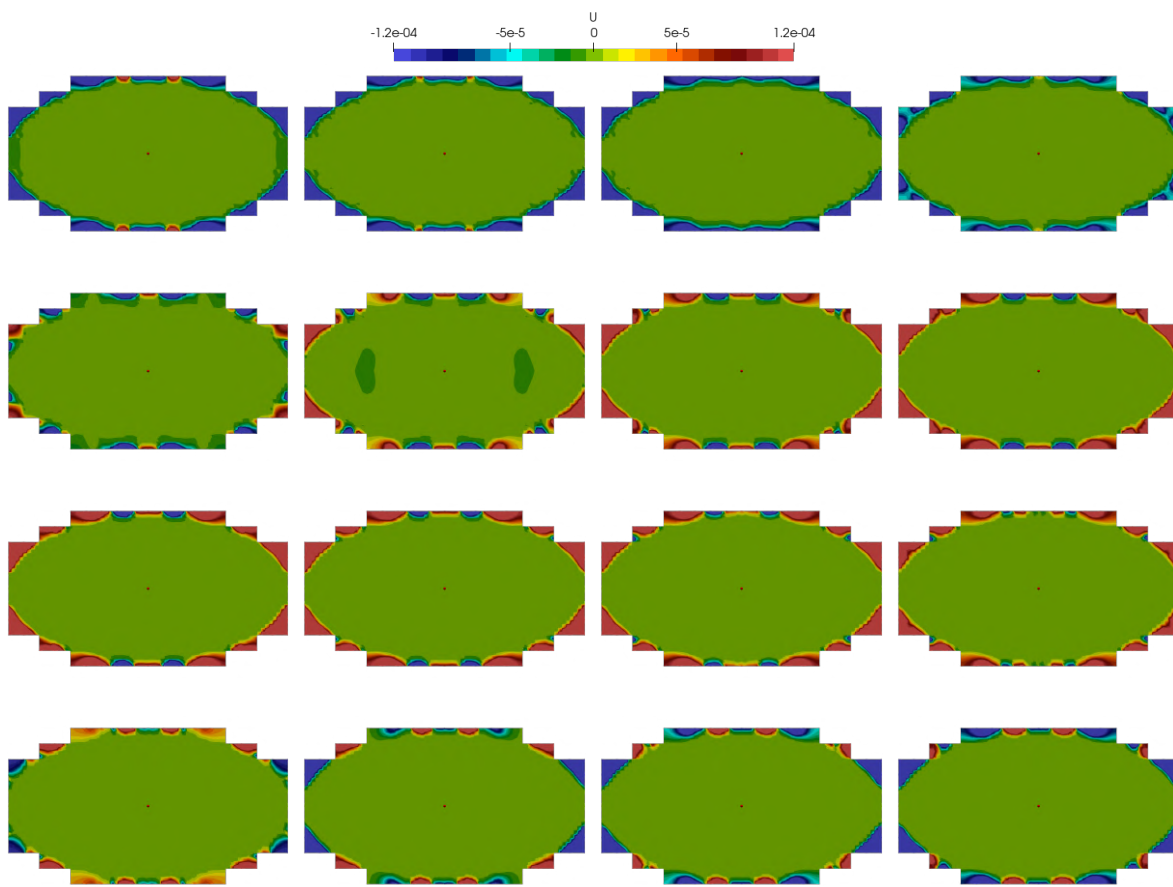


Figure 3.21: Displacement field on the cross-section of the ellipsoid at P_0 between $t = 0.67$ and $t = 0.82$ in increments of $\Delta t = 0.01$.

Chapter 4

Conclusion

This thesis covers potential approaches to diagonalizing the mass matrices of cut cells in the spectral cell method, including moment fitting and different mass lumping schemes. Various examples and evaluation methods are presented as well, revealing mixed results and the need for further research in this field.

After covering the relevant theoretical background of the SCM's roots, the finite cell method and the spectral element method, the mass matrices of cut cells are identified as the critical subject to focus on, as they contain non-zero off-diagonal entries. Without changing the basis functions of such elements, the two possible approaches of addressing the issue are either modifying the numerical integration scheme to lead to an inherently diagonal matrix, or applying a lumping scheme to the non-diagonal mass matrix after it is computed.

Although moment fitting preserves the consistent nature of the mass matrix, its reduced integration accuracy can lead to negative masses and a divergent solution in time. Used successfully for the linear FEM, row sum lumping suffers from the same issue at higher basis orders, rendering both approaches infeasible for use in the SCM.

The remaining two lumping schemes, density scaling and HRZ lumping, guarantee the positive-definiteness of the mass matrix. Even though both of these techniques were successfully applied to 2D models of linear elasticity in the literature [JDGD14], accurate results for 3D models of the acoustic wave equation could not be achieved in this thesis, hinting at the fact that these approaches have limited applicability. Both schemes are similar in nature but HRZ lumping is shown to produce less errors than density scaling.

The most important shortcoming of HRZ lumping is the fact that it introduces an artificial coupling between the physical and fictitious domains in cut cells, leading to spurious oscillations propagating into the rest of the model. Furthermore, displacement magnitudes become amplified in the fictitious domain.

As none of the mentioned approaches to diagonalization yield satisfactory results, further research is encouraged to evaluate other lumping schemes, or address the accuracy of moment fitting.

It is important to note, that although adequate diagonal mass matrices of cut cells could not be derived, global mass matrices with only few non-zero off-diagonal entries still lead to superior performance at time stepping when using direct linear solvers. As a result, on

models with a relatively small ratio of cut cells the SCM without lumping offers faster time integration than the FCM, while providing similar accuracy and convergence.

List of Figures

| | | |
|-----|---|----|
| 2.1 | Example of an FCM mesh embedding the physical domain | 9 |
| 2.2 | Embedding mesh subdivided with a quadtree of depth 2. | 11 |
| 2.3 | Lagrange basis functions on a set of fourth order Gauss-Lobatto points. | 13 |
| 2.4 | Simple 1D bar with two sample points. | 14 |
| 2.5 | Displacement history at the two sample points for a reference mesh on the left 2.5a and a coarse one on the right 2.5b. | 14 |
| 2.6 | Time of arrival based on the envelope of the displacement at P_1 of the coarse solution. | 15 |
| 2.7 | Example time of arrival offset as a result of a long time of observation and spurious oscillations. | 15 |
| 2.8 | Time of arrival computed from a squared envelope. | 16 |
| 2.9 | Time of arrival based on a cubic interpolating spline. | 16 |
| 3.1 | Square cell cut in half along the y -axis with a Lagrange basis of order 2, and the corresponding Gauss-Lobatto points. | 21 |
| 3.2 | Embedded axis-aligned bar and its discretization. | 24 |
| 3.3 | Example bottom left cell for different bar heights. | 25 |
| 3.4 | Relative time-of-flight error for p-refinement on different fill ratios without lumping. The embedded cases $r \neq 0$ overlap within line width. | 26 |
| 3.5 | Displacement history at the two points for $r = \{0, 4\}$, $p = \{1, 4\}$ and no lumping. | 26 |
| 3.6 | Relative time-of-flight error for p-refinement on different fill ratios with HRZ lumping. | 27 |
| 3.7 | Displacement history at the two points for $r = 1$, $p \in \{1, \dots, 4\}$ and HRZ lumping. | 27 |
| 3.8 | Cropped top view of the displacement field at $t = \{0.04, 0.08, 0.12, 0.16, 0.2\}$ with HRZ lumping (left) and without lumping (right). The fill ratio $\eta = \frac{1}{2}$ and basis order $p = 4$ is identical for both cases. | 28 |

| | | |
|------|---|----|
| 3.9 | Relative time-of-flight error for p-refinement on different fill ratios with density scaling. | 29 |
| 3.10 | Displacement history at the two points for $r \in \{1, \dots, 4\}$, $p = 4$ and density scaling. | 29 |
| 3.11 | Relative time-of-flight error for varying fictitious exponents β | 30 |
| 3.12 | Displacement history at the two points for $\eta = \frac{1}{4}$, $p = 4$ and HRZ lumping. | 31 |
| 3.13 | Cropped top view of the displacement field at $t = \{0.04, 0.08, 0.12, 0.16, 0.2\}$, $\beta = \{3.5, 4.5\}$ with HRZ lumping. The fill ratio $\eta = \frac{1}{4}$ and basis order $p = 4$ is identical for both cases. | 31 |
| 3.14 | Displacement fields of the rotated bar example at different time points. Note, that the physical and fictitious domains have separate ranges and color maps on account of large differences in magnitude. | 32 |
| 3.15 | Displacement history at the two sample points of the rotated bar. | 33 |
| 3.16 | Displacement along an extended cross-section of the bar, cutting elements along their diagonals. | 33 |
| 3.17 | Geometry of the physical domain. | 34 |
| 3.18 | Histogram of the cells' fill ratios. | 34 |
| 3.19 | Displacement histories at the two sample points on the ellipsoid's major axis with HRZ lumping (left) and without lumping (right). | 34 |
| 3.20 | Displacement fields on the surface of the ellipsoid at $t = \{0.2, 0.4, 0.6, 0.8\}$, computed with HRZ lumping (left) and no lumping as reference (right). | 35 |
| 3.21 | Displacement field on the cross-section of the ellipsoid at P_0 between $t = 0.67$ and $t = 0.82$ in increments of $\Delta t = 0.01$ | 36 |

List of Tables

| | | |
|-----|---|----|
| 3.1 | Constant parameters for the bar example. | 24 |
| 3.2 | Cell fill ratios for different bar heights. | 25 |

Bibliography

- [DPYR08] A. Düster, J. Parvizian, Z. Yang, and E. Rank. The finite cell method for three-dimensional problems of solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 197(45-48):3768 – 3782, 2008.
- [PDR07] J. Parvizian, A. Düster, E. Rank. Finite cell method h- and p-extension for embedded domain problems in solid mechanics *Computational Mechanics*, 41(121-133), 2007
- [BBO04] I. Babuška, U. Banerjee, J. Osborn. Generalized Finite Element Methods: Main Ideas, Results, and Perspective. , 2004
- [CLE06] Y. Chen, J. Lee, A. Eskandarian. Meshless Methods in Solid Mechanics. , 2006
- [MKO13] B. Müller, F. Kummer, M. Oberlack. Highly accurate surface and volume integration on implicit domains by means of moment-fitting , 2013
- [JHD15] J. Meysam, S. Hubrich, A. Düster. Numerical integration of discontinuities on arbitrary domains based on moment fitting , 2015
- [MQ94] F. Maggio, A. Quarteroni. Acoustic wave simulation by spectral methods *East-West J. Numer. Math.*, 2 (129-150), 1994
- [Z96] G.W. Zumbusch. Symmetric Hierarchical Polynomials for the h-p-Version of Finite Elements *Houston Journal of Mathematics*, (529-540), 1996
- [K96] B.D. Keister. Multidimensional Quadrature Algorithms *Computers in Physics*, 10 (119-122), 1996
- [AW10] M. Ainsworth, H.A. Wajtid. Optimally Blended Spectral-Finite Element Scheme for Wave Propagation and Nonstandard Reduced Integration *SIAM Journal on Numerical Analysis*, 48 (346-371), 2010
- [LB13] M.G. Larson, F. Bengzon. The Finite Element Method: Theory, Implementation and Applications. , 2013
- [ZW14] S. Zhao, G.W. Wei. A unified discontinuous Galerkin framework for time integration. *Mathematical Methods in the Applied Sciences*, 37(1042-1071), 2014
- [KB14] U. Köcher, M. Bause. Variational Space-Time Methods for the Wave Equation *IEEE transactions on magnetics*, 47(1530-1533), 2014
- [CFL67] R. Courant, K. Friedrichs, H. Lewy. On the Partial Difference Equations of Mathematical Physics. *IBM Journal of Research and Development*, 11(215-234), 1967

- [N59] N.M. Newmark. A Method of Computation for Structural Dynamics. *Journal of the Engineering Mechanics Division*, 85, 1959
- [A88] K.E. Atkinson. An Introduction to Numerical Analysis: Second Edition. , 1988
- [HRZ76] E. Hinton, T. Rock, and O.C. Zienkiewicz. A note on mass lumping and related processes in the finite element method. *Earthquake Engineering and Structural Dynamics*, 4(245-249), 1976.
- [DG19] S. Duczek, H. Gravenkamp. Critical Assessment of different mass lumping schemes for higher order serendipity finite elements. *Computational Methods in Applied Mechanical Engineering*, 350(836-897), 2019.
- [H00] T.J.R. Hughes. The finite element method: Linear static and dynamic finite element analysis. , 2000.
- [D14] S. Duczek. Higher order finite elements and the fictitious domain concept for wave propagation analysis. , 2014.
- [HSL98] N.E. Huang, Z.S. Shen, S.R. Long, M.C. Wu, H.H. Shih, Q. Zheng, N. Yen, C.C. Tung, H.H. Liu. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis *Proceedings: Mathematical, Physical and Engineering Sciences*, 454 (903-995), 1998
- [DJDG14] S. Duczek, M. Joulaiian, A. Düster, U. Gabbert. Numerical analysis of Lamb waves using the finite and spectral cell methods *Int. J. Numer. Meth. Engng.*, 99 (26-53), 2014
- [JDGD14] M. Joulaiian, S. Duczek, U. Gabbert, A. Düster. Finite and spectral cell method for wave propagation in heterogeneous materials. *Computational Mechanics*, 54(661-675), 2014.
- [ZL13] W. Zhang, J. Luo Full-waveform Velocity Inversion Based on the Acoustic Wave Equation *American Journal of Computational Mechanics*, 3 (13-20), 2013
- [KKAR19] L. Kudela, S. Kollmannsberger, U. Almac, E. Rank. Direct structural analysis of domains defined by point clouds ,2019
- [A79] O.M. Alifanov. Inverse Heat Transfer Problems ,1979
- [BS94] I. Babuška, M. Suri. The p and h-p Versions of the Finite Element Method, Basic Principles and Properties , 1994