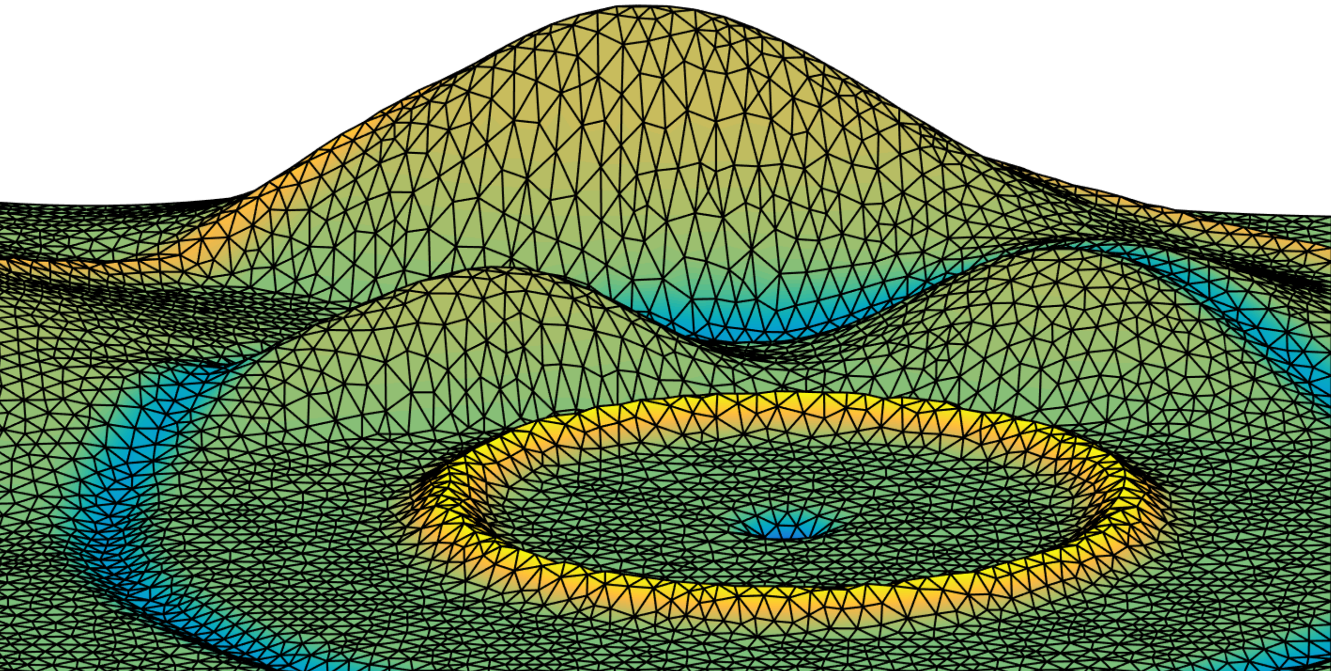
Finite Element Methods for Seismic Modelling

Sjoerd Geevers



Finite Element Methods for Seismic Modelling

Sjoerd Geevers

Graduation Committee

Chairman
Prof. Dr. J.N. Kok (Universiteit Twente)

Promotor
Prof. Dr. Ir. J.J.W. van der Vegt (Universiteit Twente)

Members
Prof. Dr. W.A. Mulder (Shell Global Solutions Int. BV, TU Delft)
Prof. Dr. Ir. B. Koren (TU Eindhoven)
Prof. Dr. C.W. Oosterlee (CWI, TU Delft)
Prof. Dr. J.G.M. Kuerten (Universiteit Twente)
Prof. Dr. H.J. Zwart (Universiteit Twente)

This work was funded by the Shell Global Solutions International B.V. under contract no. PT45999.

It was carried out in the group of Mathematics of Computational Science (MACS), Department of Applied Mathematics, Faculty of Electrical Engineering, Mathematics, and Computer Science (EEMCS) and MESA+ Institute for Nanotechnology of the University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.

ISBN: 978-90-365-4613-3
DOI: 10.3990/1.9789036546133

FINITE ELEMENT METHODS FOR SEISMIC MODELLING

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
Prof. Dr. T.T.M. Palstra,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op vrijdag 21 september 2018 om 16.45 uur

door

Sjoerd Geevers

geboren op 20 maart 1991
te Lemsterland, Nederland.

Dit proefschrift is goedgekeurd door de promotor Prof. Dr. Ir. J.J.W. van der Vegt.

Summary

New and more efficient finite element methods for modelling seismic wave propagation are presented and analysed in this dissertation.

Seismic modelling is a useful tool for better understanding seismic behaviour in complex rock structures, but it is also a key aspect of full waveform inversion, which is a powerful technique for imaging the structure of the earth's subsurface. The great advantage of finite element methods over other wave modelling methods, like the popular finite difference method, is that it accurately captures the effect of complex topographies, such as mountainous areas and rough seabeds, without refining the grid resolution. Even so, these methods require a huge amount of computational power and making them more efficient is of great value for many industrial applications.

The most promising finite element methods for wave propagation modelling are the discontinuous Galerkin method and the mass-lumped finite element method, since they allow for explicit time-stepping. In this dissertation, new and sharp bounds for the time step size and penalty term for the discontinuous Galerkin method are presented. These parameter bounds can be efficiently computed and guarantee stability of the method. It is also shown that these new bounds significantly reduce the number of time steps, and therefore the computation time, compared to other parameter estimates available in literature. Furthermore, it is shown that the new penalty term bound also results in a higher accuracy.

While the stability properties of standard global time-stepping methods are well-known, little is known about the stability of local time-stepping methods. In this dissertation, it is shown that the stability of local time-stepping algorithms is in fact questionable. In particular, it is shown for a basic local time-stepping algorithm that instabilities can always occur, unless the local time step size is applied everywhere. This, however, would turn the algorithm back into a standard time-stepping scheme.

Besides the discontinuous Galerkin method, new mass-lumped finite

element methods are also presented in this dissertation. First, a new accuracy condition for the construction of mass-lumped elements is presented. This condition is less restrictive than the condition that has been used for several decades and has so far led to new mass-lumped tetrahedral elements of degrees 2, 3, and 4. The new tetrahedral elements of degree 2 and 3 are shown to be much more efficient than those available in literature, and mass-lumped tetrahedral elements of degree 4 had not been found so far. It is also shown that the new mass-lumped tetrahedral elements result in a much more efficient scheme than the discontinuous Galerkin method for elements of degree 4 or less. New quadrature rules for the stiffness matrix are presented that make the mass-lumped finite element methods even more efficient, since they allow for a faster computation of the stiffness matrix and can handle spatial parameters that vary within the element.

A dispersion analysis is presented in this dissertation that compares the discontinuous Galerkin and mass-lumped finite element methods in terms of accuracy and computation time. The analysis not only shows which method is the most efficient for a given accuracy, but also how many elements per wavelength are required for a given accuracy, and how much the accuracy of the method is affected by element distortions. From the results, it follows that the new degree-2 mass-lumped finite element method is the most efficient for a dispersion error between 0.01 and 0.001, while the new higher-degree mass-lumped finite element methods are more efficient for smaller dispersion errors.

Concluding, the finite element methods and algorithms presented in this dissertation allow for a much faster modelling of seismic waves. This makes the use of finite element methods much more attractive for geophysical applications or other industrial applications that involve solving wave propagation problems.

Samenvatting

In dit proefschrift worden nieuwe en efficiëntere eindige elementen methodes voor het modelleren van seismische golven gepresenteerd en geanalyseerd.

Seismisch modelleren is nuttig om het seismische gedrag in complexe aardstructuren beter te kunnen begrijpen en is tevens een hoofdonderdeel van volledige seismische inversie, een krachtig hulpmiddel om de inwendige structuur van de aarde mee in kaart te kunnen brengen. Het grote voordeel van eindige elementen methodes ten opzichte van andere modelleer technieken, zoals de populaire eindige differentie methode, is dat ze nauwkeurig het effect van een complexe topografie, zoals bergachtige gebieden en grillige zeebodems, kunnen modelleren zonder dat het rekenrooster hoeft te worden verfijnd. Echter, deze methodes vergen enorme rekenkracht en ze efficiënter maken is daarom van grote waarde voor veel industriële toepassingen.

De meestbelovende eindige elementen methodes zijn de discontinue Galerkin methode en de massa-geconcentreerde eindige elementen methode, aangezien zij resulteren in een expliciet tijdstappenschema. In dit proefschrift worden nieuwe en scherpe afschattingen voor de tijdstapgrootte en de stabiliteitsterm van de discontinue Galerkin methode gepresenteerd. Deze parameter afschattingen kunnen efficiënt worden berekend en garanderen dat de methode stabiel is. Er wordt ook aangetoond dat deze nieuwe afschattingen resulteren in een aanzienlijk kleiner aantal tijdstappen, en daarmee minder rekentijd, vergeleken met andere afschattingen die in de literatuur te vinden zijn. Tevens wordt aangetoond dat de nieuwe stabiliteitstermafschatting resulteert in hogere nauwkeurigheid.

Terwijl er veel bekend is over de stabiliteit van standaard globale tijdstappenschema's, is er maar relatief weinig bekend over de stabiliteit van lokale tijdstappenschema's. In dit proefschrift wordt aangetoond dat de stabiliteit van laatstgenoemde schema's twijfelachtig is. Om precies te zijn, er wordt aangetoond dat een eenvoudig basisschema met lokale tijdstappen altijd instabiel kan zijn, tenzij de lokale tijdstapgrootte overal wordt

toegepast. In dat geval echter, komt het schema op hetzelfde neer als een globaal tijdstappenschema.

Naast de Discontinue Galerkin methode worden er ook nieuwe massa-geconcentreerde eindige elementen methodes gepresenteerd in dit proefschrift. Allereerst wordt er een nieuwe nauwkeurigheidseis voor massa-geconcentreerde elementen gepresenteerd. Deze eis is minder streng dan de eis die al sinds vele decennia gebruikt wordt en heeft tot dusverre geleid tot nieuwe massa-geconcentreerde viervlakkige elementen van graad 2, 3 en 4. De nieuwe viervlakkige elementen van graad 2 en 3 zijn aanzienlijk efficiënter dan de voorgaande versies en massa-geconcentreerde viervlakkige elementen van graad 4 waren tot dusver nog niet gevonden. De nieuwe massa-geconcentreerde viervlakkige elementen zijn ook aantoonbaar efficiënter dan de discontinue Galerkin methodes voor graad 4 of lager. Nieuwe quadratuurregels voor de stijfheidsmatrix worden gepresenteerd die de massa-geconcentreerde elementen nog efficiënter maken, aangezien ze leiden tot een snellere berekening van de stijfheidsmatrix en ze om kunnen gaan met ruimtelijke parameters die binnen het element variëren zonder convergentiesnelheid te verliezen.

Door middel van een dispersie analyse, gepresenteerd in dit proefschrift, worden de discontinue Galerkin en massa-geconcentreerde eindige elementen methodes met elkaar te vergelijken in termen van nauwkeurigheid en rekensnelheid. De analyse laat niet alleen zien welke methode efficiënter is voor een gegeven nauwkeurigheid, maar ook hoeveel elementen per golflengte nodig zijn voor een gegeven nauwkeurigheid en hoeveel de nauwkeurigheid wordt beïnvloed door vervormingen in het rekenrooster. Uit de analyse volgt dat de nieuwe tweedegraads massa-geconcentreerde eindige elementen methode de meest efficiënte methode is voor een dispersiefout tussen 0.01 en 0.001 terwijl de nieuwe hogere-graads massa-geconcentreerde eindige elementen methodes efficiënter zijn voor kleinere dispersiefouten.

Kort samengevat, de eindige elementen methodes en algoritmes die in dit proefschrift worden beschreven maken het mogelijk om seismische golven te modelleren in een veel kortere rekentijd. Dit maakt het gebruik van eindige elementen methodes een stuk aantrekkelijker voor geofysische toepassingen of andere industriële toepassingen waarin golfproblemen moeten worden opgelost.

Contents

Chapter 1

Introduction

1.1 Earthquakes and seismic inversion

Earthquakes occur everywhere and anytime. Many of them are so small that we do not notice them. Those that can be clearly felt occur around 10,000 times a year [70], and great earthquakes, with a magnitude of 8 or more on the scale of Richter, occur only once a year [69]. The lightest earthquakes that most people can still feel have a magnitude of 3, which roughly means the earth is shaking with an amplitude of 1 mm at 100 km distance from the earthquake epicentre [61]. An increase of one magnitude corresponds to a shaking amplitude ten times as large, so an earthquake of magnitude 6 already shakes with an amplitude of one meter at 100 km distance and can cause severe damage, like the central Italy earthquakes of 2016 [73, 72]. A magnitude of 7 or more can be catastrophic. Such recent great earthquakes include the Nepal earthquake of April 2015 (scale 7.8, >9,000 deaths [74]), the Tohoku earthquake and tsunami of March 2011 (scale 8.9, >18,000 deaths [75]), and the Haiti earthquake of January 2010 (scale 7.0, 100,000-316,000 deaths [71]). The most powerful earthquake ever recorded is the Great Chilean earthquake of 1960 with a magnitude of 9.5 [70].

In the past, earthquakes were believed to be caused by the struggling of the tortured god Loki (Norse mythology [68]), the trident of an angry Poseidon (Greek mythology [24]), or a giant catfish called Namazu thrashing about (Japanese mythology [26]). Nowadays, scientists agree that most natural occurring earthquakes are caused by the rupture between tectonic plates. When the fault plane between two tectonic plates contains a lot of irregularities, movement along the fault results in a continuous increase

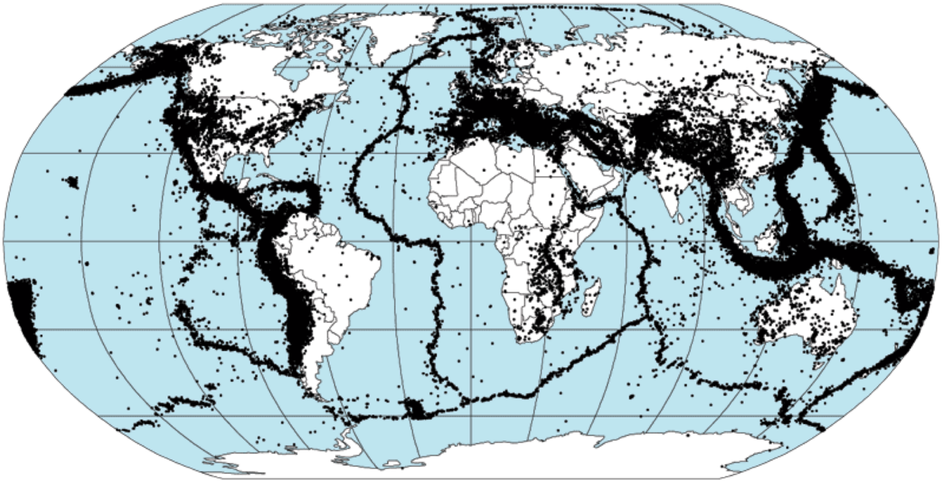


Figure 1.1: Earthquake epicentres in 1963-1998. Source: NASA, DTAM project team [56].

and release of elastic energy. This elastic energy is released in the form of heat, cracking rock, and seismic waves, which literally means 'waves of shaking earth', thereby causing an earthquake [57].

Because of the devastating effects, a large branche of seismology sciences is dedicated to understanding and predicting earthquakes. Despite considerable effort, this remains a challenging task [76]. Rupture dynamics are still poorly understood and monitoring the geometry of the earth's subsurface remains difficult. The latter can usually only be achieved by processing seismic data. Based on the refractions and reflections of seismic waves at geological interfaces, it is possible to obtain a model of the earth's subsurface. This technique is known as seismic inversion or seismic imaging and is somewhat comparable to the way a bat can produce an image of its surroundings by producing ultrasonic sounds and listening to the returning echoes. It is the main technique to image the deep interior of the earth, and while it can not directly be used to forecast earthquakes, it may help to get a better understanding of the earth's subsurface.

Another application for which seismic inversion is often used, is oil and gas reservoir exploration. In the beginning of the 20th century, shortly after seismic data were used to obtain the first images of the earth's interior, engineers started using artificially created earthquakes to observe the upper few kilometres of the earth's crust. Such seismic surveys led to the discovery of an oil reservoir for the first time in Texas 1924 [66]. Since then, seismic imaging has become a standard method for subsurface exploration and has up to this day led to the discovery of a great number of petroleum and gas reservoirs.

A seismic survey typically consists of an artificially created earthquake, using, for example, dynamite, specialized air guns, or a seismic vibrator [66, 65]. The refracted and reflected waves are then detected by an array of receivers, usually consisting of geophones or hydrophones. By measuring the time it takes for the wave to travel from the source to the receivers, a basic image of the geological interfaces can be reconstructed. For simple geological structures, mapping the geological interface by measuring reflected waves is similar to the echolocation used by bats. When the rock structure between the earth's surface and the reservoir is more complicated, however, more advanced inversion algorithms are required to detect and map the reservoir.

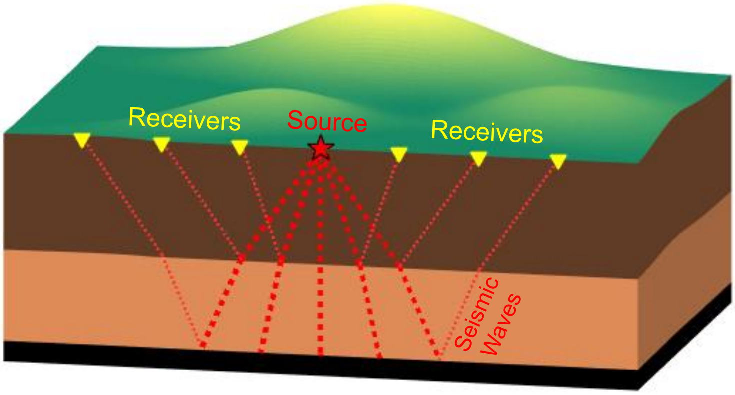


Figure 1.2: Illustration of a seismic survey

The high commercial interest and the complexity of the inversion problem has resulted in a huge number of different seismic imaging techniques, ranging from ray-based Kirchhoff methods, where seismic waves are approximated by rays in a way similar to geometric optics, to one-way methods, where waves are approximated by assuming they mainly move in the vertical direction, to two-way reverse time migration and full waveform inversion methods, where the wavefield is accurately solved on the entire 3D domain [28, 78]. While ray-based and one one-way wave propagation methods are much faster, their accuracy is limited when the geometry is complex. Full waveform inversion methods remain accurate by modelling the complete waveform on the entire 3D domain.

The main idea of full waveform inversion is to optimise the image of the geological structure and rock parameters, such that the wavefield emitted from the source matches best with the receiver data. Obtaining the optimal image is usually an iterative process, where each update is based on a cross-correlation of the wavefield emitted from the source and the backward propagated receiver data misfit [78]. Solving these wave propagation problems requires huge computational power, since the wavefield has to be solved accurately on large 3D domains, typically several kilometres deep and up to tens of kilometres wide. Improving the efficiency of wave propagation modelling is therefore of enormous interest for the petroleum and gas industry, and many algorithms have been developed and analysed for several decades. Yet, because of the great commercial interest and because of the enormous difficulty of the problem, more efficient algorithms are still in need.

Finding such algorithms has also been the main goal of this research project. During the last four years, I extensively studied numerous numerical algorithms and found several ways to model seismic waves more efficiently. These new methods are presented in this dissertation and will be briefly introduced in this chapter.

1.2 Numerical methods for seismic modelling

Most of the numerical methods for seismic modelling belong to one of the following three categories: spectral methods, finite difference methods, or finite element methods. An extensive overview of these methods can be found in [77]. Spectral methods approximate the wave field using Fourier modes. They are very efficient for certain specific geometries, such as a layered earth with purely horizontal interfaces, but are usually not applicable to more generic geometries.

Finite difference methods, instead, approximate the wave field on a uniform grid. They can be applied to generic geological structures and are very easy to implement, which makes them currently the most popular choice for large-scale modelling and inverse problems. For complex topographies and sharp contrasts in the rock material, however, they require a very fine grid resolution to remain accurate.

Finite element methods can handle complex geological interfaces more efficiently by carefully subdividing the earth's geometry into simple geometries called elements. The wave field is then approximated on each element using simple interpolation functions, usually polynomials up to a certain degree. Finite elements methods are more difficult to implement compared to finite difference methods and require more computation time and memory to model simple rock structures. For complex geological structures, however, they can remain very accurate without having to refine the element mesh as long as the elements are well-aligned with the geological interfaces. Finite element methods can therefore significantly outperform finite difference methods in such cases [81]. In the context of seismic inversion, this is especially relevant in case of complex topographies such as mountainous areas and rough seabeds, since the subsurface geometry is initially unknown.

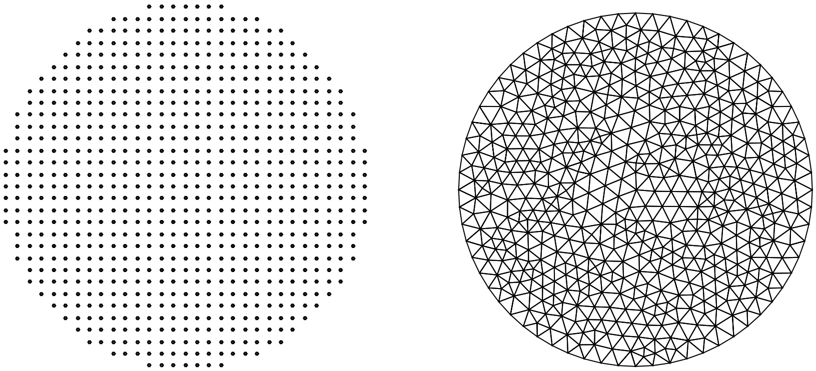


Figure 1.3: Finite difference grid (left) and triangular element mesh (right) for a circular domain

The goal of this research project was to further improve the efficiency of these finite element methods. To achieve this, I focused on finite element methods that allow for explicit time-stepping, which means that, given the wave field at times t and $t - \Delta t$, the wave field at time $t + \Delta t$ can be computed without having to solve a large and complex linear system of equations. If this property is not satisfied, the resulting time-stepping scheme is called implicit. There exist many implicit time-stepping schemes for wave propagation modelling, even most standard finite element schemes result in an implicit time-stepping scheme, but since implicit time-stepping requires solving a large linear system of equations at each time step, none of these are suitable for large-scale industrial applications.

Explicit time-stepping methods, on the other hand, are much faster and can compete with the popular finite difference methods. Explicit time-stepping is only possible, however, when the mass matrix that appears in the finite element discretization is (block)-diagonal. There are two types of finite element methods that achieve this: mass-lumped finite element methods and discontinuous Galerkin (DG) methods. The latter approximates the wave field with interpolation functions that can be discontinuous at the element interfaces. The most promising among the existing DG methods for seismic modelling is the symmetric interior penalty discontinuous Galerkin (SIPDG) method, although the performance of this method heavily depends on the choice of a parameter called the penalty term.

Mass-lumped finite element methods, on the other hand, approximate the wave field with specific continuous interpolation functions that allow for an accurate diagonal approximation of the mass matrix. These interpolation functions consist of standard piecewise polynomial functions up to a given degree p plus some additional higher-degree bubble functions. The efficiency of mass-lumped finite element methods strongly depends on the choice of these higher-degree bubble functions.

It is not clear which of these methods is more efficient. Therefore, both types of finite element methods are still actively studied today.

1.3 Overview of this dissertation

During this research, I extensively studied and compared SIPDG and mass-lumped finite element methods and found several ways to improve them. An overview of the topics that were addressed during this research are listed below.

- **Penalty term estimates for the SIPDG method (Chapter 2)**

  New and sharper lower bounds for the penalty terms used in SIPDG methods are derived. When the penalty term is too small, the method becomes unstable, which means the results are often useless. If this parameter is chosen too large, the method requires more computations and becomes less accurate. Lower bounds for the penalty term already existed, but are not always sharp. Using the new bounds guarantees stability and significantly reduces the number of time steps compared to other penalty term bounds.

- **Time step estimate for the SIPDG method (Chapter 2)**

  An efficient way to compute a sharp upper bound for the time step size for SIPDG methods is derived. If the time step size is too large, the method becomes unstable, while a smaller time step size results in more computation time. Currently, the time step size is chosen using model problems, but this does not always guarantee stability or results in an overly small time step size. This new upper bound guarantees stability in all cases and results in a time step size that is close to optimal.

- **Stability analysis of a basic local time-stepping algorithm (Chapter 3)**

  An interesting research question is if a similar bound for the time step size can also be derived for the basic local time-stepping algorithm, given in [22]. This algorithm allows for using a smaller time step size at only those regions where a smaller time step size is required. A proof is given that a similar upper bound does not exist for this algorithm. In particular, it is shown that for this basic local time-stepping method, instabilities can always be present, unless local time-stepping is applied everywhere, which would make the scheme ineffective.

- **Dispersion properties and comparisons of SIPDG and mass-lumped finite element methods (Chapter 4)**

  The efficiency of existing mass-lumped finite element methods and SIPDG methods is compared using a dispersion analysis. In particular, a dispersion analysis is used to determine which method is the most efficient for a given accuracy, how many elements per wavelength are required for a given accuracy, and how much is the accuracy of the method affected by element distortions. This dispersion analysis also demonstrates the significant gain in efficiency of the SIPDG method when using the new penalty term bound.

- **New mass-lumped finite element methods (Chapter 5)**

  A new accuracy condition for the construction of mass-lumped tetrahedral elements is derived. This new condition is less severe than the condition that has been used for several decades and has so far resulted in new mass-lumped tetrahedral elements of degrees 2 to 4. Numerical tests show that these new elements are significantly more efficient than the old mass-lumped elements or the DG elements.

- **Quadrature rules for the stiffness matrices of mass-lumped tetrahedra (Chapter 6)**

> An accuracy condition for the quadrature rules for computing the stiffness matrices of mass-lumped elements is derived and several new quadrature rules for mass-lumped tetrahedra are presented. These quadrature rules allow for a more efficient implementation of mass-lumped tetrahedral elements and can handle material parameters that vary within the element without a loss of convergence rate.

While this research was focused on seismic waves, many of the results also hold for other wave propagation problems, such as the acoustic wave equation or the electromagnetic wave equations.

Each of the topics listed above is addressed in detail in one of the following chapters of this dissertation. Furthermore, an overview of the main conclusions of this research project is given in Chapter 7.

Chapter 2

Sharp Penalty Term and Time Step Bounds for the Interior Penalty Discontinuous Galerkin Method for Linear Hyperbolic Problems[1]

Abstract

We present *sharp* and *sufficient* bounds for the interior penalty term and time step size to ensure stability of the symmetric interior penalty discontinuous Galerkin (SIPDG) method combined with an explicit time-stepping scheme. These conditions hold for generic meshes, including unstructured nonconforming heterogeneous meshes of mixed element types, and apply to a large class of linear hyperbolic problems, including the acoustic wave equation, the (an)isotropic elastic wave equations, and Maxwell's equations. The penalty term bounds are computed elementwise, while bounds for the time step size are computed at weighted submeshes requiring only a small number of elements and faces. Numerical results illustrate the sharpness of these bounds.

2.1 Introduction

Realistic wave problems often involve large three-dimensional domains, with complex boundary layers, sharp material interfaces and detailed internal structures. Solving such problems therefore requires a numerical

[1] Published as: Geevers, S., & van der Vegt, J. J. W. (2017). Sharp Penalty Term and Time Step Bounds for the Interior Penalty Discontinuous Galerkin Method for Linear Hyperbolic Problems. *SIAM Journal on Scientific Computing*, 39(5), A1851-A1878.

method that is efficient in terms of both memory usage and computation time and is flexible enough to deal with interfaces and internal structures without leading to an unnecessary overhead.

A standard finite difference scheme therefore falls short, since it cannot efficiently deal with complex material interfaces, and since small internal structures impose global restrictions on the grid resolution. Finite element methods overcome these problems, since they can be applied to unstructured meshes. However, the finite element method, combined with an explicit time-stepping scheme, requires solving mass matrix-vector equations during every time step. This significantly increases the computational time when the mass matrix is not (block)-diagonal. To obtain diagonal mass matrices without losing the order of accuracy, several mass-lumping techniques have been developed; see, for example, [42, 11, 15, 53]. However, for higher order elements, these techniques require additional quadrature points and degrees of freedom to maintain the optimal order of accuracy.

An alternative method is the discontinuous Galerkin (DG) finite element method. This method is similar to the conforming finite element method but allows its approximation functions to be discontinuous at the element boundaries, which naturally results in a block-diagonal mass matrix. Additional advantages of this method are that it also supports meshes with hanging nodes, and that the extension to arbitrary higher order polynomial basis functions is straightforward and can be adapted elementwise. The downside of the DG method, however, is that the discontinuous basis functions can result in significantly more degrees of freedom.

Still, because of its advantages, numerous DG schemes have already been developed and analyzed for linear wave problems, including the symmetric interior penalty discontinuous Galerkin (SIPDG) method; see, for example, [36, 37, 5]. The advantage of the SIPDG method is that it is based on the second order formulation of the problem, while schemes based on a first order formulation require solving additional variables leading to more memory usage. The SIPDG and several alternatives have also been compared and analyzed in [7], from which it follows that the SIPDG method is one of the most attractive options because of its consistency, stability, and optimal convergence rate.

However, to efficiently apply the SIPDG method with an explicit time-stepping scheme, the interior penalty term needs to be sufficiently large and the time step size needs to be sufficiently small. If the penalty term is set too small or the time step size too large, the SIPDG scheme will become unstable. On the other hand, increasing the penalty term will lead to a more severe time step restriction, and a smaller time step size results in a longer computation time. For this reason, there have been multiple studies on finding suitable choices for these parameters.

In [64, 27], for example, sufficient conditions have been derived for the penalty term, for triangular and tetrahedral meshes, while the results of [27] have been sharpened in [55] for tetrahedral meshes. However, the numerical results in Section 2.7 illustrate that these estimates are still not always very sharp. In [55] they also studied the effects of the penalty term, element shape, and polynomial order on the CFL condition for tetrahedral elements, although these results may not give sufficient conditions for nonuniform grids. Penalty term conditions for regular square and cubic meshes have been studied in [3, 20, 1], where [20, 1] also studied the CFL conditions for these meshes. However, the analysis in these studies only holds for uniform homogeneous meshes. For generic heterogeneous meshes, sharp parameter conditions have remained an open problem.

In this chapter we solve these problems by deriving sufficient conditions for both the penalty term and time step size, which lead to sharp estimates, and which hold for generic meshes, including unstructured nonconforming heterogeneous meshes of mixed element types with different types of boundary conditions. These conditions also apply to a large class of linear wave problems, including the acoustic wave equation, Maxwell's equations, and (an)isotropic elastic wave equations. We compare our estimates to some of the existing ones and illustrate the sharpness of our parameter estimates with several numerical tests.

This chapter is constructed as follows: in Section 2.2 we introduce some tensor notation, such that we can present the general linear hyperbolic model in Section 2.3, and present the symmetric interior penalty discontinuous Galerkin method in Section 2.4. After this, we derive sufficient conditions for the penalty parameter in Section 2.5 and sufficient conditions for the time step size in Section 2.6. Finally, we compare and test the sharpness of our estimates in Section 2.7 and give a summary in Section 2.8.

2.2 Some tensor notation

Before we present the linear hyperbolic model, it is useful to define some tensor notation. Let M and N be two nonnegative integers. Also, let $A \in \mathbb{R}^{n_1\times\cdots\times n_N}$ be a tensor of order N and $B \in \mathbb{R}^{m_1\times\cdots\times m_M}$ be a tensor of order M. We define the tensor product $AB \in \mathbb{R}^{n_1\times\cdots\times n_N\times m_1\times\cdots\times m_M}$, which is of order $N+M$, as follows:

$$[AB]_{i_1,\dots,i_N,j_1,\dots,j_M} := A_{i_1,\dots,i_N} B_{j_1,\dots,j_M},$$

for $(i_1,\dots,i_N,j_1,\dots,j_M) \in (\mathbb{N}_{n_1},\dots,\mathbb{N}_{n_N},\mathbb{N}_{m_1},\dots,\mathbb{N}_{m_M})$, where $\mathbb{N}_n := \{1,\dots,n\}$. Now let $A \in \mathbb{R}^{n_1\times\cdots\times n_N\times p}$ be a tensor of order $N+1$ and $B \in \mathbb{R}^{p\times m_1\times\cdots\times m_M}$ be a tensor of order $M+1$. We define the dot product $A\cdot B \in \mathbb{R}^{n_1\times\cdots\times n_N\times m_1\times\cdots\times m_M}$, which is of order $N+M$, as follows:

$$[A\cdot B]_{i_1,\dots,i_N,j_1,\dots,j_M} := \sum_{k=1}^{p} A_{i_1,\dots,i_N,k} B_{k,j_1,\dots,j_M},$$

for $(i_1,\dots,i_N,j_1,\dots,j_M) \in (\mathbb{N}_{n_1},\dots,\mathbb{N}_{n_N},\mathbb{N}_{m_1},\dots,\mathbb{N}_{m_M})$. For the double dot product, let $A \in \mathbb{R}^{n_1\times\cdots\times n_N\times p_1\times p_2}$ be a tensor of order $N+2$ and $B \in \mathbb{R}^{p_2\times p_1\times m_1\times\cdots\times m_M}$ be a tensor of order $M+2$. We define $A:B \in \mathbb{R}^{n_1\times\cdots\times n_N\times m_1\times\cdots\times m_M}$, which is of order $N+M$, as follows:

$$[A:B]_{i_1,\dots,i_N,j_1,\dots,j_M} := \sum_{k_1=1}^{p_1}\sum_{k_2=1}^{p_2} A_{i_1,\dots,i_N,k_1,k_2} B_{k_2,k_1,j_1,\dots,j_M},$$

for $(i_1,\dots,i_N,j_1,\dots,j_M) \in (\mathbb{N}_{n_1},\dots,\mathbb{N}_{n_N},\mathbb{N}_{m_1},\dots,\mathbb{N}_{m_M})$. Now let $A \in \mathbb{R}^{n_1\times\cdots\times n_N}$ be a tensor of order N again. We define the transpose $A^t \in \mathbb{R}^{n_N\times\cdots\times n_1}$ as follows:

$$A^t_{i_N,\dots,i_1} := A_{i_1,\dots,i_N},$$

for $(i_N,\dots,i_1) \in (\mathbb{N}_{n_N},\dots,\mathbb{N}_{n_1})$. A tensor A is called symmetric if $A = A^t$ and we define $\mathbb{R}^{n_1\times\cdots\times n_N}_{sym}$ to be the set of symmetric tensors in $\mathbb{R}^{n_1\times\cdots\times n_N}$. Now let $B \in \mathbb{R}^{n_1\times\cdots\times n_N}$ be a tensor of the same size as A. We define the inner product as follows:

$$(A,B) := \sum_{i_1=1}^{n_1}\cdots\sum_{i_N=1}^{n_N} A_{i_1,\dots,i_N} B_{i_1,\dots,i_N}.$$

The corresponding norm of a tensor is given by

$$\|A\|^2 := (A,A).$$

In the next section we will present the general linear hyperbolic problem, which we will solve using the SIPDG method.

2.3 A general linear hyperbolic model

Let $\Omega \subset \mathbb{R}^d$ be a d-dimensional open domain with a Lipschitz boundary $\partial\Omega$, and let $(0,T)$ be the time domain. Also, let $\{\Gamma_d, \Gamma_n\}$ be a partition of $\partial\Omega$, corresponding to Dirichlet and von Neumann boundary conditions, respectively. We define the following linear hyperbolic problem2:

$$
\begin{aligned}
\rho\partial_t^2\mathbf{u} &= \nabla\cdot C:\nabla\mathbf{u}+\mathbf{f} && \text{in } \Omega\times(0,T), && \text{(2.1a)}\\
\hat{\mathbf{n}}\cdot C:\hat{\mathbf{n}}\mathbf{u} &= \mathbf{0} && \text{on } \Gamma_d\times(0,T), && \text{(2.1b)}\\
\hat{\mathbf{n}}\cdot C:\nabla\mathbf{u} &= \mathbf{0} && \text{on } \Gamma_n\times(0,T), && \text{(2.1c)}\\
\mathbf{u}|_{t=0} &= \mathbf{u}_0 && \text{in } \Omega, && \text{(2.1d)}\\
\partial_t\mathbf{u}|_{t=0} &= \mathbf{v}_0 && \text{in } \Omega, && \text{(2.1e)}
\end{aligned}
$$

where $\mathbf{u} : \Omega\times(0,T)\to\mathbb{R}^m$ is a vector of m variables that are to be solved, ∇ is the gradient operator, $\rho : \Omega\to\mathbb{R}^+$ is a positive scalar field, $C : \Omega\to\mathbb{R}^{d\times m\times m\times d}_{sym}$ a fourth order tensor field, $\mathbf{f} : \Omega\times(0,T)\to\mathbb{R}^m$ the external volume force, and $\hat{\mathbf{n}} : \partial\Omega\to\mathbb{R}^d$ the outward pointing normal unit vector.

We make some assumptions on the material parameters. First, we assume that there exist positive constants ρ_{min}, ρ_{max} such that

$$
0<\rho_{min}\leq\rho\leq\rho_{max} \qquad \text{in } \Omega. \qquad \text{(2.2)}
$$

We also assume that the tensor field is symmetric, $C = C^t$, and that there exist linear subspaces $\Sigma_0, \Sigma_1 \subset \mathbb{R}^{d\times m}$, with $\Sigma_0+\Sigma_1=\mathbb{R}^{d\times m}$ and $\Sigma_0\perp\Sigma_1$, and constants c_{min}, c_{max} such that C is nonnegative and bounded in the following sense:

$$
\begin{aligned}
0<c_{min}\leq\frac{\boldsymbol{\sigma}^t:C:\boldsymbol{\sigma}}{\|\boldsymbol{\sigma}\|^2}&\leq c_{max} && \text{in } \Omega,\ \text{for all } \boldsymbol{\sigma}\in\Sigma_1/\{\mathbf{0}\}, && \text{(2.3a)}\\
C:\boldsymbol{\sigma}&=\mathbf{0} && \text{in } \Omega,\ \text{for all } \boldsymbol{\sigma}\in\Sigma_0. && \text{(2.3b)}
\end{aligned}
$$

By $\Sigma_0+\Sigma_1=\mathbb{R}^{d\times m}$ we mean that any $\boldsymbol{\sigma}\in\mathbb{R}^{d\times m}$ can be written as $\boldsymbol{\sigma}=\boldsymbol{\sigma}_0+\boldsymbol{\sigma}_1$ for some $\boldsymbol{\sigma}_0\in\Sigma_0, \boldsymbol{\sigma}_1\in\Sigma_1$, and by $\Sigma_0\perp\Sigma_1$ we mean that $(\boldsymbol{\sigma}_0,\boldsymbol{\sigma}_1)=0$ for all $\boldsymbol{\sigma}_0\in\Sigma_0, \boldsymbol{\sigma}_1\in\Sigma_1$.

By choosing the correct tensor and scalar field we can obtain a number of hyperbolic problems, including Maxwell's equations, the acoustic wave equation and the (an)isotropic elastic wave equations. We illustrate this in the following examples, where we define the tensor and vector fields using Cartesian coordinates.

Example 2.3.1. *Consider the acoustic wave equation written in the following dimensionless form:*

$$\partial_t^2 p = \nabla \cdot c^2 \nabla p,$$

where $p : \Omega \times (0,T) \to \mathbb{R}$ is the pressure field and $c : \Omega \to \mathbb{R}^+$ the acoustic wave velocity. We can write these equations in the form of (2.1) by setting $m = 1$, $\mathbf{u} = p$, $\mathbf{f} = \mathbf{0}$, $\rho = 1$, and

$$C_{i,1,1,j} := \delta_{ij} c^2$$

for $i, j = 1, \ldots, d$, where δ is the Kronecker delta function.

Example 2.3.2. *Consider Maxwell's equations in a domain with zero conductivity written in the following dimensionless form:*

$$\epsilon \partial_t^2 E = -\nabla \times (\mu^{-1} \nabla \times E) + \partial_t j,$$

where $E : \Omega \times (0,T) \to \mathbb{R}^3$ is the electric field, $\epsilon : \Omega \to \mathbb{R}^+$ the relative electric permittivity, $\mu : \Omega \to \mathbb{R}^+$ the relative magnetic permeability, and $j : \Omega \times (0,T) \to \mathbb{R}^3$ the applied current density. We can write these equations in the form of (2.1) by setting $d = 3$, $m = 3$, $\mathbf{u} = E$, $\mathbf{f} = \partial_t j$, $\rho = \epsilon$, and

$$C_{i_D,i_M,j_M,j_D} := \mu^{-1}\big(\delta_{i_D,j_D}\delta_{i_M,j_M} - \delta_{i_D,j_M}\delta_{i_M,j_D}\big)$$

for $i_D, i_M, j_D, j_M = 1, 2, 3$, where δ is the Kronecker delta function.

Example 2.3.3. *Consider the linear anisotropic elastic wave equations. They can immediately be written in the form of (2.1) with $\mathbf{u} : \Omega \times (0,T) \to \mathbb{R}^d$ being the displacement vector and $C : \Omega \to \mathbb{R}^{d\times d\times d\times d}_{sym}$ being the elasticity tensor, which has the additional symmetries*

$$C_{i_D,i_M,j_M,j_D} = C_{i_M,i_D,j_M,j_D} = C_{i_D,i_M,j_D,j_M}$$

for $i_D, i_M, j_D, j_M = 1, \ldots, d$. For the special case of isotropic elasticity, we can write

$$C_{i_D,i_M,j_M,j_D} = \lambda \delta_{i_D,i_M}\delta_{j_D,j_M} + \mu(\delta_{i_D,j_D}\delta_{i_M,j_M} + \delta_{i_D,j_M}\delta_{i_M,j_D})$$

for $i_D, i_M, j_D, j_M = 1, \ldots, d$, where δ is the Kronecker delta function and λ, μ are the Lamé parameters.

In the next section we present the DG method that we use to solve these linear hyperbolic problems.

2.4 A discontinuous Galerkin method

To explain the DG method, we first present the weak formulation of (2.1). After that, we introduce the tesselation of the domain, the discrete function spaces, and the trace operators. We then present the symmetric interior penalty discontinuous Galerkin (SIPDG) method and derive some of its properties.

2.4.1 The weak formulation

Define the following function space:

$$U := \left\{\mathbf{u} \in L^2(\Omega)^m \;\middle|\; C : \nabla\mathbf{u} \in L^2(\Omega)^{d\times m},\ \hat{\mathbf{n}} \cdot C : \hat{\mathbf{n}}\mathbf{u} = 0 \text{ on } \Gamma_d\right\}.$$

Assume that $\mathbf{u}_0 \in U$, $\mathbf{v}_0 \in L^2(\Omega)^m$, and $\mathbf{f} \in L^2\big(0,T;L^2(\Omega)^m\big)$. The weak formulation of (2.1) is finding $\mathbf{u} \in L^2\big(0,T;U\big)$, with $\partial_t\mathbf{u} \in L^2\big(0,T;L^2(\Omega)^m\big)$ and $\partial_t(\rho\partial_t\mathbf{u}) \in L^2\big(0,T;U^{-1}\big)$, such that $\mathbf{u}|_{t=0} = \mathbf{u}_0$, $\partial_t\mathbf{u}|_{t=0} = \mathbf{v}_0$, and

$$\langle\partial_t(\rho\partial_t\mathbf{u}), \mathbf{w}\rangle + a(\mathbf{u},\mathbf{w}) = (\mathbf{f},\mathbf{w}), \quad \text{a.e. } t \in (0,T), \text{ for all } \mathbf{w} \in U. \tag{2.4}$$

Here $(\cdot,\cdot)$ denotes the inner product of $L^2(\Omega)^m$, $\langle\cdot,\cdot\rangle$ denotes the pairing between U^{-1} and U, and $a(\cdot,\cdot) : U \times U \to \mathbb{R}$ is the semielliptic bilinear operator given by

$$a(\mathbf{u},\mathbf{w}) := \int_\Omega (\nabla\mathbf{u})^t : C : \nabla\mathbf{w}\, dx.$$

Using (2.3) it can be shown that U is a separable Hilbert space, and from (2.2) it follows that the norm $\|\mathbf{u}\|_\rho^2 := (\rho\mathbf{u},\mathbf{u})$ is equivalent to the standard $L^2(\Omega)^m$ inner product. Using these properties, it can be proven, in a way analogous to the proof of [46, Chapter 3, Theorem 8.1] that (2.4) is well-posed and has a unique solution.

2.4.2 Tesselation, discrete function space, and trace operators

Let $\mathcal{T}_h$ be a set of nonoverlapping open domains in $\mathbb{R}^d$, referred to as elements, such that every element $e \in \mathcal{T}_h$ fits inside a d-dimensional sphere of radius h, and such that $\overline{\Omega} := \bigcup_{e\in\mathcal{T}_h} \overline{e}$, where $\overline{e}$ and $\overline{\Omega}$ are the closures of e and Ω, respectively. We call $\mathcal{T}_h$ the tesselation of Ω. Using the tesselation we define the set of faces $\mathcal{F}_h := \mathcal{F}_{h,in} \cup \mathcal{F}_{h,b}$ and the union of all faces $\Gamma_h := \bigcup_{f\in\mathcal{F}_h} f$, where $\mathcal{F}_{h,in} := \big\{\partial e \cap \partial e'\big\}_{e,e'\in\mathcal{T}_h}$ is the set of all internal faces, $\mathcal{F}_{h,b} := \big\{\partial e \cap \partial\Omega\big\}_{e\in\mathcal{T}_h}$ is the set of all boundary faces, and ∂e denotes the element boundary. Furthermore, we let $\{\mathcal{F}_{h,d}, \mathcal{F}_{h,n}\}$ be the partition of $\mathcal{F}_{h,b}$ corresponding to the Dirichlet and Neumann boundary conditions, such that $\bigcup_{f\in\mathcal{F}_{h,d}} f = \Gamma_d$ and $\bigcup_{f\in\mathcal{F}_{h,n}} f = \Gamma_n$.

We use these sets of elements and faces to construct the discrete function space. To do this, let $e \in \mathcal{T}_h$ be an element with $\tilde{e}$ the corresponding reference element, which depends only on the shape of e. For every reference element $\tilde{e}$ we define a discrete function space $\tilde{U}_e : \tilde{e} \to \mathbb{R}^m$, such that $\tilde{U}_e = [\mathcal{P}^{K_e}(\tilde{e})]^m$, where $\mathcal{P}^{K_e}(\tilde{e})$ is a finite set of polynomial functions on $\tilde{e}$, which depends only on the degree K_e and $\tilde{e}$. Now let $\phi_e : \tilde{e} \to e$ be an invertible polynomial mapping, such that $|\mathbf{J}_e| := |\det(\nabla\phi_e)| \geq |\mathbf{J}_e|_{min} > 0$ in e, for some positive constant $|\mathbf{J}_e|_{min}$. Using this mapping and the reference function space $\tilde{U}_e$, we can construct the function space on the physical element U_e as follows:

$$U_e := \big\{u : e \to \mathbb{R}^m \mid u = \tilde{u} \circ \phi_e^{-1}, \text{ for some } \tilde{u} \in \tilde{U}_e\big\}.$$

This can then be used to construct the discrete finite element space:

$$U_h := \big\{u \in [L^2(\Omega)]^m \bigm| u|_e \in U_e, e \in \mathcal{T}_h\big\}.$$

The functions in the finite element space are continuous within every element, but can be discontinuous at the faces between two elements. To construct a discrete version of the bilinear form a, which can deal with these discontinuities, we introduce trace operators $\{\!\{\cdot\}\!\}$ and $[\![\cdot]\!]$ given below:

$$\begin{aligned} \{\!\{\phi\}\!\}\big|_f &:= \frac{1}{|\mathcal{T}_f|}\sum_{e\in\mathcal{T}_f} \phi|_{\partial e\cap f}, && f \in \mathcal{F}_h, \\ [\![\mathbf{u}]\!]\big|_f &:= \sum_{e\in\mathcal{T}_f} (\hat{\mathbf{n}}\mathbf{u})|_{\partial e\cap f}, && f \in \mathcal{F}_h, \end{aligned}$$

where $\mathcal{T}_f$ is the set of elements adjacent to f, $|\mathcal{T}_f|$ is the number of elements adjacent to f, and $\hat{\mathbf{n}}|_{\partial e}$ is the outward pointing normal vector of element e. The first trace operator is the average of traces, while the second operator is known as the jump operator. Using the first trace operator $\{\!\{\cdot\}\!\}$, we can construct the numerical flux operator $(\cdot)^* : U_h \to [L^2(\Gamma_h)]^m$, which assigns a unique value for $\mathbf{u} \in U_h$ at the faces, as follows:

$$\mathbf{u}^*|_f = \begin{cases} \{\!\{\mathbf{u}\}\!\} & f \in \mathcal{F}_{h,in} \cap \mathcal{F}_{h,n}, \\ \mathbf{0} & f \in \mathcal{F}_d. \end{cases} \tag{2.5}$$

In order to ensure that the discrete bilinear form a_h remains semielliptic, we also introduce penalty terms $\eta_e \in \mathbb{R}^+$ for every element $e \in \mathcal{T}_h$, and a penalty scaling function $\nu_h \in \bigotimes_{e\in\mathcal{T}_h} L^\infty(\partial e)$, with $\nu_h > 0$, which means $\nu|_{\partial e} : \partial e \to \mathbb{R}^+$ for all $e \in \mathcal{T}_h$. The penalty terms η_e are positive dimensionless constants for which lower bounds will be derived in Section 2.5. The function ν_h scales with order h^{-1} and is chosen as follows:

$$\nu_h|_{\partial e\cap f} := \left(\frac{|\mathbf{J}_f|}{|\mathbf{J}_e|}\right)|_{\partial e\cap f}, \qquad e \in \mathcal{T}_h, f \in \mathcal{F}_e,$$

where $\mathcal{F}_e$ denotes the faces adjacent to element e, $|\mathbf{J}_e| := |\det(\nabla\phi_e)|$ is the reference-to-physical element scale, and $|\mathbf{J}_f|$ is the reference-to-physical face scale. The face scale satisfies $|\mathbf{J}_f| = 1$ in 1D, $|\mathbf{J}_f| = |\partial_1\phi_f|$ in 2D, and $|\mathbf{J}_f| = |\partial_1\phi_f \times \partial_2\phi_f|$ in 3D, where $\phi_f : \tilde{f} \to f$ is the reference-to-physical face mapping, and $\partial_i\phi_f$ is the derivative of ϕ_f in reference coordinate i, assuming Cartesian reference coordinates. In our numerical tests we use this scaling function, although the stability analysis in this chapter holds for arbitrary positive functions ν_h.

Finally, to ensure that the discrete version of a is well defined we also make the following additional assumptions on the material parameters ρ and C:

$$\rho|_e \in W^{1,\infty}(e), \quad C|_e \in W^{1,\infty}(e)^{d\times m\times m\times d}_{sym}, \qquad e \in \mathcal{T}_h,$$

where $W^{1,\infty}$ denotes the Sobolev space of differentiable functions with uniformly bounded weak derivatives. These assumptions together with the trace inequality imply that the element traces of C and ρ are well defined and bounded.

We have now introduced the function spaces, operators, and parameter assumptions needed to present the DG method in the next subsection.

2.4.3 The symmetric interior penalty discontinuous Galerkin method

We present a DG method, which is known as the symmetric interior penalty discontinuous Galkerkin (SIPDG) method. The SIPDG method is solving $\mathbf{u} : [0,T] \to U_h$ such that

$$(\rho\partial_t^2\mathbf{u},\mathbf{w}) + a_h(\mathbf{u},\mathbf{w}) = (\mathbf{f},\mathbf{w}), \qquad \mathbf{w} \in U_h, t \in [0,T], \tag{2.6a}$$

$$(\rho\mathbf{u}|_{t=0},\mathbf{w}) = (\rho\mathbf{u}_0,\mathbf{w}) \qquad \mathbf{w} \in U_h, \tag{2.6b}$$

$$(\rho\partial_t\mathbf{u}|_{t=0},\mathbf{w}) = (\rho\mathbf{v}_0,\mathbf{w}) \qquad \mathbf{w} \in U_h, \tag{2.6c}$$

where $a_h : U_h \times U_h \to \mathbb{R}$ is the discrete version of the elliptic operator, given by

$$a_h(\mathbf{u}, \mathbf{w}) := a_h^{(C)}(\mathbf{u}, \mathbf{w}) + a_h^{(DG)}(\mathbf{u}, \mathbf{w}) + a_h^{(DG)}(\mathbf{w}, \mathbf{u}) + a_h^{(IP)}(\mathbf{u}, \mathbf{w}), \quad (2.7)$$

with

$$\begin{aligned}
a_h^{(C)}(\mathbf{u}, \mathbf{w}) &:= \sum_{e \in \mathcal{T}_h} a_e^{(C)}(\mathbf{u}, \mathbf{w}) := \sum_{e \in \mathcal{T}_h} \int_e (\nabla \mathbf{u})^t : C : \nabla \mathbf{w} \, dx, \\
a_h^{(DG)}(\mathbf{u}, \mathbf{w}) &:= \sum_{e \in \mathcal{T}_h} a_{\partial e}^{(DG)}(\mathbf{u}, \mathbf{w}) := \sum_{e \in \mathcal{T}_h} \int_{\partial e} (\mathbf{u}^* - \mathbf{u})\hat{\mathbf{n}} : C : \nabla \mathbf{w} \, ds, \\
a_h^{(IP)}(\mathbf{u}, \mathbf{w}) &:= \sum_{e \in \mathcal{T}_h} \eta_e a_{\partial e}^{(IP)}(\mathbf{u}, \mathbf{w}) \\
&:= \sum_{e \in \mathcal{T}_h} \eta_e \int_{\partial e} (\mathbf{u}^* - \mathbf{u})\hat{\mathbf{n}} : \nu_h C : \hat{\mathbf{n}}(\mathbf{w}^* - \mathbf{w}) \, ds,
\end{aligned}$$

for all $\mathbf{u}, \mathbf{w} \in U_h$. The bilinear form $a_h^{(C)}$ is the same as the original elliptic operator a and is the part that remains when both input functions are continuous. The bilinear form $a_h^{(DG)}$ can be interpreted as the additional part that results from partial integration of the elliptic operator a when the first input function is discontinuous. Finally, the bilinear form $a^{(IP)}$ is the part that contains the interior penalty terms needed to ensure stability of the scheme.

Using the definition of the numerical flux in (2.5), we can rewrite the bilinear forms $a_h^{(DG)}$ and $a_h^{(IP)}$, as follows:

$$a_h^{(DG)}(\mathbf{u}, \mathbf{w}) = \sum_{f \in \mathcal{F}_{h,in} \cup \mathcal{F}_{h,d}} - \int_f [\![\mathbf{u}]\!]^t : \{\!\{C : \nabla \mathbf{w}\}\!\} \, ds, \quad (2.8a)$$

$$a_h^{(IP)}(\mathbf{u}, \mathbf{w}) = \sum_{f \in \mathcal{F}_{h,in} \cup \mathcal{F}_{h,d}} \epsilon_f \int_f [\![\mathbf{u}]\!]^t : \{\!\{\eta_h \nu_h C\}\!\} : [\![\mathbf{w}]\!] \, ds, \quad (2.8b)$$

for all $\mathbf{u}, \mathbf{w} \in U_h$. Here $\epsilon_f := 1/2$ for internal faces and $\epsilon_f := 1$ for faces in $\mathcal{F}_{h,d}$, and $\eta_h \in \bigotimes_{e \in \mathcal{T}_h} L^\infty(\partial e)$ is defined by $\eta_h|_{\partial e} := \eta_e$ for all $e \in \mathcal{T}_h$. This scheme conforms with existing SIPDG schemes, except for a possible deviation in the interior penalty part $a_h^{(IP)}$. For example, for the acoustic wave equation given in Example 2.3.1, this scheme is equivalent to the one in [36] when choosing their penalty term $\mathbf{a}$ as $\mathbf{a}|_f = \epsilon_f \{\!\{\eta \nu_h c\}\!\}|_f$ for all $f \in \mathcal{F}_h$.

Since the bilinear form is symmetric, $a_h(u,w) = a_h(w,u)$ for all $\mathbf{u}, \mathbf{w} \in U_h$, we can obtain, by substituting $\mathbf{w} = \partial_t \mathbf{u}$ into (2.6a), the following energy equation:

$$\partial_t E_h = (\mathbf{f}, \partial_t \mathbf{u}), \qquad\qquad t \in [0,T],$$

where $E_h := \frac{1}{2}\left\|\rho^{1/2}\mathbf{w}\right\|_0^2 + \frac{1}{2}a_h(\mathbf{u},\mathbf{u})$ is the discrete energy, with $\|\cdot\|_0$ the $[L^2(\Omega)]^m$ norm. In the absence of an external force $\mathbf{f}$ this implies that the discrete energy is conserved.

However, for this energy to be well defined, in the sense that it is always nonnegative, the discrete bilinear form needs to remain semielliptic: $a_h(\mathbf{u},\mathbf{u}) \geq 0$ for all $\mathbf{u} \in U_h$. This then implies that any nonzero discrete eigenmode cannot grow unbounded in the absence of an external force. In case $\mathbf{u}$ is a zero discrete eigenmode, we can substitute $\mathbf{w} = \mathbf{u}$ into (2.6a) to obtain $\partial_t^2 \|\rho^{1/2}\mathbf{u}\|_0^2 = 2(\mathbf{f},\mathbf{u})$. This implies that, in the absence of an external force, zero eigenmodes grow at most linearly in time. This behavior can correspond to physical rigid motions, or, when there is a discrepancy between the physical and discrete zero modes, a linear drift of a spurious mode. For acoustic and elastic waves these spurious modes are absent, while for electromagnetic waves, these modes have been analyzed in, for example, [10]. However, even if there are spurious modes, we will not consider their drift as numerical instability, since the numerical error is expected to grow linearly in time anyway due to dispersion errors.

In the next section we will find sufficient lower bounds for the penalty term to make sure a_h is semielliptic. In particular, we will show there that a_h satisfies a coercivity condition that is commonly used to show optimal convergence in the energy-norm.

2.5 Sufficient penalty term estimates

In this section we derive a sufficient lower bound for the penalty term and a positive constant $c_{coer} > 0$, where c_{coer} is independent of the mesh $\mathcal{T}_h$, such that

$$a_h(\mathbf{u},\mathbf{u}) \geq c_{coer}\left|\mathbf{u}\right|_{1,h}^2, \qquad\qquad \mathbf{u} \in U_h, \tag{2.9}$$

where $|\cdot|_{1,h}$ is the discrete seminorm defined by $\left|\mathbf{u}\right|_{1,h}^2 := \sum_{e\in\mathcal{T}_h} \left|\mathbf{u}\right|_{1,e}^2$, with

$$\left|\mathbf{u}\right|_{1,e}^2 := \int_e \|C^{1/2} : \nabla\mathbf{u}\|^2 \, dx \; + \; \eta_e \int_{\partial e} \left\|\nu_h^{1/2} C^{1/2} : \hat{\mathbf{n}}(\mathbf{u}^* - \mathbf{u})\right\|^2 \, ds.$$

Here $C^{1/2} \in \bigotimes_{e\in\mathcal{T}_h} W^{1,\infty}(e)^{d\times m\times m\times d}_{sym}$ is a tensor field such that $C^{1/2} : C^{1/2} = C$. The existence of such a tensor field follows from Lemma 2.A.1. The numerical flux $\mathbf{u}^*$ is defined in (2.5), although the stability analysis in this chapter holds for arbitrary linear flux operators. Note that (2.9) is satisfied when

$$a_e(\mathbf{u},\mathbf{u}) \geq c_{coer}\big|\mathbf{u}\big|^2_{1,e}, \qquad e\in\mathcal{T}_h, \mathbf{u}\in U_h, \tag{2.10}$$

where $a_e(\mathbf{u},\mathbf{w}) := a_e^{(C)}(\mathbf{u},\mathbf{w}) + a_{\partial e}^{(DG)}(\mathbf{u},\mathbf{w}) + a_{\partial e}^{(DG)}(\mathbf{w},\mathbf{u}) + \eta_e a_{\partial e}^{(IP)}(\mathbf{u},\mathbf{w})$. Since we can write $|\mathbf{u}|^2_{1,e} = a_e^{(C)}(\mathbf{u},\mathbf{u}) + \eta_e a_{\partial e}^{(IP)}(\mathbf{u},\mathbf{u})$, it remains to bound $a_{\partial e}^{(DG)}(\mathbf{u},\mathbf{u})$ in terms of $a_e^{(C)}(\mathbf{u},\mathbf{u})$ and $a_{\partial e}^{(IP)}(\mathbf{u},\mathbf{u})$. In order to do this, we first introduce the auxiliary bilinear form $a_{\partial e}^{(C*)} : U_h \times U_h \to \mathbb{R}$ defined by

$$a_{\partial e}^{(C*)}(\mathbf{u},\mathbf{w}) := \int_{\partial e} (\nabla \mathbf{u}^t) : \nu_h^{-1} C : \nabla \mathbf{w}\, ds.$$

Note that this operator is similar to $a_e^{(C)}$, but integrates over the element boundary instead of the interior. Next, we show that $a_{\partial e}^{(DG)}(\mathbf{u},\mathbf{u})$ can be bounded in terms of $a_{\partial e}^{(C*)}(\mathbf{u},\mathbf{u})$ and $a_{\partial e}^{(IP)}(\mathbf{u},\mathbf{u})$:

Lemma 2.5.1. *Consider an arbitrary element $e\in\mathcal{T}_h$, and let $c>0$ be an arbitrary positive constant. Then the following inequality holds:*

$$|2a_{\partial e}^{(DG)}(\mathbf{u},\mathbf{u})| \leq c^{-1} a_{\partial e}^{(C*)}(\mathbf{u},\mathbf{u}) + c a_{\partial e}^{(IP)}(\mathbf{u},\mathbf{u}), \qquad \mathbf{u}\in U_h. \tag{2.11}$$

Proof. Take an arbitrary function $u \in U_h$. We can write

$$2a_{\partial e}^{(DG)}(\mathbf{u},\mathbf{u}) = \int_{\partial e} 2\Big(c^{1/2}\nu_h^{1/2} C^{1/2} : \hat{\mathbf{n}}(\mathbf{u}^* - \mathbf{u}),\ c^{-1/2}\nu_h^{-1/2} C^{1/2} : \nabla\mathbf{u}\Big)\, ds.$$

Using the Cauchy–Schwarz and the Cauchy inequalities, we can then obtain

$$\begin{aligned}\big|2a_{\partial e}^{(DG)}(\mathbf{u},\mathbf{u})\big| &\leq c \int_{\partial e} \big\|\nu_h^{1/2} C^{1/2} : \hat{\mathbf{n}}(\mathbf{u}^* - \mathbf{u})\big\|^2\, ds \\ &\quad + c^{-1}\int_{\partial e} \big\|\nu_h^{-1/2} C^{1/2} : \nabla \mathbf{u}\big\|^2\, ds \\ &= c^{-1} a_{\partial e}^{(C*)}(\mathbf{u},\mathbf{u}) + c a_{\partial e}^{(IP)}(\mathbf{u},\mathbf{u}).\end{aligned}$$

□

We now construct the following constant:

$$\kappa_e^* := \sup_{\mathbf{u}\in U_e,\, a_e^{(C)}(\mathbf{u},\mathbf{u})\neq 0} \frac{a_{\partial e}^{(C*)}(\mathbf{u},\mathbf{u})}{a_e^{(C)}(\mathbf{u},\mathbf{u})},$$

where U_e is the discrete function space restricted to element e. From its definition, we can immediately obtain the inequality $a_{\partial e}^{(C*)}(\mathbf{u},\mathbf{u}) \leq \kappa_e^* a_e^{(C)}(\mathbf{u},\mathbf{u})$ for any $\mathbf{u} \in U_h$. Using this property and Lemma 2.5.1 we can prove in Theorem 2.5.6 that κ_e^* is a sufficient lower bound for η_e to ensure a_e to be coercive.

However, before we give this proof, we first show that κ_e^* is well defined and show how it can be computed efficiently. To do this, consider an arbitrary $e \in \mathcal{T}_h$, and let $\{\mathbf{w}_i\}_{i=1}^n$ be a set of basis functions spanning U_e. Using these basis functions we can define positive semidefinite matrices $A_e, A_{\partial e}^* \in \mathbb{R}_{sym}^{n\times n}$ as follows:

$$[A_e]_{ij} = a_e^{(C)}(\mathbf{w}_i, \mathbf{w}_j), \qquad i,j = 1,\dots,n \tag{2.12a}$$

$$[A_{\partial e}^*]_{ij} = a_{\partial e}^{(C*)}(\mathbf{w}_i, \mathbf{w}_j), \qquad i,j = 1,\dots,n. \tag{2.12b}$$

If matrix A had been positive definite, then we could have obtained, using Lemma 2.A.5, the following relation:

$$\kappa_e^* = \sup_{\mathbf{u}\in U_e,\, a_e^{(C)}(\mathbf{u},\mathbf{u})\neq 0} \frac{a_{\partial e}^{(C*)}(\mathbf{u},\mathbf{u})}{a_e^{(C)}(\mathbf{u},\mathbf{u})} = \sup_{\underline{\mathbf{u}}\in\mathbb{R}^n, \underline{\mathbf{u}}\neq 0} \frac{\underline{\mathbf{u}}^t A_{\partial e}^* \underline{\mathbf{u}}}{\underline{\mathbf{u}}^t A_e \underline{\mathbf{u}}} = \lambda_{max}(A_e^{-1} A_{\partial e}^*),$$

where $\lambda_{max}(A_e^{-1}A_{\partial e}^*)$ denotes the largest eigenvalue of $A_e^{-1}A_{\partial e}^*$. However, the matrix A_e is only positive semidefinite, so we need to obtain some intermediate results before we can show that a similar type of relation still holds. First, we show that the kernel of A_e is a subset of the kernel of $A_{\partial e}^*$.

Lemma 2.5.2. *Let $e \in \mathcal{T}_h$ be an arbitrary element, and let $A_e, A_{\partial e}^*$ be matrices defined as in (2.12). Then* $\mathrm{Ker}(A_e) \subset \mathrm{Ker}(A_{\partial e}^*)$.

Proof. Let $\underline{\mathbf{u}} \in \mathrm{Ker}(A_e)$, and define $\mathbf{u} \in U_e$ as follows: $\mathbf{u} := \sum_{i=1}^n \underline{\mathbf{u}}_i \mathbf{w}_i$. Then

$$0 = \underline{\mathbf{u}}^t A_e \underline{\mathbf{u}} = a_e^{(C)}(\mathbf{u},\mathbf{u}) = \int_e \|C^{1/2} : \nabla\mathbf{u}\|^2 \, dx.$$

From this it follows that $C : \nabla\mathbf{u} = \mathbf{0}$ in e. Now let $\tilde{\mathbf{u}} := \mathbf{u} \circ \phi_e$ be the reference function, with ϕ_e the reference-to-physical element mapping. Since

ϕ_e is assumed to be a polynomial function, such that $|\mathbf{J}_e| := |\det(\nabla\phi_e)| \geq |\mathbf{J}_e|_{min} > 0$ in e, for some constant $|\mathbf{J}_e|_{min}$, it follows that $\phi_e^{-1} \in C^\infty(e)^d$. Furthermore, since the reference function $\tilde{\mathbf{u}}$ is also assumed to be polynomial, this implies that $\mathbf{u} = \tilde{\mathbf{u}} \circ \phi_e^{-1} \in C^\infty(e)^d$. Because we assumed that the spatial parameter C satisfies $C|_e \in W^{1,\infty}(e)^{d\times m\times m\times d}_{sym}$, it then follows from the trace theorem that $C : \nabla\mathbf{u} = \mathbf{0}$ is also satisfied on the boundary ∂e. This implies $a_{\partial e}^{(C*)}(\mathbf{u},\mathbf{w}) = 0$ for any $\mathbf{w} \in U_e$ and therefore $\underline{\mathbf{u}} \in \mathrm{Ker}(A^*_{\partial e})$. □

Now let k be the rank of A_e, and let $V_e \in \mathbb{R}^{n\times n}$ be a nonsingular matrix such that $V_e^t A_e V_e = D_e$, where $D_e \in \mathbb{R}^{n\times n}_{sym}$ is a diagonal matrix with the last $n-k$ entries being zero. Such a matrix decomposition can be obtained from, for example, a symmetric Gauss elimination procedure or a singular value decomposition. We then use these matrices to construct matrices $\tilde{D}_e, \tilde{A}^*_{\partial e} \in \mathbb{R}^{k\times k}_{sym}$ as follows:

$$[\tilde{D}_e]_{ij} = [D_e]_{ij}, \qquad i,j = 1,\ldots,k, \tag{2.13a}$$

$$[\tilde{A}^*_{\partial e}]_{ij} = [V_e^t A^*_{\partial e} V_e]_{ij}, \qquad i,j = 1,\ldots,k. \tag{2.13b}$$

Using these matrices $\tilde{D}_e$ and $\tilde{A}^*_{\partial e}$ we can compute κ_e^* and show that it is well defined.

Lemma 2.5.3. *Let $e \in \mathcal{T}_h$ be an arbitrary element, and let $\tilde{D}_e, \tilde{A}^*_{\partial e}$ be the matrices defined as in (2.13). The constant κ_e^* is well defined and satisfies*

$$\kappa_e^* = \lambda_{max}\big(\tilde{D}_e^{-1}\tilde{A}^*_{\partial e}\big), \tag{2.14}$$

where $\lambda_{max}\big(\tilde{D}_e^{-1}\tilde{A}^*_{\partial e}\big)$ denotes the largest eigenvalue of $\tilde{D}_e^{-1}\tilde{A}^*_{\partial e}$.

Proof. First, consider the decomposition $V_e^t A_e V_e = D_e$ which was used to construct $\tilde{D}_e$ and $\tilde{A}^*_{\partial e}$. Since matrix A_e has rank k and the last $n-k$ entries of D_e are zero, and since V_e is nonsingular, this implies that the last $n-k$ columns of V_e span the kernel of A_e. From Lemma 2.5.2 it follows that these columns are also in the kernel of $A^*_{\partial e}$. Now let $\underline{\mathbf{w}} \in \mathbb{R}^n$, and let $\underline{\tilde{\mathbf{w}}} \in \mathbb{R}^k$ be the vector composed of the first k entries of $\underline{\mathbf{w}}$. We can then obtain the following relation:

$$\underline{\mathbf{w}}^t V_e^t A^*_{\partial e} V_e \underline{\mathbf{w}} = \underline{\tilde{\mathbf{w}}}^t \tilde{A}^*_{\partial e} \underline{\tilde{\mathbf{w}}}. \tag{2.15}$$

Since A_e is positive semidefinite, it also follows that all entries of $\tilde{D}_e$ are strictly positive. Furthermore, since $A^*_{\partial e}$ is positive semidefinite, the matrix

$\tilde{A}^*_{\partial e}$ will be positive semidefinite as well. Using these properties, we can prove (2.14) as follows:

$$\begin{aligned}\kappa_e^* &:= \sup_{\mathbf{u}\in U_e,\, a_e^{(C)}(\mathbf{u},\mathbf{u})\neq 0} \frac{a_{\partial e}^{(C*)}(\mathbf{u},\mathbf{u})}{a_e^{(C)}(\mathbf{u},\mathbf{u})} = \sup_{\underline{\mathbf{u}}\in\mathbb{R}^n, \underline{\mathbf{u}}^t A_e \underline{\mathbf{u}}\neq 0} \frac{\underline{\mathbf{u}}^t A^*_{\partial e}\underline{\mathbf{u}}}{\underline{\mathbf{u}}^t A_e \underline{\mathbf{u}}} \\ &= \sup_{\underline{\mathbf{w}}\in\mathbb{R}^n, \underline{\mathbf{w}}^t D_e \underline{\mathbf{w}}\neq 0} \frac{\underline{\mathbf{w}}^t V_e^t A^*_{\partial e} V_e \underline{\mathbf{w}}}{\underline{\mathbf{w}}^t D_e \underline{\mathbf{w}}} = \sup_{\tilde{\underline{\mathbf{w}}}\in\mathbb{R}^k, \tilde{\underline{\mathbf{w}}}\neq \underline{\mathbf{0}}} \frac{\tilde{\underline{\mathbf{w}}}^t \tilde{A}^*_{\partial e}\tilde{\underline{\mathbf{w}}}}{\tilde{\underline{\mathbf{w}}}^t \tilde{D}_e \tilde{\underline{\mathbf{w}}}} = \lambda_{max}\big(\tilde{D}_e^{-1}\tilde{A}^*_{\partial e}\big).\end{aligned}$$

In the third step we substituted $\underline{\mathbf{u}}$ by $V_e\underline{\mathbf{w}}$, in the fourth step we used (2.15), and in the last step we used Lemma 2.A.5 combined with the fact that $\tilde{D}_e$ is positive definite and $\tilde{A}^*_{\partial e}$ is positive semidefinite. □

Remark 2.5.4. *A symmetric Gauss elimination procedure or a singular value decomposition algorithm usually does not give the exact decomposition $V_e^t A_e V_e = D_e$, but only a numerical approximation. The diagonal entries of D_e are then considered to be 0 when they are smaller than a given tolerance.*

Remark 2.5.5. *The largest eigenvalue $\lambda_{max}\big(\tilde{D}_e^{-1}\tilde{A}^*_{\partial e}\big)$ can be efficiently obtained using a power iteration method.*

We can now derive the following sufficient estimate for the penalty term.

Theorem 2.5.6. *Let $e \in \mathcal{T}_h$ be an arbitrary element, and let $c_\kappa \geq 1$ be an arbitrary constant. If $\eta_e \geq c_\kappa \kappa_e^*$, then $a_e(\mathbf{u},\mathbf{u}) \geq 0$ for all $\mathbf{u} \in U_h$. Moreover, if $c_\kappa > 1$, then*

$$a_e(\mathbf{u},\mathbf{u}) \geq c_{coer}|\mathbf{u}|^2_{1,e}, \qquad \mathbf{u}\in U_h, \tag{2.16}$$

where

$$c_{coer} := \sup_{x\in[1,c_\kappa]} \min\left\{1 - x^{-1}, \frac{c_\kappa - x}{c_\kappa}\right\} > 0.$$

Proof. Take an arbitrary function $\mathbf{u} \in U_h$ and scalar $x \in [1, c_\kappa]$. We can then derive the following inequality:

$$\begin{aligned} a_e(\mathbf{u},\mathbf{u}) &= a_e^{(C)}(\mathbf{u},\mathbf{u}) + 2a_{\partial e}^{(DG)}(\mathbf{u},\mathbf{u}) + \eta_e a_{\partial e}^{(IP)}(\mathbf{u},\mathbf{u}) \\ &\geq a_e^{(C)}(\mathbf{u},\mathbf{u}) - x^{-1}(\kappa_e^*)^{-1} a_{\partial e}^{(C*)}(\mathbf{u},\mathbf{u}) + (\eta_e - x\kappa_e^*) a_{\partial e}^{(IP)}(\mathbf{u},\mathbf{u}) \\ &\geq (1 - x^{-1}) a_e^{(C)}(\mathbf{u},\mathbf{u}) + (c_\kappa - x)\kappa_e^* a_{\partial e}^{(IP)}(\mathbf{u},\mathbf{u}). \end{aligned}$$

In the second line we used Lemma 2.5.1 with $c = x\kappa_e^*$, and in the last line we used the definition of κ_e^*. Now note that we can write $|\mathbf{u}|_{1,e}^2 = a_e^{(C)}(\mathbf{u},\mathbf{u}) + c_\kappa \kappa_e^* a_{\partial e}^{(DG)}(\mathbf{u},\mathbf{u})$. Combining this with the inequality above gives

$$a_e(\mathbf{u},\mathbf{u}) \geq \min\left\{1 - x^{-1}, \frac{c_\kappa - x}{c_\kappa}\right\} |\mathbf{u}|_{1,e}^2 \geq 0.$$

Taking the supremum over all $x \in [1, c_\kappa]$ results in (2.16). □

The penalty term estimate depends on the constant κ_e^*. However, this constant does not include any effects of the normal vector on the positivity of the bilinear operator, which may cause the penalty term estimate to be less sharp. Therefore, we consider an additional penalty term estimate which does include the effect of the normal vector, and is shown to be considerably sharper in Section 2.7. To do this, we first define the tensor field $\mathbf{c}_{\hat{\mathbf{n}}} \in \bigotimes_{e\in\mathcal{T}_h} L^\infty(\partial e)_{sym}^{m\times m}$ as follows:

$$\mathbf{c}_{\hat{\mathbf{n}}}|_{\partial e} := (\hat{\mathbf{n}} \cdot C \cdot \hat{\mathbf{n}})|_{\partial e}, \qquad\qquad e \in \mathcal{T}_h.$$

where $\hat{\mathbf{n}}|_{\partial e}$ is the outward pointing normal vector of element e. We also define the following function space:

$$\hat{U}_h := \Big\{ \hat{\mathbf{u}} \in \bigotimes_{e\in\mathcal{T}_h} L^2(\partial e)^m \;\Big|\; \hat{\mathbf{u}}|_{\partial e} = (\hat{\mathbf{n}} \cdot C : \nabla \mathbf{u})|_{\partial e},$$
$$\text{for some } u \in U_h, \text{ for all } e \in \mathcal{T}_h \Big\}.$$

From Lemma 2.A.2 it follows that there exists a pseudoinverse $\mathbf{c}_{\hat{\mathbf{n}}}^{-1} \in \bigotimes_{e\in\mathcal{T}_h} L^\infty(\partial e)_{sym}^{m\times m}$ such that $\mathbf{c}_{\hat{\mathbf{n}}}^{-1} \cdot \mathbf{c}_{\hat{\mathbf{n}}} \cdot \hat{\mathbf{u}} = \mathbf{c}_{\hat{\mathbf{n}}} \cdot \mathbf{c}_{\hat{\mathbf{n}}}^{-1} \cdot \hat{\mathbf{u}} = \hat{\mathbf{u}}$ for all $\hat{\mathbf{u}} \in \hat{U}_h$. We use this tensor field to define an alternative auxiliary bilinear operator $a_{\partial e}^{(C**)} : U_h \times U_h \to \mathbb{R}$ as follows:

$$a_{\partial e}^{(C**)}(\mathbf{u},\mathbf{w}) := \int_{\partial e} (\hat{\mathbf{n}} \cdot C : \nabla \mathbf{u})^t \cdot \nu_h^{-1} \mathbf{c}_{\hat{\mathbf{n}}}^{-1} \cdot (\hat{\mathbf{n}} \cdot C : \nabla \mathbf{w})\, ds.$$

The penalty term estimate and the coercivity result are obtained in the same way as before, except that we now use $a_{\partial e}^{(C**)}$ instead of $a_{\partial e}^{(C*)}$. We start again by deriving a bound on $a_{\partial e}^{(DG)}$:

Lemma 2.5.7. *Consider an arbitrary element $e \in \mathcal{T}_h$, and let $c > 0$ be an arbitrary positive constant. Then the following inequality holds:*

$$|2a_{\partial e}^{(DG)}(\mathbf{u},\mathbf{u})| \leq c^{-1} a_{\partial e}^{(C**)}(\mathbf{u},\mathbf{u}) + c a_{\partial e}^{(IP)}(\mathbf{u},\mathbf{u}), \qquad \mathbf{u} \in U. \tag{2.17}$$

Proof. From Lemma 2.A.2 it follows that $\mathbf{c}_{\hat{\mathbf{n}}}$ and $\mathbf{c}_{\hat{\mathbf{n}}}^{-1}$ are positive semidefinite tensor fields, and therefore there exist symmetric positive semidefinite tensor fields $\mathbf{c}_{\hat{\mathbf{n}}}^{1/2}, \mathbf{c}_{\hat{\mathbf{n}}}^{-1/2} \in \bigotimes_{e\in\mathcal{T}_h} L^\infty(\partial e)^{m\times m}_{sym}$ such that $\mathbf{c}_{\hat{\mathbf{n}}}^{1/2}\cdot\mathbf{c}_{\hat{\mathbf{n}}}^{1/2} = \mathbf{c}_{\hat{\mathbf{n}}}$ and $\mathbf{c}_{\hat{\mathbf{n}}}^{-1/2}\cdot\mathbf{c}_{\hat{\mathbf{n}}}^{-1/2} = \mathbf{c}_{\hat{\mathbf{n}}}^{-1}$, and such that $\mathbf{c}_{\hat{\mathbf{n}}}^{-1/2}\cdot\mathbf{c}_{\hat{\mathbf{n}}}^{1/2}\cdot\hat{\mathbf{u}} = \mathbf{c}_{\hat{\mathbf{n}}}^{1/2}\cdot\mathbf{c}_{\hat{\mathbf{n}}}^{-1/2}\cdot\hat{\mathbf{u}} = \hat{\mathbf{u}}$ for all $\hat{\mathbf{u}} \in \hat{U}_h$.

Now take an arbitrary function $u \in U_h$, and define the function $\hat{\mathbf{u}} \in \hat{U}_h$ as follows:

$$\hat{\mathbf{u}}|_{\partial e} := (\hat{\mathbf{n}}\cdot C : \nabla\mathbf{u})|_{\partial e}, \qquad\qquad e \in \mathcal{T}_h.$$

We can then write

$$\begin{aligned}
2a^{(DG)}_{\partial e}(\mathbf{u},\mathbf{u}) &= \int_{\partial e} 2\Big(c^{1/2}\nu_h^{1/2}(\mathbf{u}^* - \mathbf{u}),\ c^{-1/2}\nu_h^{-1/2}\hat{\mathbf{u}}\Big)\,ds \\
&= \int_{\partial e} 2\Big(c^{1/2}\nu_h^{1/2}(\mathbf{u}^* - \mathbf{u}),\ c^{-1/2}\nu_h^{-1/2}\mathbf{c}_{\hat{\mathbf{n}}}^{1/2}\cdot\mathbf{c}_{\hat{\mathbf{n}}}^{-1/2}\cdot\hat{\mathbf{u}}\Big)\,ds \\
&= \int_{\partial e} 2\Big(c^{1/2}\nu_h^{1/2}\mathbf{c}_{\hat{\mathbf{n}}}^{1/2}\cdot(\mathbf{u}^* - \mathbf{u}),\ c^{-1/2}\nu_h^{-1/2}\mathbf{c}_{\hat{\mathbf{n}}}^{-1/2}\cdot\hat{\mathbf{u}}\Big)\,ds.
\end{aligned}$$

Using the Cauchy–Schwarz and the Cauchy inequalities, we can then obtain

$$\begin{aligned}
\big|2a^{(DG)}_{\partial e}(\mathbf{u},\mathbf{u})\big| &\le c\int_{\partial e}\big\|\nu_h^{1/2}\mathbf{c}_{\hat{\mathbf{n}}}^{1/2}\cdot(\mathbf{u}^* - \mathbf{u})\big\|^2\,ds \\
&\quad + c^{-1}\int_{\partial e}\big\|\nu_h^{-1/2}\mathbf{c}_{\hat{\mathbf{n}}}^{-1/2}\cdot\hat{\mathbf{u}}\big\|^2\,ds \\
&= c^{-1}a^{(C**)}_{\partial e}(\mathbf{u},\mathbf{u}) + ca^{(IP)}_{\partial e}(\mathbf{u},\mathbf{u}).
\end{aligned}$$

□

We now use the bilinear operator $a^{(C**)}_{\partial e}$ to construct the following constant:

$$\kappa_e^{**} := \sup_{\mathbf{u}\in U_e,\ a_e^{(C)}(\mathbf{u},\mathbf{u})\neq 0} \frac{a^{(C**)}_{\partial e}(\mathbf{u},\mathbf{u})}{a_e^{(C)}(\mathbf{u},\mathbf{u})}.$$

The proof of the existence of this constant and the way to compute it is analogous to κ_e^*. In a similar way as before we can use this constant to obtain the following sufficient penalty term estimate.

Theorem 2.5.8. *Let $e \in \mathcal{T}_h$ be an arbitrary element, and let $c_\kappa \ge 1$ be an arbitrary constant. If $\eta_e \ge c_\kappa\kappa_e^{**}$, then $a_e(\mathbf{u},\mathbf{u}) \ge 0$ for all $\mathbf{u} \in U_h$. Moreover, if $c_\kappa > 1$, then*

$$a_e(\mathbf{u},\mathbf{u}) \ge c_{coer}|\mathbf{u}|^2_{1,e}, \qquad\qquad \mathbf{u} \in U_h, \tag{2.18}$$

where

$$c_{coer} := \sup_{x\in[1,c_\kappa]} \min\left\{1 - x^{-1}, \frac{c_\kappa - x}{c_\kappa}\right\}.$$

Proof. The proof is analogous to that of Theorem 2.5.6. □

We have now derived conditions for the penalty term to ensure that a_h is positive semidefinite and showed how the penalty term can be computed. In the next section we will derive time step estimates to ensure that the local time-stepping scheme is stable.

2.6 Sufficient time step estimates

We start by rewriting the DG method as a linear system of ordinary differential equations. We then show how we can obtain sufficient upper bounds for the spectral radius of $M^{-1}A$, and therefore sufficient lower bounds for the time step size for a large class of explicit time integration schemes, by splitting the mass mass matrix M and stiffness matrix A into multiple parts. Finally, we introduce weighted mesh decompositions to explain how this splitting of matrices can be done efficiently.

2.6.1 A system of ordinary differential equations

Let $\{\mathbf{w}_i\}_{i=1}^N$ be a linear basis of U_h and define, for $\mathbf{u} \in U_h$, the vector $\underline{\mathbf{u}} \in \mathbb{R}^N$ such that $\mathbf{u} = \sum_{i=1}^N \underline{\mathbf{u}}_i \mathbf{w}_i$. We can rewrite the DG method, given in (2.6), as the following system of ordinary differential equations: we solve $\underline{\mathbf{u}} : [0,T] \to \mathbb{R}^N$, such that

$$M_h \partial_t^2 \underline{\mathbf{u}} + A_h \underline{\mathbf{u}} = \underline{\mathbf{f}}_h^*, \qquad t \in [0,T], \tag{2.19a}$$

$$\underline{\mathbf{u}}|_{t=0} = \underline{\mathbf{u}}_{0,h} := M_h^{-1} \underline{\mathbf{u}}_{0,h}^*, \tag{2.19b}$$

$$\partial_t \underline{\mathbf{u}}|_{t=0} = \underline{\mathbf{v}}_{0,h} := M_h^{-1} \underline{\mathbf{v}}_{0,h}^*, \tag{2.19c}$$

where $M_h, A_h \in \mathbb{R}^{N\times N}$ are matrices, $\underline{\mathbf{u}}^*_{0,h}, \underline{\mathbf{v}}^*_{0,h} \in \mathbb{R}^N$ are vectors, and $\underline{\mathbf{f}}^*_h : [0,T] \to \mathbb{R}^N$ is a vector function, defined as follows:

$$\begin{aligned}
[M_h]_{ij} &:= (\rho\mathbf{w}_i, \mathbf{w}_j), && i,j = 1,\dots,N,\\
[A_h]_{ij} &:= a_h(\mathbf{w}_i, \mathbf{w}_j), && i,j = 1,\dots,N,\\
[\underline{\mathbf{u}}^*_{0,h}]_i &:= (\rho\mathbf{u}_0, \mathbf{w}_i), && i = 1,\dots,N,\\
[\underline{\mathbf{v}}^*_{0,h}]_i &:= (\rho\mathbf{v}_0, \mathbf{w}_i), && i = 1,\dots,N,\\
[\underline{\mathbf{f}}^*_h(t)]_i &:= (\mathbf{f}(t), \mathbf{w}_i), && i = 1,\dots,N,\ t \in [0,T].
\end{aligned}$$

For a large class of explicit time integrators, including the leap-frog or central difference scheme, the Lax–Wendroff schemes, and explicit Runge–Kutta schemes, the time step size condition is of the form

$$\Delta t \leq \frac{c_{method}}{\sqrt{\lambda_{max}(M^{-1}A)}}, \tag{2.20}$$

where $c_{method} > 0$ is a constant, depending only on the type of time integration method, and $\lambda_{max}(M^{-1}A)$ is the largest eigenvalue of $M^{-1}A$, which is also known as the spectral radius of $M^{-1}A$. For example, the stability condition for the leap-frog scheme is well known to be (2.20) with $c_{method} = 2$. Because of the form of (2.20), it remains to find an upper estimate for the spectral radius. In the next section we show how this can be done by splitting the matrices M and A into multiple parts.

2.6.2 Spectral radius estimates by splitting matrices

In order to obtain a bound for the spectral radius we first introduce the mapping $\mathcal{I} : \mathbb{R}^{N\times N}_{sym} \to \mathbb{R}^{N\times N}_{sym}$, which maps a symmetric matrix to a diagonal matrix with entries 0 and 1 to indicate the nonzero rows or columns of the input matrix:

$$[\mathcal{I}(S)]_{ij} = \begin{cases} 1 & i = j, \text{ and } S_{ik} \neq 0 \text{ for somes } k \in \{1,..,N\},\\ 0 & \text{otherwise.} \end{cases}$$

We also define $\mathcal{I}^*(S)$ as the matrix $\mathcal{I}(S)$ with all zero-columns removed. Using these definitions we can formulate the following theorem.

Theorem 2.6.1. *Let $M \in \mathbb{R}^{N\times N}_{sym}$ be a symmetric positive definite matrix and $A \in \mathbb{R}^{N\times N}_{sym}$ a symmetric positive semidefinite matrix. Also let*

$M_{(i)}, A_{(i)} \in \mathbb{R}^{N\times N}_{sym}$ for $i = 1, \dots, n$, be symmetric matrices such that

$$\sum_{i=1}^{n} M_{(i)} = M, \qquad \sum_{i=1}^{n} A_{(i)} = A, \tag{2.21a}$$

$$M'_{(i)} \succ 0, \qquad I_{(i)} A_{(i)} I_{(i)} = A_{(i)}, \qquad i = 1, \dots, n, \tag{2.21b}$$

where $M'_{(i)} := (I^*_{(i)})^t M_{(i)} I^*_{(i)}$, $I_{(i)} := \mathcal{I}(M_{(i)})$ and $I^*_{(i)} := \mathcal{I}^*(M_{(i)})$. Then

$$\lambda_{max}(M^{-1}A) \leq \max_{i=1,\dots,n} \lambda_{max}\big((M'_{(i)})^{-1} A'_{(i)}\big), \tag{2.22}$$

where $A'_{(i)} := (I^*_{(i)})^t A_{(i)} I^*_{(i)}$, and $\lambda_{max}(\cdot)$ denotes the largest eigenvalue in magnitude.

Remark 2.6.2. *The matrices $M'_{(i)}$ and $A'_{(i)}$ are the submatrices of $M_{(i)}$ and $A_{(i)}$, respectively, obtained by removing all rows and columns corresponding to the zero rows and columns of $M_{(i)}$. The condition $I_{(i)} A_{(i)} I_{(i)} = A_{(i)}$ means that any zero column or row of $M_{(i)}$ is also a zero column or row of $A_{(i)}$, and the condition $M'_{(i)} \succ 0$ means that the submatrices of M are positive definite.*

Proof. For any $\underline{\mathbf{u}} \in \mathbb{R}^n$, define the following set of indices:

$$\mathbb{I}(\underline{\mathbf{u}}) := \big\{ i \in \{1, \dots, n\} \bigm| I_{(i)} \underline{\mathbf{u}} \neq \underline{\mathbf{0}} \big\}.$$

Using Lemma 2.A.5 and Lemma 2.A.3 we can then bound the largest eigenvalue as follows:

$$\begin{aligned}
\lambda_{max}(M^{-1}A) &= \sup_{\underline{\mathbf{u}} \in \mathbb{R}^N,\, \underline{\mathbf{u}} \neq \underline{\mathbf{0}}} \frac{\underline{\mathbf{u}}^t A \underline{\mathbf{u}}}{\underline{\mathbf{u}}^t M \underline{\mathbf{u}}} \\
&= \sup_{\underline{\mathbf{u}} \in \mathbb{R}^N,\, \underline{\mathbf{u}} \neq \underline{\mathbf{0}}} \frac{\sum_{i=1}^n \underline{\mathbf{u}}^t A_{(i)} \underline{\mathbf{u}}}{\sum_{i=1}^n \underline{\mathbf{u}}^t M_{(i)} \underline{\mathbf{u}}} = \sup_{\underline{\mathbf{u}} \in \mathbb{R}^N,\, \underline{\mathbf{u}} \neq \underline{\mathbf{0}}} \frac{\sum_{i \in \mathbb{I}(\underline{\mathbf{u}})} \underline{\mathbf{u}}^t A_{(i)} \underline{\mathbf{u}}}{\sum_{i \in \mathbb{I}(\underline{\mathbf{u}})} \underline{\mathbf{u}}^t M_{(i)} \underline{\mathbf{u}}} \\
&\leq \sup_{\underline{\mathbf{u}} \in \mathbb{R}^N,\, \underline{\mathbf{u}} \neq \underline{\mathbf{0}}} \max_{i \in \mathbb{I}(\underline{\mathbf{u}})} \frac{|\underline{\mathbf{u}}^t A_{(i)} \underline{\mathbf{u}}|}{\underline{\mathbf{u}}^t M_{(i)} \underline{\mathbf{u}}} = \max_{i=1,\dots,n} \sup_{\underline{\mathbf{u}} \in \mathbb{R}^N,\, I_{(i)}\underline{\mathbf{u}} \neq \underline{\mathbf{0}}} \frac{|\underline{\mathbf{u}}^t A_{(i)} \underline{\mathbf{u}}|}{\underline{\mathbf{u}}^t M_{(i)} \underline{\mathbf{u}}}.
\end{aligned}$$

Using Lemma 2.A.5 again, we can obtain, for any $i = 1, .., n$, the following:

$$\sup_{\underline{\mathbf{u}} \in \mathbb{R}^N,\, I_{(i)}\underline{\mathbf{u}} \neq \underline{\mathbf{0}}} \frac{|\underline{\mathbf{u}}^t A_{(i)} \underline{\mathbf{u}}|}{\underline{\mathbf{u}}^t M_{(i)} \underline{\mathbf{u}}} = \sup_{\underline{\mathbf{u}} \in \mathbb{R}^{N_i},\, \underline{\mathbf{u}} \neq \underline{\mathbf{0}}} \frac{|\underline{\mathbf{u}}^t A'_{(i)} \underline{\mathbf{u}}|}{\underline{\mathbf{u}}^t M'_{(i)} \underline{\mathbf{u}}} = \lambda_{max}\big((M'_{(i)})^{-1} A'_{(i)}\big),$$

where N_i is the number of nonzero columns of $M_{(i)}$. Combining these results gives (2.22). □

To apply the above theorem it remains to find a decomposition of the matrices M and A such that (2.21) is satisfied. For continuous finite elements such a decomposition can be easily obtained from the element matrices,

$$M = \sum_{e \in \mathcal{T}_h} M_e, \qquad A = \sum_{e \in \mathcal{T}_h} A_e,$$

where M_e and A_e are the element matrices corresponding to the mass matrix M and stiffness matrix A, respectively. Using Theorem 2.6.1 we then obtain the following estimate for the spectral radius:

$$\lambda_{max}(M^{-1}A) \leq \max_{e \in \mathcal{T}_h} \lambda_{max}\big((M_e')^{-1} A_e'\big),$$

where $M_e' := (I_e^*)^t M_e I_e^*$, $A_e' := (I_e^*)^t A_e I_e^*$ and $I_e^* := \mathcal{I}^*(M_e)$. In other words, the largest eigenvalue of the global matrix is bounded by the supremum over all elements of the largest eigenvalue of the element matrix. This result was already mentioned by [40]. For discontinuous elements, however, a suitable decomposition of the matrices is less straightforward due to the face integral terms. In the next subsection we show how we can decompose the matrices for discontinuous elements, using a weighted mesh decomposition.

2.6.3 A weighted mesh decomposition

We define a weighted submesh $\omega : \mathcal{T}_h \cup \mathcal{F}_h \to [0,1]$ to be a function that assigns to every element and face a weight value ω_e and ω_f between 0 and 1, such that if $\omega_f > 0$ for a certain face f, then $\omega_e > 0$ for the adjacent elements $e \in \mathcal{T}_f$. We call a set of weighted submeshes $\mathcal{W}_h$ a weighted mesh decomposition of $\mathcal{T}_h$ if the sum of all weighted submeshes adds up to one for every face and element: $\sum_{\omega \in \mathcal{W}_h} \omega_e = 1$ for all $e \in \mathcal{T}_h$ and $\sum_{\omega \in \mathcal{W}_h} \omega_f = 1$ for all $f \in \mathcal{F}_h$. An illustration of a weighted mesh decomposition is given in Figure 2.1.

We can use a weighted submesh to construct bilinear forms $(\cdot,\cdot)_\omega, a_\omega : U_h \times U_h \to \mathbb{R}$ as follows:

$$\begin{aligned}
(\mathbf{u},\mathbf{w})_\omega &:= \sum_{e \in \mathcal{T}_h} \omega_e \int_e \rho \mathbf{u} \cdot \mathbf{w} \, dx, \\
a_\omega(\mathbf{u},\mathbf{w}) &:= a_\omega^{(C)}(\mathbf{u},\mathbf{w}) + a_\omega^{(DG)}(\mathbf{u},\mathbf{w}) + a_\omega^{(DG)}(\mathbf{w},\mathbf{u}) + a_\omega^{(IP)}(\mathbf{u},\mathbf{w})
\end{aligned}$$

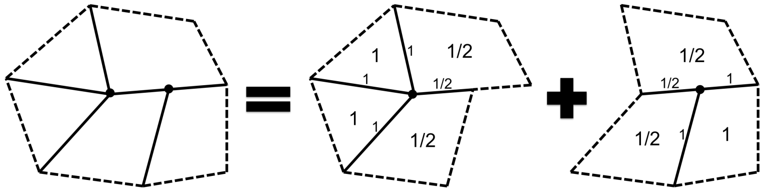


Figure 2.1: A weighted mesh decomposition. The larger numbers denote the element weights, while the smaller numbers denote the face weights. Weight values of elements and faces outside the illustrated subdomains are zero.

with

$$a_\omega^{(C)}(\mathbf{u},\mathbf{w}) := \sum_{e\in\mathcal{T}_h} \omega_e \int_e (\nabla\mathbf{u})^t : C : \nabla\mathbf{w}\, dx,$$
$$a_\omega^{(DG)}(\mathbf{u},\mathbf{w}) := \sum_{e\in\mathcal{T}_h}\sum_{f\in\mathcal{F}_e} \omega_f \int_{\partial e\cap f} (\mathbf{u}^* - \mathbf{u})\hat{\mathbf{n}} : C : \nabla\mathbf{w}\, ds,$$
$$a_\omega^{(IP)}(\mathbf{u},\mathbf{w}) := \sum_{e\in\mathcal{T}_h}\sum_{f\in\mathcal{F}_e} \omega_f\eta_e \int_{\partial e\cap f} (\mathbf{u}^* - \mathbf{u})\hat{\mathbf{n}} : \nu_h C : \hat{\mathbf{n}}(\mathbf{w}^* - \mathbf{w})\, ds.$$

Note that $(\mathbf{u},\mathbf{w}) = \sum_{\omega\in\mathcal{W}_h}(\mathbf{u},\mathbf{w})_\omega$ and $a_h(\mathbf{u},\mathbf{w}) = \sum_{\omega\in\mathcal{W}_h} a_\omega(\mathbf{u},\mathbf{w})$, for all $\mathbf{u},\mathbf{w}\in U_h$.

For the numerical tests, we will in particular consider a weighted mesh decomposition based on the vertices, as illustrated in Figure 2.2. The vertex-based mesh decomposition is given by $\mathcal{W}_h := \{\omega^{(q)}\}_{q\in\mathcal{Q}}$, with

$$\omega_e^{(q)} := \begin{cases} \frac{1}{|\mathcal{Q}_e|} & e\in\mathcal{T}_q, \\ 0 & \text{otherwise}, \end{cases} \qquad \omega_f^{(q)} := \begin{cases} \frac{1}{|\mathcal{Q}_f|} & e\in\mathcal{F}_q, \\ 0 & \text{otherwise}, \end{cases} \tag{2.23}$$

where $|\mathcal{Q}_e|$, $|\mathcal{Q}_f|$ are the number of vertices adjacent to element e and face f, respectively, and $\mathcal{T}_q$, $\mathcal{F}_q$ are the set of elements and faces adjacent to q, respectively.

Now let $\{\mathbf{w}_i\}_{i=1}^N$ be a linear basis of U_h, such that every basis function is nonzero on only a single element e_i. We can use a weighted mesh decomposition $\mathcal{W}_h$ to decompose the mass matrix and stiffness matrix as follows:

$$M = \sum_{\omega\in\mathcal{W}_h} M_\omega, \qquad A = \sum_{\omega\in\mathcal{W}_h} A_\omega,$$

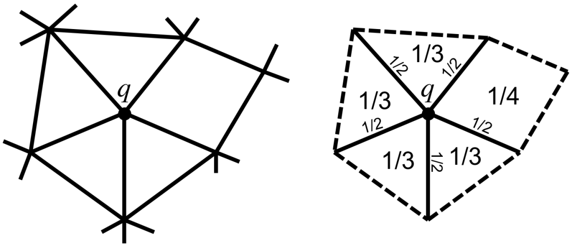


Figure 2.2: Illustration of a vertex-based mesh decomposition. For every vertex q, a weighted submesh $\omega^{(q)}$ is created assigning nonzero values only for elements and faces directly adjacent to the vertex.

where $[M_\omega]_{ij} := (\mathbf{w}_i, \mathbf{w}_j)_\omega$ and $[A_\omega]_{ij} := a_\omega(\mathbf{w}_i, \mathbf{w}_j)$, for $i, j = 1, \ldots, N$. Using Theorem 2.6.1 we can immediately obtain the following estimate for the spectral radius and therefore the time step size.

Theorem 2.6.3. *Let $\mathcal{W}_h$ be a weighted mesh decomposition. Then*

$$\lambda_{max}(M^{-1}A) \leq \max_{\omega \in \mathcal{W}_h} \lambda_{max}\big((M'_\omega)^{-1} A'_\omega\big), \tag{2.24}$$

where $M'_\omega := (I^*_\omega)^t M_\omega I^*_\omega$, $A'_\omega := (I^*_\omega)^t A_\omega I^*_\omega$, and $I^*_\omega := \mathcal{I}^*(M_\omega)$, and where $\lambda_{max}(\cdot)$ denotes the largest eigenvalue in magnitude.

Remark 2.6.4. *When the weighted submeshes ω are nonzero for only a few elements and faces, then M'_ω and A'_ω are relatively small matrices. The largest eigenvalue $\lambda_{max}\big((M'_\omega)^{-1} A'_\omega\big)$ can then be efficiently computed in parallel for each submatrix, using a power iteration method requiring only a relatively small number of iterations.*

In the next section we show several numerical results illustrating the sharpness of the penalty term and time step estimates.

2.7 Numerical results

2.7.1 Computing the spectral radius for periodic meshes

To test the sharpness of the penalty term estimates and time step estimates we consider a d-dimensional cubic domain of the form $(0, N)^d$ with periodic boundary conditions. We then create a uniform mesh of N^d unit cubes, after which we subdivide every cube into smaller elements and choose basis function sets and material parameters for every subelement. These subelements, basis functions and material parameters are chosen identically for every cube. An illustration of such a mesh is given in Figure 2.3. The advantage of such a uniform periodic mesh is that we can rather easily obtain the exact spectral radius by using a Fourier analysis, in a way similar to the von Neumann method for finite difference schemes.

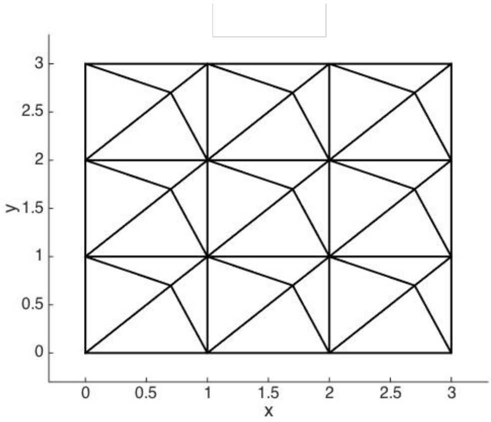


Figure 2.3: Square 2D mesh consisting of 3^2 unit cubes, where each square is identically subdivided into four distorted triangles.

To apply a Fourier analysis we first choose a linear basis $\{\mathbf{w}_i\}_{i=1}^{N^d\times M}$ of the discrete function space U, such that the linear basis is of the form

$$\{\mathbf{w}_i\}_{i=1}^{N^d\times M} = \bigcup_{\mathbf{k}\in\mathbb{Z}_N^d} \{\mathbf{w}_{\mathbf{k},i}\}_{i=1}^{M},$$

where $\mathbf{k}\in\mathbb{Z}_N^d$ is the identifier of unit cube $(k_1, k_1+1)\times\cdots\times(k_d, k_d+1)$ and $\{\mathbf{w}_{\mathbf{k},i}\}_{i=1}^{M}$ is a linear basis of the discrete space U restricted to this cube. We can then define the submatrices $M_{\mathbf{k},\mathbf{l}}, A_{\mathbf{k},\mathbf{l}} \in \mathbb{R}^{M\times M}$ as follows:

$$\begin{aligned}
[M_{\mathbf{k},\mathbf{l}}]_{ij} &:= (\rho\mathbf{w}_{\mathbf{k},i}, \mathbf{w}_{\mathbf{l},j}), & i,j &= 1,\dots,M,\ \mathbf{k},\mathbf{l}\in\mathbb{Z}_N^d,\\
[A_{\mathbf{k},\mathbf{l}}]_{ij} &:= a_h(\rho\mathbf{w}_{\mathbf{k},i}, \mathbf{w}_{\mathbf{l},j}), & i,j &= 1,\dots,M,\ \mathbf{k},\mathbf{l}\in\mathbb{Z}_N^d,
\end{aligned}$$

By construction of the mesh, most of these submatrices are identical. Fix any $\mathbf{l}\in\mathbb{Z}_N^d$. Then the submatrices $M_{\mathbf{k},\mathbf{k}+\mathbf{l}}$ are identical for any $\mathbf{k}\in\mathbb{Z}_N^d$. The same holds for $A_{\mathbf{k},\mathbf{k}+\mathbf{l}}$. Moreover, by definition of the mass matrix, $M_{\mathbf{k},\mathbf{k}+\mathbf{l}}$ is only nonzero when $\mathbf{l}=\mathbf{0}$, and by construction of the stiffness matrix, $A_{\mathbf{k},\mathbf{k}+\mathbf{l}}$ is only nonzero when $|\mathbf{l}|\leq 1$. Therefore, we only have to consider the submatrices $M_0 := M_{\mathbf{k},\mathbf{k}}$, $A_0 := A_{\mathbf{k},\mathbf{k}}$, and $A_i^{\pm} := A_{\mathbf{k},\mathbf{k}\pm\mathbf{e}_i}$ for $i=1,\dots,d$, where $\mathbf{k}$ is an arbitrary vector in $\mathbb{Z}_N^d$ and $\mathbf{e}_i$ is the unit vector in direction i.

Now let $\underline{\mathbf{u}} \in \mathbb{R}^{N^d \times M}$ be a vector of coefficients, and let $\underline{\mathbf{u}}_{\mathbf{k}} \in \mathbb{R}^M$ be the vector of coefficients corresponding to cube $\mathbf{k}$. Suppose that $\underline{\mathbf{w}} = M^{-1}A\underline{\mathbf{u}}$. We can then write

$$\underline{\mathbf{w}}_{\mathbf{k}} = M_0^{-1}\left(A_0\underline{\mathbf{u}}_{\mathbf{k}} + \sum_{i=1}^{d}(A_i^+\underline{\mathbf{u}}_{\mathbf{k}+\mathbf{e}_i} + A_i^-\underline{\mathbf{u}}_{\mathbf{k}-\mathbf{e}_i})\right), \qquad \mathbf{k} \in \mathbb{Z}_N^d.$$

Define $\underline{\mathbf{u}}^{(\mathbf{z})} \in \mathbb{R}^{N^d \times M}$ as follows:

$$\underline{\mathbf{u}}_{\mathbf{k}}^{(\mathbf{z})} = e^{\hat{\imath}(\mathbf{z}\cdot\mathbf{k}/N)2\pi}\underline{\mathbf{u}}_0, \qquad \mathbf{k} \in \mathbb{Z}_N^d, \tag{2.25}$$

where $\underline{\mathbf{u}}_0 \in \mathbb{R}^M$ is an arbitrary vector of coefficients corresponding to a single cube, $\hat{\imath} := \sqrt{-1}$ is the imaginary number, and $\mathbf{z} \in \mathbb{Z}_N^d$ is a vector of integers. Then $\underline{\mathbf{w}}^{(\mathbf{z})} := M^{-1}A\underline{\mathbf{u}}^{(\mathbf{z})}$ satisfies

$$\underline{\mathbf{w}}_{\mathbf{k}}^{(\mathbf{z})} = e^{\hat{\imath}(\mathbf{z}\cdot\mathbf{k}/N)2\pi}Z^{(\mathbf{z})}\underline{\mathbf{u}}_0, \qquad \mathbf{k} \in \mathbb{Z}_N^d, \tag{2.26}$$

where

$$Z^{(\mathbf{z})} := M_0^{-1}\left(A_0 + \sum_{i=1}^{d}(e^{\hat{\imath}(z_i/N)2\pi}A_i^+ + e^{-\hat{\imath}(z_i/N)2\pi}A_i^-)\right), \qquad \mathbf{z} \in \mathbb{Z}_N^d.$$

From (2.25) and (2.26) it follows that if $(\lambda, \underline{\mathbf{u}}_0)$ is an eigenpair of $Z^{(\mathbf{z})}$, then $(\lambda, \underline{\mathbf{u}}^{(\mathbf{z})})$ is an eigenpair of $M^{-1}A$. Since $Z^{(\mathbf{z})}$ has M eigenpairs and since there are N^d possible choices for $\mathbf{z}$, every eigenvalue of $M^{-1}A$ is an eigenvalue of $Z^{(\mathbf{z})}$ for some $\mathbf{z} \in \mathbb{Z}_N^d$. For the time step estimates we are only interested in the largest eigenvalue $\lambda_{max}(M^{-1}A)$, which we can then compute by

$$\lambda_{max}(M^{-1}A) = \sup_{\mathbf{z}\in\mathbb{Z}_N^d} \lambda_{max}(Z^{(\mathbf{z})}).$$

For the numerical tests that we will present here, we have taken $N = 2$, since in most cases the largest eigenvalue $\lambda_{max}(M^{-1}A)$ no longer increases significantly for $N > 2$.

2.7.2 Sharpness of the penalty term and time step estimates

For testing the sharpness of our parameter estimates we use polynomial basis functions up to degree p for simplicial elements, and polynomials up to degree p in the direction of each reference coordinate for quadrilateral and hexahedral elements. First, we consider several regular homogeneous meshes for the acoustic wave equation in 1D, 2D and 3D. After that we test on meshes with deformed elements and meshes with piecewise linear parameter fields. We also test on meshes for electromagnetic and elastic wave problems, including heterogeneous meshes with sharp material contrasts and meshes with sharp contrasts in primary and secondary wave velocities.

To test the sharpness of the parameters we first compute the penalty terms as in Theorem 2.5.6 or Theorem 2.5.8 with $c_\kappa = 1$. We will refer to the first penalty terms as η_e^* and to the second as η_e^{**}. We then find the smallest scale $c_{min} \in [0, 1]$ such that the stiffness matrix A is still positive semidefinite when using the downscaled penalty terms $\eta_{min,e} := c_{min}\eta_e$. We compute c_{min} accurate to two decimal places using the bisection method.

After we have computed η and c_{min}, we consider the time step condition for the leap-frog scheme and use this to compute the time step size in the three ways given below:

$$\Delta t(\eta_{min}) := \frac{2}{\sqrt{\lambda_{max}(M^{-1}A_{min})}}, \tag{2.27a}$$

$$\Delta t(\eta) := \frac{2}{\sqrt{\lambda_{max}(M^{-1}A)}}, \tag{2.27b}$$

$$\Delta t_{est}(\eta) := \frac{2}{\sqrt{\sup_{\omega\in\mathcal{W}_h} \lambda_{max}\big((M'_\omega)^{-1}A'_\omega\big)}}, \tag{2.27c}$$

Here A_{min} is the stiffness matrix that results from using the downscaled penalty terms $\eta_{min,e}$, and M'_ω and A'_ω are the submatrices corresponding to the weighted submesh ω. These time step sizes can be interpreted as follows: $\Delta t(\eta_{min})$ is the largest allowed time step size when using the minimum downscaled penalty terms $\eta_{min,e}$, $\Delta t(\eta)$ is the largest allowed time step size when using the penalty term estimates η_e, and $\Delta t_{est}(\eta)$ is the time step estimate when using the penalty term estimates η_e and a weighted mesh decomposition.

For our time step estimate $\Delta t_{est}(\eta)$ we will use the vertex-based mesh decomposition as given in (2.23). We will measure the sharpness of the penalty term by $\Delta t(\eta_{min})/\Delta t(\eta)$, and we will measure the sharpness of our time step estimate by $\Delta t(\eta)/\Delta t_{est}(\eta)$.

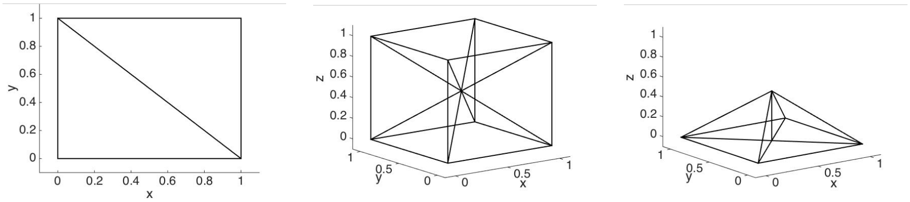
Regular meshes

For the first tests we consider the acoustic wave equation as given in Example 2.3.1, with $c = 1$. We use meshes of the form described in Section

2.7.1, with element subdivisions as listed below. An illustration of some of the element subdivisions is given in Figure 2.4.

- *1D*: mesh constructed from unit intervals.
- *square*: 2D mesh constructed from unit squares.
- *tri*: 2D mesh constructed from unit squares, with each square subdivided into two triangles.
- *cube*: 3D mesh constructed from unit cubes.
- *tet*: 3D mesh constructed from unit cubes, with each cube subdivided into six pyramids, and every pyramid subdivided into four tetrahedra.

The results of the parameter estimates, when using the penalty term η^*, are given in Table 2.1. The results when using η^{**} are given in Table 2.2. From these tables we can already see that our second penalty term estimate η^{**} is in general much sharper than the first estimate η^*. This is true especially for cubes, where η^* causes a reduction in the largest allowed time step size of more than 2. Also, for square and tetrahedral meshes the first penalty term estimate causes a reduction in the time step size of more than a factor 1.5. On the other hand, when using η^{**} the largest allowed time step size is never reduced more than a factor 1.2, and for many of the regular meshes $\Delta t(\eta^{**}_{min})/\Delta t(\eta^{**})$ is even below 1.01. For the 1D, square, and cubic meshes, this penalty term and corresponding time step estimate even coincide with the analytic results derived in [1]. In general, the time step estimate does not reduce the time step size by a factor more than 1.2 when using η^{**} and not more than 1.3 when using η^*.



(a) A square subdivided into two triangles.

(b) A cube subdivided into six pyramids.

(c) A pyramid subdivided into four tetrahedra.

Figure 2.4

Mesh	p	c^*_{min}	$\Delta t(\eta^*_{min})$	$\Delta t(\eta^*)$	$\Delta t_{est}(\eta^*)$	$\frac{\Delta t(\eta^*_{min})}{\Delta t(\eta^*)}$	$\frac{\Delta t(\eta^*)}{\Delta t_{est}(\eta^*)}$
1D	1	1.00	0.5774	0.5774	0.5774	1.00	1.00
	2	1.00	0.2582	0.2582	0.2582	1.00	1.00
	3	1.00	0.1533	0.1533	0.1533	1.00	1.00
square	1	0.25	0.4082	0.2357	0.2019	1.73	1.17
	2	0.33	0.1826	0.1170	0.0956	1.56	1.22
	3	0.38	0.1084	0.0694	0.0554	1.56	1.25
tri	1	0.67	0.2579	0.2273	0.1948	1.13	1.17
	2	0.69	0.1406	0.1250	0.1048	1.12	1.19
	3	0.70	0.0906	0.0739	0.0621	1.23	1.19
cube	1	0.14	0.3333	0.1361	0.1172	2.45	1.16
	2	0.20	0.1491	0.0678	0.0554	2.20	1.22
	3	0.23	0.0885	0.0405	0.0322	2.19	1.26
tet	1	0.38	0.1035	0.0635	0.0560	1.63	1.13
	2	0.44	0.0598	0.0384	0.0336	1.56	1.14
	3	0.48	0.0360	0.0243	0.0212	1.48	1.15

Table 2.1: Parameter estimates on regular meshes, using penalty term η^*.

For the case of tetrahedral meshes we can compare our penalty term estimates with the estimate derived in [55]. The penalty term derived there is equivalent to $\eta_e = p(p+2)$, with the penalty scaling function given by $\nu_h|_{\partial e \cap f} = 1/(\epsilon_f \min_{e \in \mathcal{T}_f} d_{i,e})$, where $d_{i,e}$ is the diameter of the inscribed sphere of e and $\epsilon_f \in \{1/2, 1\}$ is defined as in (2.8b). Their analysis can be readily extended to triangles by replacing the trace inverse inequality for tetrahedra by the trace inverse inequality for triangles, given in Theorem 3 of [79], which is equivalent to setting $\eta_e = p(p+1)$. The results of these estimates are given in Table 2.3, from which we can see that this penalty term estimate has a similar sharpness as η^*, but is significantly less sharp than η^{**}, having a time step size more than 1.5 times smaller than when using η^{**} for $p = 2, 3$ on tetrahedra and $p = 3$ on triangles.

Since η^{**} is significantly sharper than η^*, we will only use η^{**} throughout the following numerical tests.

Meshes with deformed elements

In this subsection we consider the acoustic wave equation with $c = 1$ again, but now using deformed elements. For the penalty term we will only use

Mesh	p	c^{**}_{min}	$\Delta t(\eta^{**}_{min})$	$\Delta t(\eta^{**})$	$\Delta t_{est}(\eta^{**})$	$\frac{\Delta t(\eta^{**}_{min})}{\Delta t(\eta^{**})}$	$\frac{\Delta t(\eta^{**})}{\Delta t_{est}(\eta^{**})}$
1D	1	1.00	0.5774	0.5774	0.5774	1.00	1.00
	2	1.00	0.2582	0.2582	0.2582	1.00	1.00
	3	1.00	0.1533	0.1533	0.1533	1.00	1.00
square	1	1.00	0.4082	0.4082	0.4082	1.00	1.00
	2	1.00	0.1826	0.1826	0.1826	1.00	1.00
	3	1.00	0.1084	0.1084	0.1084	1.00	1.00
tri	1	1.00	0.2582	0.2582	0.2427	1.00	1.06
	2	0.96	0.1406	0.1399	0.1275	1.01	1.10
	3	0.96	0.0906	0.0896	0.0755	1.01	1.19
cube	1	1.00	0.3333	0.3333	0.3333	1.00	1.00
	2	1.00	0.1491	0.1491	0.1491	1.00	1.00
	3	1.00	0.0885	0.0885	0.0885	1.00	1.00
tet	1	0.75	0.1040	0.0918	0.0803	1.13	1.14
	2	0.74	0.0599	0.0510	0.0455	1.17	1.15
	3	0.81	0.0359	0.0320	0.0279	1.12	1.15

Table 2.2: Parameter estimates on regular meshes, using penalty term η^{**}.

Mesh	p	$\Delta t(\eta)$	$\Delta t(\eta^{*})$	$\Delta t(\eta^{**})$	$\frac{\Delta t(\eta^{*})}{\Delta t(\eta)}$	$\frac{\Delta t(\eta^{**})}{\Delta t(\eta)}$
tri	1	0.2280	0.2273	0.2582	1.00	1.13
	2	0.1002	0.1250	0.1399	1.25	1.40
	3	0.0567	0.0739	0.0896	1.30	1.58
tet	1	0.0689	0.0635	0.0918	0.92	1.33
	2	0.0327	0.0384	0.0510	1.17	1.56
	3	0.0196	0.0243	0.0320	1.24	1.63

Table 2.3: Parameter estimates on a regular tetrahedral mesh, using the penalty term, here denoted by η, derived in [55], and using penalty terms η^{*} and η^{**}.

η^{**}. An overview of the different meshes is listed below, and an illustration of the element subdivisions is given in Figures 2.5 and 2.6.

- *rectangular*$[x]$: 2D mesh constructed from unit squares, with each square subdivided into 2×2 rectangles adjacent to a central node at (x, x).

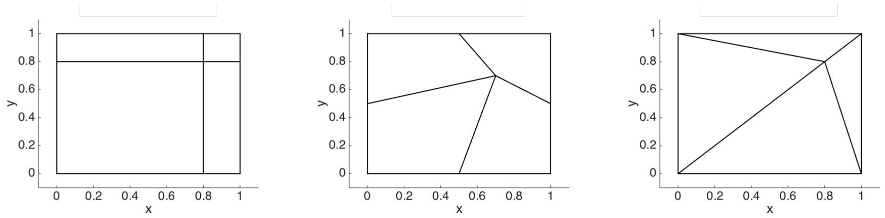
- *quadrilateral*$[x]$: 2D mesh constructed from unit squares, with each square subdivided into 2×2 smaller uniform squares, after which the central node at (0.5,0.5) is moved to (x, x).

- *triangular*$[x]$: 2D mesh constructed from unit squares, with each square subdivided into four triangles adjacent to the central node at (x, x).

- *cuboid*$[x]$: 3D mesh constructed from unit cubes, with each cube subdivided into $2 \times 2 \times 2$ cuboids adjacent to the central node at (x, x, x).

- *hexahedral*$[x]$: 3D mesh constructed from unit cubes, with each cube subdivided into $2 \times 2 \times 2$ smaller uniform cubes, after which the central node at $(0.5, 0.5, 0.5)$ is moved to (x, x, x).

- *tetrahedral*$[x]$: 3D mesh constructed from unit cubes, with each cube subdivided into six pyramids adjacent to the central node at (x, x, x), and with each pyramid subdivided into four tetrahedra.



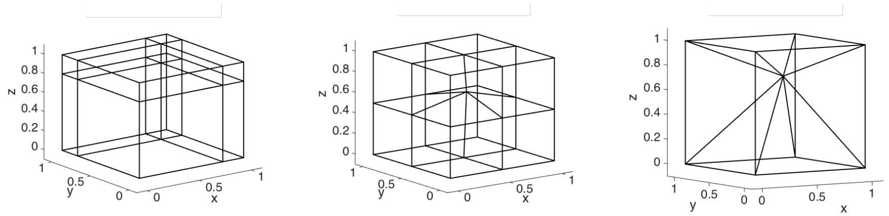
(a) A square divided into four rectangles with a central node at (0.8,0.8).

(b) A square divided into four quadrilaterals with a central node at (0.7,0.7).

(c) A square divided into four triangles with a central node at (0.7,0.7).

Figure 2.5

The results of the parameter estimates are given in Tables 2.4 and 2.5. In all cases, the penalty term estimate causes a reduction in the time step size of no more than a factor 1.1, with respect to the time step size using the minimal downscaled penalty term η^{**}_{min}. The time step estimates for triangular meshes causes a reduction of no more than 1.25 and for the tetrahedral meshes a reduction of no more than 1.2 with respect to the largest allowed time step size. For quadrilateral and hexahedral meshes the time step estimate tends to get less sharp for higher order polynomial



(a) A cube divided into eight cuboids with a central node at (0.8,0.8,0.8).

(b) A cube divided into eight hexahedra with a central node at (0.6,0.6,0.6).

(c) A cube divided into six pyramids with a central node at (0.7,0.7,0.7).

Figure 2.6

Mesh	x	p	$\frac{\Delta t(\eta^{**}_{min})}{\Delta t(\eta^{**})}$	$\frac{\Delta t(\eta^{**})}{\Delta t_{est}(\eta^{**})}$
triangular$[x]$	$\{0.5, 0.7, 0.9\}$	1	[1.04, 1.06]	[1.14, 1.20]
		2	[1.05, 1.09]	[1.18, 1.20]
		3	[1.05, 1.09]	[1.19, 1.21]
rectangular$[x]$	$\{0.5, 0.7, 0.9\}$	1	[1.00, 1.00]	[1.00, 1.11]
		2	[1.00, 1.00]	[1.00, 1.28]
		3	[1.00, 1.00]	[1.00, 1.37]
quadrilateral$[x]$	$\{0.5, 0.6, 0.7\}$	1	[1.00, 1.05]	[1.00, 1.20]
		2	[1.00, 1.04]	[1.00, 1.26]
		3	[1.00, 1.05]	[1.00, 1.31]

Table 2.4: Parameter estimates on deformed 2D meshes, using penalty term η^{**}. Intervals denote the range in which the time step ratios are found.

basis functions and more strongly deformed meshes, but in all of our tests $\Delta t(\eta^{**})/\Delta t_{est}(\eta^{**})$ remains below 1.4.

Since it is hard to compute the element and face integrals for general quadrilateral and hexahedral meshes exactly, we approximate them using the Gauss–Legendre quadrature rule with $p+1$ points in every direction, where p is the polynomial order of the basis functions. We do not consider meshes of the type *quadrilateral*$[x]$ with $x \geq 0.75$ and *hexahedral*$[x]$ with $x \geq 2/3$ since in those cases one of the elements no longer has a well-defined mapping.

Mesh	x	p	$\frac{\Delta t(\eta^{**}_{min})}{\Delta t(\eta^{**})}$	$\frac{\Delta t(\eta^{**})}{\Delta t_{est}(\eta^{**})}$
tetrahedral$[x]$	$\{0.5, 0.7, 0.9\}$	1	[1.06, 1.13]	[1.12, 1.14]
		2	[1.08, 1.17]	[1.14, 1.17]
		3	[1.07, 1.12]	[1.14, 1.15]
cuboid$[x]$	$\{0.5, 0.7, 0.9\}$	1	[1.00, 1.00]	[1.00, 1.11]
		2	[1.00, 1.00]	[1.00, 1.28]
		3	[1.00, 1.00]	[1.00, 1.37]
hexahedral$[x]$	$\{0.5, 0.6, 0.65\}$	1	[1.00, 1.09]	[1.00, 1.17]
		2	[1.00, 1.07]	[1.00, 1.25]
		3	[1.00, 1.10]	[1.00, 1.28]

Table 2.5: Parameter estimates on deformed 3D meshes, using penalty term η^{**}. Intervals denote the range in which the time step ratios are found.

Meshes with piecewise linear parameters

We now consider the acoustic wave equation with piecewise linear parameter fields ρ and c instead of constant parameters. An overview of the different meshes is listed below.

- *squarePL*$[\rho_0, c_0]$: 2D mesh constructed from unit squares, with each square subdivided into 2×2 smaller squares and with piecewise linear parameters ρ, c such that $\rho = c = 1$ at $y = 0$ and $y = 1$ and $\rho = \rho_0, c = c_0$ at $y = 0.5$.

- *triPL*$[\rho_0, c_0]$: 2D mesh constructed from unit squares, with each unit square subdivided into four uniform triangles and with piecewise linear parameters ρ, c such that $\rho = c = 1$ at the boundary and $\rho = \rho_0, c = c_0$ at the center of each square.

- *cubicPL*$[\rho_0, c_0]$: 3D mesh constructed from unit cubes, with each cube subdivided into $2 \times 2 \times 2$ smaller cubes and with piecewise linear parameters ρ, c such that $\rho = c = 1$ at $z = 0$ and $z = 1$ and $\rho = \rho_0, c = c_0$ at $z = 0.5$.

- *tetraPL*$[\rho_0, c_0]$: 3D mesh constructed from unit cubes, with each cube subdivided into 24 uniform tetrahedra and with piecewise linear parameters ρ, c such that $\rho = c = 1$ at the boundary and $\rho = \rho_0, c = c_0$ at the center of each cube.

Mesh	(ρ_0, c_0)	p	$\frac{\Delta t(\eta^{**}_{min})}{\Delta t(\eta^{**})}$	$\frac{\Delta t(\eta^{**})}{\Delta t_{est}(\eta^{**})}$
triPL$[\rho_0, c_0]$	$\{(1,1),(5,5),(10,1),...$ $(1,10),(10,10)\}$	1	[1.01,1.09]	[1.04,1.17]
		2	[1.01,1.11]	[1.06,1.19]
		3	[1,01,1.09]	[1.08,1.20]
squarePL$[\rho_0, c_0]$	$\{(1,1),(5,5),(10,1),...$ $(1,10),(10,10)\}$	1	[1.00,1.00]	[1.00,1.00]
		2	[1.00,1.00]	[1.00,1.00]
		3	[1.00,1.00]	[1.00,1.04]
tetraPL$[\rho_0, c_0]$	$\{(1,1),(5,5),(10,1),...$ $(1,10),(10,10)\}$	1	[1.12,1.19]	[1.13,1.14]
		2	[1.12,1.19]	[1.13,1.15]
		3	[1.10,1.12]	[1.14,1.15]
cubicPL$[\rho_0, c_0]$	$\{(1,1),(5,5),(10,1),...$ $(1,10),(10,10)\}$	1	[1.00,1.00]	[1.00,1.00]
		2	[1.00,1.00]	[1.00,1.00]
		3	[1.00,1.00]	[1.00,1.03]

Table 2.6: Parameter estimates on 2D and 3D meshes for the acoustic wave equation with piecewise linear parameters ρ and c, using penalty term η^{**}. Intervals denote the range in which the time step ratios are found.

The results of the parameter estimates are given in Table 2.6. The penalty term estimate cause a reduction in the time step size of no more than a factor 1.2, with respect to the time step size using the minimal downscaled penalty term η^{**}_{min} for triangular and tetrahedral elements and no reduction at all for square and cubic elements. The time step estimates for the triangular and tetrahedral meshes causes a reduction in the time step size of no more than 1.2 with respect to the largest allowed time step size, while for the square and cubic meshes they often cause no reduction at all or a reduction not larger than 1.05.

Meshes for electromagnetic and elastic wave problems

In this subsection we consider the electromagnetic wave equations as given in Example 2.3.2 and the 3D isotropic elastic wave equations as given in Example 2.3.3. We use the electromagnetic wave equations to test heterogeneous media with sharp contrasts in material parameters, while we use the elastic wave equations to test media with large contrasts in primary and secondary wave velocities. An overview of the different meshes is listed below.

- *tetraEM*$[\mu_1, \mu_2]$: 3D mesh constructed from unit cubes, with each cube subdivided into 24 smaller uniform tetrahedra, where all tetrahedra below the surface $x = z$ have a relative magnetic permeability of μ_1 and all tetrahedra above this surface have a permeability of μ_2. For all tetrahedra the relative electric permittivity ϵ equals 1.

- *cubicEM*$[\mu_1, \mu_2]$: 3D mesh constructed from unit cubes, with each cube subdivided into $2 \times 2 \times 2$ smaller uniform cubes, where the bottom four cubes have a relative magnetic permeability of μ_1 and the top four cubes a permeability of μ_2. For all cubes the relative electric permittivity ϵ equals 1.
- *tetraISO*$[\lambda, \mu]$: 3D mesh constructed from unit cubes, with each cube subdivided into 24 smaller uniform tetrahedra, and with all tetrahedra having Lamé parameters λ and μ and mass density $\rho = 1$.
- *cubicISO*$[\lambda, \mu]$: 3D mesh constructed from unit cubes, with each cube having Lamé parameters λ and μ and mass density $\rho = 1$.

Mesh	μ_1^{-1}	μ_2^{-1}	p	$\frac{\Delta t(\eta^{**}_{min})}{\Delta t(\eta^{**})}$	$\frac{\Delta t(\eta^{**})}{\Delta t_{est}(\eta^{**})}$
tetraEM$[\mu_1, \mu_2]$	1	$\{1, 10, 100\}$	1	[1.04, 1.05]	[1.14, 1.15]
			2	[1.04, 1.07]	[1.15, 1.15]
			3	[1.00, 1.04]	[1.15, 1.16]
cubicEM$[\mu_1, \mu_2]$	1	$\{1, 10, 100\}$	1	[1.00, 1.07]	[1.29, 1.29]
			2	[1.00, 1.00]	[1.31, 1.35]
			3	[1.00, 1.01]	[1.33, 1.35]

Table 2.7: Parameter estimates on a 3D electromagnetic mesh, using penalty term η^{**}. Intervals denote the range in which the time step ratios are found.

The results of the parameter estimates are given in Tables 2.7 and 2.8. For the heterogeneous electromagnetic problems, the penalty term estimate causes a reduction in the time step size of no more than a factor 1.1, with respect to the time step size using the minimal downscaled penalty term η^{**}_{min}. For the isotropic elastic meshes this reduction is no more than a factor 1.2. The time step estimates for all the tetrahedral meshes causes a reduction in the time step size of no more than 1.25 with respect to the largest allowed time step size, while for the cubic meshes this reduction remains below a factor 1.4.

Mesh	(λ, μ)	p	$\frac{\Delta t(\eta^{**}_{min})}{\Delta t(\eta^{**})}$	$\frac{\Delta t(\eta^{**})}{\Delta t_{est}(\eta^{**})}$
tetraISO$[\lambda, \mu]$	$\{(1,0), (0,1), \ldots$	1	[1.00, 1.11]	[1.14, 1.15]
	$(10,1), (100,1)\}$	2	[1.00, 1.14]	[1.14, 1.15]
		3	[1.00, 1.13]	[1.13, 1.15]
cubicSO$[\lambda, \mu]$	$\{(1,0), (0,1), \ldots$	1	[1.05, 1.20]	[1.24, 1.28]
	$(10,1), (100,1)\}$	2	[1.00, 1.10]	[1.23, 1.31]
		3	[1.00, 1.07]	[1.24, 1.33]

Table 2.8: Parameter estimates on a 3D isotropic elastic tetrahedral mesh, using penalty term η^{**}. Intervals denote the range in which the time step ratios are found.

2.8 Conclusion

We have derived sharp and sufficient conditions for the penalty parameter in Theorem 2.5.6 and Theorem 2.5.8 to guarantee stability of the SIPDG method for linear wave problems. In addition, we derived sufficient upper bounds for the spectral radius and the time step size in Theorem 2.6.3 and Section 2.6.3, by introducing a weighted mesh decomposition. These conditions hold for generic meshes, including unstructured nonconforming heterogeneous meshes of mixed element types, with different types of boundary conditions. Moreover, the estimates hold for general linear hyperbolic partial differential equations, including the acoustic wave equation, the (an)isotropic elastic wave equations, and Maxwell's equations. Both the penalty term and time step size can be efficiently computed in parallel using a power iteration method on each element and submesh, respectively.

To test the sharpness of our estimates we have considered several semi-uniform meshes made of N^d uniform cubes or squares, which are then uniformly subdivided into smaller meshes, including meshes with deformed elements and heterogeneous media. From the results we can see that the penalty term estimate given in Theorem 2.5.8 is significantly sharper than the estimate in Theorem 2.5.6 for 2D and 3D meshes. Especially for cubic elements, we see that downscaling the penalty estimate constructed using Theorem 2.5.6 allows a time step size more than twice as large, while the penalty term in Theorem 2.5.8 does not allow any downscaling at all. Furthermore, the penalty term estimate from Theorem 2.5.8 allows a time step size more than a factor 1.5 times larger than when using the penalty term estimate derived in [55], when using square or cubic basis functions on tetrahedra or cubic basis functions on triangles. For regular square and cubic meshes, this same penalty term estimate, together with the time step estimate from Theorem 2.6.3 using the weighted mesh decomposition of (2.23), conforms with the largest allowed time step sizes analytically derived in [1].

In general we see that downscaling the penalty term from Theorem 2.5.8 does not increase the maximum allowed time step size by more than a factor 1.2, even when the material parameters are nonconstant within the elements, while our time step estimate, based on a vertex-based weighted mesh decomposition, does not become smaller than a factor 1.2 compared to the maximum allowed time step size for triangular and tetrahedral meshes, and not smaller than a factor 1.4 for quadrilateral and hexahedral meshes.

2.A Linear algebra

Lemma 2.A.1. *Let $C \in \mathbb{R}^{d\times m\times m\times d}_{sym}$ be a symmetric tensor such that*

$$\boldsymbol{\sigma}^t : C : \boldsymbol{\sigma} \geq 0, \qquad\qquad \boldsymbol{\sigma} \in \mathbb{R}^{d\times m}.$$

Then there exists a symmetric tensor field $C^{1/2} \in \mathbb{R}^{d\times m\times m\times d}_{sym}$ such that $C^{1/2} : C^{1/2} = C$.

Proof. The result follows from the fact that $\boldsymbol{\sigma} \to C : \boldsymbol{\sigma}$ is a linear, self-adjoint, and positive semidefinite operator on a finite-dimensional subspace $\mathbb{R}^{d\times m}$. □

Lemma 2.A.2. *Let $\hat{\mathbf{n}} \in \mathbb{R}^d$ be a vector and $C \in \mathbb{R}^{d\times m\times m\times d}_{sym}$ be a symmetric tensor such that*

$$\boldsymbol{\sigma}^t : C : \boldsymbol{\sigma} \geq 0, \qquad\qquad \boldsymbol{\sigma} \in \mathbb{R}^{d\times m}. \tag{2.28}$$

Also, define the second order tensor $\mathbf{c}_{\hat{\mathbf{n}}} := \hat{\mathbf{n}}\cdot C\cdot\hat{\mathbf{n}}$ and the following function space:

$$U := \left\{\mathbf{u} \in \mathbb{R}^m \;\middle|\; \mathbf{u} = \hat{\mathbf{n}} \cdot C : \boldsymbol{\sigma},\ \textit{for some } \boldsymbol{\sigma} \in \mathbb{R}^{d\times m}\right\}.$$

There exists a pseudoinverse $\mathbf{c}_{\hat{\mathbf{n}}}^{-1} \in \mathbb{R}^{m\times m}$ such that $\mathbf{c}_{\hat{\mathbf{n}}}^{-1}\cdot\mathbf{c}_{\hat{\mathbf{n}}}\cdot\mathbf{u} = \mathbf{c}_{\hat{\mathbf{n}}}\cdot\mathbf{c}_{\hat{\mathbf{n}}}^{-1}\cdot\mathbf{u} = \mathbf{u}$, for all $\mathbf{u} \in U$. Moreover, both $\mathbf{c}_{\hat{\mathbf{n}}}$ and $\mathbf{c}_{\hat{\mathbf{n}}}^{-1}$ are symmetric and positive semidefinite.

Proof. The fact that $\mathbf{c}_{\hat{\mathbf{n}}}$ is symmetric follows from its definition and the fact that C is symmetric. The fact that this tensor is also positive semidefinite follows from (2.28) and by writing

$$\mathbf{u} \cdot \mathbf{c}_{\hat{\mathbf{n}}} \cdot \mathbf{u} = \mathbf{u}\hat{\mathbf{n}} : C : \hat{\mathbf{n}}\mathbf{u} \geq 0, \qquad \mathbf{u} \in \mathbb{R}^m.$$

To prove that the pseudoinverse with respect to U exists and is symmetric positive semidefinite, it is sufficient to show that

$$\mathbf{u} \cdot \mathbf{c}_{\hat{\mathbf{n}}} \cdot \mathbf{u} > 0, \qquad \mathbf{u} \in U, \mathbf{u} \neq \mathbf{0}. \tag{2.29}$$

Since $\mathbf{c}_{\hat{\mathbf{n}}}$ is positive semidefinite we then only need to show that

$$\mathbf{u} \in U \ \wedge \ \mathbf{u} \cdot \mathbf{c}_{\hat{\mathbf{n}}} \cdot \mathbf{u} = 0 \quad \Longrightarrow \quad \mathbf{u} = \mathbf{0}. \tag{2.30}$$

Now take an arbitrary $\mathbf{u} \in U$, and suppose that $\mathbf{u} \cdot \mathbf{c}_{\hat{\mathbf{n}}} \cdot \mathbf{u} = 0$. Using Lemma 2.A.1 we can write

$$0 = \mathbf{u} \cdot \mathbf{c}_{\hat{\mathbf{n}}} \cdot \mathbf{u} = \|C^{1/2} : \hat{\mathbf{n}}\mathbf{u}\|^2.$$

From this it follows that $C : \hat{\mathbf{n}}\mathbf{u} = \mathbf{0}$. Because of the definition of U we can write $\mathbf{u} = \hat{\mathbf{n}} \cdot C : \boldsymbol{\sigma}$ for some $\boldsymbol{\sigma} \in \mathbb{R}^{d\times m}$, and therefore $C : \hat{\mathbf{n}}(\hat{\mathbf{n}} \cdot C : \boldsymbol{\sigma}) = \mathbf{0}$. Applying the double dot product with $\boldsymbol{\sigma}^t$ on the left side gives

$$0 = \boldsymbol{\sigma}^t : C : \hat{\mathbf{n}}(\hat{\mathbf{n}} \cdot C : \boldsymbol{\sigma}) = (\boldsymbol{\sigma}^t : C \cdot \hat{\mathbf{n}}) \cdot (\hat{\mathbf{n}} \cdot C : \boldsymbol{\sigma}) = \|\mathbf{u}\|^2.$$

Therefore, $\mathbf{u} = \mathbf{0}$, which proves (2.30). □

Lemma 2.A.3. *Let $\{a_i\}_{i=1}^n$ be a set of nonnegative scalars, and let $\{b_i\}_{i=1}^n$ be a set of positive scalars. Then*

$$\frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i} \leq \max_{i=1,\ldots,n} \frac{a_i}{b_i}.$$

Proof.

$$\frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i} = \sum_{i=1}^n \frac{a_i}{b_i} \frac{b_i}{\sum_{j=1}^n b_j} \leq \max_{k=1,\ldots,n} \frac{a_k}{b_k} \sum_{i=1}^n \frac{b_i}{\sum_{j=1}^n b_j} = \max_{k=1,\ldots,n} \frac{a_k}{b_k}.$$

□

Lemma 2.A.4. *Let $A \in \mathbb{R}^{n\times n}_{sym}$ be a symmetric matrix and $M \in \mathbb{R}^{n\times n}_{sym}$ a positive definite matrix. Then there exists a diagonalization $M^{-1}A = VDV^{-1}$, such that D is a real diagonal matrix and V satisfies $V^tMV = I$, where I is the identity matrix.*

Proof. Since M is symmetric positive definite, there exists a symmetric positive definite matrix $M^{1/2}$ such that $M^{1/2}M^{1/2} = M$. Define $\tilde{A} := M^{-1/2}AM^{-1/2}$. Since A is symmetric, $\tilde{A}$ is symmetric as well and can be diagonalised as $\tilde{A} = \tilde{V}D\tilde{V}^{-1}$ with D a diagonal real matrix and $\tilde{V}$ satisfying $\tilde{V}^t\tilde{V} = I$. Now define $V := M^{-1/2}\tilde{V}$. Then $M^{-1}A = V^{-1}DV$ and $V^tMV = I$. □

Lemma 2.A.5. *Let $A \in \mathbb{R}^{n\times n}_{sym}$ be a symmetric matrix and $M \in \mathbb{R}^{n\times n}_{sym}$ a positive definite matrix. Then*

$$\sup_{\underline{\mathbf{u}}\in\mathbb{R}^n, \underline{\mathbf{u}}\neq\underline{\mathbf{0}}} \frac{|\underline{\mathbf{u}}^tA\underline{\mathbf{u}}|}{\underline{\mathbf{u}}^tM\underline{\mathbf{u}}} = \lambda_{max}\big(M^{-1}A\big),$$

where $\lambda_{max}\big(M^{-1}A\big)$ denotes the largest eigenvalue of $M^{-1}A$ in magnitude.

Proof. From Lemma 2.A.4 it follows that there exist a diagonal matrix D and a nonsingular matrix V such that $MVDV^{-1} = A$ and $V^tMV = I$. Now consider an arbitrary $\underline{\mathbf{u}} \in \mathbb{R}^n, \underline{\mathbf{u}} \neq \underline{\mathbf{0}}$. We set $\underline{\mathbf{w}} := V^{-1}\underline{\mathbf{u}}$ to obtain

$$\frac{|\underline{\mathbf{u}}^tA\underline{\mathbf{u}}|}{\underline{\mathbf{u}}^tM\underline{\mathbf{u}}} = \frac{|\underline{\mathbf{w}}^tV^tMVDV^{-1}V\underline{\mathbf{w}}|}{\underline{\mathbf{w}}^tV^tMV\underline{\mathbf{w}}} = \frac{|\underline{\mathbf{w}}^tD\underline{\mathbf{w}}|}{\underline{\mathbf{w}}^t\underline{\mathbf{w}}}.$$

The lemma follows from this equality and the following relation:

$$\sup_{\underline{\mathbf{w}}\in\mathbb{R}^n, \underline{\mathbf{w}}\neq\underline{\mathbf{0}}} \frac{|\underline{\mathbf{w}}^tD\underline{\mathbf{w}}|}{\underline{\mathbf{w}}^t\underline{\mathbf{w}}} = \max_{i=1,\dots,N} |D_{ii}| = \lambda_{max}\big(M^{-1}A\big).$$

□

Chapter 3

A Note on the Stability of an Explicit Local Time-Stepping Method

Abstract

We analyse the stability of a basic local time-stepping method derived in [Diaz, J., & Grote, M. J. (2009). Energy conserving explicit local time stepping for second-order wave equations. *SIAM Journal on Scientific Computing*, 31(3), 1985-2014] for solving linear hyperbolic problems. We consider the simplest case, namely the second-order leap-frog scheme with two local time steps at a certain subdomain. We proof that there is always a case for which this scheme is unstable, except when local time-stepping is applied to the entire domain, which would make the scheme ineffective. This result also explains the instabilities that were already observed in [Diaz, J., & Grote, M. J. (2009)].

3.1 Introduction

Wave propagation problems in industrial applications are often solved using fully explicit numerical schemes, since they do not require solving a large sparse system at every time step. Such schemes include finite element schemes like the mass-lumped finite element scheme and discontinuous Galerkin scheme for the spatial discretization, combined with an explicit time integration scheme like the leap-frog scheme, higher order Lax–Wendroff schemes, or explicit Runge–Kutta schemes. For these schemes to be stable, a sufficiently small time step size is required, of which the upper bound is dictated by the smallest mesh resolution with respect to the wave velocity. Ideally, the element size is scaled optimally with respect to the wavelength. This, however, is not always possible near complex material interfaces and boundary layers due to limitations of the mesh generator. At such surfaces, overly small elements can appear, which result in an overly severe time step restriction globally, even though only a small part of the grid or mesh is locally refined.

To overcome this problem, explicit local time-stepping methods have been developed, in which the time step size can be adapted locally [25, 22, 34, 8, 35, 33, 23]. However, there is no analytical proof yet for the stability of any of these schemes. One of the reasons for this might be the increased complexity of the resulting fully discrete schemes. In this chapter, we therefore provide a stability analysis for one of the simplest and most commonly used schemes, namely the second-order leap-frog scheme. This scheme has been extended to a local time-stepping scheme in [22]. We consider the case in which the time step size is refined by a factor 2 at a region of the grid or mesh where the resolution has been refined locally up to a factor 2. We proof that there are always cases for which the local time-stepping scheme becomes unstable. These instabilities were already observed but not explained in [22]. Our theory shows that, even when extending the region in which local time-stepping is applied, this will still not result in strict numerical stability, unless this region contains the entire mesh, which would turn the scheme back into a standard single time-stepping scheme.

3.2 A local time-stepping scheme

Let Ω be an open domain and $\rho, c : \Omega \to \mathbb{R}^+$ positive scalar fields. We consider the following model wave problem

$$\begin{aligned} \rho\partial_t^2 u &= \nabla \cdot c\nabla u, && \text{in } \Omega \times (0,T), \\ u &= 0, && \text{on } \partial\Omega \times (0,T), \end{aligned}$$

with initial conditions for u and $\partial_t u$ at $t = 0$, where $u : \Omega \times (0,T) \to \mathbb{R}$ is the wave field and ∇ is the gradient operator. Solving this problem using a finite difference or finite element scheme typically results in the following ODE system:

$$M\partial_t^2 u + Au = 0, \qquad \text{in } (0,T),$$

where M is often referred to as the mass matrix and A as the stiffness matrix. This system of equations can be solved using a leap-frog scheme:

$$M\frac{u^{n+1} - 2u^n + u^{n-1}}{\Delta t^2} + Au^n = 0, \qquad n = 1, 2, ..,$$

given u^0 and u^1, where Δt is the time step size and u^n is the approximation of $u(n\Delta t)$. To obtain u^{n+1}, given u^n and u^{n-1}, we need to solve a system of equations of the form $Mx = b$. This becomes trivial when M is diagonal, which is the case for finite difference schemes, mass-lumped finite element schemes, and Discontinuous Galerkin schemes using orthogonal basis functions. Furthermore, when using orthonormal basis functions, the resulting mass matrix becomes the identity matrix, and the leap-frog scheme will have the following form:

$$\frac{u^{n+1} - 2u^n + u^{n-1}}{\Delta t^2} + Au^n = 0, \qquad n = 1, 2, .. . \tag{3.1}$$

Now consider the same leap-frog scheme using a time step twice as small. We can write this scheme as

$$\frac{u^{r+1/2} - 2u^r + u^{r-1/2}}{(\Delta t/2)^2} + Au^r = 0, \qquad r = \frac{1}{2}, 1, \frac{3}{2}, .. . \tag{3.2}$$

Summing (3.2) over $r = n - 1/2$, $r = n + 1/2$, and two times $r = n$, and then substituting $u^{n-1/2} + u^{n+1/2}$ by $2u^n - (\Delta t/2)^2 Au^n$ results in

$$\frac{u^{n+1} - 2u^n + u^{n-1}}{\Delta t^2} + \tilde{A}u^n = 0, \qquad n = 1, 2, .. , \tag{3.3}$$

with $\tilde{A} := A - (1/16)\Delta t^2 A^2$. This scheme has the same form as the original leap-frog scheme, given in (3.1), and is equivalent to the original leap-frog scheme using a time step twice as small.

The second-order local time-stepping method of [22], using two local time steps, can be interpreted as something in between scheme (3.1) and (3.3), and can be written as (3.3) with $\tilde{A} := A - (1/16)\Delta t^2 APA$, where P is a diagonal matrix with the diagonal entries corresponding to the local time-stepping region equal to one, and the other entries equal to zero. Note that if P is zero, meaning no local time-stepping is applied, we obtain the standard leap-frog scheme (3.1), while if P is the identity matrix, meaning we apply local time-stepping everywhere, this scheme is equivalent to (3.3), which in turn is equivalent to the standard leap-frog scheme using a time step twice as small.

3.3 Stability analysis

For the stability analysis we will assume that A is symmetric positive semidefinite, which is the case for most mass-lumped finite element methods and for symmetric discontinuous Galerkin methods like the symmetric interior penalty discontinuous Galerkin method [36].

It is well-known that the leap-frog scheme, given in (3.1), is stable iff $\sigma(\Delta t^2 A) \subset [0,4]$, where $\sigma(A)$ denotes the set of eigenvalues of A. Since A is symmetric positive semidefinite, stability is ensured when

$$\Delta t^2 \sigma_{max}(A) \leq 4,$$

or equivalently, when $\Delta t \leq 2/\sqrt{\sigma_{max}(A)}$, where $\sigma_{max}(A)$ denotes the spectral radius of A.

For the local time-stepping scheme, we assume that the spectral radius approximately satisfies $\sigma_{max}(A) \approx 4\sigma_{max}(A_{coarse})$, where $A_{coarse} := (I - P)A(I-P)$ is the stiffness matrix restricted to the coarse part of the mesh and I is the identity matrix. This means that the fine part of the mesh requires a time step about twice as small as the coarse part, which is the case when for example the mesh is locally refined by a factor of 2. We therefore want to use a time step size Δt satisfying

$$\Delta t^2 \sigma_{max}(A) = 16 - \epsilon, \tag{3.4a}$$

$$\Delta t^2 \sigma_{max}(A_{coarse}) = 4 - \epsilon', \tag{3.4b}$$

where $\epsilon, \epsilon' > 0$ are small positive constants. Property (3.4a) corresponds to the leap-frog stability condition when using the local time step size $\Delta t/2$, while property (3.4b) corresponds to the leap-frog stability condition when using the global time step size Δt considering only the coarse part of the mesh. For an efficient scheme, ϵ and ϵ' should be as small as possible.

We now present the main result of the stability analysis.

Theorem 3.3.1. *Consider the local time-stepping scheme given in (3.3) with $\tilde{A} = A - (1/16)\Delta t^2 APA$ and with Δt satisfying (3.4). If*

(i) $(8, u^*)$ is an eigenpair of $\Delta t^2 A$,

(ii) $(I - P)u^* \neq 0$,

then the local time-stepping scheme is unstable.

Proof. Since the local time-stepping scheme has the same form as the leap-frog scheme, but uses $\tilde{A}$ instead of A, it is stable iff $\sigma(\Delta t^2 \tilde{A}) \subset [0,4]$. Since $\tilde{A}$ is symmetric, this implies that the scheme is stable iff

$$\Delta t^2 \frac{u^t \tilde{A} u}{\|u\|^2} \in [0,4], \qquad u \in \mathbb{R}^n,$$

where n is the number of degrees of freedom. Now set $u = u^*$ and suppose $(I-P)u^* \neq 0$. Then

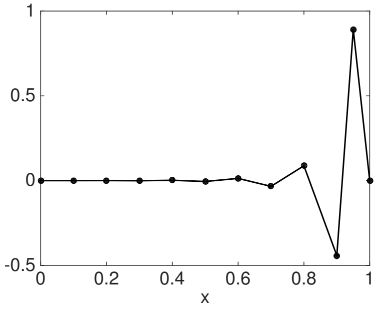
$$\Delta t^2 \frac{u^t \tilde{A} u}{\|u\|^2} = 8 - 4\frac{u^t P u}{\|u\|^2} = 4 + 4\frac{u^t (I-P) u}{\|u\|^2} = 4 + 4\frac{\|(I-P)u\|^2}{\|u\|^2} > 4,$$

which implies that the scheme is unstable. □

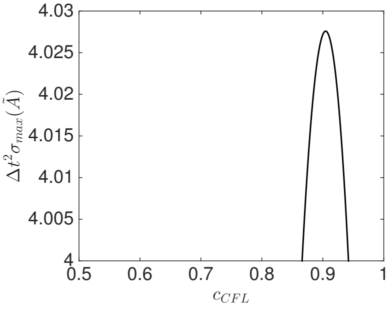
This theorem states that if there exists an eigenpair of $\Delta t^2 A$ of the form $(8, u^*)$ and if u^* is non-zero at the coarse part of the mesh, then the local time-stepping scheme is unstable.

Although there exist cases in which the first condition is not met, this cannot be guaranteed in general. The reason is that usually, the eigenvalues of A can take any value between 0 and $\sigma_{max}(A)$, and therefore, the eigenvalues of $\Delta t^2 A$ can take any value between 0 and $16 - \epsilon > 8$.

The second condition, which is u^* being non-zero at the part of the mesh where no local time-stepping is applied, is satisfied in most cases, since eigenmodes are usually not strictly zero on an entire subdomain.



(a) Unstable mode of the local time-stepping scheme with $N = 10$, $m = 0$, and $\Delta t = 0.9\Delta x_{coarse}$.



(b) Instabilities of the local time-stepping scheme with $N = 10$, $m = 0$, and $\Delta t = c_{CFL}\Delta x_{coarse}$, for different CFL numbers c_{CFL}.

To demonstrate the instability of the local time-stepping scheme, we consider a simple example in 1D with $\Omega := [0,1]$ and $\rho, c = 1$ and use the linear mass-lumped finite element method for the spatial discretization. Since suboptimal scaled elements usually only appear near sharp material interfaces and complex boundary layers, we consider a mesh divided uniformly into N elements of size $\Delta x_{coarse} = 1/N$, and split the last element into two smaller elements of size $\Delta x_{fine} = 1/(2N)$. We can bound the eigenvalues of A by the eigenvalues of the element matrices [40] and

	$\Delta t^2 \sigma_{max}(\tilde{A})$		
m	$N = 10$	$N = 20$	$N = 100$
0	$4 + 2.72 \times 10^{-2}$	$4 + 2.72 \times 10^{-2}$	$4 + 2.72 \times 10^{-2}$
1	$4 + 4.58 \times 10^{-4}$	$4 + 4.58 \times 10^{-4}$	$4 + 4.58 \times 10^{-4}$
2	$4 + 6.64 \times 10^{-6}$	$4 + 6.64 \times 10^{-6}$	$4 + 6.64 \times 10^{-6}$

Table 3.1: Instabilities of the local time-stepping scheme with $\Delta t = 0.9\Delta x_{coarse}$ for different mesh sizes N and number of added LTS nodes m.

	$N = 10$		$N = 100$	
m	unstable c_{CFL} region	length	unstable c_{CFL} region	length
0	$(0.8661, 0.9421)$	0.0760	$(0.8661, 0.9421)$	0.0760
1	$(0.8955, 0.9051)$	0.0096	$(0.8955, 0.9051)$	0.0096
2	$(0.8997, 0.9008)$	0.0011	$(0.8997, 0.9008)$	0.0011

Table 3.2: Unstable CFL numbers c_{CFL} for the local time-stepping scheme with $\Delta t = c_{CFL}\Delta x_{coarse}$ and $N = 10$ and with LTS applied to m additional nodes.

obtain $\sigma_{max}(A) \leq 4/(\Delta x_{fine})^2$ and $\sigma_{max}(A_{coarse}) \leq 4/(\Delta x_{coarse})^2$. This implies (3.4) is satisfied if $\Delta t \leq \Delta x_{coarse}$, which is the well-known CFL condition. We apply local time-stepping at the nodes on the refined region $[1 - \Delta x_{coarse}, 1]$ plus the m neighbouring nodes and use a global time step size $\Delta t = 0.9\Delta x_{coarse}$.

An illustration of an unstable mode is given in Figure (3.1a) and the instability for different CLF numbers is shown in Figure (3.1b). The size of the instability and the regions of unstable CLF numbers are shown in Table 3.1 and 3.2 for different values of N and m .

These results confirm that the local time-stepping scheme can be unstable, even if the local time-stepping region is larger than the region where the mesh is refined. Stability can only be guaranteed when local time-stepping is applied to the entire mesh, but this would turn it into a standard time-stepping scheme, hence making it ineffective.

The results also show that the instabilities hardly change when increasing the mesh size N, although the possible instabilities decrease exponentially when expanding the region on which LTS is applied. This behaviour seems related to the fact that the higher frequency modes induced by local mesh refinement, like the mode shown in Figure 3.1a, decrease exponentially when moving away from the refined region. When the instability becomes sufficiently small, it will have no significant effect on the accuracy. Therefore, while it is not possible to guarantee strict stability, it might still be possible to obtain bounds on the instabilities and guarantee that possible instabilities will have no significant effect when using local time-stepping.

Chapter 4

Dispersion Properties of Explicit Finite Element Methods for Wave Propagation Modelling on Tetrahedral Meshes[1]

Abstract

We analyse the dispersion properties of two types of explicit finite element methods for modelling acoustic and elastic wave propagation on tetrahedral meshes, namely mass-lumped finite element methods and symmetric interior penalty discontinuous Galerkin methods, both combined with a suitable Lax–Wendroff time integration scheme. The dispersion properties are obtained semi-analytically using standard Fourier analysis. Based on the dispersion analysis, we give an indication of which method is the most efficient for a given accuracy, how many elements per wavelength are required for a given accuracy, and how sensitive the accuracy of the method is to poorly shaped elements.

4.1 Introduction

Realistic wave propagation problems often involve large three-dimensional domains consisting of heterogeneous materials with complex geometries and sharp interfaces. Solving such problems requires a numerical method that is efficient in terms of computation time and is flexible enough to capture the effect of a complex geometry.

[1] Published as: Geevers, S., Mulder, W.A. & van der Vegt, J.J.W. (2018). Dispersion Properties of Explicit Finite Element Methods for Wave Propagation Modelling on Tetrahedral Meshes. *Journal of Scientific Computing.* https://doi.org/10.1007/s10915-018-0709-7

Standard finite difference methods fall short, since they rely on Cartesian grids that cannot efficiently capture the effect of complex interfaces and boundary layers. Finite element methods can overcome this problem when the elements are aligned with those surfaces. However, the accuracy of the finite element method quickly deteriorates when the elements are poorly shaped or are poorly aligned with the geometry. Obtaining a high quality mesh is therefore quintessential. While both hexahedral and tetrahedral elements are commonly used for three-dimensional problems, tetrahedral elements have a big advantage in this respect, since they offer more geometric flexibility and since robust tetrahedral mesh generators based on the Delaunay criterion are available [58, 67].

Apart from the construction of a high-quality mesh, finite element methods for wave propagation problems also require a (block)-diagonal mass matrix to enable explicit time-stepping. A diagonal mass matrix can be obtained with nodal basis functions and a quadrature rule, if the quadrature points coincide with the basis function nodes. This technique is known as mass-lumping. For quadrilaterals and hexahedra, mass-lumping is achieved by combining tensor-product basis functions with a Gauss-Lobatto quadrature rule, resulting in a scheme known as the spectral element method [59, 63, 43]. For triangles and tetrahedra, an efficient linear mass-lumped scheme is obtained by combining standard Lagrangian basis functions with a Newton–Cotes quadrature rule. For higher-degree triangles and tetrahedra, however, this approach results in an unstable, unsolvable, or inaccurate scheme. To remain accurate and stable, the space of the triangle or tetrahedron is enriched with higher-degree bubble functions. This approach has led to accurate mass-lumped triangles of degree 2 [29], 3 [16], 4 [51], 5 [11], 6 [53], 7-9 [49, 17] and tetrahedra of degree 2 [51] and 3 [11].

Another way to obtain a (block)-diagonal mass matrix is by using discontinuous basis functions. The resulting schemes are known as Discontinuous Galerkin (DG) methods. The first DG methods for wave propagation problems were based on a first-order formulation of the wave equation [60, 13]. In [62] and [36], DG methods were introduced that were based on the original second-order formulation of the wave problem. The advantage of finite element methods based on the second-order formulation is that they do not need to compute or store the auxiliary variables that appear in the first-order formulation. Moreover, they can be combined with a leap-frog or higher-order Lax–Wendroff time integration scheme that only requires K stages for a $2K$-order accuracy. We focus on the symmetric interior penalty discontinuous Galerkin (SIPDG) method, presented and analysed in [36], which is based on the second-order formulation of the wave problem and which also remains energy-conservative on the discrete level. To remain accurate and stable, face integrals and interior penalty parameters are added to the discrete operator. We consider two choices for the penalty parameter: the penalty term derived in [55], based on the trace inequality of [79], and a recently developed sharper estimate [32], based on a more involved trace inequality.

To effectively apply these methods, it is crucial to know the required mesh resolution for a given accuracy. It is also useful to know which method is the most efficient for a given accuracy and how the mesh quality and material parameters, such as the P/S-wave velocity ratio for elastic waves, affect the accuracy. A practical and common measure for the accuracy of these type of methods is the amount of numerical dispersion and dissipation. In this chapter, we will focus mainly on the numerical dispersion, since the methods we consider are all energy-conservative and therefore do not suffer from numerical dissipation. We do, however, also investigate the spurious modes that appear when projecting a physical wave onto the discrete space.

The dispersion properties of DG methods based on the first-order formulation of the wave problem have already been analysed for Cartesian meshes [39, 2], triangles [39, 47], and tetrahedra [41]. For the SIPDG method, these properties have already been analysed for Cartesian meshes in [3, 21] and triangles in [6] and for the mass-lumped finite element method this has already been analysed for quadrilaterals and hexahedra in [52, 14, 19] and for triangles in [48]. However, a dispersion analysis of the mass-lumped finite element and SIPDG methods for tetrahedra is, to the best of our knowledge, still missing, even though most realistic wave problems involve three-dimensional domains for which tetrahedral elements are particularly suitable. In this chapter, we therefore present an extensive dispersion analysis of these methods for tetrahedra. This analysis is based on standard Fourier analysis. We use the analysis to obtain estimates for the required number of elements per wavelength and estimate the computational cost to obtain an indication of which method is the most efficient for a given accuracy. We consider both acoustic and elastic waves and also look at the effect of poorly shaped elements and high P/S-wave velocity ratios on the accuracy of the methods.

This chapter is organised as follows: in Section 4.2, we introduce the tensor notation used in this chapter. The acoustic and elastic wave equations are presented in Section 4.3 and the mass-lumped and discontinuous Galerkin finite element methods are presented in Section 4.4. In Section 4.5, we explain how we analyse the dispersion properties of these methods. The results of this analysis are presented in Section 4.6 and the main conclusions are summarised in Section 4.7.

4.2 Some tensor notation

Before we present the acoustic and elastic wave equations, we explain the tensor notation that we use throughout this chapter. We let the dot product of two tensors denote the summation over the last index of the left and first index of the right tensor. For the double dot product we also sum over the last-but-one index of the left and second index of the right tensor. A concatenation of two tensors denotes the standard tensor product. To give some examples, let $\hat{\mathbf{n}} \in \mathbb{R}^d, \mathbf{u} \in \mathbb{R}^m$ be two vectors, $\boldsymbol{\sigma} \in \mathbb{R}^{d\times m}$ a second-order tensor, and $C \in \mathbb{R}^{d\times m\times m\times d}$ a fourth-order tensor. Then

$$[\hat{\mathbf{n}}\mathbf{u}]_{ij} := \hat{n}_i u_j, \qquad [\boldsymbol{\sigma}\cdot\mathbf{u}]_i := \sum_{l=1}^{m} \sigma_{il} u_l,$$

$$[C:\boldsymbol{\sigma}]_{ij} := \sum_{k=1}^{d}\sum_{l=1}^{m} C_{ijlk}\sigma_{kl}, \qquad [\hat{\mathbf{n}}\cdot C]_{qji} := \sum_{k=1}^{d} \hat{n}_k C_{kqji},$$

for all $i = 1,\ldots,d$ and $j,q = 1,\ldots,m$.

In the next section we will use this tensor notation to present the acoustic and isotropic elastic wave equations.

4.3 The acoustic and isotropic elastic wave equations

Let $\Omega \subset \mathbb{R}^3$ be a three-dimensional open domain with a Lipschitz boundary $\partial\Omega$, and let $(0,T)$ be the time domain. Also, let $\{\Gamma_d, \Gamma_n\}$ be a partition of $\partial\Omega$, corresponding to Dirichlet and von Neumann boundary conditions, respectively. We define the following linear hyperbolic problem4:

$$\begin{aligned}
\rho\partial_t^2\mathbf{u} &= \nabla\cdot C:\nabla\mathbf{u} + \mathbf{f} && \text{in } \Omega\times(0,T), && \text{(4.1a)}\\
C:\hat{\mathbf{n}}\mathbf{u} &= \mathbf{0} && \text{on } \Gamma_d\times(0,T), && \text{(4.1b)}\\
\hat{\mathbf{n}}\cdot C:\nabla\mathbf{u} &= \mathbf{0} && \text{on } \Gamma_n\times(0,T), && \text{(4.1c)}\\
\mathbf{u}|_{t=0} &= \mathbf{u}_0 && \text{in } \Omega, && \text{(4.1d)}\\
\partial_t\mathbf{u}|_{t=0} &= \mathbf{v}_0 && \text{in } \Omega, && \text{(4.1e)}
\end{aligned}$$

where $\mathbf{u} : \Omega\times(0,T) \to \mathbb{R}^m$ is a vector of m variables that are to be solved, ∇ is the gradient operator, $\rho : \Omega \to \mathbb{R}^+$ is a positive scalar field,

$C : \Omega \to \mathbb{R}^{3\times m\times m\times 3}$ a fourth-order tensor field, $\mathbf{f} : \Omega \times (0, T) \to \mathbb{R}^m$ the source field, and $\hat{\mathbf{n}} : \partial\Omega \to \mathbb{R}^3$ the outward pointing normal unit vector.

By choosing the appropriate tensor and scalar field we can obtain the acoustic wave equation and the isotropic elastic wave equations.

Case 1 (Isotropic elastic wave equations)**.** *To obtain the isotropic elastic wave equations, set $m = 3$ and*

$$C_{ijqp} = \lambda\delta_{ij}\delta_{pq} + \mu(\delta_{ip}\delta_{jq} + \delta_{iq}\delta_{jp}),$$

for $i, j, p, q = 1, 2, 3$, where δ is the Kronecker delta. Equation (4.1a) then becomes

$$\rho\partial_t^2\mathbf{u} = \nabla\lambda(\nabla \cdot \mathbf{u}) + \nabla \cdot \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^t) + \mathbf{f},$$

where $\mathbf{u} : \Omega \times (0, T) \to \mathbb{R}^3$ is the displacement field, $\rho : \Omega \to \mathbb{R}^+$ is the mass density, $\lambda, \mu : \Omega \to \mathbb{R}^+$ are the Lamé parameters, and $\mathbf{f} : \Omega \times (0, T) \to \mathbb{R}^3$ is the external volume force. The superscript t denotes the transposed.

Case 2 (Acoustic wave equation)**.** *To obtain the acoustic wave equation, set $m = 1$, $u = p$, $\rho = (\tilde{\rho}\tilde{c}^2)^{-1}$, and*

$$C_{i11j} := \frac{1}{\tilde{\rho}}\delta_{ij},$$

for $i, j = 1, 2, 3$, where δ is the Kronecker delta. Equation (4.1a) then becomes

$$\frac{1}{\tilde{\rho}\tilde{c}^2}\partial_t^2 p = \nabla \cdot \frac{1}{\tilde{\rho}}\nabla p + f,$$

where $p : \Omega \times (0, T) \to \mathbb{R}$ is the pressure field, $\tilde{\rho} : \Omega \to \mathbb{R}^+$ the mass density, $\tilde{c} : \Omega \to \mathbb{R}^+$ the acoustic velocity field, and $f = \nabla \cdot (\tilde{\rho}^{-1}\tilde{\mathbf{f}})$ the source term with $\tilde{\mathbf{f}} : \Omega \times (0, T) \to \mathbb{R}^3$ the external volume force.

These equations can be solved with the finite element methods described in the next section.

4.4 The discontinuous Galerkin and mass-lumped finite element method

4.4.1 The classical finite element method

Let $\mathcal{T}_h$ be a tetrahedral tessellation of Ω, with h denoting the radius of the smallest sphere that can contain each element and let $\mathcal{U}_h$ be the finite element space consisting of continuous element-wise polynomial basis functions satisfying boundary condition (4.1b). The classical conforming finite element formulation of (4.1) is finding $\mathbf{u}_h : [0,T] \to \mathcal{U}_h$ such that $\mathbf{u}_h|_{t=0} = \Pi_h \mathbf{u}_0$, $\partial_t \mathbf{u}_h|_{t=0} = \Pi_h \mathbf{v}_0$ and

$$(\rho \partial_t^2 \mathbf{u}_h, \mathbf{w}) + a(\mathbf{u}_h, \mathbf{w}) = (\mathbf{f}, \mathbf{w}), \qquad \mathbf{w} \in \mathcal{U}_h, t \in [0,T], \tag{4.2}$$

where $(\cdot,\cdot)$ denotes the standard L^2 inner product, $\Pi_h : L^2(\Omega) \to \mathcal{U}_h$ denotes the weighted L^2-projection operator defined such that $(\rho \Pi_h \mathbf{u}, \mathbf{w}) = (\rho \mathbf{u}, \mathbf{w})$ for all $\mathbf{w} \in \mathcal{U}_h$, and $a : H^1(\Omega)^m \times H^1(\Omega)^m \to \mathbb{R}$ is the (semi)-elliptic operator given by

$$a(\mathbf{u}, \mathbf{w}) := \int_\Omega (\nabla \mathbf{u})^t : C : \nabla \mathbf{w} \; dx.$$

Let $\{\mathbf{w}^{(i)}\}_{i=1}^n$ be the set of basis functions spanning $\mathcal{U}_h$, and let, for any $\mathbf{u} \in L^2(\Omega)^m$, the vector $\underline{\mathbf{u}} \in \mathbb{R}^n$ be defined such that $\sum_{i=1}^n \underline{u}_i \mathbf{w}^{(i)} = \Pi_h \mathbf{u}$. Also, let $M, A \in \mathbb{R}^{n \times n}$ be the mass matrix and stiffness matrix, respectively, defined by $M_{ij} := (\rho \mathbf{w}^{(i)}, \mathbf{w}^{(j)})$ and $A_{ij} := a(\mathbf{w}^{(i)}, \mathbf{w}^{(j)})$, and let $\underline{\mathbf{f}}^* : [0,T] \to \mathbb{R}^n$ be given by $\underline{f}_i^* := (\mathbf{f}, \mathbf{w}^{(i)})$. The finite element method can then be formulated as finding $\underline{\mathbf{u}}_h : [0,T] \to \mathbb{R}^n$ such that $\underline{\mathbf{u}}_h|_{t=0} = \underline{\mathbf{u}_0}$, $\partial_t \underline{\mathbf{u}}_h|_{t=0} = \underline{\mathbf{v}_0}$, and

$$M \partial_t^2 \underline{\mathbf{u}}_h + A \underline{\mathbf{u}}_h = \underline{\mathbf{f}}^*, \qquad t \in [0,T]. \tag{4.3}$$

The main drawback of the classical conforming finite element approach is that when an explicit time integration scheme is applied, a system of equations of the form $M\underline{\mathbf{x}} = \underline{\mathbf{b}}$ needs to be solved at every time step, with M not (block)-diagonal. For large-scale problems, this results in a very inefficient time stepping scheme. This problem can be circumvented by lumping the mass matrix into a diagonal matrix or by using discontinuous basis functions.

4.4.2 Mass lumping

When using nodal basis functions, the mass matrix can be lumped into a diagonal matrix by taking the sum over each row. This is equivalent to replacing the inner product $(\cdot,\cdot)$ by $(\cdot,\cdot)_h^{(L)}$, in which the element integrals are approximated by a quadrature rule with quadrature points that coincide with the nodes of the basis functions. We can write

$$(\mathbf{u}, \mathbf{w})_h^{(L)} = \sum_{e \in \mathcal{T}_h} \sum_{\mathbf{x} \in \mathcal{Q}_e} \omega_{e,\mathbf{x}} \rho(\mathbf{x}) \mathbf{u}(\mathbf{x}) \cdot \mathbf{w}(\mathbf{x}),$$

where $\mathcal{Q}_e$ denotes the quadrature points on element e and $\omega_{e,\mathbf{x}}$ denote the quadrature weights. Let $\{\mathbf{x}^{(i)}\}_{i=1}^{n}$ denote the global set of integration points and define $\mathbf{w}^{(i)}$ to be the nodal basis function corresponding to $\mathbf{x}^{(i)}$, so $\mathbf{w}^{(i)}(\mathbf{x}^{(j)}) = \delta_{ij}$, with δ the Kronecker delta. Then the mass matrix becomes diagonal with entries $M_{ii} = \sum_{e\in\mathcal{T}_{\mathbf{x}^{(i)}}} \omega_{e,\mathbf{x}^{(i)}}\rho(\mathbf{x}^{(i)})$, where $\mathcal{T}_{\mathbf{x}}$ denotes the set of elements containing or adjacent to $\mathbf{x}$.

For quadrilaterals and hexahedra, mass-lumping is achieved by using tensor-product basis functions and Gauss-Lobatto integration points. The resulting scheme is known as the spectral element method. For triangles and tetrahedra, mass-lumping is less straight-forward. Combining standard Lagrangian basis functions with a Newton–Cotes quadrature rule results in an efficient mass-lumped scheme for linear tetrahedra, but for higher-degree basis functions, this approach results either in an unstable scheme due to non-positive quadrature weights or in a scheme with a reduced order of convergence. This problem can be resolved by enriching the finite element space with higher-degree bubble functions and by adding integration points to the interior of the elements and faces. For example, by enriching the space of the quadratic tetrahedron with 3 degree-4 face bubble functions and 1 degree-4 interior bubble function, an enriched degree-2 mass-lumped tetrahedron that remains third-order accurate can be obtained [51].

In this chapter, we will analyse the standard linear mass-lumped finite element method, the mass-lumped finite element method of degree 2 derived in [51], and the 2 versions of degree 3 mass-lumped finite element methods derived in [11]. We will refer to these methods as ML1, ML2, ML3a and ML3b, respectively.

4.4.3 The symmetric interior penalty discontinuous Galerkin method

Another way to obtain a (block)-diagonal mass matrix is by allowing the finite element space $\mathcal{U}_h$ to be discontinuous at the faces. When choosing basis functions that have support on only a single element, the mass matrix becomes block-diagonal with each block corresponding to a single element. When using orthogonal basis functions, the mass matrix even becomes strictly diagonal. In order to keep the finite element method stable and consistent with the analytic solution, the elliptic operator needs to be augmented. This can be accomplished with the symmetric interior penalty method [36], where a is replaced by the discrete (semi)-elliptic operator

$a_h^{(DG)} : \mathcal{U}_h \times \mathcal{U}_h \to \mathbb{R}$, given by

$$a_h^{(DG)}(\mathbf{u},\mathbf{w}) := a_h^{(C)}(\mathbf{u},\mathbf{w}) - a_h^{(D)}(\mathbf{u},\mathbf{w}) - a_h^{(D)}(\mathbf{w},\mathbf{u}) + a_h^{(IP)}(\mathbf{u},\mathbf{w})$$

with

$$\begin{aligned}
a_h^{(C)}(\mathbf{u},\mathbf{w}) &:= \sum_{e\in\mathcal{T}_h} \int_e (\nabla\mathbf{u})^t : C : \nabla\mathbf{w}\; d\mathbf{x}, \\
a_h^{(D)}(\mathbf{u},\mathbf{w}) &:= \sum_{f\in\mathcal{F}_{h,in}\cup\mathcal{F}_{h,d}} \int_f [\![\mathbf{u}]\!]^t : \{\!\{C : \nabla\mathbf{w}\}\!\}\; d\mathbf{s}, \\
a_h^{(IP)}(\mathbf{u},\mathbf{w}) &:= \sum_{f\in\mathcal{F}_{h,in}\cup\mathcal{F}_{h,d}} \int_f [\![\mathbf{u}]\!]^t : \{\!\{\alpha_h C\}\!\} : [\![\mathbf{w}]\!]\; d\mathbf{s},
\end{aligned}$$

where $\mathcal{F}_{h,in}$ and $\mathcal{F}_{h,d}$ are the internal faces and boundary faces on Γ_d, respectively, $\alpha_h \in \bigotimes_{e\in\mathcal{T}} L^\infty(\partial e)$ is the penalty function, and $\{\!\{\cdot\}\!\}$, $[\![\cdot]\!]$ are the average trace operator and jump operator, respectively, defined as

$$\{\!\{\phi\}\!\}\big|_f := \frac{1}{|\mathcal{T}_f|}\sum_{e\in\mathcal{T}_f} \phi|_{\partial e\cap f}, \qquad [\![\mathbf{u}]\!]\big|_f := \sum_{e\in\mathcal{T}_f} (\hat{\mathbf{n}}\mathbf{u})|_{\partial e\cap f},$$

for all faces $f \in \mathcal{F}$, where $\mathcal{T}_f$ denotes the set of elements adjacent to face f, and $\hat{\mathbf{n}}|_{\partial e}$ denotes the outward pointing normal unit vector of element e. The bilinear form $a_h^{(C)}$ is the same as the original elliptic operator a and is the part that remains when both input functions are continuous. The bilinear form $a_h^{(D)}$ can be interpreted as the additional part that results from partial integration of the elliptic operator a when the first input function is discontinuous. Finally, the bilinear form $a_h^{(IP)}$ is the part that contains the interior penalty function needed to ensure stability of the scheme.

The penalty term can have a significant impact on the performance of the SIPDG method, since a larger penalty term results in a more restrictive bound on the time step size, but also because it can have a significant effect on the accuracy, as we will show in Section 4.6. Several lower bounds for the penalty term are based on the trace inequality of [79], including [64, 27, 55], among which we found the bound in [55] to be the sharpest. Recently, a sharper penalty term bound was presented in [32], which is based on a more involved trace inequality. In this chapter, we will consider both the penalty term of [32], given by (4.4a), and the one of [55], given by (4.4b):

$$\alpha_h|_{\partial e\cap f} := \frac{\nu_h|_{\partial e\cap f}}{|\mathcal{T}_f|} \sup_{\substack{\mathbf{u}\in\mathcal{P}^p(e)^m\\ C:\nabla\mathbf{u}\neq\mathbf{0}}} \frac{\displaystyle\int_{\partial e} (\hat{\mathbf{n}}\cdot C:\nabla\mathbf{u})\cdot\nu_h^{-1}\mathbf{c}_{\hat{\mathbf{n}}}^{-1}\cdot(\hat{\mathbf{n}}\cdot C:\nabla\mathbf{u})\,ds}{\displaystyle\int_e (\nabla\mathbf{u})^t : C : \nabla\mathbf{u}\,d\mathbf{x}}, \tag{4.4a}$$

$$\alpha_h|_{\partial e\cap f} := \frac{p(p+2)}{\min_{e\in\mathcal{T}_f} d_e}, \tag{4.4b}$$

for all $e \in \mathcal{T}_h$, $f \subset \partial e$, where p denotes the degree of the polynomial basis functions, $\mathcal{P}^p(e)$ denotes the space of polynomial functions of degree p or less in element e, $\nu_h|_{\partial e\cap f} := |f|/|e|$ is a scaling function of order h^{-1}, with $|e|, |f|$ the volume of e and area of f, respectively, $\mathbf{c}_{\hat{\mathbf{n}}}^{-1}$ denotes the (pseudo)-inverse of the second-order tensor $\mathbf{c}_{\hat{\mathbf{n}}} := \hat{\mathbf{n}}\cdot C\cdot\hat{\mathbf{n}}$, where $\hat{\mathbf{n}}$ is the outward pointing normal unit vector, and d_e denotes the diameter of the inscribed sphere of e. Although the first version requires more preprocessing time, it allows for an approximately 1.5 times larger time step [32].

We will refer to the SIPDG method with $p = 1, 2, 3$ using the penalty term as defined by (4.4a) as DG1a, DG2a, and DG3a, respectively, and to the same methods using the penalty term as defined by (4.4b) as DG1b, DG2b, and DG3b.

4.4.4 The Lax–Wendroff time integration scheme

To solve the resulting set of ODE's (4.3) in time, we use the Lax–Wendroff method [45, 20], which is based on Taylor expansions in time and substitutes the time derivatives by matrix-vector operators using the original equations (4.3). For the second-order formulation, the resulting scheme is also known as Dablain's scheme [18]. The advantage of this scheme is that it is time-reversible, energy-conservative, and only requires K stages for a $2K$-order of accuracy.

To introduce the scheme, let $\Delta t > 0$ denote the time step size, and let $\underline{\mathbf{U}}_h(t_i)$ denote the approximation of $\underline{\mathbf{u}}_h$ at time $t_i := i\Delta t$ for $i = 0, \ldots, N_T$ with N_T the total number of time steps. The order-$2K$ Lax–Wendroff method can be written as

$$\underline{\mathbf{U}}_h(t_{i+1}) = -\underline{\mathbf{U}}_h(t_{i-1}) + 2\sum_{k=0}^{K} \frac{1}{(2k)!}\Delta t^{2k}(\partial_t^{2k}\underline{\mathbf{U}}_h)(t_i), \quad i = 1, \ldots, N_T - 1, \tag{4.5}$$

with $\underline{\mathbf{U}}_h(t_0) = \underline{\mathbf{U}}_h(0) := \underline{\mathbf{u_0}}$ and $\underline{\mathbf{U}}_h(t_1) := \sum_{k=0}^{2K+1} \frac{1}{k!}\Delta t^k (\partial_t^k \underline{\mathbf{U}}_h)(0)$, and where $(\partial_t^k \underline{\mathbf{U}}_h)(t_i)$ is recursively defined by

$$(\partial_t^k \underline{\mathbf{U}}_h)(0) := \begin{cases} \underline{\mathbf{u_0}} & k = 0, \\ \underline{\mathbf{v_0}} & k = 1, \\ -M^{-1}A(\partial_t^{k-2}\underline{\mathbf{U}}_h)(0) + \partial_t^{k-2}\underline{\mathbf{f}}(0) & k \geq 2, \end{cases}$$

and

$$(\partial_t^k \underline{\mathbf{U}}_h)(t_i) := \begin{cases} \underline{\mathbf{U}}_h(t_i) & k = 0, \\ -M^{-1}A(\partial_t^{k-2}\underline{\mathbf{U}}_h)(t_i) + \partial_t^{k-2}\underline{\mathbf{f}}(t_i) & k = 2, 4, 6, \dots, 2K, \end{cases}$$

for $i \geq 1$, with $\underline{\mathbf{f}} := M^{-1}\underline{\mathbf{f}}^*$. In case $K = 1$, this scheme reduces to the standard leap-frog or central difference scheme. When there is no source term, (4.5) simplifies to

$$\underline{\mathbf{U}}_h(t_{i+1}) = -\underline{\mathbf{U}}_h(t_{i-1}) + 2\sum_{k=0}^{K} \frac{1}{(2k)!}\Delta t^{2k}(-M^{-1}A)^k \underline{\mathbf{U}}_h(t_i), \tag{4.6}$$

for $i = 1, \dots, N_T - 1$.

For the dispersion analysis, we will choose K equal to the polynomial degree p of the spatial discretization, since this will result in a $2p$-order convergence rate of the dispersion error as shown in Section 4.6.

4.5 Dispersion analysis

A common measure for the quality of a numerical method for wave propagation modelling is the amount of numerical dispersion and dissipation. Numerical dispersion refers in this context to the discrepancy between the numerical and physical wave propagation speed and numerical dissipation is the loss of energy in the numerical scheme. Since the schemes that we consider are all energy-conservative, they do not suffer from numerical dissipation. However, when projecting a physical wave onto the discrete space, this results in a superposition of a well-matching numerical wave and several numerical waves that have a completely different shape and frequency. We compute the number of these non-matching or spurious waves and refer to it as the eigenvector error, since it is related to the accuracy of the eigenvectors of $M^{-1}A$, while the dispersion error is related to the accuracy of the eigenvalues of $M^{-1}A$.

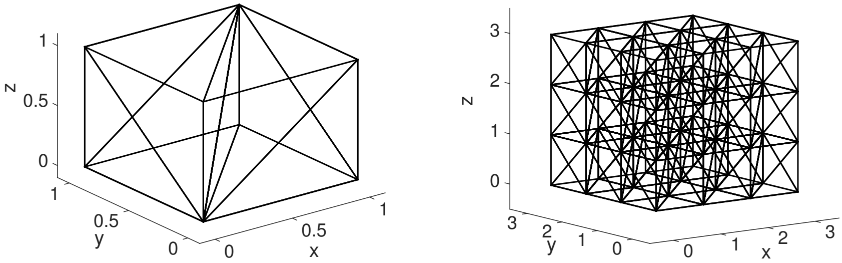


Figure 4.1: Unit cell subdivided into tetrahedra (left), and periodic mesh made from 3 × 3 × 3 copies of this unit cell (right).

We analyse the dispersion and eigenvector error using standard Fourier analysis, which is also known in this context as plane wave analysis. The main idea of this analysis is to compare physical plane waves with numerical plane waves on a homogeneous periodic domain, free from external forces, using a periodic mesh. To obtain a periodic tetrahedral mesh we subdivide a small cell into tetrahedra and repeat this pattern to fill the entire domain as illustrated in Figure 4.1. By using Fourier modes, we can then efficiently compute the numerical plane waves and their dispersion properties by solving eigenvalue problems on only a single cell.

Our analysis is similar to [19], but with the following extensions:

- We extend the analysis to parallellepiped cells, since this allows for a more regular tetrahedral mesh.

- We also compute the number of spurious modes that appear in the projection of the physical wave.

- In the three-dimensional elastic case, there are two distinct secondary or shear waves with the same wave vector. To compute the dispersion and eigenvector error in this case, we consider the two best matching numerical plane waves.

We explain the dispersion analysis in more detail in the following subsections. First, we show how we can derive an analytical expression for the numerical plane waves using Fourier modes. After that, we show how we use this to compute the numerical dispersion and eigenvector error. In the last subsection we explain how we estimate the computational cost for each method.

4.5.1 Analytic expression for the numerical plane waves

We first consider a periodic cubic domain of the form $\Omega := [0, N)^3$, with N a positive integer, and later extend the results to parallelepiped domains which allow for more regular tetrahedral meshes. The physical plane wave has the following form:

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{a}e^{\hat{\imath}(\boldsymbol{\kappa}\cdot\mathbf{x}-\omega t)}, \qquad \mathbf{x} \in \Omega, t \in [0, T], \tag{4.7}$$

where $\hat{\imath} := \sqrt{-1}$ is the imaginary number, $\boldsymbol{\kappa} \in \mathbb{R}^3$ is the wave vector, $\omega \in \mathbb{R}$ is the angular velocity, and $\mathbf{a} \in \mathbb{R}^m$ is the amplitude vector. The wave vector must be of the form $\boldsymbol{\kappa} = \boldsymbol{\kappa}_{\mathbf{z}} = \frac{2\pi}{N}\mathbf{z}$, with $\mathbf{z} \in \mathbb{Z}_N^3$, in order to satisfy the periodic boundary conditions.

The numerical plane wave can be written in a similar form when using a periodic mesh. To obtain a periodic tetrahedral mesh, we subdivide the unit cell $\Omega_0 := [0, 1)^3$ into tetrahedra and repeat this pattern $N \times N \times N$ times to fill the entire domain as illustrated in Figure 4.1. We equip the mesh with a translation-invariant set of basis functions where each basis function has minimal support. In case of mass-lumping we use nodal basis functions and in case of DG we use basis functions that have support on only a single element. The numerical plane wave $\underline{\mathbf{U}}_h$ of the fully discrete scheme then has the form

$$\underline{\mathbf{U}}_h(\Omega_{\mathbf{k}}, t_i) = \underline{\mathbf{U}}_{h,\Omega_0}e^{\hat{\imath}(\boldsymbol{\kappa}\cdot\mathbf{x}_{\mathbf{k}}-\omega_h t)}, \qquad i = 0, \dots, N_T, \mathbf{k} \in \mathbb{Z}_N^3. \tag{4.8}$$

Here, $\underline{\mathbf{U}}_h(\Omega_{\mathbf{k}}, t_i)$ denotes the coefficients of the basis functions corresponding to cell $\Omega_{\mathbf{k}} := \mathbf{k} + \Omega_0$ at time t_i. In case of mass-lumping, these basis functions are the nodal basis functions corresponding to the nodes on $\Omega_{\mathbf{k}} = \mathbf{k} + [0, 1)^3$, while in case of DG, these are the basis functions that have support on one of the tetrahedra in $\Omega_{\mathbf{k}}$. The vector $\underline{\mathbf{U}}_{h,\Omega_0} \in \mathbb{R}^{n_0}$ denotes the basis function coefficients corresponding to cell Ω_0 at time 0 and $\mathbf{x}_{\mathbf{k}} = \mathbf{k}$ are the coordinates of the front-left-bottom vertex of cell $\Omega_{\mathbf{k}}$.

To show that this is indeed a numerical plane wave, let $M^{(\Omega_{\mathbf{k}},\Omega_{\mathbf{m}})}$, $A^{(\Omega_{\mathbf{k}},\Omega_{\mathbf{m}})} \in \mathbb{R}^{n_0\times n_0}$, for $\mathbf{k}, \mathbf{m} \in \mathbb{Z}_N^3$, be submatrices of M and A, respectively, defined as follows:

$$\begin{aligned} M_{ij}^{(\Omega_{\mathbf{k}},\Omega_{\mathbf{m}})} &:= \left(\rho\mathbf{w}^{(\Omega_{\mathbf{k}},i)}, \mathbf{w}^{(\Omega_{\mathbf{m}},j)}\right)_h, && i, j = 1, \dots, n_0, \\ A_{ij}^{(\Omega_{\mathbf{k}},\Omega_{\mathbf{m}})} &:= a_h\left(\mathbf{w}^{(\Omega_{\mathbf{k}},i)}, \mathbf{w}^{(\Omega_{\mathbf{m}},j)}\right), && i, j = 1, \dots, n_0, \end{aligned}$$

where $\{\mathbf{w}^{(\Omega_{\mathbf{k}},i)}\}_{i=0}^{n_0}$ denote the basis functions corresponding to cell $\Omega_{\mathbf{k}}$ and where $a_h = a_h^{(DG)}$ and $(\cdot, \cdot)_h = (\cdot, \cdot)$ in case of the DG method and $a_h = a$ and $(\cdot, \cdot)_h = (\cdot, \cdot)_h^{(L)}$ in case of the mass-lumped method.

Since the basis functions are translation invariant, the submatrices $M^{(\Omega_{\mathbf{k}},\Omega_{\mathbf{k}+\Delta\mathbf{k}})}$ and $A^{(\Omega_{\mathbf{k}},\Omega_{\mathbf{k}+\Delta\mathbf{k}})}$ are the same for any $\mathbf{k}\in\mathbb{Z}_N^3$ with $\Delta\mathbf{k}\in\mathbb{Z}_N^3$ fixed. Furthermore, the submatrices $M^{(\Omega_{\mathbf{k}},\Omega_{\mathbf{m}})}$ are only non-zero when $\mathbf{k}=\mathbf{m}$, since the mass matrix is diagonal in case of mass-lumping and block-diagonal, with each block corresponding to an element, in case of DG. The submatrices $A^{(\Omega_{\mathbf{k}},\Omega_{\mathbf{k}+\Delta\mathbf{k}})}$ are only non-zero when $\Delta\mathbf{k}\in\{-1,0,1\}^3$, since the nodal basis functions for mass-lumping and the local basis functions for DG do not interact when they are two or more cells apart. This implies that we only need to consider the submatrices $M^{(\Omega_0)}:=M^{(\Omega_0,\Omega_0)}$ and $A^{(\Omega_0,\Omega_{\Delta\mathbf{k}})}$ for $\Delta\mathbf{k}=\{-1,0,1\}^3$.

Now let $\boldsymbol{\kappa}=\boldsymbol{\kappa}_{\mathbf{z}}:=\frac{2\pi}{N}\mathbf{z}$, for some $\mathbf{z}\in\mathbb{Z}_N^3$, and let $\underline{\mathbf{U}}_{h,0}\in\mathbb{R}^{N^3\times n_0}$ be the numerical wave at time $t=0$:

$$\underline{\mathbf{U}}_{h,0}(\Omega_{\mathbf{k}}):=\underline{\mathbf{U}}_{h,\Omega_0}e^{\hat{\imath}(\boldsymbol{\kappa}\cdot\mathbf{x}_{\mathbf{k}})},\qquad\qquad \mathbf{k}\in\mathbb{Z}_N^3. \tag{4.9}$$

Then $M^{-1}A\underline{\mathbf{U}}_{h,0}$ satisfies

$$\begin{aligned}\big(M^{-1}A\underline{\mathbf{U}}_{h,0}\big)(\Omega_{\mathbf{k}}) &= M_{inv}^{(\Omega_0)}\left(\sum_{\Delta\mathbf{k}\in\{-1,0,1\}^3}A^{(\Omega_0,\Omega_{\Delta\mathbf{k}})}\underline{\mathbf{U}}_{h,0}(\Omega_{\mathbf{k}+\Delta\mathbf{k}})\right)\\ &= M_{inv}^{(\Omega_0)}\left(\sum_{\Delta\mathbf{k}\in\{-1,0,1\}^3}e^{\hat{\imath}(\boldsymbol{\kappa}\cdot\mathbf{x}_{\Delta\mathbf{k}})}A^{(\Omega_0,\Omega_{\Delta\mathbf{k}})}\right)\underline{\mathbf{U}}_{h,\Omega_0}e^{\hat{\imath}(\boldsymbol{\kappa}\cdot\mathbf{x}_{\mathbf{k}})}\\ &= M_{inv}^{(\Omega_0)}A^{(\boldsymbol{\kappa})}\underline{\mathbf{U}}_{h,\Omega_0}e^{\hat{\imath}(\boldsymbol{\kappa}\cdot\mathbf{x}_{\mathbf{k}})},\end{aligned}$$

for all $\mathbf{k}\in\mathbb{Z}_N^3$, with $M_{inv}^{(\Omega_0)}$ the inverse of $M^{(\Omega_0)}$ and

$$A^{(\boldsymbol{\kappa})}:=\sum_{\Delta\mathbf{k}\in\{-1,0,1\}^3}e^{\hat{\imath}(\boldsymbol{\kappa}\cdot\mathbf{x}_{\Delta\mathbf{k}})}A^{(\Omega_0,\Omega_{\Delta\mathbf{k}})}.$$

This implies that if $(s_h,\underline{\mathbf{U}}_{h,\Omega_0})$ is an eigenpair of $S^{(\boldsymbol{\kappa})}:=M_{inv}^{(\Omega_0)}A^{(\boldsymbol{\kappa})}$, then $(s_h,\underline{\mathbf{U}}_{h,0})$ is an eigenpair of $M^{-1}A$. In other words, we can obtain eigenpairs of $M^{-1}A$ by computing the eigenpairs of a small matrix $S^{(\boldsymbol{\kappa})}\in\mathbb{R}^{n_0\times n_0}$. Note that since $M^{(\Omega_0)}$ is symmetric positive definite, and $A^{(\boldsymbol{\kappa})}$ is Hermitian, $S^{(\boldsymbol{\kappa})}$ has n_0 distinct eigenpairs. Since there are N^3 choices for $\mathbf{z}\in\mathbb{Z}_N^3$ and $S^{(\boldsymbol{\kappa}_{\mathbf{z}})}$ has n_0 eigenpairs, we can obtain all of the $N^3\times n_0$ eigenpairs of $M^{-1}A$ in this way.

Now consider the numerical plane wave in (4.8) with $(s_h,\underline{\mathbf{U}}_{h,\Omega_0})$ an eigenpair of $S^{(\boldsymbol{\kappa})}$, so with $(s_h,\underline{\mathbf{U}}_{h,0})$ an eigenpair of $M^{-1}A$. We can rewrite

$\underline{\mathbf{U}}_h$ as $\underline{\mathbf{U}}_h(t) = \underline{\mathbf{U}}_{h,0}e^{-\hat{\imath}(\omega t)}$. If we then substitute this wave into (4.6) we obtain

$$\cos(\Delta t\omega_h)\underline{\mathbf{U}}_h(t_i) = \sum_{k=0}^{K} \frac{1}{(2k)!}(-\Delta t^2 s_h)^k \underline{\mathbf{U}}_h(t_i), \qquad i = 1, \ldots, n_T - 1.$$

From this, it follows that $\underline{\mathbf{U}}_h$ in (4.8) is a discrete plane wave if $(s_h, \underline{\mathbf{U}}_{h,\Omega_0})$ is an eigenpair of $S^{(\boldsymbol{\kappa})}$ and if ω_h satisfies $\cos(\Delta t\omega_h) = \sum_{k=0}^{K} \frac{1}{(2k)!}(-\Delta t^2 s_h)^k$, so if

$$\omega_h = \pm\frac{1}{\Delta t}\arccos\left(\sum_{k=0}^{K} \frac{1}{(2k)!}(-\Delta t^2 s_h)^k\right). \tag{4.10}$$

It remains to determine the time step size Δt. In the appendix we show that the numerical scheme is stable, if

$$\Delta t \leq \sqrt{c_K/\sigma_{max}(M^{-1}A)}, \tag{4.11}$$

where $\sigma_{max}(M^{-1}A)$ denotes the spectral radius of $M^{-1}A$ and c_K is a constant, given by

$$c_K := \inf\left\{x \geq 0 \mid \left|\sum_{k=0}^{K} \frac{1}{(2k)!}(-x)^k\right| > 1\right\}. \tag{4.12}$$

To obtain a bound on the spectral radius, recall that we can write every eigenpair of $M^{-1}A$ in the form of $(s_h, \underline{\mathbf{U}}_{h,0})$, with $\underline{\mathbf{U}}_{h,0}$ given in (4.9) and with $(s_h, \underline{\mathbf{U}}_{h,\Omega_0})$ an eigenpair of $S^{(\boldsymbol{\kappa}_\mathbf{z})}$ for some $\mathbf{z} \in \mathbb{Z}_N^3$. This implies that $\sigma_{max}(M^{-1}A)$ is equal to $\sup_{\mathbf{z}\in\mathbb{Z}_N^3} \sigma_{max}(S^{(\boldsymbol{\kappa}_\mathbf{z})})$. We can therefore bound $\sigma_{max}(M^{-1}A)$ as follows:

$$\sigma_{max}(M^{-1}A) = \sup_{\mathbf{z}\in\mathbb{Z}_N^3} \sigma_{max}(S^{(\boldsymbol{\kappa}_\mathbf{z})}) \leq \sup_{\boldsymbol{\kappa}\in\mathcal{K}_0} \sigma_{max}(S^{(\boldsymbol{\kappa})}) =: s_{h,max}, \tag{4.13}$$

with $\mathcal{K}_0 := [0, 2\pi)^3 \supset \{\boldsymbol{\kappa}_\mathbf{z}\}_{\mathbf{z}\in\mathbb{Z}_N^3}$ the space of all distinct wave vectors $\boldsymbol{\kappa}$.

The constants c_K can be computed numerically. For example, $c_K = 4, 12, 7.57$ for $K = 1, 2, 3$, respectively. For higher values of K, see, for example, [53], where his σ_t satisfies $c_K = 2\sigma_t$.

We can extend the results of this section to parallelepiped cells by applying a linear transformation $\mathbf{x} \to \mathbf{T}\cdot\mathbf{x}$, with $\mathbf{T} \in \mathbb{R}^{3\times 3}$ a second-order tensor. The parallelepiped domain is then given by $\Omega = \mathbf{T} \cdot (0, N)^3$ and the cells are given by $\Omega_0 = \mathbf{T} \cdot [0, 1)^3$ and $\Omega_\mathbf{k} = \mathbf{x}_\mathbf{k} + \Omega_0$, with $\mathbf{x}_\mathbf{k} = \mathbf{T} \cdot \mathbf{k}$ the front-left-bottom vertex. The wave vectors $\boldsymbol{\kappa}_\mathbf{z}$ are of the form $\boldsymbol{\kappa}_\mathbf{z} = \frac{2\pi}{N}(\mathbf{T}^{-t} \cdot \mathbf{z})$ and the wave vector space $\mathcal{K}_0$ is given by $\mathcal{K}_0 := \mathbf{T}^{-t} \cdot [0, 2\pi)^3$, with $\mathbf{T}^{-t}$ the transposed inverse of $\mathbf{T}$.

4.5.2 Computing the dispersion and eigenvector error

To explain how we compute the dispersion error, we first consider the acoustic wave equation. Let $\boldsymbol{\kappa}$ be a given wave vector and let $\mathbf{u}^{(\boldsymbol{\kappa})}$ be the acoustic plane wave given by $\mathbf{u}^{(\boldsymbol{\kappa})}(\mathbf{x},t) = e^{\hat{\imath}(\boldsymbol{\kappa}\cdot\mathbf{x}-\omega t)}$. The angular velocity is given by $\omega = \pm c|\boldsymbol{\kappa}|$ with c the acoustic wave propagation speed. We compare this plane wave with the numerical plane waves.

To do this, we use the results from the previous subsection. There, we showed that for any eigenpair $(s_h, \underline{\mathbf{U}}_{h,\Omega_0})$ of $S^{(\boldsymbol{\kappa})}$ we can obtain a numerical plane wave in the form of (4.8) with angular velocity $\pm\omega_h$ given by (4.10). Since $S^{(\boldsymbol{\kappa})}$ has n_0 eigenpairs, this means we can obtain n_0 discrete plane waves $\{\underline{\mathbf{U}}_h^{(\boldsymbol{\kappa},i)}\}_{i=1}^{n_0}$, with angular velocities $\{\pm\omega_h^{(\boldsymbol{\kappa},i)}\}_{i=1}^{n_0}$, for a given wave vector $\boldsymbol{\kappa}$. The corresponding wave propagation speeds $\{c_h^{(\boldsymbol{\kappa},i)}\}_{i=1}^{n_0}$ can be computed by $c_h^{(\boldsymbol{\kappa},i)} = |\omega_h^{(\boldsymbol{\kappa},i)}|/|\boldsymbol{\kappa}|$ and we can order the numerical plane waves such that

$$|c - c_h^{(\boldsymbol{\kappa},1)}| \le |c - c_h^{(\boldsymbol{\kappa},2)}| \le \dots .$$

We consider $\underline{\mathbf{U}}_h^{(\boldsymbol{\kappa},1)}$ to be the matching numerical plane wave and $\underline{\mathbf{U}}_h^{(\boldsymbol{\kappa},i)}$, with $i > 1$, to be spurious modes. We then define the dispersion error as follows

$$e_{disp}(\boldsymbol{\kappa}) = \frac{|c - c_h^{(\boldsymbol{\kappa},1)}|}{c}.$$

The complete procedure for computing $e_{disp}(\boldsymbol{\kappa})$ in the acoustic case is given by

a. Compute all eigenpairs $(s_h^{(\boldsymbol{\kappa},i)}, \underline{\mathbf{U}}_{h,\Omega_0}^{(\boldsymbol{\kappa},i)})$ of $S^{(\boldsymbol{\kappa})} := M_{inv}^{(\Omega_0)} A^{(\boldsymbol{\kappa})}$.

b. Compute $\omega_h^{(\boldsymbol{\kappa},i)} = \frac{1}{\Delta t}\arccos\left(\sum_{k=0}^{K}\frac{1}{(2k)!}(-\Delta t^2 s_h^{(\boldsymbol{\kappa},i)})^k\right)$, the angular velocities.

c. Compute the wave propagation speeds $c_h^{(\boldsymbol{\kappa},i)} = \omega_h^{(\boldsymbol{\kappa},i)}/|\boldsymbol{\kappa}|$ and order everything such that $|c - c_h^{(\boldsymbol{\kappa},1)}| \le |c - c_h^{(\boldsymbol{\kappa},2)}| \le \dots$.

d. Compute $e_{disp}(\boldsymbol{\kappa}) = |c - c_h^{(\boldsymbol{\kappa},1)}|/c$.

Now let $\mathbf{u}_0^{(\boldsymbol{\kappa})}(\mathbf{x}) := \mathbf{u}^{(\boldsymbol{\kappa})}(\mathbf{x},0) = e^{\hat{\imath}(\boldsymbol{\kappa}\cdot\mathbf{x})}$ be the acoustic plane wave at $t = 0$. Also, let $\underline{\mathbf{u}}_0^{(\boldsymbol{\kappa})}$ be the projection of $\mathbf{u}_0^{(\boldsymbol{\kappa})}$ onto the numerical space, and let $\underline{\mathbf{U}}_{h,0}^{(\boldsymbol{\kappa},i)}$ be the discrete plane wave at $t = 0$. In the ideal case, $\underline{\mathbf{u}}_0^{(\boldsymbol{\kappa})}$

is equal to $\underline{\mathbf{U}}_{h,0}^{(\boldsymbol{\kappa},1)}$ up to a constant. In most cases, however, the projection $\underline{\mathbf{u}}_0^{(\boldsymbol{\kappa})}$ is a superposition of a well-matching plane wave $\underline{\mathbf{U}}_{h,0}^{(\boldsymbol{\kappa},1)}$ and several other plane waves $\underline{\mathbf{U}}_{h,0}^{(\boldsymbol{\kappa},i)}$, for $i > 1$, that may have a completely different shape and velocity. We can compute the number of these spurious waves by computing the projection error.

To do this, we let $\underline{\mathbf{U}}_0^{(\boldsymbol{\kappa})} \in \operatorname{span}\{\underline{\mathbf{U}}_{h,0}^{(\boldsymbol{\kappa},1)}\}$ denote the projection of $\underline{\mathbf{u}}_0^{(\boldsymbol{\kappa})}$ onto $\operatorname{span}\{\underline{\mathbf{U}}_{h,0}^{(\boldsymbol{\kappa},1)}\}$, such that $(\underline{\mathbf{U}}_0^{(\boldsymbol{\kappa})}, \underline{\mathbf{U}}_{h,0}^{(\boldsymbol{\kappa},1)})_M = (\underline{\mathbf{u}}_0^{(\boldsymbol{\kappa})}, \underline{\mathbf{U}}_{h,0}^{(\boldsymbol{\kappa},1)})_M$, with inner product $(\underline{\mathbf{u}}, \underline{\mathbf{v}})_M := \underline{\mathbf{u}}^t M \underline{\mathbf{v}}$. We then define the projection error as

$$e_{vec}(\boldsymbol{\kappa}) := \frac{\|\underline{\mathbf{u}}_0^{(\boldsymbol{\kappa})} - \underline{\mathbf{U}}_0^{(\boldsymbol{\kappa})}\|_M}{\|\underline{\mathbf{u}}_0^{(\boldsymbol{\kappa})}\|_M},$$

where $\|\underline{\mathbf{u}}\|_M := \sqrt{\underline{\mathbf{u}}^t M \underline{\mathbf{u}}}$. We refer to this as the eigenvector error, since it is related to the accuracy of $\underline{\mathbf{U}}_{h,0}^{(\boldsymbol{\kappa},1)}$, which is an eigenvector of $M^{-1}A$ [52].

Since the physical plane wave, the mesh, and the set of basis functions are all translation invariant, we can efficiently compute this error by only considering $\mathbf{u}_{\Omega_0}^{(\boldsymbol{\kappa})}$, the part of $\mathbf{u}_0^{(\boldsymbol{\kappa})}$ restricted to cell Ω_0. We define $\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa})}$ to be the projection of $\mathbf{u}_{\Omega_0}^{(\boldsymbol{\kappa})}$ onto the discrete space restricted to Ω_0 and define $\underline{\mathbf{U}}_{\Omega_0}^{(\boldsymbol{\kappa})} \in \operatorname{span}\{\underline{\mathbf{U}}_{h,\Omega_0}^{(\boldsymbol{\kappa},1)}\}$ the projection of $\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa})}$ onto $\operatorname{span}\{\underline{\mathbf{U}}_{h,\Omega_0}^{(\boldsymbol{\kappa},1)}\}$ such that $(\underline{\mathbf{U}}_{\Omega_0}^{(\boldsymbol{\kappa})}, \underline{\mathbf{U}}_{h,\Omega_0}^{(\boldsymbol{\kappa},1)})_{M_0} = (\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa})}, \underline{\mathbf{U}}_{h,\Omega_0}^{(\boldsymbol{\kappa},1)})_{M_0}$, with $M_0 := M^{(\Omega_0)}$. We can then compute $e_{vec}(\boldsymbol{\kappa})$ by

$$e_{vec}(\boldsymbol{\kappa}) = \frac{\|\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa})} - \underline{\mathbf{U}}_{\Omega_0}^{(\boldsymbol{\kappa})}\|_{M_0}}{\|\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa})}\|_{M_0}}.$$

The complete procedure for computing $e_{vec}(\boldsymbol{\kappa})$ in the acoustic case is given by

A. Compute all eigenpairs $(s_h^{(\boldsymbol{\kappa},i)}, \underline{\mathbf{U}}_{h,\Omega_0}^{(\boldsymbol{\kappa},i)})$ of $S^{(\boldsymbol{\kappa})} := M_{inv}^{(\Omega_0)} A^{(\boldsymbol{\kappa})}$.

B. Compute $\omega_h^{(\boldsymbol{\kappa},i)} = \frac{1}{\Delta t} \arccos\left(\sum_{k=0}^{K} \frac{1}{(2k)!}(-\Delta t^2 s_h^{(\boldsymbol{\kappa},i)})^k\right)$, the angular velocities.

C. Compute the wave propagation speeds $c_h^{(\boldsymbol{\kappa},i)} = \omega_h^{(\boldsymbol{\kappa},i)}/|\boldsymbol{\kappa}|$ and order everything such that $|c - c_h^{(\boldsymbol{\kappa},1)}| \leq |c - c_h^{(\boldsymbol{\kappa},2)}| \leq \ldots$.

D. Compute $\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa})}$: the projection of $\mathbf{u}_{\Omega_0}^{(\boldsymbol{\kappa})}$ onto the discrete space of cell Ω_0.

E. Compute $\underline{\mathbf{U}}_{\Omega_0}^{(\boldsymbol{\kappa})}$: the projection of $\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa})}$ onto span$\{\underline{\mathbf{U}}_{h,\Omega_0}^{(\boldsymbol{\kappa},1)}\}$

F. Compute $e_{vec}(\boldsymbol{\kappa}) = \|\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa})} - \underline{\mathbf{U}}_{\Omega_0}^{(\boldsymbol{\kappa})}\|_{M_0} / \|\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa})}\|_{M_0}$.

For the isotropic elastic case, the procedure is very similar. Let $\boldsymbol{\kappa}$ be the wave vector and let $\mathbf{u}^{(\boldsymbol{\kappa})}$ denote the elastic plane wave of the form $\mathbf{u}^{(\boldsymbol{\kappa})}(\mathbf{x}, t) = \mathbf{a}e^{\hat{\imath}(\boldsymbol{\kappa}\cdot\mathbf{x}-\omega t)}$, with $\mathbf{a}$ the amplitude vector, $\omega = \pm c|\boldsymbol{\kappa}|$ the angular velocity, and c the elastic wave propagation speed. In the elastic isotropic case, we have to distinguish between longitudinal or primary waves, where $\mathbf{a}$ is parallel with $\boldsymbol{\kappa}$ and the propagation speed is $c = c_P = \sqrt{(\lambda + 2\mu)/\rho}$, and transversal, shear or secondary waves, where $\mathbf{a}$ is perpendicular to $\boldsymbol{\kappa}$ and the propagation speed is $c = c_S = \sqrt{\mu/\rho}$.

For the analysis, we will only consider the secondary wave, since the wavelength $\lambda = 2\pi/|\kappa| = 2\pi c/\omega$ of this wave is shorter and therefore governs the required mesh resolution. In 3D, there are two linear independent amplitude vectors, $\mathbf{a}^{(\boldsymbol{\kappa},1)}$ and $\mathbf{a}^{(\boldsymbol{\kappa},2)}$, that are perpendicular to $\boldsymbol{\kappa}$ and we will refer to the corresponding secondary plane waves as $\underline{\mathbf{u}}^{(\boldsymbol{\kappa},1)}$ and $\underline{\mathbf{u}}^{(\boldsymbol{\kappa},2)}$. We will compare these physical plane waves with the numerical plane waves in a similar way as for the acoustic case.

Since, for a given $\boldsymbol{\kappa}$ and $\omega = \pm c_S|\boldsymbol{\kappa}|$, there are two linearly independent secondary waves, we compare the secondary wave velocity $c = c_S$ with the wave propagation speed of the two best matching numerical plane waves. In particular, we define the dispersion error as

$$e_{disp}(\boldsymbol{\kappa}) = \frac{|c - c_h^{(\boldsymbol{\kappa},2)}|}{c}.$$

The procedures for computing this error is the same as for the acoustic case, with step d replaced by

d*. Compute $e_{disp}(\boldsymbol{\kappa}) = |c - c_h^{(\boldsymbol{\kappa},2)}|/c$.

The eigenvector is now computed by

$$e_{vec}(\boldsymbol{\kappa}) = \sup_{\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa})} \in \text{span}\{\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa},1)}, \underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa},2)}\}} \frac{\|\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa})} - \underline{\mathbf{U}}_{\Omega_0}^{(\boldsymbol{\kappa})}\|_{M_0}}{\|\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa})}\|_{M_0}},$$

where $\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa},i)}$ is the projection of $\mathbf{u}_{\Omega_0}^{(\boldsymbol{\kappa},i)}$ onto the discrete space of cell Ω_0, and $\underline{\mathbf{U}}_{\Omega_0}^{(\boldsymbol{\kappa})} \in \text{span}\{\underline{\mathbf{U}}_{h,\Omega_0}^{(\boldsymbol{\kappa},1)}, \underline{\mathbf{U}}_{h,\Omega_0}^{(\boldsymbol{\kappa},2)}\}$ is the projection of $\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa})} \in \text{span}\{\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa},1)}, \underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa},2)}\}$ onto span$\{\underline{\mathbf{U}}_{h,\Omega_0}^{(\boldsymbol{\kappa},1)}, \underline{\mathbf{U}}_{h,\Omega_0}^{(\boldsymbol{\kappa},2)}\}$. In other words, we compute the worst possible projection error for a linear combination of $\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa},1)}$ and $\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa},2)}$ projected onto the span of the two best-matching numerical plane waves $\underline{\mathbf{U}}_{h,\Omega_0}^{(\boldsymbol{\kappa},1)}$ and $\underline{\mathbf{U}}_{h,\Omega_0}^{(\boldsymbol{\kappa},2)}$. We can efficiently compute this by

$$e_{vec}(\boldsymbol{\kappa}) = \sqrt{\sigma_{max}(B^{-1}R)},$$

where $\sigma_{max}(B^{-1}R)$ denotes the largest eigenvalue of $B^{-1}R$ and $B, R \in \mathbb{R}^{2\times 2}$ are matrices given by $B_{ij} := (\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa},i)}, \underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa},j)})_{M_0}$ and $R_{ij} := (\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa},i)} - \underline{\mathbf{U}}_{\Omega_0}^{(\boldsymbol{\kappa},i)}, \underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa},j)} - \underline{\mathbf{U}}_{\Omega_0}^{(\boldsymbol{\kappa},j)})_{M_0}$, with $\underline{\mathbf{U}}_{\Omega_0}^{(\boldsymbol{\kappa},i)}$ the projection of $\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa},i)}$ onto the span of $\{\underline{\mathbf{U}}_{h,\Omega_0}^{(\boldsymbol{\kappa},1)}, \underline{\mathbf{U}}_{h,\Omega_0}^{(\boldsymbol{\kappa},2)}\}$.

The procedure for computing $e_{vec}(\boldsymbol{\kappa})$ is the same as for the acoustic case, with steps D-F replaced by

D*. Compute $\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa},i)}$: the projection of $\mathbf{u}_{\Omega_0}^{(\boldsymbol{\kappa},i)}$ onto the discrete space of cell Ω_0, for $i = 1, 2$.

E*. Compute $\underline{\mathbf{U}}_{\Omega_0}^{(\boldsymbol{\kappa},i)}$: the projection of $\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa},i)}$ onto span$\{\underline{\mathbf{U}}_{h,\Omega_0}^{(\boldsymbol{\kappa},1)}, \underline{\mathbf{U}}_{h,\Omega_0}^{(\boldsymbol{\kappa},2)}\}$, for $i = 1, 2$.

F*. Compute $B_{ij} := (\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa},i)}, \underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa},j)})_{M_0}$ and $R_{ij} := (\underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa},i)} - \underline{\mathbf{U}}_{\Omega_0}^{(\boldsymbol{\kappa},i)}, \underline{\mathbf{u}}_{\Omega_0}^{(\boldsymbol{\kappa},j)} - \underline{\mathbf{U}}_{\Omega_0}^{(\boldsymbol{\kappa},j)})_{M_0}$, for $i, j = 1, 2$, and use this to compute the eigenvector error $e_{vec}(\boldsymbol{\kappa}) = \sqrt{\sigma_{max}(B^{-1}R)}$.

So far, we only considered the dispersion error and eigenvector error for a given wave vector $\boldsymbol{\kappa}$. For a given wavelength $\lambda = 2\pi/|\boldsymbol{\kappa}|$, we define the dispersion and eigenvector error as the worst case among all wave vectors of length $|\boldsymbol{\kappa}| = \lambda/(2\pi)$, so among wave vectors in all possible directions:

$$e_{disp}(\lambda) := \sup_{\boldsymbol{\kappa}\in\mathbb{R}^3,\ |\boldsymbol{\kappa}|=\lambda/(2\pi)} e_{disp}(\boldsymbol{\kappa}), \tag{4.14a}$$

$$e_{vec}(\lambda) := \sup_{\boldsymbol{\kappa}\in\mathbb{R}^3,\ |\boldsymbol{\kappa}|=\lambda/(2\pi)} e_{vec}(\boldsymbol{\kappa}). \tag{4.14b}$$

We can use these errors to determine the required number of elements per wavelength. To compute these errors, we use a search algorithm, which requires the computation of $e_{disp}(\boldsymbol{\kappa})$ and $e_{vec}(\boldsymbol{\kappa})$ for a large number of wave vectors $\boldsymbol{\kappa}$. The complete procedure for computing the dispersion and eigenvector error is given by:

1. Construct a cell Ω_0 and subdivide it into tetrahedra.
2. Compute the submatrices $M^{(\Omega_0)}$ and $A^{(\Omega_0,\Omega_{\Delta\mathbf{k}})}$ for $\Delta\mathbf{k} \in \{-1, 0, 1\}^3$.

3. Compute $s_{h,max}$, given by (4.13). This is done with a search algorithm which requires the computation of $\sigma_{max}(S^{(\boldsymbol{\kappa})})$, with $S^{(\boldsymbol{\kappa})} := M_{inv}^{(\Omega_0)} A^{(\boldsymbol{\kappa})}$, for a large number of wave vectors $\boldsymbol{\kappa}$.

4. Compute $\Delta t \leq \sqrt{c_K / s_{h,max}}$, with c_K given by (4.12).

5. For a given wavelength λ, compute the errors $e_{disp}(\lambda)$ and $e_{vec}(\lambda)$ given in (4.14). For each λ, this requires the computation of $e_{disp}(\boldsymbol{\kappa})$ and $e_{vec}(\boldsymbol{\kappa})$, using steps a-d and A-F, for a large number of wave vectors $\boldsymbol{\kappa}$.

4.5.3 Estimating the computational cost

To compare the efficiency of the different methods, we also compute the number of degrees of freedom n_{vec}, the number of non-zero entries of the stiffness matrix n_{mat}, and the estimated computational cost n_{comp}, for each wavelength λ.

We define n_{vec} to be the number of degrees of freedom per λ^3-volume. This is computed by

$$n_{vec} = n_0 \frac{\lambda^3}{|\Omega_0|},$$

where n_0 is the number of basis functions corresponding to cell Ω_0, and $|\Omega_0|$ is the volume of Ω_0.

We define n_{mat} to be the number of non-zero entries of the stiffness matrix per λ^3-volume. In case of mass-lumping, we estimate this number by

$$n_{mat}^{(ML)} = \left(\sum_{q \in \mathcal{Q}_{\Omega_0}} \sum_{q' \in \mathcal{N}(q)} |\mathcal{U}_q||\mathcal{U}_{q'}| \right) \frac{\lambda^3}{|\Omega_0|},$$

where $|\mathcal{U}_q|$ is the number of degrees of freedom per node ($|\mathcal{U}_q| = 1$ in the acoustic and $|\mathcal{U}_q| = 3$ in the elastic case), $\mathcal{Q}_{\Omega_0}$ is the set of nodes on Ω_0, and $\mathcal{N}(q)$ are the neighbouring nodes of q that are connected with q through an element.

In case of the SIPDG method, we estimate this number by

$$n_{mat}^{(DG)} = \left(\sum_{e \in \mathcal{T}_{\Omega_0}} \sum_{e' \in \mathcal{N}(e)} |\mathcal{U}_e||\mathcal{U}_{e'}| \right) \frac{\lambda^3}{|\Omega_0|},$$

where $|\mathcal{U}_e|$ is the number of basis functions with support on element e, $\mathcal{T}_{\Omega_0}$ are the elements in Ω_0, and $\mathcal{N}(e)$ are the neighbouring elements of e that are connected with e through a face.

To estimate the computational cost we look at the size of the matrix times the number of matrix-vector products. The resulting estimates gives a rough estimate of the relative CPU time of the different methods, since it estimates the number of computations when using a globally assembled matrix.

We define the computational cost n_{comp} as the number of non-zero matrix entries per λ^3-volume times the number of matrix-vector products during one oscillation in time. The duration of one oscillation is $T_0 = \lambda/c$, with c the wave propagation speed. The number of matrix-vector products during one oscillation is the number of stages of the Lax–Wendroff scheme K times the number of time steps $N_{\Delta t} = T_0/\Delta t = \lambda/(c\Delta t)$, where $\Delta t = \sqrt{c_K/s_{h,max}}$, with c_K given by (4.12) and $s_{h,max}$ given by (4.13). We use this to compute n_{comp} as follows:

$$n_{comp} = n_{mat} K N_{\Delta t}.$$

4.6 Results and comparisons

Table 4.1: Analysed finite element methods

Method	Description
ML1	Linear mass-lumped finite element method
ML2	Degree-2 mass-lumped finite element method [51]
ML3a, ML3b	Degree-3 mass-lumped finite element methods [11]
DGX	Symmetric Interior Penalty Discontinous Galerkin method [36] of degree $X = 1, 2, 3$
DGXa	DGX with penalty term derived in [32] and given by (4.4a)
DGXb	DGX with penalty term derived in [55] and given by (4.4b)

An overview of the different finite element methods that we analyse is given in Table 4.1. Each method is combined with an order-$2p$ Lax–Wendroff time integration scheme, where p denotes the degree of the spatial discretization.

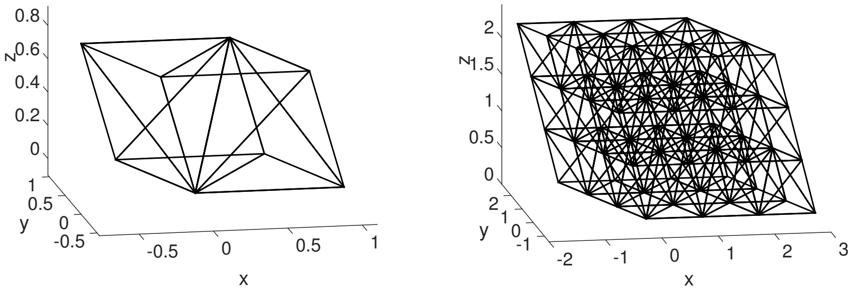


Figure 4.2: Tetragonal disphenoid honeycomb restricted to cell Ω_0 (left) and restricted to $3 \times 3 \times 3$ cells (right).

To analyse the dispersion properties of these methods, we use standard Fourier analysis, as explained in Section 4.5. We consider a periodic mesh of congruent nearly-regular equifacial tetrahedra, known as the tetragonal disphenoid honeycomb. To obtain this mesh, we slice the unit cell $\Omega_0 := [0, 1)^3$ into 6 tetrahedra with the planes $x = y$, $x = z$, and $y = z$ and then apply the linear transformation $\mathbf{x} \to \mathbf{T} \cdot \mathbf{x}$, with

$$\mathbf{T} := \begin{bmatrix} 1 & -1/3 & -1/3 \\ 0 & \sqrt{8/9} & -\sqrt{2/9} \\ 0 & 0 & \sqrt{2/3} \end{bmatrix}. \tag{4.15}$$

An illustration of this mesh is given in Figure 4.2.

4.6.1 Acoustic waves on a regular mesh

We first consider the acoustic wave model with $c = \rho = 1$. Figure 4.3 illustrates the dispersion and eigenvector error with respect to the number of elements per wavelength $N_E := \sqrt[3]{\lambda^3/|e|_{av}}$, with λ the wavelength and $|e|_{av}$ the average element volume. The eigenvector error for ML1 is always zero, since it has only one degree of freedom per cell $\Omega_{\mathbf{k}}$ and therefore allows only one numerical plane wave for a given wave vector. From this figure we can obtain the order of convergence, which is $2p$ for the dispersion error and $p + 1$ for the eigenvector error. These convergence rates are typical for symmetric finite element methods for eigenvalue problems, see, for example, [9] and the references therein. The $2p$-order superconvergence of the dispersion error is also in accordance with the results of [52, 3, 21].

By extrapolating the results shown in Figure 4.3 we can obtain approximations of the errors of the form $e = \alpha(N_E)^{-\beta}$, where α is the leading

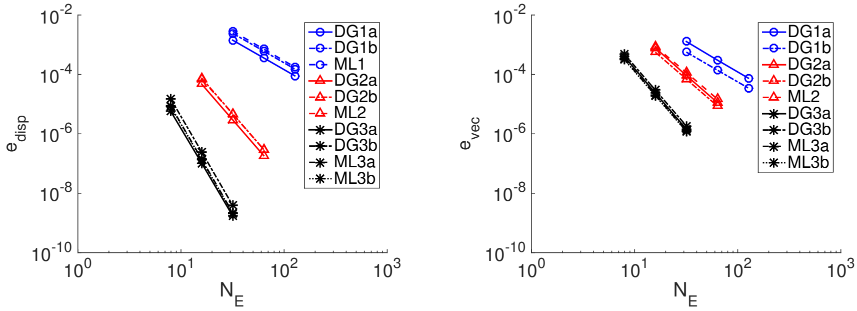


Figure 4.3: Dispersion error (left) and eigenvector error (right) for the acoustic wave model.

constant and β is the order of convergence. The approximations are given in Table 4.2.

Table 4.2: Approximation of the dispersion and eigenvector error for the acoustic case.

Method	e_{disp}	e_{vec}
DG1a	$1.45(N_E)^{-2}$	$1.20(N_E)^{-2}$
DG1b	$2.46(N_E)^{-2}$	$0.56(N_E)^{-2}$
ML1	$2.87(N_E)^{-2}$	0
DG2a	$3.00(N_E)^{-4}$	$2.89(N_E)^{-3}$
DG2b	$4.83(N_E)^{-4}$	$2.22(N_E)^{-3}$
ML2	$4.82(N_E)^{-4}$	$3.78(N_E)^{-3}$
DG3a	$1.77(N_E)^{-6}$	$1.46(N_E)^{-4}$
DG3b	$3.98(N_E)^{-6}$	$1.88(N_E)^{-4}$
ML3a	$2.25(N_E)^{-6}$	$1.26(N_E)^{-4}$
ML3b	$2.15(N_E)^{-6}$	$1.22(N_E)^{-4}$

We can use these results to obtain estimates for the number of elements per wavelength required for a given accuracy, but we can also use them to obtain other properties, such as the number of time steps or the computational cost required for a given accuracy. An overview for a dispersion error of 0.01 and 0.001 is given in Table 4.3 and 4.4, respectively, and the relation between the accuracy and the computational cost is illustrated in

Figure 4.4.

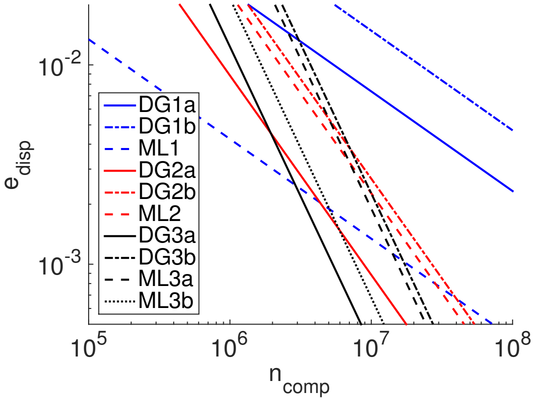


Figure 4.4: Dispersion error of different finite element methods for the acoustic wave model plotted against the estimated computational cost.

Figure 4.4 shows that for linear elements, the mass-lumped method ML1 is significantly more efficient than the DG methods DG1a and DG1b, while for quadratic elements, DG2a is significantly more efficient than ML2 and DG2b, and for cubic functions, DG3a is slightly more efficient than ML3b and significantly more efficient than DG3b and ML3a. In all cases, the DG methods using the sharper penalty term given by (4.4a) are significantly more efficient than those using the penalty term given by (4.4b). For a dispersion error of around 0.01 and higher, the linear mass-lumped method ML1 performs best in terms of computational cost, while for a dispersion error below 0.001 the best method is the DG method with cubic basis functions DG3a or the second degree-3 mass-lumped finite element method ML3b.

Tables 4.3 and 4.4 also show that for the case $p = 1$, the eigenvector error is always smaller than the dispersion error, but that for higher-order elements, the eigenvector error can become almost 5 times as large when the dispersion error is 0.01 and 10 times as large when the dispersion error is 0.001. This is due to the fact that the dispersion error converges with a faster rate (order $2p$) than the eigenvector error (order $p + 1$) for higher-degree methods.

Table 4.3: Number of elements per wavelength N_E, number of degrees of freedom n_{vec}, size of the global matrix n_{mat}, number of time steps $N_{\Delta t}$, computational cost n_{comp} and eigenvector error e_{vec} for a dispersion error of 0.01 for different finite element methods for the acoustic wave model. The numbers are accurate up to two decimal places.

	$e_{disp} = 0.01$					
Method	N_E	n_{vec}	n_{mat}	$N_{\Delta t}$	n_{comp}	e_{vec}
DG1a	12	7000	140×10^3	39	5.4×10^6	0.0083
DG1b	16	15000	310×10^3	72	22×10^6	0.0023
ML1	17	810	12×10^3	15	0.18×10^6	0
DG2a	4.2	720	36×10^3	12	0.88×10^6	0.040
DG2b	4.7	1000	51×10^3	26	2.7×10^6	0.022
ML2	4.7	860	39×10^3	29	2.3×10^6	0.037
DG3a	2.4	270	27×10^3	14	1.1×10^6	0.046
DG3b	2.7	400	40×10^3	31	3.7×10^6	0.035
ML3a	2.5	370	31×10^3	36	3.3×10^6	0.034
ML3b	2.4	360	30×10^3	18	1.7×10^6	0.034

4.6.2 The effect of mesh distortions

We also investigate the effect of the mesh quality on the dispersion error. To do this, we first create meshes of very flat elements by scaling the regular disphenoid mesh in the z-direction. After that, we create distorted meshes by displacing some of the vertices of the disphenoid honeycomb.

To create flat elements, we scale the disphenoid mesh in the z-direction by a factor T_z. The effect on the dispersion error is illustrated in Figure 4.5. For a mesh flattened by a factor 2, the dispersion error does not grow more than a factor 2.5, but flattening the mesh by a factor 10 increases the error by a factor between 10 and 100. In all cases, the mesh resolution remains the same and even becomes smaller in the z-direction. This means that the mesh quality can have a strong effect on the accuracy of the method and that using flat tetrahedra can significantly reduce the accuracy. The methods using lower-order elements are more sensitive to the mesh quality than the higher-order methods.

To create distorted meshes, we displace some of the vertices of the disphenoid mesh. In particular, we create a distorted mesh using the fol-

Table 4.4: Same as Table 4.3, but for a dispersion error of 0.001.

	$e_{disp} = 0.001$					
Method	N_E	n_{vec}	n_{mat}	$N_{\Delta t}$	n_{comp}	e_{vec}
DG1a	38	220000	4400×10^3	120	540×10^6	0.00083
DG1b	50	490000	9700×10^3	230	2200×10^6	0.00023
ML1	54	26000	390×10^3	47	18×10^6	0
DG2a	7.4	4100	200×10^3	22	8.8×10^6	0.0071
DG2b	8.3	5800	290×10^3	46	27×10^6	0.0038
ML2	8.3	4800	220×10^3	52	23×10^6	0.0065
DG3a	3.5	840	84×10^3	21	5.3×10^6	0.010
DG3b	4.0	1300	130×10^3	46	17×10^6	0.0075
ML3a	3.6	1200	98×10^3	52	15×10^6	0.0074
ML3b	3.6	1100	96×10^3	27	7.7×10^6	0.0073

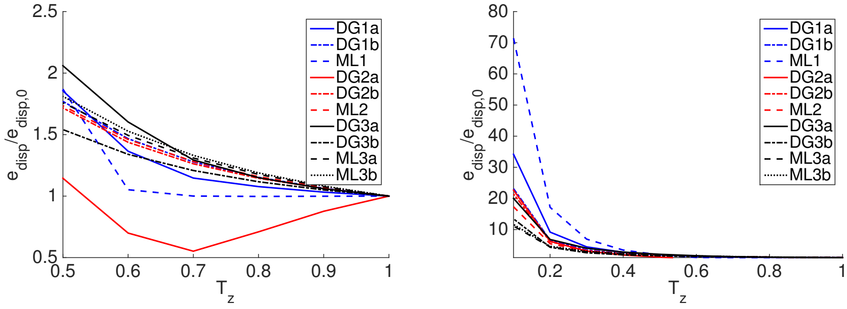


Figure 4.5: Relative dispersion error for the disphenoid mesh scaled in the z-direction by a factor T_z for the acoustic wave model. Here, $e_{disp,0}$ denotes the error for the original mesh.

lowing steps:

1. Slice the cube $[0, 0.5)^3$ into 6 tetrahedra with the planes $x = y$, $x = z$, $y = z$.

2. Repeat this pattern $2 \times 2 \times 2$ times to pack the unit cell $[0, 1)$ with 48 tetrahedra.

3. Displace the central node by moving it from $(0.5, 0.5, 0.5)$ to $\big(0.5(1+$

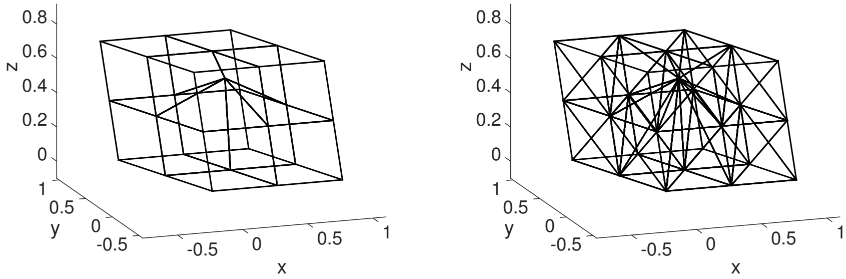


Figure 4.6: Repeated subcells with a small distortion (left) and corresponding tetrahedral mesh (right).

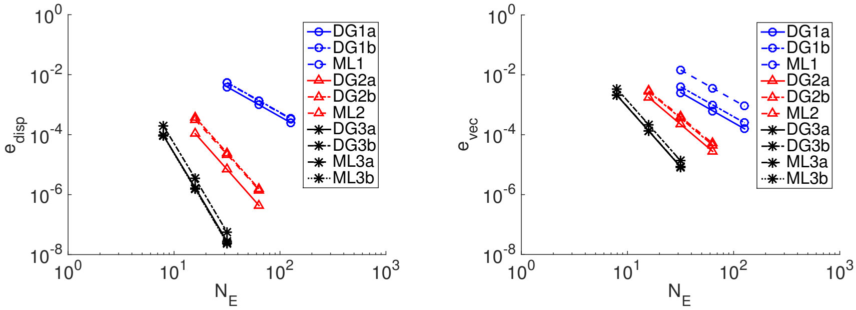


Figure 4.7: Dispersion error (left) and eigenvector error (right) for the acoustic wave model for a distorted mesh with distortion $\delta = 0.9$.

$\delta), 0.5(1+\delta), 0.5(1+\delta)\big)$, where $\delta \in [0,1)$ denotes the size of the distortion.

4. Apply the transformation $\mathbf{x} \to \mathbf{T} \cdot \mathbf{x}$, with $\mathbf{T}$ defined as in (4.15).

In case of zero distortion, $\delta = 0$, we obtain the original disphenoid honeycomb, scaled by a factor 0.5. When the distortion δ approaches 1, some of the elements become completely flat with zero volume.

An illustration of the mesh with distortion $\delta = 0.4$ is given in Figure 4.6. In Figure 4.7, the dispersion and eigenvector error are plotted against the number of elements per wavelength for a heavily distorted mesh with $\delta = 0.9$. These results show that the order of convergence remains $2p$ for the dispersion and $p+1$ for the eigenvector error, even though the mesh is distorted. The distortion does, however, affect the leading constant of the errors. The effect of the mesh distortion on the dispersion error is

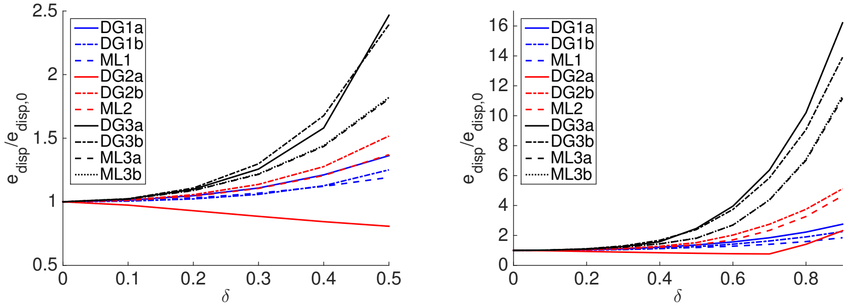


Figure 4.8: Relative dispersion error for meshes with a distortion δ. Here, $e_{disp,0}$ denotes the error of the regular mesh with $\delta = 0$.

illustrated in Figure 4.8. Again, the accuracy is not significantly affected by small distortions, but large distortions can reduce the accuracy by an order of magnitude.

4.6.3 Elastic waves and the effect of the P/S-wave velocity ratio

Besides the acoustic wave model, we also consider the isotropic elastic wave model. Figure 4.9 illustrates the dispersion and eigenvector error with respect to the number of elements per wavelength for the isotropic elastic wave model with $\mu = \rho = 1$ and $\lambda = 2$, so with a P/S-wave velocity ratio of 2. Again, the order of convergence is $2p$ for the dispersion error and $p + 1$ for the eigenvector error.

By extrapolating these results we can again obtain approximations of the errors of the form $e = \alpha(N_E)^{-\beta}$, which are given in Table 4.5. Figure 4.10 illustrates the relation between the dispersion error and the computational cost, based on these results. The relative performance of the different methods is similar to the acoustic case.

We also look at the influence of the P/S-wave velocity ratio c_P/c_S on the dispersion error, where $c_S = \sqrt{\mu}$ denotes the S-wave velocity and $c_P = \sqrt{\lambda + 2\mu}$ denotes the P-wave velocity. This relation is illustrated in Figure 4.11. This figure shows that the DG methods are not really sensitive to the c_P/c_S ratio, since the dispersion error never grows more than a factor 1.5. The higher-order mass-lumped methods are slightly more sensitive, with a dispersion error becoming around 3 times as large for $c_P/c_S = 10$, compared to $c_P/c_S = 2$, while the linear mass-lumped

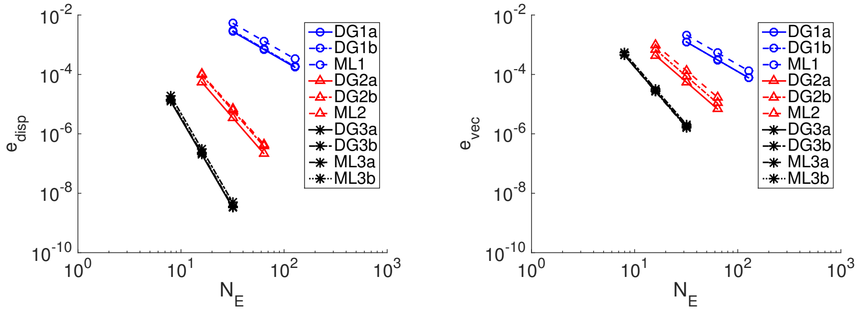


Figure 4.9: Dispersion error (left) and eigenvector error (right) for the isotropic elastic wave model with a P/S-wave velocity ratio of 2.

Table 4.5: Approximation of the dispersion and eigenvector error for the elastic wave model with a P/S-wave velocity ratio of 2.

Method	e_{disp}	e_{vec}
DG1a	$2.81(N_E)^{-2}$	$1.25(N_E)^{-2}$
DG1b	$3.00(N_E)^{-2}$	$1.25(N_E)^{-2}$
ML1	$5.39(N_E)^{-2}$	$2.16(N_E)^{-2}$
DG2a	$3.55(N_E)^{-4}$	$1.76(N_E)^{-3}$
DG2b	$6.20(N_E)^{-4}$	$2.77(N_E)^{-3}$
ML2	$7.29(N_E)^{-4}$	$4.39(N_E)^{-3}$
DG3a	$3.32(N_E)^{-6}$	$1.79(N_E)^{-4}$
DG3b	$5.04(N_E)^{-6}$	$2.11(N_E)^{-4}$
ML3a	$3.63(N_E)^{-6}$	$1.66(N_E)^{-4}$
ML3b	$3.58(N_E)^{-6}$	$1.69(N_E)^{-4}$

method is very sensitive, with a dispersion error becoming almost 40 times as large in this case.

4.7 Conclusions

We analysed the dispersion properties of two types of explicit finite element methods for modelling wave propagation on tetrahedral meshes, namely mass-lumped finite elements methods and symmetric interior penalty discontinuous Galerkin (SIPDG) methods, both for degrees $p = 1, 2, 3$ and

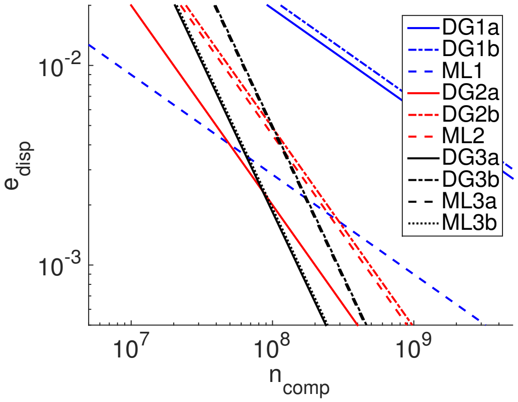


Figure 4.10: Dispersion error of different finite element methods for the isotropic elastic wave model with a P/S-wave velocity ratio of 2, plotted against the estimated computational cost. The graphs of DG3a and ML3b and of DG3b and ML3a are almost identical.

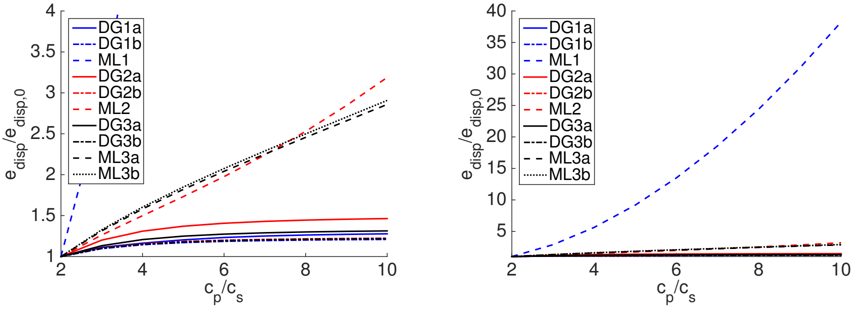


Figure 4.11: Relative dispersion error for the isotropic elastic wave model with different c_P/c_S ratios. Here, $e_{disp,0}$ denotes the error for the original mesh with $c_P/c_S = 2$.

combined with an order-$2p$ Lax–Wendroff time integration method. The analysed methods are listed in Table 1.

The dispersion properties are obtained semi-analytically using standard Fourier analysis. We used this to give an indication of which method is the most efficient for a given accuracy, how many elements per wavelength are required for a given accuracy, and how sensitive the accuracy of the method is to poorly shaped elements and high P/S-wave velocity ratios.

Based on the results we draw the following conclusions with regard to efficiency:

- The linear mass-lumped method is the most efficient method for a dispersion error of around 1% when using approximately regular tetrahedra. Heavily distorted elements, however, can significantly reduce its accuracy.

- The degree-3 SIPDG method, with the penalty term derived in [32] and given by (4.4a), and the second degree-3 mass-lumped finite element method of [11] are the most efficient methods for a dispersion error of around 0.1% and less.

- The SIPDG methods using the sharper penalty term bound derived in [32] are significantly more efficient than those using the penalty term of [55], which is based on the trace inequality of [79].

The required number of elements for a given accuracy can be obtained from the approximations given in Tables 4.2 and 4.5. We also draw the following conclusions with regard to accuracy:

- Higher-order methods suffer more from spurious modes for the same dispersion error. This is due to the fact that for higher-order methods, the convergence rate of the dispersion error, $2p$, is larger than the convergence rate of the eigenvector, $p+1$.

- All methods are significantly affected by a poor mesh quality, although lower-order methods are more sensitive to this than higher-order methods. Flattening the tetrahedra by a factor 10 reduces the accuracy of the methods by 1-2 orders of magnitude, even though the mesh resolution remains the same and even improves in one direction.

- The SIPDG methods are not really sensitive to high P/S-wave velocity ratios, while the accuracy of the higher-order mass-lumped methods reduces slightly when the P/S-wave velocity ratio is increased. The accuracy of the linear mass-lumped method, however, reduces by an order of magnitude when the P/S-wave velocity ratio is raised from 2 to 10.

4.A Stability of the Lax–Wendroff method

Theorem 4.A.1. *Consider the following time integration scheme:*

$$U(t_{i+1}) = -U(t_{i-1}) + 2\beta U(t_i), \qquad i = 1, 2, \dots,$$

where $\beta \in \mathbb{R}$ is a constant and $\{U(t_i)\}_{i\geq 0}$ is a sequence of scalars representing a scalar variable $u(t)$ at time slots $t_i = i\Delta t$, with Δt the time step size. This scheme is stable, by which we mean that the solution grows at most linearly in time, iff $|\beta| \leq 1$.

Proof. If $\beta \in (-1, 1)$, then the two independent solutions of the time integration scheme are given by $U(t_n) = e^{\pm \hat{\imath}(t_n\omega)}$, where ω satisfies $\cos(\omega\Delta t) = \beta$ and $\hat{\imath} := \sqrt{-1}$ is the imaginary number. Otherwise, if $\beta = 1$ (or $\beta = -1$), then the two independent solutions are given by $U(t_n) = 1, n$ (or $U(t_n) = (-1)^n, n(-1)^n$). Finally, if $\beta \geq 1$ (or $\beta < -1$), then the two independent solutions are given by $U(t_n) = e^{\pm t_n\omega}$ (or $U(t_n) = -e^{\pm t_n\omega}$), where ω satisfies $\cosh(\omega\Delta t) = \beta$ (or $-\cosh(\omega\Delta t) = \beta$). Therefore, the scheme grows at most linearly in time iff $\beta \in [-1, 1]$. □

Theorem 4.A.2. *Consider the order-$2K$ Lax–Wendroff time integration method given by*

$$\underline{\mathbf{U}}(t_{i+1}) = -\underline{\mathbf{U}}(t_{i-1}) + 2\sum_{k=0}^{K} \frac{1}{(2k)!}\Delta t^{2k}(-M^{-1}A)^k\underline{\mathbf{U}}(t_i), \quad i = 1, 2, \ldots,$$

where M and A are symmetric positive definite matrices, and $\{\underline{\mathbf{U}}(t_i)\}_{i\geq 0}$ is a sequence of vectors representing a vector variable $\underline{\mathbf{u}}(t)$ at time slots $t_i = i\Delta t$, with Δt the time step size. This scheme is stable, in the sense that the solution grows at most linearly in time, if $\Delta t \leq \sqrt{c_K/\sigma_{max}(M^{-1}A)}$, where $\sigma_{max}(M^{-1}A)$ denotes the spectral radius of $M^{-1}A$ and c_K is defined as

$$c_K := \inf\left\{x \geq 0 \mid \left|\sum_{k=0}^{K} \frac{1}{(2k)!}(-x)^k\right| > 1\right\}.$$

Proof. We can rewrite the time integration scheme as

$$\underline{\mathbf{U}}(t_{i+1}) = -\underline{\mathbf{U}}(t_{i-1}) + 2B\underline{\mathbf{U}}(t_i),$$

where $B := \sum_{k=0}^{K} \frac{1}{(2k)!}\Delta t^{2k}(-M^{-1}A)^k$. Since M and A are symmetric positive definite, we can diagonalise $M^{-1}A$ as VDV^{-1}, with D a diagonal matrix with only positive real values on the diagonal. We can then diagonalise B as $B = V\left(\sum_{k=0}^{K} \frac{1}{(2k)!}\Delta t^{2k}(-D)^k\right)V^{-1}$. Using this diagonalisation we can decouple the matrix-vector equations into scalar equations of the form

$$U(t_{i+1}) = -U(_{i-1}) + 2\beta U(t_i), \qquad i = 1, 2, \ldots,$$

with

$$\beta = \sum_{k=0}^{K} \frac{1}{(2k)!}(-s\Delta t^2)^k \qquad \text{for some eigenvalue } s \text{ of } M^{-1}A.$$

From the definition of c_K, it follows that $|\beta| \leq 1$ for all possible β, if $\Delta t^2 \sigma_{max}(M^{-1}A) \leq c_K$, so if $\Delta t \leq \sqrt{c_K/\sigma_{max}(M^{-1}A)}$. From Theorem 4.A.1 it then follows that this scheme is stable. □

Remark 4.A.3. *The values of c_K can be computed numerically. For example, $c_K = 4, 12, 7.57$ for $K = 1, 2, 3$, respectively.*

Chapter 5

New Higher-Order Mass-Lumped Tetrahedral Elements for Wave Propagation Modelling[1]

Abstract

We present a new accuracy condition for the construction of continuous mass-lumped elements. This condition is less restrictive than the one currently used and enabled us to construct new mass-lumped tetrahedral elements of degrees 2 to 4. The new degree-2 and degree-3 tetrahedral elements require 15 and 32 nodes per element, respectively, while currently, these elements require 23 and 50 nodes, respectively. The new degree-4 elements require 60, 61, or 65 nodes per element. Tetrahedral elements of this degree had not been found until now. We prove that our accuracy condition results in a mass-lumped finite element method that converges with optimal order in the L^2-norm and energy-norm. A dispersion analysis and several numerical tests confirm that our elements maintain the optimal order of accuracy and show that the new mass-lumped tetrahedral elements are more efficient than the current ones.

5.1 Introduction

Wave propagation modelling has many applications in the fields of structural mechanics, electromagnetism and geosciences. In many of these applications, waves need to be modelled on a large and complex 3D geometry that requires a fast and robust numerical algorithm.

[1] Accepted for publication as: Geevers, S., Mulder, W.A. & van der Vegt, J.J.W. (2018). New Higher-Order Mass-Lumped Tetrahedral Elements for Wave Propagation Modelling. *SIAM Journal on Scientific Computing.*

The oldest and most popular algorithm is the finite difference method, which approximates the wave field on a uniform grid. This method is relatively easy to implement and is very efficient on simple geometries. However, its accuracy quickly deteriorates if the grid points are not aligned with sharp material interfaces and boundaries of the domain. A good alignment is often not possible with uniform grids.

Unstructured meshes, on the other hand, offer more geometric flexibility and can be properly aligned with many complex geometries. Such meshes can be used with finite element methods. While more difficult to implement and requiring more computations, the finite element method can remain accurate on very complex geometries when using a proper mesh. When applied with mass lumping, the finite element method can in such cases become more efficient than the finite difference method [81].

Mass lumping is important for applying the finite element method to wave propagation problems, since it allows for explicit time-stepping. When using an explicit time integration scheme, the finite element method requires the solution of a linear system $Mx = b$, with M the mass matrix, at every time step. When using the classical finite element method, the mass matrix is large and sparse, but not (block)-diagonal. This makes the numerical scheme very inefficient for large-scale simulations. Mass lumping avoids this problem by lumping the mass matrix M into a diagonal matrix. Usually, this is done with nodal basis functions and an inexact quadrature rule for M with quadrature points that coincide with the basis functions nodes.

For quadrilaterals and hexahedra, mass lumping is relatively straightforward and is accomplished by using tensor product basis functions and Gauss–Lobatto quadrature points. The resulting method is known as the spectral element method. Quadrilaterals and hexahedra, however, offer less geometric flexibility than triangles and tetrahedra.

For linear triangular and tetrahedral elements, mass lumping is done using standard Lagrangian basis functions and a Newton–Cotes integration rule. For higher-degree triangular and tetrahedral elements, however, this approach results in instabilities, a singular mass matrix, or a suboptimal convergence rate. The Newton–Cotes rule for quadratic triangular elements, for example, has zero weights at the vertices, resulting in a singular mass matrix. This can be resolved by enriching the quadratic element space with a cubic bubble function that vanishes on all edges and by adding an additional node at the centre of the triangle [29]. By enriching the element space with higher-degree bubble functions and combining it with a suitable quadrature rule, mass-lumped triangular elements were also obtained for degrees 3 [16, 15], 4 [51], 5 [11], 6 [53], and 7 to 9 [49, 17]. For tetrahedra, mass lumping can be accomplished in a similar way by adding higher-degree face and internal bubble functions to the element space. So far, this has resulted in mass-lumped tetrahedral elements of degrees 2 [51] and 3 [11].

In this chapter, we show that the accuracy condition that was imposed on the quadrature rules of these higher-degree triangular and tetrahedral mass-lumped elements is too strong. This condition is that the quadrature rule of a degree-p element should be exact for polynomials up to degree $p+p'-2$ [12], where $p' > p$ is the highest polynomial degree of the functions in the enriched element space. Instead, we show that for $p \geq 2$ the quadrature rule only needs to be exact for functions in $\tilde{U} \otimes \mathcal{P}_{p-2}$, with $\tilde{U}$ the enriched element space and $\mathcal{P}_{p-2}$ the set of polynomials up to degree $p-2$. We prove that by satisfying this condition, the finite element method can maintain an optimal order of convergence in the L^2-norm and energy-norm.

This new accuracy condition enabled us to develop several new mass-lumped tetrahedral elements of degrees 2 to 4. The new elements of degrees 2 and 3 require 15 and 32 nodes per element, respectively, while the current versions require 23 and 50 nodes, respectively. Our degree-4 elements require 60, 61 or 65 nodes. Mass-lumped tetrahedral elements of this degree had not been found until now. A dispersion analysis and various numerical tests confirm the optimal order of convergence of these methods and show that the new mass-lumped tetrahedral elements are significantly more efficient than the current ones.

Although this chapter focuses on wave propagation problems, more generally, mass lumping is useful for solving any type of evolution problem that requires explicit time-stepping. It is also useful for efficiently computing higher-order derivatives, which appear, for example, in the Korteweg–de Vries equation [50].

This chapter is constructed as follows: In Section 5.2, we present the scalar wave equation and the classical finite element method. In Section 5.3, we explain mass lumping. The stability is analyzed in Section 5.3.4. In Section 5.4, we present our new accuracy condition for the quadrature rule for the mass matrix and prove that, if this condition is satisfied, the mass-lumped finite element method can maintain an optimal order of convergence. This condition enabled us to derive several new mass-lumped tetrahedral elements of degrees 2 to 4, presented in Section 5.5. We analyze the dispersion properties of these new methods in Section 5.6 and test the methods numerically in Section 5.7. In both sections we compare the new methods with existing finite element methods. Finally, we present our main conclusions in Section 5.8.

5.2 The scalar wave equation and classical finite element method

In this chapter, we mainly focus on the scalar wave equation, which serves as a model problem for more complex wave problems such as the elastic wave equations and Maxwell's equations. Let $\Omega \subset \mathbb{R}^3$ be a 3D open bounded domain, with Lipschitz boundary $\partial\Omega$, and let $(0, T)$ be the time domain. The scalar wave equation can be written as

$$\begin{aligned} \rho\partial_t^2 u &= \nabla \cdot c\nabla u + f && \text{in } \Omega \times (0,T), && (5.1\text{a})\\ u &= 0 && \text{on } \partial\Omega, && (5.1\text{b})\\ u|_{t=0} &= u_0 && \text{in } \Omega, && (5.1\text{c})\\ \partial_t u|_{t=0} &= v_0 && \text{in } \Omega, && (5.1\text{d}) \end{aligned}$$

where $u : \Omega \times (0,T) \to \mathbb{R}$ is the unknown scalar field, ∇ is the gradient operator, $\rho, c : \Omega \to \mathbb{R}^+$ are positive scalar fields, and $f : \Omega \times (0,T) \to \mathbb{R}$ is the source term. We assume that the parameters ρ and c are bounded by $\rho_0 \leq \rho \leq \rho_1$ and $c_0 \leq c \leq c_1$ for some positive scalars $\rho_0, \rho_1, c_0, c_1 \in \mathbb{R}^+$.

This equation can be solved with the finite element method, which is based on the weak formulation of (5.1). Assume the initial conditions satisfy $u_0 \in H_0^1(\Omega)$ and $v_0 \in L^2(\Omega)$ and assume the source term satisfies $f \in L^2\big(0,T; L^2(\Omega)\big)$. Here, $L^2(\Omega)$ denotes the space of square integrable functions on Ω, H_0^1 denotes the Sobolev space of functions on Ω that are zero on $\partial\Omega$ and have square integrable weak derivatives, and $L^2(0,T;U)$, with U a Banach space, denotes the Bochner space consisting of functions $f : (0,T) \to U$ such that $\|f(t)\|_U$ is square integrable in $(0,T)$. The weak formulation of (5.1) is finding $u \in L^2\big(0,T; H_0^1(\Omega)\big)$, with $\partial_t u \in L^2\big(0,T; L^2(\Omega)\big)$ and $\partial_t(\rho\partial_t u) \in L^2\big(0,T; H^{-1}(\Omega)\big)$, such that $u|_{t=0} = u_0$, $\partial_t u|_{t=0} = v_0$, and

$$\langle \partial_t(\rho\partial_t u), w\rangle + a(u,w) = (f,w) \quad \text{for all } w \in H_0^1(\Omega), \text{ a.e. } t \in (0,T). \quad (5.2)$$

Here, $\langle \cdot,\cdot \rangle$ denotes the pairing between $H^{-1}(\Omega)$ and $H_0^1(\Omega)$, $(\cdot,\cdot)$ denotes the $L^2(\Omega)$ inner product, and $a(\cdot,\cdot) : H_0^1(\Omega) \times H_0^1(\Omega) \to \mathbb{R}$ is the elliptic operator given by

$$a(u,w) := \int_\Omega c\nabla u \cdot \nabla w \, dx.$$

Because of the boundedness of ρ, it follows that the norm $\|u\|_\rho^2 := (\rho u, u)$ is equivalent to the standard $L^2(\Omega)$-norm. It can then be proven, in a way analogous to [46, Chapter 3, Theorem 8.1], that (5.2) is well-posed and has a unique solution.

The solution of (5.2) can be approximated by the finite element method. Let $\mathcal{T}_h$ be a tetrahedral tessellation of Ω, with h the diameter of the smallest sphere that can contain each element in $\mathcal{T}_h$, and let U_h denote the finite element space consisting of continuous functions that are polynomial of degree at most p when restricted to a single element:

$$U_h = \{u \in H_0^1(\Omega) \mid u|_e \in \mathcal{P}_p(e) \text{ for all } e \in \mathcal{T}_h\},$$

where $\mathcal{P}_p$ denotes the set of all polynomials of degree p or less. The classical conforming finite element method is finding $u_h : [0,T] \to U_h$, such that $u_h|_{t=0} = \Pi_h u_0$, $\partial_t u_h|_{t=0} = \Pi_h v_0$, and

$$(\rho \partial_t^2 u_h, w) + a(u_h, w) = (f, w) \qquad \text{for all } w \in U_h, \text{ a.e. } t \in (0,T), \tag{5.3}$$

where $\Pi_h : L^2(\Omega) \to U_h$ is the weighted L^2 projection operator defined such that $(\rho \Pi_h u, w) = (\rho u, w)$ for all $w \in U_h$.

This can be rewritten as a set of ODEs using a linear basis $\{w_i\}_{i=1}^n$ of U_h. For any function $u \in U_h$ we define $\underline{\mathbf{u}} \in \mathbb{R}^n$ as the vector of coefficients such that $u = \sum_{i=1}^n \underline{u}_i w_i$. The finite element method can then be formulated as solving $\underline{\mathbf{u}}_h : [0,T] \to \mathbb{R}^n$, such that $\underline{\mathbf{u}}_h|_{t=0} = \underline{\mathbf{\Pi_h u_0}}$, $\partial_t \underline{\mathbf{u}}_h|_{t=0} = \underline{\mathbf{\Pi_h v_0}}$, and

$$M \partial_t^2 \underline{\mathbf{u}}_h + A \underline{\mathbf{u}}_h = \underline{\mathbf{f}}^* \qquad \text{for a.e. } t \in (0,T), \tag{5.4}$$

where $M, A \in \mathbb{R}^{n \times n}$ are the mass matrix and stiffness matrix, respectively, given by $M_{ij} := (\rho w_i, w_j)$, $A_{ij} := a(w_i, w_j)$ for all $i, j = 1, \ldots, n$, and $\underline{\mathbf{f}}^* \in L^2(0,T;\mathbb{R}^n)$ is the source vector, given by $\underline{\mathbf{f}}_i^* := (f, w_i)$, for $i = 1, \ldots, n$, a.e. $t \in (0,T)$.

When using an explicit time integration scheme, a system of the form $M\underline{\mathbf{x}} = \underline{\mathbf{b}}$ needs to be solved at every time step. Typically, the mass matrix M is large and sparse, but not (block)-diagonal, resulting in a very inefficient numerical scheme. A diagonal mass matrix can be obtained by a technique known as mass lumping. We will discuss this in the next section.

5.3 Mass lumping

Mass lumping is usually done with nodal basis functions and an inexact quadrature rule for the mass matrix. A diagonal matrix is obtained when the integration points coincide with the nodes of the basis functions. However, when using elements of degree $p \geq 2$, this technique does not result in a stable and accurate finite element scheme. For example, for standard quadratic Lagrangian basis functions combined with a Newton–Cotes quadrature rule, the weights at the vertices of the quadratic tetrahedral element become negative, resulting in unstable modes.

To overcome such problems, the elements are enriched with higher-degree face and interior bubble functions. These enriched elements are still affine-equivalent to a reference element $\tilde{e}$. We can therefore write the discrete space in the form

$$U_h = H_0^1(\Omega) \cap U(\mathcal{T}_h, \tilde{U}),$$

where

$$U(\mathcal{T}_h, \tilde{U}) := \{u \in H^1(\Omega) \mid u \circ \phi_e \in \tilde{U} \text{for all } e \in \mathcal{T}_h\},$$

with $\phi_e : \tilde{e} \to e$ the reference-to-physical element mapping, and $\tilde{U}$ the reference space. If $\tilde{U} = \mathcal{P}_p(\tilde{e})$ we obtain the standard elements of degree p. To obtain enriched elements, we set $\tilde{U} = \mathcal{P}_p(\tilde{e}) \oplus \tilde{U}^+ := \{u \mid u = w + u^+ \text{ for some } w \in \mathcal{P}_p(\tilde{e}), u^+ \in \tilde{U}^+\}$, with $\tilde{U}^+$ a space of higher-degree face and interior bubble functions.

A nodal basis and quadrature rule for U_h can be constructed from a nodal basis and quadrature rule for the reference space $\tilde{U}$. In the next two subsections we will discuss this in more detail.

5.3.1 Nodes and nodal basis functions

A nodal basis for a space U_h consists of a set of nodes $\mathcal{Q}_h$ and corresponding basis functions $\{w_{\mathbf{x}}\}_{\mathbf{x} \in \mathcal{Q}_h}$, such that $\operatorname{span}\{w_{\mathbf{x}}\}_{\mathbf{x} \in \mathcal{Q}_h} = U_h$ and $w_{\mathbf{x}}(\mathbf{y}) = \delta_{\mathbf{xy}}$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{Q}_h$, where $\delta_{\mathbf{xy}}$ denotes the Kronecker delta. This means that each basis function equals one at one particular node and zero at all the other nodes.

A common way to construct such a nodal basis for the space $U(\mathcal{T}_h, \tilde{U})$ is using a nodal basis $\{\tilde{w}_{\tilde{\mathbf{x}}}\}_{\tilde{\mathbf{x}} \in \tilde{\mathcal{Q}}}$ for the reference space $\tilde{U}$. The element nodes $\mathcal{Q}_e$ are obtained by mapping the reference nodes to the physical element: $\mathcal{Q}_e := \{\phi_e(\tilde{\mathbf{x}})\}_{\tilde{\mathbf{x}} \in \tilde{\mathcal{Q}}}$. The nodal basis functions of this element, $\{w_{e,\mathbf{x}}\}_{\mathbf{x} \in \mathcal{Q}_e}$, are obtained by mapping the reference basis functions to the physical element. We can write these functions as $w_{e,\mathbf{x}} := \tilde{w}_{\phi_e^{-1}(\mathbf{x})} \circ \phi_e^{-1}$. The set of global nodes $\mathcal{Q}_h$ is the union of all element nodes and the corresponding global basis functions are obtained by concatenating the corresponding element basis functions. Formally, we define the global nodal basis functions $\{w_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{Q}_h}$ as follows:

$$w_{\mathbf{x}}|_e := \begin{cases} \tilde{w}_{\phi_e^{-1}(\mathbf{x})} \circ \phi_e^{-1}, & e \in \mathcal{T}_{\mathbf{x}}, \\ 0, & \text{otherwise}, \end{cases} \tag{5.5}$$

for all $\mathbf{x} \in \mathcal{Q}_h$, where $\mathcal{T}_{\mathbf{x}}$ denotes the set of elements containing or adjacent to $\mathbf{x}$. To ensure that these global basis functions are well defined and continuous, we need to impose the following additional conditions on $\tilde{\mathcal{Q}}$ and $\{\tilde{w}_{\tilde{\mathbf{x}}}\}_{\tilde{\mathbf{x}}\in\tilde{\mathcal{Q}}}$:

$$\tilde{w}_{\tilde{\mathbf{x}}}|_{\tilde{f}} = 0 \qquad\qquad \text{for all } \tilde{f} \in \tilde{\mathcal{F}}, \tilde{\mathbf{x}} \in \tilde{\mathcal{Q}} \setminus \tilde{f}, \tag{5.6}$$

and

$$\tilde{\mathcal{Q}} = s(\tilde{\mathcal{Q}}) \qquad\qquad \text{for all } s \in \mathcal{S}, \tag{5.7a}$$

$$\tilde{w}_{\tilde{\mathbf{x}}} = \tilde{w}_{s(\tilde{\mathbf{x}})} \circ s \qquad\qquad \text{for all } s \in \mathcal{S}, \tag{5.7b}$$

where $\tilde{\mathcal{F}}$ is the set of reference faces and $\mathcal{S}$ is the set of all affine mappings that map $\tilde{e}$ onto itself. Condition (5.6) implies that if a basis function is zero at the nodes on a face, then it should be zero on the entire face, and condition (5.7) implies that the set of element nodes and basis functions are symmetric and do not depend on the choice of ϕ_e. A proof that $\{w_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{Q}_h}$ is indeed a set of well-defined and continuous nodal basis functions is given in Lemma 5.A.2 and Theorem 5.A.3.

It remains to incorporate the Dirichlet boundary condition $u_h|_{\partial\Omega} = 0$. If $u_h \in U(\mathcal{T}_h, \tilde{U}) = \operatorname{span}\{w_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{Q}_h}$, then, because of (5.6), this condition is satisfied when $u_h = 0$ at all nodes on $\partial\Omega$. A nodal basis for U_h therefore consists of all interior nodes $\mathcal{Q}_h \setminus \partial\Omega$ and corresponding basis functions $\{w_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{Q}_h\setminus\partial\Omega}$.

5.3.2 Quadrature rule

To obtain a diagonal mass matrix, we approximate the integrals with an inexact quadrature rule with integration points that coincide with the nodes of the nodal basis.

Let $\mathcal{Q}_e := \{\phi_e(\tilde{\mathbf{x}})\}_{\tilde{\mathbf{x}}\in\tilde{\mathcal{Q}}}$ be the set of nodes on e, and let $\{\omega_{e,\mathbf{x}}\}_{\mathbf{x}\in\mathcal{Q}_e}$ be a set of corresponding weights. Together, the weights and nodes form a quadrature rule for the element. The quadrature rule is used to approximate the integrals of the mass matrix at the element as follows:

$$(\rho u, w)_e = \int_e \rho u w \; dx \approx \sum_{\mathbf{x}\in\mathcal{Q}_e} \omega_{e,\mathbf{x}} \rho_e(\mathbf{x}) u(\mathbf{x}) w(\mathbf{x}) =: (\rho u, w)_{\mathcal{Q}_e}, \tag{5.8}$$

where $\rho_e := \rho|_e$ denotes the scalar field ρ restricted to element e. We assume that ρ is continuous within each element, which implies that the approximation above is well defined. The global product $(\rho u, w)$ is then approximated by

$$(\rho u, w) \approx (\rho u, w)_{\mathcal{Q}_h} := \sum_{e\in\mathcal{T}_h} (\rho u, w)_{\mathcal{Q}_e}. \tag{5.9}$$

Now let $w_{\mathbf{x}}, w_{\mathbf{y}}$, with $\mathbf{x}, \mathbf{y} \in \mathcal{Q}_h$, be nodal basis functions as described in the previous subsection. The corresponding mass matrix entry is given by

$$(\rho w_{\mathbf{x}}, w_{\mathbf{y}})_{\mathcal{Q}_h} = \delta_{\mathbf{x}\mathbf{y}} \sum_{e\in\mathcal{T}_{\mathbf{x}}} \omega_{e,\mathbf{x}} \rho_e(\mathbf{x}), \tag{5.10}$$

This implies that the mass matrix is diagonal with entries of the form $\sum_{e\in\mathcal{T}_{\mathbf{x}}} \omega_{e,\mathbf{x}} \rho_e(\mathbf{x})$.

The quadrature rules can be constructed from a reference quadrature rule. This rule consists of the reference nodes $\tilde{\mathcal{Q}}$ and a set of weights $\{\tilde\omega_{\tilde{\mathbf{x}}}\}_{\tilde{\mathbf{x}}\in\tilde{\mathcal{Q}}}$ and approximates integrals on the reference element as follows:

$$\int_{\tilde e} \tilde\rho \tilde u \tilde w \; d\tilde x \approx \sum_{\tilde{\mathbf{x}}\in\tilde{\mathcal{Q}}} \omega_{\tilde{\mathbf{x}}} \tilde\rho(\tilde{\mathbf{x}}) \tilde u(\tilde{\mathbf{x}}) \tilde v(\tilde{\mathbf{x}}) =: (\tilde\rho \tilde u, \tilde w)_{\tilde{\mathcal{Q}}}.$$

We can use this to approximate the integral of the physical element by

$$(\rho u, w)_e = \frac{|e|}{|\tilde e|} \int_{\tilde e} \tilde\rho \tilde u \tilde w \; d\tilde x \approx \frac{|e|}{|\tilde e|} (\tilde\rho \tilde u, \tilde w)_{\tilde{\mathcal{Q}}},$$

with $|e|$ the volume of e, $|\tilde e|$ the volume of $\tilde e$, and $\tilde\rho := \rho\circ\phi_e$, $\tilde u := u\circ\phi_e$, $\tilde w := w\circ\phi_e$. This approximation is the same as (5.8) when $\omega_{e,\mathbf{x}} = (|e|/|\tilde e|)\tilde\omega_{\phi_e^{-1}(\mathbf{x})}$.

Now that we have introduced the quadrature rules for the mass matrix, we can present the mass-lumped finite element method.

5.3.3 Mass-lumped finite element method

Assume that $\rho \in \mathcal{C}^0(\mathcal{T}_h)$, $u_0 \in H_0^1(\Omega) \cap \mathcal{C}_0^0(\Omega)$, $v_0 \in \mathcal{C}_0^0(\Omega)$, and $f \in L^2\big(0,T;\mathcal{C}^0(\overline{\Omega})\big)$. Here, $\mathcal{C}^0(\mathcal{T}_h)$ denotes the set of functions that are in $\mathcal{C}^0(\overline{e})$ when restricted to e. The mass-lumped finite element method is finding $u_h : [0,T] \to U_h$, such that $u_h|_{t=0} = I_h u_0$, $\partial_t u_h|_{t=0} = I_h v_0$, and

$$(\rho\partial_t^2 u_h, w)_{\mathcal{Q}_h} + a(u_h, w) = (f, w)_{\mathcal{Q}_h} \quad \text{for all } w \in U_h, \text{ a.e. } t \in (0,T), \tag{5.11}$$

where $I_h : \mathcal{C}^0(\overline{\Omega}) \to U(\mathcal{T}_h, \tilde{U})$ denotes the interpolation of a continuous function by a function in $U(\mathcal{T}_h, \tilde{U})$ through the nodes of $\mathcal{Q}_h$.

To write this as a set of ODEs, let $\{\mathbf{x}^{(i)}\}_{i=1}^n = \mathcal{Q}_h \setminus \partial\Omega$ be a numbering of all interior nodes, and define $w_i := w_{\mathbf{x}^{(i)}}$ for all $i = 1, 2, \ldots, n$. Then the mass-lumped finite element method can be formulated as solving $\underline{\mathbf{u}}_h : [0,T] \to \mathbb{R}^n$ such that $\underline{\mathbf{u}}_h|_{t=0} = \underline{\mathbf{I_h u_0}}$, $\partial_t \underline{\mathbf{u}}_h|_{t=0} = \underline{\mathbf{I_h v_0}}$, and

$$M\partial_t^2 \underline{\mathbf{u}}_h + A\underline{\mathbf{u}}_h = \underline{\mathbf{f}}^* \qquad \text{for a.e. } t \in (0,T), \tag{5.12}$$

where $M_{ij} := (\rho w_i, w_j)_{\mathcal{Q}_h}$, $A_{ij} := a(w_i, w_j)$ for all $i, j = 1, \ldots, n$, and $\underline{\mathbf{f}}_i^* := (f, w_i)_{\mathcal{Q}_h}$, for $i = 1, \ldots, n$, a.e. $t \in (0,T)$. From (5.10) it follows that M is now a diagonal matrix that can be written as

$$M_{ij} = \delta_{ij} \sum_{e \in \mathcal{T}_{\mathbf{x}^{(i)}}} \omega_{e,\mathbf{x}^{(i)}} \rho_e(\mathbf{x}^{(i)}), \qquad i, j = 1, \ldots, n. \tag{5.13}$$

This set of ODEs can be efficiently solved using an explicit time integration scheme such as the second-order leap-frog scheme or a higher-order Dablain scheme [18], which is a type of Lax–Wendroff scheme [45] for second-order wave equations.

In the next sections we analyze the stability and accuracy of the mass-lumped finite element method and derive conditions for the quadrature rules.

5.3.4 Stability of the mass-lumped finite element method

To analyze the stability of the mass-lumped finite element method, we look at the behavior of the discrete energy. Consider the mass-lumped method given in (5.11) and substitute $w = \partial_t u$ to obtain

$$\partial_t E_h = (f, \partial_t u)_{\mathcal{Q}_h} \qquad \text{for a.e. } t \in (0,T),$$

where $E_h := \frac{1}{2}(\rho\partial_t u, \partial_t u)_{\mathcal{Q}_h} + \frac{1}{2}a(u,u)$ is the discrete energy. This implies that the discrete energy remains bounded when the source term f is bounded and that the discrete energy is conserved when there is no source term.

For stability it then remains to show that the discrete energy is a well-defined energy. This means that $(\rho v, v)_{\mathcal{Q}_h} + a(u,u) > 0$ for all $u, v \in U_h$, $(u,v) \neq 0$, which is the case when $(\rho u, u)_{\mathcal{Q}_h} > 0$ for any $u \in U_h$, $u \neq 0$. Since we can write $(\rho u, u)_{\mathcal{Q}_h} = \underline{\mathbf{u}}^t M \underline{\mathbf{u}}$, this is satisfied when M is positive definite. From (5.13) it follows that this is the case when all weights of the quadrature rules are strictly positive, which is the case when the weights of the reference quadrature rule are strictly positive.

5.4 Accuracy of the mass-lumped finite element method

5.4.1 A less restrictive condition on the accuracy of the quadrature rule

Let $U_h = U(\mathcal{T}_h, \tilde{U})$, with $\tilde{U} = \mathcal{P}_p(\tilde{e}) \oplus \tilde{U}^+$, be the finite element space constructed as in Section 5.3, where $p \geq 2$ denotes the degree of the finite element method and $\tilde{U}^+ \subset \mathcal{P}_{p'}(\tilde{e})$ is the space of higher-degree face and interior bubble functions. Also, let the quadrature rule for the mass matrix be based on a reference element quadrature rule as described in Section 5.3.2. We will prove that an optimal convergence rate of the mass-lumped finite element method is obtained when all weights of the reference quadrature rule, $\{\tilde{\omega}_{\tilde{\mathbf{x}}}\}_{\tilde{\mathbf{x}}\in\tilde{\mathcal{Q}}}$, are strictly positive and

$$\int_{\tilde{e}} \tilde{f}\, d\tilde{x} = \sum_{\tilde{\mathbf{x}}\in\tilde{\mathcal{Q}}} \tilde{\omega}_{\tilde{\mathbf{x}}} \tilde{f}(\mathbf{x}) \qquad \text{for all } \tilde{f} \in \mathcal{P}_{p-2}(\tilde{e}) \otimes \tilde{U}, \tag{5.14}$$

where $\mathcal{P}_{p-2}(\tilde{e}) \otimes \tilde{U} := \{f \mid f = wu \text{ for some } w \in \mathcal{P}_{p-2}(\tilde{e}), u \in \tilde{U}\}$. This means that the quadrature rule of the reference element should be exact for products of the reference basis functions and polynomials of degree $p-2$. Until now, the condition used for the accuracy of the quadrature rule was

$$\int_{\tilde{e}} \tilde{f}\, d\tilde{x} = \sum_{\tilde{\mathbf{x}}\in\tilde{\mathcal{Q}}} \tilde{\omega}_{\tilde{\mathbf{x}}} \tilde{f}(\mathbf{x}), \qquad \text{for all } \tilde{f} \in \mathcal{P}_{p+p'-2}(\tilde{e}), \tag{5.15}$$

(see, for example, [16, 51, 11]), so it was imposed that the reference quadrature rule should be exact for functions in $\mathcal{P}_{p+p'-2}(\tilde{e})$, with p' the highest polynomial degree of the enriched space, which turns out to be significantly more restrictive for tetrahedral elements. By using (5.14) instead of (5.15) we are able to develop new mass-lumped elements that require significantly fewer nodes.

In the next subsections we will prove that the convergence rate of the mass-lumped finite element method remains optimal under the less severe condition (5.14). The novel part of the proofs is the bounds on the integration error, derived in Section 5.4.3. This is the only part where we explicitly use condition (5.14). Using these bounds, we can prove optimal convergence in a rather standard way.

5.4.2 Some norms and interpolation properties

For the convergence analysis, we use multiple interpolation properties, which we will present in this subsection. Also, to make the analysis more readable, we will use C to denote some positive constant that may depend on the regularity of the mesh, the reference space $\tilde{U}$, the reference quadrature rule, the domain Ω, and the parameters ρ, c, but does not depend on the mesh resolution h, the time interval $(0,T)$, or the choice of the functions that appear in the inequality.

Let $H^k(\Omega)$, with $k \geq 1$, denote the Sobolev space, consisting of functions with square integrable order-k weak derivatives equipped with norm

$$\|u\|_k^2 := \sum_{|\boldsymbol{\alpha}|\leq k} \|D^{\boldsymbol{\alpha}}u\|_0^2, \qquad k \geq 1,$$

where $\|\cdot\|_0$ denotes the standard $L^2(\Omega)$-norm, and $D^{\boldsymbol{\alpha}} := \partial_1^{\alpha_1}\partial_2^{\alpha_2}\partial_3^{\alpha_3}$ denotes a higher-order partial derivative of order $|\boldsymbol{\alpha}| := \alpha_1 + \alpha_2 + \alpha_3$. Also let $H^k(\mathcal{T}_h)$, with $k \geq 1$, denote the broken Sobolev space, consisting of functions that belong to $H^k(e)$ when restricted to element e, for all $e \in \mathcal{T}_h$. We equip this space with the norm

$$\|u\|_{\mathcal{T}_h,k}^2 := \sum_{e\in\mathcal{T}_h} \|u\|_{e,k}^2 := \sum_{e\in\mathcal{T}_h} \left(\sum_{|\boldsymbol{\alpha}|\leq k} \|D^{\boldsymbol{\alpha}}\mathbf{u}\|_e^2 \right), \qquad k \geq 1.$$

Now let $I_h : \mathcal{C}^0(\overline{\Omega}) \to U(\mathcal{T}_h, \tilde{U})$ denote the interpolation by a function in $U(\mathcal{T}_h, \tilde{U})$ through the nodes of $\mathcal{Q}_h$. This interpolation operator is well defined for functions in $H^2(\mathcal{T}_h) \cap H^1(\Omega)$, since $H^2(e) \subset \mathcal{C}^0(\overline{e})$ when e is a 3D element, and therefore $H^2(\mathcal{T}_h) \cap H^1(\Omega) \subset \mathcal{C}^0(\overline{\Omega})$. For this interpolation operator, we can present the following approximation properties:

Lemma 5.4.1. *Let $p \geq 2$ be the degree of the finite element space and let $u \in H^1(\Omega) \cap H^k(\mathcal{T}_h)$ with $k \geq 2$. Then*

$$\|u - I_h u\|_{\mathcal{T}_h,l} \leq Ch^{\min(p+1,k)-l} \|u\|_{\mathcal{T}_h,\min(p+1,k)}, \qquad l \leq \min(p+1,k).$$

Proof. This result follows from [12, Theorem 3.1.6]. □

Now assume that the weights for the reference quadrature rule are all strictly positive. For any function in $H^1(\Omega) \cap H^2(\mathcal{T}_h)$, we can then define the following discrete L^2 seminorm:

$$|u|^2_{\mathcal{Q}_h} := (u,u)_{\mathcal{Q}_h}.$$

This discrete seminorm is well defined, since $H^1(\Omega) \cap H^2(\mathcal{T}_h) \subset \mathcal{C}^0(\overline{\Omega})$ as mentioned before. This becomes a full norm, $\|\cdot\|_{\mathcal{Q}_h}$, that is equivalent to the L^2-norm, for functions in U_h:

Lemma 5.4.2. *If all the weights of the reference quadrature rule are strictly positive, then*

$$C^{-1}\|u\|_0 \leq \|u\|_{\mathcal{Q}_h} \leq C\|u\|_0 \qquad \textit{for all } u \in U_h. \tag{5.16}$$

Proof. Since the function space of the reference element $\tilde{U} := \text{span}\{\tilde{w}_{\tilde{\mathbf{x}}}\}_{\tilde{\mathbf{x}} \in \tilde{\mathcal{Q}}}$ is finite-dimensional, and since all weights of the reference quadrature rule $\{\tilde{\omega}_{\tilde{\mathbf{x}}}\}_{\tilde{\mathbf{x}} \in \tilde{\mathcal{Q}}}$ are positive, there exists a constant $C > 0$ depending on the reference quadrature rule and function space $\tilde{U}$, such that

$$C^{-1}\|\tilde{u}\|_{\tilde{e}} \leq \|\tilde{u}\|_{\tilde{\mathcal{Q}}} \leq C\|\tilde{u}\|_{\tilde{e}} \qquad \text{for all } \tilde{u} \in \tilde{U}.$$

where $\|\tilde{u}\|^2_{\tilde{\mathcal{Q}}} := (\tilde{u},\tilde{u})_{\tilde{\mathcal{Q}}}$. Then (5.16) follows from the relations

$$\|u\|_0^2 = \sum_{e \in \mathcal{T}_h} \frac{|e|}{|\tilde{e}|}\|\tilde{u}_e\|^2_{\tilde{e}}, \qquad \|u\|^2_{\mathcal{Q}_h} = \sum_{e \in \mathcal{T}_h} \frac{|e|}{|\tilde{e}|}\|\tilde{u}_e\|^2_{\tilde{\mathcal{Q}}},$$

where $\tilde{u}_e := u \circ \phi_e$. □

Now let $\Pi_{h,q} : L^2(\Omega) \to \mathcal{P}_q(\mathcal{T}_h)$ denote the L^2-projection onto the space of piecewise *nonconforming* polynomials of at most degree q:

$$\mathcal{P}_q(\mathcal{T}_h) := \{u \in L^2(\Omega) \mid u|_e \in \mathcal{P}_q(e) \text{ for all } e \in \mathcal{T}_h\}.$$

We then present the following interpolation properties:

Lemma 5.4.3. *Let $u \in H^k(\mathcal{T}_h)$ with $k \geq 2$, and let $q \geq 0$. Then*

$$\|u - \Pi_{h,q}u\|_0 \leq Ch^{\min(q+1,k)}\|u\|_{\mathcal{T}_h,\min(q+1,k)}. \tag{5.17}$$

Furthermore, if also $u \in H^1(\Omega)$, if $p \geq \max(q,2)$ is the degree of the finite element space, and if all the weights of the reference quadrature rule are strictly positive, then

$$|u - \Pi_{h,q}u|_{\mathcal{Q}_h} \leq Ch^{\min(q+1,k)}\|u\|_{\mathcal{T}_h,\min(q+1,k)}. \tag{5.18}$$

Proof. The first inequality, (5.17), follows from [12, Theorem 3.1.6]. The second inequality can be derived as follows:

$$\begin{aligned}|u - \Pi_{h,q}u|_{\mathcal{Q}_h} &= |I_hu - \Pi_{h,q}u|_{\mathcal{Q}_h}\\ &\leq C\|I_hu - \Pi_{h,q}u\|_0\\ &\leq C(\|I_hu - u\|_0 + \|u - \Pi_{h,q}u\|_0)\\ &\leq Ch^{\min(q+1,k)}\|u\|_{\mathcal{T}_h,\min(q+1,k)}\end{aligned}$$

where we used Lemma 5.4.2 in the second line, the triangle inequality in the third line, and Lemma 5.4.1 and (5.17) in the last line. □

5.4.3 Bounds on the integration error

In this section we will derive some useful bounds on the error of the quadrature rules for the mass matrix. The proofs of these bounds will be the only cases where we explicitly use the accuracy condition of the quadrature rule, given in (5.14). Using these results we can prove optimal order of convergence of the mass-lumped finite element method in a rather standard way.

Let $u, w \in H^2(\mathcal{T}_h)$, and let $r_h(u,w) := (u,w) - (u,w)_{\mathcal{Q}_h}$ be the integration error of the mass matrix. We can derive the following bounds on r_h:

Lemma 5.4.4. *Let $p \geq 2$ be the degree of the finite element space, $u \in H^k(\Omega)$ with $k \geq 2$, and $w \in U_h$. If the reference quadrature rule satisfies (5.14) and if all its weights are strictly positive, then*

$$|r_h(u,w)| \leq Ch^{\min(p,k)}\|u\|_{\min(p,k)}\|w\|_1. \tag{5.19}$$

and

$$|r_h(u,w)| \leq Ch^{\min(p+1,k)}\|u\|_{\min(p+1,k)}\|w\|_{\mathcal{T}_h,2}. \tag{5.20}$$

Proof. Using (5.14) and the fact that $\mathcal{P}_{p-2}(\tilde{e}) \otimes \tilde{U} \supset \mathcal{P}_p(\tilde{e})$ for $p \geq 2$, we can write

$$\begin{aligned} r_h(u,w) &= r_h\big((u-\Pi_{h,p-1}u) + (\Pi_{h,p-1}u - \Pi_{h,p-2}u) + \Pi_{h,p-2}u, \\ &\qquad (w-\Pi_{h,0}w) + \Pi_{h,0}w\big) \\ &= r_h(u-\Pi_{h,p-1}u, w) + r_h(\Pi_{h,p-1}u - \Pi_{h,p-2}u, w - \Pi_{h,0}w). \end{aligned}$$

From this, the Cauchy–Schwarz inequality, and Lemma 5.4.3, we can then obtain (5.19).

Using (5.14), we can also write

$$\begin{aligned} r_h(u,w) = r_h\Big(&\big[(u-\Pi_{h,p}u) + (\Pi_{h,p}u - \Pi_{h,p-1}u) + (\Pi_{h,p-1}u - \Pi_{h,p-2}u) + \\ &\Pi_{h,p-2}u\big], \big[(w-\Pi_{h,1}w) + (\Pi_{h,1}w - \Pi_{h,0}w) + \Pi_{h,0}w\big]\Big) \\ = {}& r_h(u-\Pi_{h,p}u, w) + r_h(\Pi_{h,p}u - \Pi_{h,p-1}u, w - \Pi_{h,0}w) + \\ & r_h(\Pi_{h,p-1}u - \Pi_{h,p-2}u, w - \Pi_{h,1}w). \end{aligned}$$

From this, the Cauchy–Schwarz inequality, and Lemma 5.4.3, we can then obtain (5.20). □

5.4.4 Optimal convergence for a related elliptic problem

To prove optimal convergence of the mass-lumped finite element method, we first prove optimal convergence for a related elliptic problem.

Let $v \in H^2(\mathcal{T}_h)$. The elliptic problem related to (5.2), is finding $u \in H_0^1(\Omega)$ such that

$$a(u,w) = (v,w) \qquad \text{for all } w \in H_0^1(\Omega). \tag{5.21}$$

This problem is well defined since a is coercive and bounded with respect to the $H^1(\Omega)$-norm, which follows from the boundedness of c and Poincaré's inequality.

The related mass-lumped method for solving this problem is finding $u_h \in U_h$ such that

$$a(u_h,w) = (v,w)_{\mathcal{Q}_h} \qquad \text{for all } w \in U_h. \tag{5.22}$$

In the next theorems we prove optimal convergence of this method in the H^1-norm and L^2-norm.

Theorem 5.4.5 (Optimal Convergence in the H^1-norm)**.** *Let u be the solution of (5.21) and u_h the solution of (5.22), with $p \geq 2$ the degree of the finite element space. Also, let $k_u, k_v \geq 2$, $u \in H^{k_u}(\Omega)$, and $v \in H^{k_v}(\Omega)$. If the reference quadrature rule satisfies (5.14) and if all its weights are strictly positive, then*

$$\|u - u_h\|_1 \leq Ch^{\min(p,k_u-1,k_v)}(\|u\|_{\min(p+1,k_u)} + \|v\|_{\min(p,k_v)}). \tag{5.23}$$

Proof. By definition of u and u_h, we have

$$a(u - u_h, w) = r_h(v, w) \qquad \text{for all } w \in U_h.$$

By choosing $w = I_h u - u_h$ we can then obtain

$$a(I_h u - u_h, I_h u - u_h) = -a(u - I_h u, I_h u - u_h) + r_h(v, I_h u - u_h). \tag{5.24}$$

From the coercivity of a it follows that

$$\|I_h u - u_h\|_1^2 \leq Ca(I_h u - u_h, I_h u - u_h). \tag{5.25}$$

From the boundedness of a and Lemma 5.4.1 it follows that

$$|a(u - I_h u, I_h u - u_h)| \leq Ch^{\min(p,k_u-1)}\|u\|_{\min(p+1,k_u)}\|I_h u - u_h\|_1. \tag{5.26}$$

Using Lemma 5.4.4 we obtain

$$|r_h(v, I_h u - u_h)| \leq Ch^{\min(p,k_v)}\|v\|_{\min(p,k_v)}\|I_h u - u_h\|_1. \tag{5.27}$$

Combining (5.24), (5.25), (5.26), and (5.27) then gives

$$\|I_h u - u_h\|_1 \leq Ch^{\min(p,k_u-1,k_v)}(\|u\|_{\min(p+1,k_u)} + \|v\|_{\min(p,k_v)}). \tag{5.28}$$

From Lemma 5.4.1 it also follows that

$$\|u - I_h u\|_1 \leq Ch^{\min(p,k_u-1)}\|u\|_{\min(p+1,k_u)}. \tag{5.29}$$

Combining (5.28) and (5.29) then results in (5.23). □

To prove optimal convergence in the L^2-norm, we make the following regularity assumption: for any $v \in L^2(\Omega)$, the solution u of (5.21) is in $H^2(\Omega)$ and satisfies

$$\|u\|_2 \leq C\|v\|_0. \tag{5.30}$$

This is certainly true if $\partial\Omega$ is $\mathcal{C}^2$ and $c \in \mathcal{C}^1(\overline{\Omega})$.

Theorem 5.4.6 (Optimal Convergence in the L^2-norm)**.** *Let u be the solution of (5.21) and u_h the solution of (5.22), with $p \geq 2$ the degree of the finite element space. Also, let $k_u, k_v \geq 2$, $u \in H^{k_u}(\Omega)$, $v \in H^{k_v}(\Omega)$, and assume that the regularity condition (5.30) holds. If the reference quadrature rule satisfies (5.14) and if all its weights are strictly positive, then*

$$\|u - u_h\|_0 \leq Ch^{\min(p+1,k_u,k_v)}(\|u\|_{\min(p+1,k_u)} + \|v\|_{\min(p+1,k_v)}) \tag{5.31}$$

and

$$|u - u_h|_{\mathcal{Q}_h} \leq Ch^{\min(p+1,k_u,k_v)}(\|u\|_{\min(p+1,k_u)} + \|v\|_{\min(p+1,k_v)}). \tag{5.32}$$

Proof. Let $z \in H_0^1(\Omega)$ be the solution of

$$a(z, w) = (u - u_h, w) \qquad\qquad \text{for all } w \in H_0^1(\Omega).$$

From the regularity assumption it follows that $z \in H^2(\Omega)$ and $\|z\|_2 \leq C\|u - u_h\|_0$. Using the definition of z, u, and u_h, we can also write

$$\begin{aligned} \|u - u_h\|_0^2 &= a(u - u_h, z) \\ &= a(u - u_h, z - I_h z) + a(u - u_h, I_h z) \\ &= a(u - u_h, z - I_h z) + r_h(v, I_h z). \end{aligned} \tag{5.33}$$

Using the boundedness of a, Theorem 5.4.5, Lemma 5.4.1, and the regularity assumption, we obtain

$$\begin{aligned} |a(u - u_h, z - I_h z)| &\leq C\|u - u_h\|_1 \|z - I_h z\|_1 \\ \leq Ch^{\min(p+1,k_u,k_v+1)}&(\|u\|_{\min(p+1,k_u)} + \|v\|_{\min(p,k_v)})\|u - u_h\|_0. \end{aligned} \tag{5.34}$$

From Lemma 5.4.4, Lemma 5.4.1, and the regularity assumption it also follows that

$$|r_h(v, I_h z)| \leq Ch^{\min(p+1,k_v)}\|v\|_{\min(p+1,k_v)}\|u - u_h\|_0. \tag{5.35}$$

Combining (5.33), (5.34), and (5.35) results in (5.31).

To derive (5.32), we use Lemma 5.4.2 to obtain

$$|u - u_h|_{\mathcal{Q}_h} = \|I_h u - u_h\|_{\mathcal{Q}_h} \leq C\|I_h u - u_h\|_0.$$

Combining this inequality with (5.31) and Lemma 5.4.1 results in (5.32). □

5.4.5 Some additional norms and interpolation properties

In order to analyze the convergence for the time dependent problem, we need to introduce an additional projection operator and some additional function spaces.

Let L denote the spatial operator $L := -\nabla \cdot c\nabla$, and let $u \in H_0^1(\Omega)$ with $Lu \in \mathcal{C}^0(\overline{\Omega})$. We define the projection $\pi_h u \in U_h$ to be the solution of

$$a(\pi_h u, w) = (Lu, w)_{\mathcal{Q}_h}, \qquad \text{for all } w \in U_h.$$

We can derive the following interpolation property of this projection operator:

Lemma 5.4.7. *Let $p \geq 2$ be the degree of the finite element space, and let $c \in \mathcal{C}^{k+1}(\overline{\Omega})$ and $u \in H_0^1(\Omega) \cap H^{k+2}(\Omega)$, with $k \geq 2$. If the reference quadrature rule satisfies (5.14) and if all its weights are strictly positive, then*

$$\|u - \pi_h u\|_1 \leq Ch^{\min(p,k)}\|u\|_{\min(p+2,k+2)},$$

Moreover, if regularity condition (5.30) also holds, then

$$\|u - \pi_h u\|_0 \leq Ch^{\min(p+1,k)}\|u\|_{\min(p+3,k+2)},$$
$$|u - \pi_h u|_{\mathcal{Q}_h} \leq Ch^{\min(p+1,k)}\|u\|_{\min(p+3,k+2)}.$$

Proof. From partial integration it follows that $a(u, w) = (Lu, w)$ for all $w \in H_0^1(\Omega)$. Also, by definition of the projection we have $a(\pi_h u, w) = (Lu, w)_{\mathcal{Q}_h}$ for all $w \in U_h$. The inequalities then follow from Theorem 5.4.5 and Theorem 5.4.6 by taking $v = Lu$, $k_u = k + 2$, $k_v = k$, and using the bounds $\|Lu\|_q \leq C\|u\|_{q+2}$ for $q \leq k$. □

We also extend the spaces $H^k(\Omega)$ to Bochner spaces $L^\infty(0, T; H^k(\Omega))$, equipped with norm

$$\|u\|_{\infty,k} := \operatorname*{ess\,sup}_{t\in(0,T)} \|u\|_k.$$

5.4.6 Optimal convergence of the mass-lumped finite element method

In this section we prove the optimal convergence of the mass-lumped finite element method for the wave equation. We first derive an equation for the behavior of the numerical error and then prove optimal convergence in the energy-norm and L^2-norm.

Lemma 5.4.8 (Error Equation)**.** *Let u be the solution of (5.2) and let u_h be the solution of (5.11). If $\rho \in \mathcal{C}^0(\overline{\Omega})$, $\partial_t^2 u \in L^2(0,T;\mathcal{C}_0^0(\Omega))$, and $f \in L^2(0,T;\mathcal{C}^0(\overline{\Omega}))$, then $Lu \in L^2(0,T;\mathcal{C}^0(\overline{\Omega}))$, and*

$$(\rho\partial_t^2 e_h, w)_{\mathcal{Q}_h} + a(e_h, w) = -(\rho\partial_t^2 \epsilon_h, w)_{\mathcal{Q}_h} \tag{5.36}$$

for all $w \in U_h$ and almost every $t \in (0,T)$, where $e_h := \pi_h u - u_h$ and $\epsilon_h := u - \pi_h u$.

Proof. Since ρ is bounded and continuous, $\rho\partial_t^2 u \in L^2(0,T;\mathcal{C}_0^0(\Omega))$. Since also $f \in L^2(0,T;\mathcal{C}^0(\overline{\Omega}))$, it follows that $Lu \in L^2(0,T;\mathcal{C}^0(\overline{\Omega}))$ and $\rho\partial_t^2 u + Lu = f$. This implies

$$(\rho\partial_t^2 u, w)_{\mathcal{Q}_h} + (Lu, w)_{\mathcal{Q}_h} = (f, w)_{\mathcal{Q}_h}$$

for all $w \in U_h$ and almost every $t \in (0,T)$. Using the definition of $\pi_h u$ we can then obtain

$$(\rho\partial_t^2 u, w)_{\mathcal{Q}_h} + a(\pi_h u, w) = (f, w)_{\mathcal{Q}_h}$$

for all $w \in U_h$ and almost every $t \in (0,T)$. By definition of u_h we have

$$(\rho\partial_t^2 u_h, w)_{\mathcal{Q}_h} + a(u_h, w) = (f, w)_{\mathcal{Q}_h}$$

for all $w \in U_h$ and almost every $t \in (0,T)$. Subtracting this from the previous equality and reordering the terms results in (5.36). □

Theorem 5.4.9 (Optimal Convergence in the Energy-Norm)**.** *Let u be the solution of (5.2) and let u_h be the solution of (5.11), with $p \geq 2$ the degree of the finite element space. Let $\rho \in \mathcal{C}(\overline{\Omega})$, $f \in L^2(0,T;\mathcal{C}^0(\overline{\Omega}))$, and let $c \in \mathcal{C}^{k+1}(\overline{\Omega})$, $u, \partial_t u, \partial_t^2 u \in L^\infty(0,T;H^{k+2}(\Omega))$ for some $k \geq 2$. Also, assume that regularity condition (5.30) holds. If the reference quadrature rule satisfies (5.14) and if all its weights are strictly positive, then*

$$\begin{aligned}\|u-u_h\|_{\infty,1} + \|\partial_t u - \partial_t u_h\|_{\infty,0} \leq Ch^{\min(p,k)}\big(&\|u\|_{\infty,\min(p+3,k+2)} \\ &+ \|\partial_t u\|_{\infty,\min(p+3,k+2)} + T\|\partial_t^2 u\|_{\infty,\min(p+3,k+2)}\big).\end{aligned} \tag{5.37}$$

Proof. Define $e_h := \pi_h u - u_h$ and $\epsilon_h := u - \pi_h u$. From Lemma 5.4.8, it follows that

$$(\rho\partial_t^2 e_h, w)_{\mathcal{Q}_h} + a(e_h, w) = -(\rho\partial_t^2 \epsilon_h, w)_{\mathcal{Q}_h}, \tag{5.38}$$

for all $w \in U_h$ and almost every $t \in (0, T)$. By substituting $w = \partial_t e_h$ we can obtain

$$\partial_t E_h = -(\rho \partial_t^2 \epsilon_h, \partial_t e_h)_{\mathcal{Q}_h}, \tag{5.39}$$

for almost every $t \in (0, T)$, where $E_h := \frac{1}{2}(\rho \partial_t e_h, \partial_t e_h)_{\mathcal{Q}_h} + \frac{1}{2} a(e_h, e_h)$ is the discrete energy. Fix $T' \in (0, T)$ and integrate (5.39) over $(0, T')$ to obtain

$$E_h|_{t=T'} = E_h|_{t=0} - \int_0^{T'} (\rho \partial_t^2 \epsilon_h, \partial_t e_h)_{\mathcal{Q}_h} \, dt. \tag{5.40}$$

Using the coercivity of a, the boundedness of ρ, and Lemma 5.4.2, we can derive

$$\|e_h\|_1 + \|\partial_t e_h\|_0 \leq C E_h^{1/2}, \qquad \text{a.e. } t \in (0, T). \tag{5.41}$$

From the Cauchy–Schwarz inequality, the bounds of ρ, Lemma 5.4.7, and Lemma 5.4.2, we can also obtain

$$|(\rho \partial_t^2 \epsilon_h, \partial_t e_h)_{\mathcal{Q}_h}| \leq C h^{\min(p+1,k)} \|\partial_t^2 u\|_{\min(p+3,k+2)} \|\partial_t e_h\|_0, \tag{5.42}$$

for almost every $t \in (0, T)$. Finally, we can use Lemma 5.4.1, Lemma 5.4.7, and the boundedness of ρ and a to obtain

$$E_h^{1/2}|_{t=0} \leq C h^{\min(p,k)} \big(\|u\|_{\infty,\min(p+3,k+2)} + \|\partial_t u\|_{\infty,\min(p+3,k+2)} \big). \tag{5.43}$$

By taking the supremum of (5.40) for all $T' \in (0, T)$ and using (5.41), (5.42), and (5.43), we can obtain

$$\begin{aligned} \|e_h\|_{\infty,1} + \|\partial_t e_h\|_{\infty,0} \leq C h^{\min(p,k)} \big(& \|u\|_{\infty,\min(p+3,k+2)} \\ & + \|\partial_t u\|_{\infty,\min(p+3,k+2)} + T \|\partial_t^2 u\|_{\infty,\min(p+3,k+2)} \big). \end{aligned} \tag{5.44}$$

Using (5.44) and Lemma 5.4.7 we obtain (5.37). □

Theorem 5.4.10 (Optimal Convergence in the L^2-Norm)**.** *Let u be the solution of (5.2) and let u_h be the solution of (5.11), with $p \geq 2$ the degree of the finite element space. Let $\rho \in \mathcal{C}(\overline{\Omega})$, $f \in L^2(0, T; \mathcal{C}^0(\overline{\Omega}))$, $\partial_t^2 u \in L^2(0, T; \mathcal{C}_0^0(\Omega))$, and let $c \in \mathcal{C}^{k+1}(\overline{\Omega})$, $u, \partial_t u, \in L^\infty(0, T; H^{k+2}(\Omega))$ for some $k \geq 2$. Also, assume that regularity condition (5.30) holds. If the reference quadrature rule satisfies (5.14) and if all its weights are strictly positive, then*

$$\|u - u_h\|_{\infty,0} \leq C h^{\min(p+1,k)} \big(\|u\|_{\infty,\min(p+3,k+2)} + T \|\partial_t u\|_{\infty,\min(p+3,k+2)} \big). \tag{5.45}$$

Proof. Define $e_h := \pi_h u - u_h$ and $\epsilon_h := u - \pi_h u$. From Lemma 5.4.8, it follows that

$$(\rho\partial_t^2 e_h, w)_{\mathcal{Q}_h} + a(e_h, w) = -(\rho\partial_t^2 \epsilon_h, w)_{\mathcal{Q}_h}, \tag{5.46}$$

for all $w \in U_h$ and almost every $t \in (0,T)$. Fix $T' \in (0,T)$ and choose w as

$$w|_{t=t'} := \int_{t'}^{T'} e_h \, dt.$$

This implies that $w|_{t=T'} = 0$ and $\partial_t w = -e_h$. Using the relations

$$\begin{aligned}
(\rho\partial_t^2 e_h, w)_{\mathcal{Q}_h} &= \partial_t(\rho\partial_t e_h, w)_{\mathcal{Q}_h} + \frac{1}{2}\partial_t(\rho e_h, e_h)_{\mathcal{Q}_h}, \\
a(e_h, w) &= -\frac{1}{2}\partial_t a(w,w), \\
-(\rho\partial_t^2 \epsilon_h, w)_{\mathcal{Q}_h} &= -\partial_t(\rho\partial_t \epsilon_h, w)_{\mathcal{Q}_h} - (\rho\partial_t \epsilon_h, e_h)_{\mathcal{Q}_h},
\end{aligned}$$

we can rewrite (5.46) as

$$\frac{1}{2}\partial_t(\rho e_h, e_h)_{\mathcal{Q}_h} = \frac{1}{2}\partial_t a(w,w) - \partial_t\big(\rho\partial_t(u-u_h), w\big)_{\mathcal{Q}_h} - (\rho\partial_t\epsilon_h, e_h)_{\mathcal{Q}_h}, \tag{5.47}$$

for almost every $t \in (0,T)$. Integrating (5.47) over $(0,T')$ and using the fact that $w|_{t=T'} = 0$ and $\partial_t(u-u_h)|_{t=0,x\in\mathcal{Q}_h} = 0$ results in

$$\begin{aligned}
&\frac{1}{2}(\rho e_h, e_h)_{\mathcal{Q}_h}|_{t=T'} = \\
&\frac{1}{2}(\rho e_h, e_h)_{\mathcal{Q}_h}|_{t=0} - \frac{1}{2}a(w,w)|_{t=0} - \int_0^{T'} (\rho\partial_t\epsilon_h, e_h)_{\mathcal{Q}_h} \, dt.
\end{aligned} \tag{5.48}$$

From the boundedness of ρ and Lemma 5.4.2 it follows that

$$\|e_h\|_0 \leq C\|\rho^{1/2} e_h\|_{\mathcal{Q}_h}, \qquad \text{a.e. } t \in (0,T). \tag{5.49}$$

Because of the coercivity of a we have

$$-\frac{1}{2}a(w,w)|_{t=0} < 0. \tag{5.50}$$

From the Cauchy–Schwarz inequality, the bounds of ρ, Lemma 5.4.7, and Lemma 5.4.2, we can also obtain

$$|(\rho\partial_t\epsilon_h, e_h)_{\mathcal{Q}_h}| \leq Ch^{\min(p+1,k)}\|\partial_t u\|_{\min(p+3,k+2)}\|e_h\|_0, \tag{5.51}$$

for almost every $t \in (0,T)$. Finally, we can use Lemma 5.4.1, Lemma 5.4.7 and the boundedness of ρ to obtain

$$\|\rho^{1/2} e_h|_{t=0}\|_{\mathcal{Q}_h} \leq Ch^{\min(p+1,k)}\|u\|_{\infty,\min(p+3,k+2)}. \tag{5.52}$$

By taking the supremum of (5.48) for all $T' \in (0,T)$ and using (5.49), (5.50), (5.51), and (5.52), we can obtain

$$\|e_h\|_{\infty,0} \leq Ch^{\min(p+1,k)}\big(\|u\|_{\infty,\min(p+3,k+2)} + T\|\partial_t u\|_{\infty,\min(p+3,k+2)}\big). \tag{5.53}$$

Using (5.53) and Lemma 5.4.7 we obtain (5.45). □

5.5 Several new mass-lumped tetrahedral elements of degrees two to four

In this section, we present several novel mass-lumped tetrahedral elements for degree $p = 2, 3, 4$. The new degree-2 and degree-3 elements use 15 and 32 nodes per element, respectively, while the current elements for these degrees require 23 and 50 nodes, respectively [51, 11]. We also introduce several degree-4 elements, requiring 60, 61, and 65 nodes. Mass-lumped tetrahedral elements of degree 4 had not been found until now.

Table 5.1: Degree-2 mass-lumped tetrahedral element with 15 nodes.

Nodes	n	ω	Parameters
$\{(0,0,0)\}$	4	$\frac{17}{5040}$	-
$\{(\frac{1}{2},\frac{1}{2},0)\}$	6	$\frac{2}{315}$	-
$\{(\frac{1}{3},\frac{1}{3},0)\}$	4	$\frac{9}{560}$	-
$\{(\frac{1}{4},\frac{1}{4},\frac{1}{4})\}$	1	$\frac{16}{315}$	-
$U = \mathcal{P}_2 \oplus \mathcal{B}_f \oplus \mathcal{B}_e = \{x_1, x_1x_2, \beta_f, \beta_e\}$			

We present the mass-lumped tetrahedral elements using the reference tetrahedron with vertices at $(0,0,0)$, $(1,0,0)$, $(0,1,0)$, and $(0,0,1)$. In previous sections we used a tilde to denote coordinates and sets in the reference space, but since we only consider the reference space in this section, we will drop the tilde for readability.

Table 5.2: Degree-3 mass-lumped tetrahedral element with 32 nodes.

Nodes	n	ω	Parameters
$\{(0,0,0)\}$	4	$\frac{41-9\sqrt{2}}{41160}$	-
$\{(a,0,0)\}$	12	$\frac{8+9\sqrt{2}}{13720}$	$\frac{3-\sqrt{3(\sqrt{2}-1)}}{6}$
$\{(b,b,0)\}$	12	$\frac{10-\sqrt{2}}{1715}$	$\frac{4-\sqrt{2}}{12}$
$\{(c,c,c)\}$	4	$\frac{3}{140}$	$\frac{1}{6}$
$U=\mathcal{P}_3\oplus\mathcal{B}_f\mathcal{P}_1\oplus\mathcal{B}_e\mathcal{P}_1=\{x_1,x_1^2x_2,\beta_fx_1,\beta_ex_1\}$			
$U\otimes\mathcal{P}_1=\{x_1,x_1^2x_2,x_1^2x_2^2,\beta_fx_1,\beta_fx_1x_2,\beta_ex_1,\beta_ex_1x_2\}$			

The nodes on the reference element are described using the notation $\{\mathbf{x}\}$, which denotes the node $\mathbf{x}$ and all equivalent nodes $s(\mathbf{x})$, with $s\in\mathcal{S}$. As shown in Lemma 5.A.1, any $s\in\mathcal{S}$ can be represented by a permutation of the barycentric coordinates. In this case, the barycentric coordinates are given by the three Cartesian coordinates x_1, x_2, x_3, and the additional coordinate $x_4:=1-x_1-x_2-x_3$, so any $s\in\mathcal{S}$ can be written as $s(x_1,x_2,x_3)=(x_j,x_j,x_k)$, with $i,j,k\in\{1,2,3,4\}$, $i\neq j$, $i\neq k$, $j\neq k$. The barycentric coordinates of the node $\mathbf{x}=(\frac{1}{5},\frac{1}{5},\frac{1}{5})$, for example, are therefore given by $(\frac{1}{5},\frac{1}{5},\frac{1}{5},\frac{2}{5})$, and the set of equivalent nodes $\{\mathbf{x}\}$ consists of $(\frac{1}{5},\frac{1}{5},\frac{1}{5})$, $(\frac{2}{5},\frac{1}{5},\frac{1}{5})$, $(\frac{1}{5},\frac{2}{5},\frac{1}{5})$, and $(\frac{1}{5},\frac{1}{5},\frac{2}{5})$.

The reference function space, denoted by U, is the span of all nodal basis functions and is described in terms of $\{w\}$, which denotes the span of function w and all its equivalent functions $w\circ s$, with $s\in\mathcal{S}$. For example, all equivalent functions of $w=x_1x_2$ are x_1x_2, x_1x_3, x_1x_4, x_2x_3, x_2x_4, and x_3x_4, so $\{w\}$ is the span of these six functions.

We assign the same weight to each equivalent node, so $\omega_{\mathbf{x}}=\omega_{s(\mathbf{x})}$ for all $s\in\mathcal{S}$. From this and properties (5.6) and (5.7) it follows that if the quadrature rule is exact for a function w, then it is exact for all equivalent functions in $\{w\}$. If we can describe a function space in the form of $\{w_1,w_2,\dots,w_N\}$, by which we mean the span of $w_1,w_2,\dots,w_N$ and all their equivalent versions, this means the quadrature rule is exact when it is exact for the N functions $w_1,w_2,\dots,w_N$.

To give an example, the degree-3 element, given in Table 5.2, consists of the nodes $(0,0,0)$, $(a,0,0)$, $(b,b,0)$, (c,c,c), and all equivalent nodes,

Table 5.3: Degree-4 mass-lumped tetrahedral element with 65 nodes.

Nodes	n	ω	Parameters
$\{(0,0,0)\}$	4	0.0001216042545112321	-
$\{(a,0,0)\}$	12	0.0004704124198744411	0.1724919407749086
$\{(\frac{1}{2},0,0)\}$	6	0.0001767065925083475	-
$\{(b_1,b_1,0)\}$	12	0.001974748586596177	0.1474177969013686
$\{(b_2,b_2,0)\}$	12	0.001192465311769701	0.4540395272271067
$\{(\frac{1}{3},\frac{1}{3},0)\}$	4	0.001044697597634123	-
$\{(c_1,c_1,c_1)\}$	4	0.008841425190569096	0.1282209316290979
$\{(d,d,\frac{1}{2}-d)\}$	6	0.006891012924401557	0.08742182088664353
$\{(c_2,c_2,c_2)\}$	4	0.007499563520517103	0.3124061452070811
$\{(\frac{1}{4},\frac{1}{4},\frac{1}{4})\}$	1	0.01057967149339721	-

$$U = \mathcal{P}_4 \oplus \mathcal{B}_f(\mathcal{P}_2 \oplus \mathcal{B}_f) \oplus \mathcal{B}_e(\mathcal{P}_2 \oplus \mathcal{B}_f \oplus \mathcal{B}_e)$$
$$= \{x_1, x_1^2x_2, x_1^2x_2^2, \beta_f x_1, \beta_f x_1x_2, \beta_f^2, \beta_e x_1, \beta_e x_1x_2, \beta_e\beta_f, \beta_e^2\}$$

$$U \otimes \mathcal{P}_2 = \{x_1, x_1^2x_2, x_1^3x_2^2, x_1^3x_2^3, \beta_f x_1, \beta_f x_1^2x_2, \beta_f x_1^2x_2^2, \beta_f^2 x_1, \beta_f^2 x_1x_2, \dots$$
$$\dots, \beta_e x_1, \beta_e x_1^2x_2, \beta_e x_1^2x_2^2, \beta_e\beta_f x_1, \beta_e\beta_f x_1x_2, \beta_e^2 x_1, \beta_e^2 x_1x_2\}$$

and the function space for this element is given by $U = \mathcal{P}_3 \oplus \mathcal{B}_f\mathcal{P}_1 \oplus \mathcal{B}_e\mathcal{P}_1$, where $\mathcal{B}_f := \{\beta_f\} := \{x_1x_2x_3\}$ are the face bubble functions and $\mathcal{B}_e := \{\beta_e\} := \{x_1x_2x_3x_4\}$ is the internal bubble function and where we used the notation $\mathcal{B}_f\mathcal{P}_k := \mathcal{B}_f \otimes \mathcal{P}_k$, $\mathcal{B}_e\mathcal{P}_k := \mathcal{B}_e \otimes \mathcal{P}_k$. The quadrature rule should be exact for all functions in $U \otimes \mathcal{P}_1$, which can be written as

$$U \otimes \mathcal{P}_1 = \{x_1, x_1^2x_2, x_1^2x_2^2, \beta_f x_1, \beta_f x_1x_2, \beta_e x_1, \beta_e x_1x_2\},$$

so as the span of 7 independent functions and all their equivalents. This means the quadrature rule should be exact for these 7 functions. Since this quadrature rule also has 7 parameters, namely 4 weights and 3 position parameters a, b, c, this results in a system of 7 equations with 7 unknowns. Solving this system results in the parameters given in Table 5.2.

This approach has also been used to obtain the other elements presented in this chapter. We have not yet found a systematic way to determine a suitable function space $U \supset \mathcal{P}_p$ with a suitable configuration of the nodes.

Instead, we just tried multiple configurations and checked if the resulting weights were all positive and the resulting nodes all lay on the reference triangle.

The degree-2 element with 15 nodes, the degree-3 element with 32 nodes, and the degree-4 element with 65 nodes are given in Tables 5.1, 5.2, and 5.3, respectively. In these tables, n denotes the number of nodes in the given equivalence class. Variants of the degree-4 element, requiring only 60 and 61 nodes, are given in Section 5.B.

In the next sections we test these new mass-lumped elements and compare them with the current mass-lumped elements and several discontinuous Galerkin approximations.

5.6 Dispersion analysis

In this section we analyze the dispersion properties of the mass-lumped elements. The dispersion error is measured by the difference between the propagation speed of physical and numerical waves and is one of the main criteria to judge the quality of the finite elements for wave propagation modelling. We will use it to obtain an indication of the required mesh resolution for a given accuracy, and to compare different finite element methods in terms of accuracy and numerical cost.

For the analysis we will follow the same procedure as in [30]. We consider a homogeneous medium with $\rho, c = 1$ and consider physical plane waves of the form

$$u = e^{\hat{\imath}(\boldsymbol{\kappa}\cdot\mathbf{x}-\omega t)},$$

where $\hat{\imath} := \sqrt{-1}$ is the imaginary number, $\boldsymbol{\kappa}$ is the wave vector, and ω is the angular velocity. Since $\rho, c = 1$ we have a wave propagation speed $c_P = 1$. For a given wave vector $\boldsymbol{\kappa}$ we compute all corresponding numerical plane waves and determine the numerical wave with a propagation speed $c_{P,h} = \omega_h/|\boldsymbol{\kappa}|$ closest to the physical wave velocity. The dispersion error is defined as the relative difference $(c_P - c_{P,h})/c_P$. We then find the worst case among all possible wave directions for a fixed wavelength $\lambda = 2\pi/|\boldsymbol{\kappa}|$. We determine the dispersion error for different wavelengths and extrapolate the results to obtain a relation between the dispersion error and number of elements per wavelength.

To obtain the numerical plane waves we construct a periodic tetrahedral mesh by packing a single parallelepiped cell with tetrahedra and then repeating this pattern to fill the entire 3D space. An illustration of such a

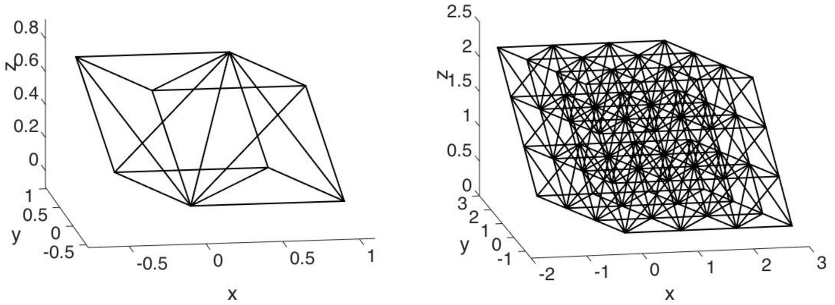


Figure 5.1: Single parallelepiped cell packed with tetrahedra (left), and a repeated pattern of these cells (right).

mesh is given in Figure 5.1. Such a periodic mesh enables us to compute the numerical plane waves using Fourier modes and by solving an eigenvalue problem related to a single cell.

To do this, let Ω_0 be the parallelepiped cell at the origin. We can write $\Omega_0 := \mathbf{T} \cdot [0,1)^3 = \{y \mid y = \mathbf{T} \cdot \mathbf{x}\text{for some } \mathbf{x} \in [0,1)^3\}$, with $\mathbf{T} \in \mathbb{R}^{3\times 3}$ the second-order tensor whose columns are the vectors of the edges of Ω_0 connected to the origin. Let $\{\mathbf{x}^{(\Omega_0,i)}\}_{i=1}^{n_0}$ be the set of nodes on Ω_0. For each $\mathbf{k} \in \mathbb{Z}^3$, we define the translated cell $\Omega_{\mathbf{k}} := \mathbf{T}\cdot\mathbf{k}+\Omega_0$ and let $\{\mathbf{x}^{(\Omega_{\mathbf{k}},i)}\}_{i=1}^{n_0}$ be the corresponding translated nodes. Then, for each node $\mathbf{x}^{(\Omega_{\mathbf{k}},i)}$, we define $w^{(\Omega_{\mathbf{k}},i)}$ to be the corresponding nodal basis function. We can then define the following submatrices:

$$\begin{aligned} M_{ij}^{(\Omega_0)} &:= \left(\rho w^{(\Omega_{\mathbf{0}},i)}, w^{(\Omega_{\mathbf{0}},j)}\right)_{\mathcal{Q}_h}, \quad && i,j = 1,\ldots,n_0, \\ A_{ij}^{(\Omega_0,\Omega_{\mathbf{k}})} &:= a\left(w^{(\Omega_{\mathbf{0}},i)}, w^{(\Omega_{\mathbf{k}},j)}\right), && \mathbf{k} \in \{-1,0,1\}^3,\ i,j = 1,\ldots,n_0. \end{aligned}$$

For each wave vector $\boldsymbol{\kappa}$ we then define the matrix

$$S^{(\boldsymbol{\kappa})} := M_{inv}^{(\Omega_0)} \left(\sum_{\mathbf{k}\in\{-1,0,1\}^3} e^{\hat{\imath}(\boldsymbol{\kappa}\cdot\mathbf{T}\cdot\mathbf{k})} A^{(\Omega_0,\Omega_{\mathbf{k}})} \right)$$

where $M_{inv}^{(\Omega_0)}$ denotes the inverse of $M^{(\Omega_0)}$. For an order-$2K$ Dablain scheme, with time step size Δt, the angular frequencies of the numerical plane waves $\{\omega_h^{(\boldsymbol{\kappa},i)}\}_{i=1}^{n_0}$ are given by

$$\omega_h^{(\boldsymbol{\kappa},i)} = \pm\frac{1}{\Delta t} \arccos\left(\sum_{k=0}^{K} \frac{1}{(2k)!} (-\Delta t^2 s_h^{(\boldsymbol{\kappa},i)})^k \right),$$

where $\{s_h^{(\boldsymbol{\kappa},i)}\}_{i=1}^{n_0}$ are the eigenvalues of $\sigma(S^{(\boldsymbol{\kappa})})$ [30]. The numerical wave propagation speed is given by $c_{P,h}^{(\boldsymbol{\kappa},i)} = |\omega_h^{(\boldsymbol{\kappa},i)}|/|\boldsymbol{\kappa}|$. The dispersion error, for a given wavelength λ, is then given by

$$e_{disp}(\lambda) := \sup_{\boldsymbol{\kappa}\in\mathbb{R}^3, |\boldsymbol{\kappa}|=2\pi/\lambda} \left(\inf_{i=1,\dots,n_0} \frac{|c_{P,h}^{(\boldsymbol{\kappa},i)} - c_P|}{c_P} \right).$$

For our dispersion analysis, we will consider a congruent, nearly regular, equifacial mesh, known as the tetragonal disphenoid honeycomb. This mesh can be obtained by a repeated pattern of cells, where a single cell can be obtained by slicing the unit cube into six tetrahedra with the planes $x_1 = x_2$, $x_1 = x_3$, and $x_2 = x_3$, and then applying the transformation $\mathbf{x} \to \mathbf{T}\cdot\mathbf{x}$, with

$$\mathbf{T} := \begin{bmatrix} 1 & -1/3 & -1/3 \\ 0 & \sqrt{8/9} & -\sqrt{2/9} \\ 0 & 0 & \sqrt{2/3} \end{bmatrix}.$$

We will analyze the relation between the dispersion error and the number of elements per wavelength $N_E := (\lambda^3/|e|_{av})^{1/3}$, where $|e|_{av} = 2\sqrt{3}/27$ denotes the average element volume. We will also look at the following quantities:

- $n_{vec} = n_0 \frac{\lambda^3}{|\Omega_0|}$: the number of degrees of freedom per λ^3-volume. Here $|\Omega_0| = 4\sqrt{3}/9$ denotes the volume of Ω_0.
- $n_{mat} = \frac{\lambda^3}{|\Omega_0|}\sum_{q\in\mathcal{Q}_{\Omega_0}} |\mathcal{N}(q)|$: the number of nonzero entries of the stiffness matrix per λ^3-volume. Here, $\mathcal{Q}_{\Omega_0}$ denotes the nodes on Ω_0 and $|\mathcal{N}(q)|$ denotes the number of nodes connected with q through an element.
- $N_{\Delta t} = T_0/\Delta t$: the number of time steps during one oscillation in time. Here $T_0 = \lambda/c_P$ denotes the duration of one oscillation and $\Delta t = \sqrt{c_K/s_{h,max}}$ is the largest allowed time step size for the order-$2K$ Dablain scheme, with c_K a constant depending on the order of the time integration scheme ($c_K = 4, 12, 7.57, 21.48$ for $K = 1, 2, 3, 4$, respectively) and

$$s_{h,max} := \sup_{\mathbf{k}\in\mathcal{K}_0} \max_{i=1,\dots,n_0} s_h^{(\boldsymbol{\kappa},i)}$$

the largest possible eigenvalue s_h, with $\mathcal{K}_0 = \mathbf{T}^{-t}\cdot[0, 2\pi)$ the space of distinct wave vectors.

- $n_{comp} = n_{mat} K N_{\Delta t}$: the estimated number of computations per λ^3-volume during one time oscillation, with K the number of stages of the order-$2K$ Dablain scheme.

Details on the dispersion analysis and how the quantities listed above are computed can be found in [30].

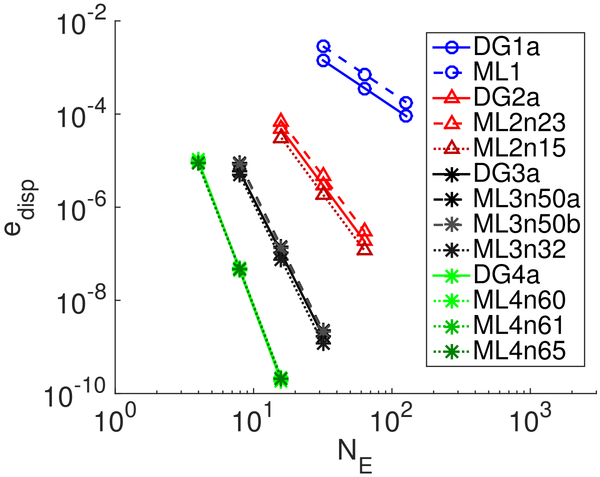


Figure 5.2: Relation between the dispersion error and number of elements per wavelength for different mass-lumped finite element methods. The graphs of ML3n50a and ML3n50b, and the graphs of the degree-four methods are almost identical.

Table 5.4: Approximation of the dispersion error. The new mass-lumped methods are marked in bold.

Method	e_{disp}	Method	e_{disp}
DG1	$1.45(n_E)^{-2}$	DG2	$3.00(n_E)^{-4}$
ML1	$2.87(n_E)^{-2}$	ML2n23	$4.82(n_E)^{-4}$
		ML2n15	$1.89(n_E)^{-4}$
DG3	$1.77(n_E)^{-6}$	DG4	$0.739(n_E)^{-8}$
ML3n50a	$2.25(n_E)^{-6}$	**ML4n60**	$0.865(n_E)^{-8}$
ML3n50b	$2.15(n_E)^{-6}$	**ML4n61**	$0.854(n_E)^{-8}$
ML3n32	$1.19(n_E)^{-6}$	**ML4n65**	$0.825(n_E)^{-8}$

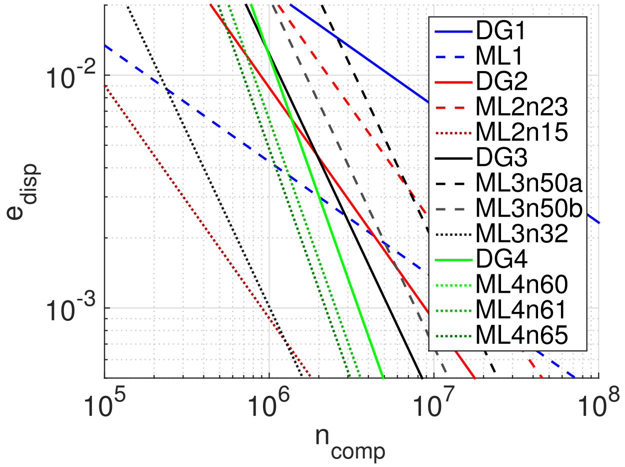


Figure 5.3: Relation between the dispersion error and estimated computational cost for different finite element methods. The new mass-lumped methods are illustrated with dotted lines. The graphs of DG4 and ML4n60 are almost identical.

We will refer to the standard linear mass-lumped finite element method as ML1. The higher-order mass-lumped methods will be referred to as ML[p]n[n], where p is the degree and n the number of nodes per element. In particular, the degree-2 method of [51] will be referred to as ML2n23 and the two versions of the degree-3 method in [11] will be referred to as ML3n50a and ML3n50b. The mass-lumped methods introduced in this chapter will be referred to as ML2n15, ML3n32, ML4n60, ML4n61, and ML4n65.

We will also compare the mass-lumped methods with the symmetric interior penalty discontinuous Galerkin (SIPDG) method, introduced and analyzed in [36]. For the penalty term, we use the lower bound derived in [32], since it was shown in [30] that this penalty term results in a significantly more efficient scheme than the penalty terms based on the more commonly used trace inequality of [79]. The quantities for the computational cost are computed in the same way as for the mass-lumped method, but now n_0 denotes the number of basis functions in Ω_0 and n_{mat} is computed as $n_{mat} = \frac{\lambda^3}{|\Omega_0|}\sum_{e\in\mathcal{T}_{\Omega_0}} |U_e|^2|\mathcal{N}(e)|$, where $\mathcal{T}_{\Omega_0}$ are the elements in Ω_0, $|U_e|$ are the number of basis functions per element, and $|\mathcal{N}(e)| = 5$ are the number of elements connected with e through a face, including e itself. We

	$e_{disp} = 0.001$				
Method	N_E	n_{vec}	n_{mat}	$N_{\Delta t}$	n_{comp}
DG1	38.0	220000	4400×10^3	120	540.00×10^6
ML1	54.0	26000	390×10^3	47	18.00×10^6
DG2	7.4	4100	200×10^3	22	8.80×10^6
ML2n23	8.3	4800	220×10^3	52	23.00×10^6
ML2n15	6.6	1200	39×10^3	11	0.90×10^6
DG3	3.5	840	84×10^3	21	5.30×10^6
ML3n50a	3.6	1200	98×10^3	52	15.00×10^6
ML3n50b	3.6	1100	96×10^3	27	7.70×10^6
ML3n32	3.2	430	26×10^3	13	1.00×10^6
DG4	2.3	420	73×10^3	12	3.50×10^6
ML4n60	2.3	370	38×10^3	23	3.50×10^6
ML4n61	2.3	390	39×10^3	16	2.50×10^6
ML4n65	2.3	410	44×10^3	13	2.20×10^6

Table 5.5: Number of elements per wavelength N_E, number of degrees of freedom n_{vec}, size of the global matrix n_{mat}, number of time steps $N_{\Delta t}$, and computational cost n_{comp} for a dispersion error of 0.001 for different finite element methods for the scalar wave equation. The new mass-lumped methods are marked in bold. The numbers are accurate up to two decimal places.

will refer to the SIPDG methods of degrees 1, 2, 3, and 4 as DG1, DG2, DG3, and DG4, respectively.

For the time integration, we combine a degree-p finite element method with an order-$2p$ Dablain time integration scheme, since this results in order-$2p$ convergence of the dispersion error.

Figure 5.2 illustrates the relation between the dispersion error and number of elements per wavelength. The dispersion error of the finite element methods converges with order $2p$, which is typical for symmetric finite element methods for eigenvalue approximations; see, for example, [9] and the references therein. Using extrapolation, we obtain formulas for the dispersion error, given in Table 5.4. These formulas can be used to determine the required resolution of the mesh given the wavelength and desired accuracy. From the leading constants we can see that the new mass-lumped methods of degrees 2 and 3 are more accurate for the same mesh resolution than the SIPDG and existing mass-lumped methods of these orders. The degree-4 mass-lumped method with 65 nodes is slightly more accurate than the versions using 60 and 61 nodes but is slightly less accurate than the degree-4 discontinuous Galerkin method.

While some methods are more accurate for the same mesh resolution, this does not necessarily mean that these methods are more efficient, since the computational cost per element can greatly differ per method. To get an idea which method is most efficient for a given accuracy, we also look at the relation between the dispersion error and the estimated computational cost. This relation is illustrated in Figure 5.3. The required computational cost of each method for a dispersion error of 0.001 is also illustrated in Table 5.5.

These results show that the new degree-2 mass-lumped method significantly outperforms the other degree-2 finite element methods, reducing the required computational cost for a given accuracy by one order of magnitude. The new degree-3 method is also significantly more efficient than the other degree-3 methods, reducing the required computational cost by more than a factor 5. These reductions in computational cost can be explained by the improved accuracy for the same mesh resolution, a reduction in the number of degrees of freedom and therefore reduction of the size of the stiffness matrix, and by a smaller number of time steps due to a larger allowed time step size.

Among the degree-4 finite element methods, the mass-lumped method using 65 nodes performs best, mainly due to a smaller number of required time steps, although these differences are relatively small.

Figure 5.3 also indicates that for a dispersion error between 1% and 0.1%, the new degree-2 mass lumped method performs best, while for smaller dispersion errors, the new degree-3 mass lumped method is most efficient. When we extrapolate the graphs, we find that the degree-4 mass-lumped method using 65 nodes will only outperform the degree-3 method for a dispersion error below 10^{-5}.

While the dispersion analysis provides useful information on the efficiency of the numerical methods, it does not include the effect of interpolation errors or inaccurate higher-frequency modes that may contaminate the numerical solution. Furthermore, the estimated computational cost is no perfect measure for the computation time, since the real computation time heavily depends on the implementation of the algorithm and the hardware that is used. In the next section we therefore also show the results of several numerical tests for the mass-lumped methods.

5.7 Numerical tests

5.7.1 Homogeneous domain

We first tested and compared the old and new mass-lumped tetrahedral element methods on a homogeneous acoustic model using unstructured tetrahedral meshes. The domain is $[-2, 2] \times [-1, 1] \times [0, 2]\,\mathrm{km}^3$ and the acoustic wave propagation speed is $c_P := 2\,\mathrm{km/s}$. A 3.5-Hz Ricker wavelet, starting from the peak, was placed at $\mathbf{x}_{src} := (0, 0, 1000)\,\mathrm{m}$, and 56 receivers were placed on a line between $x_r = -1375$ and $x_r = +1375\,\mathrm{m}$ with a 50-m interval at $y_r = 0\,\mathrm{m}$ and $z_r = 800\,\mathrm{m}$. Data were recorded for 0.6 s, counting from the time at which the wavelet peaked, but the computations already started at the negative time -0.6 s when the wavelet is approximately zero.

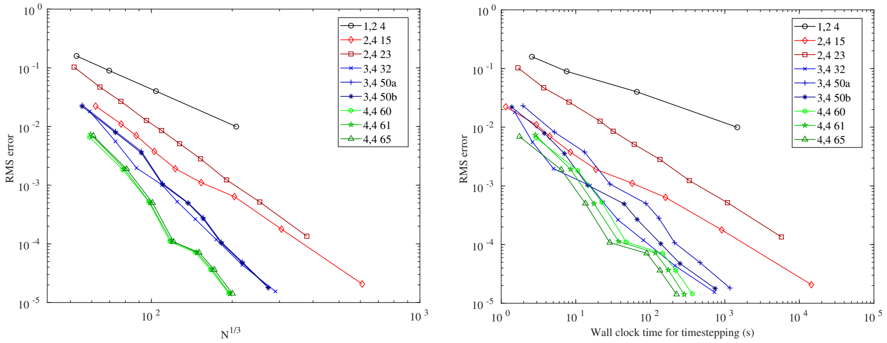


Figure 5.4: RMS errors as a function of the cube root of the number of degrees of freedom (left) and as a function of the wall clock time (right). In the legend, $p, K\ n$ refers to the element of degree p with n nodes, combined with a K-order time-stepping scheme. The older elements, apart from the one with degree 1, have degree 2 with 23 nodes and degree 3 with 50 nodes and two variants called a and b.

The exact solution, in case of an unbounded domain, is given by

$$u(\mathbf{x}, t) = \frac{w(t - r/c_P)}{4\pi r}, \qquad\qquad t \leq 0.6\,\mathrm{s},$$

where $r := |\mathbf{x} - \mathbf{x}_{src}|$ is the distance to the source and $w(t) := (1 - 2\pi^2 f^2 t^2)e^{-\pi^2 f^2 t^2}$ the Ricker-wavelet of peak frequency $f = 3.5\,\mathrm{Hz}$. This wavelet is zero up to machine precision for $t \leq -0.6\,\mathrm{s}$. For the bounded domain, on which we imposed zero Neumann boundary conditions, we add mirror sources to handle the reflections caused by the boundary conditions.

Table 5.6: Linear fits of the left graph of Figure 5.4.

Method	RMS error	Method	RMS error
1,2 4	$(4.9 \times 10^2)N^{(-1/3\times2.0)}$	2,4 15	$(4.4 \times 10^3)N^{(-1/3\times3.0)}$
		2,4 23	$(4.9 \times 10^4)N^{(-1/3\times3.3)}$
3,4 32	$(9.2 \times 10^5)N^{(-1/3\times4.4)}$	4,4 60	$(8.6 \times 10^6)N^{(-1/3\times5.1)}$
3,4 50a	$(2.9 \times 10^6)N^{(-1/3\times4.6)}$	4,4 61	$(1.3 \times 10^7)N^{(-1/3\times5.2)}$
3,4 50b	$(2.6 \times 10^6)N^{(-1/3\times4.6)}$	4,4 65	$(1.2 \times 10^7)N^{(-1/3\times5.2)}$

For the implementation of the mass-lumped methods we used the algorithm described in [54]. The time step size is based on the estimates of [55] multiplied by a factor 0.9. The simulations were carried out with OpenMP on 24 cores of two Intel® Xeon® E5-2680 v3 CPUs running at 2.50 GHz. Figure 5.4 shows the observed root mean square (RMS) errors of the receiver data for the various schemes against the number of degrees of freedom N and against wall clock time. The latter should not be taken too literally because it depends on code implementation, compiler and hardware, and even varies between runs. It can be further reduced by going to single precision, but then it becomes more difficult to measure the errors when they become small. Therefore, we ran a double-precision version of the code when preparing these figures.

Fourth-order time-stepping was used for degrees higher than one [18]. For degree 4, we also considered sixth-order time-stepping, but the errors were nearly the same as with fourth-order time-stepping for the current example.

Power-law fits, given in Table 5.6, show that the RMS errors converge with approximately order $p+1$ and confirm that the new elements maintain an optimal order of convergence. Figure 5.4 also shows that the new mass-lumped methods require fewer degrees of freedom and less computation time for the same accuracy. For the degree-2 methods, the difference in wall clock time is up to one order of magnitude, while for the degree-3 methods this difference is up to a factor 2. The degree-4 methods become more efficient for errors below 10^{-3}.

5.7.2 Elastic salt model

We also tested the methods on the more realistic salt model from [44], made elastic by replacing the water layer at the top by rock. A 3-Hz Ricker wavelet vertical force source was placed on the surface at (2000,2200,0) m and 25 receivers were placed on a line between $x_r = -1012.5$ and $x_r = 7887.5$ m with a 25-m interval at $y_r = 2200$ m and $z_r = 0$ m. An illustration of this salt model is given in Figure 5.5. Figure 5.6 displays vertical cross sections through the 3D vertical displacement wavefield of the 65-node degree-4 method, clipped at 25% of its maximum amplitude with red for positive and blue for negative values. Small amplitudes were replaced by the P-velocity to give an impression of the model. Figure 5.7 also shows seismograms of this method for the displacement in the x- and z-directions clipped at 2% of the maximum amplitude.

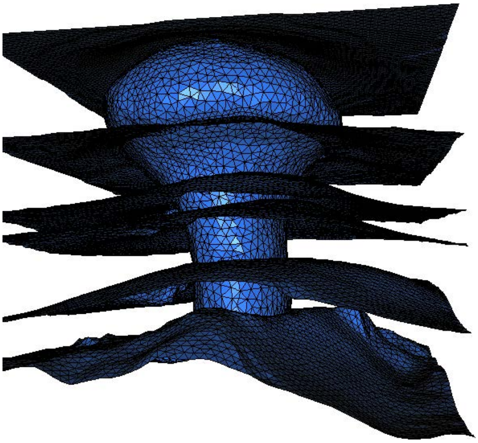
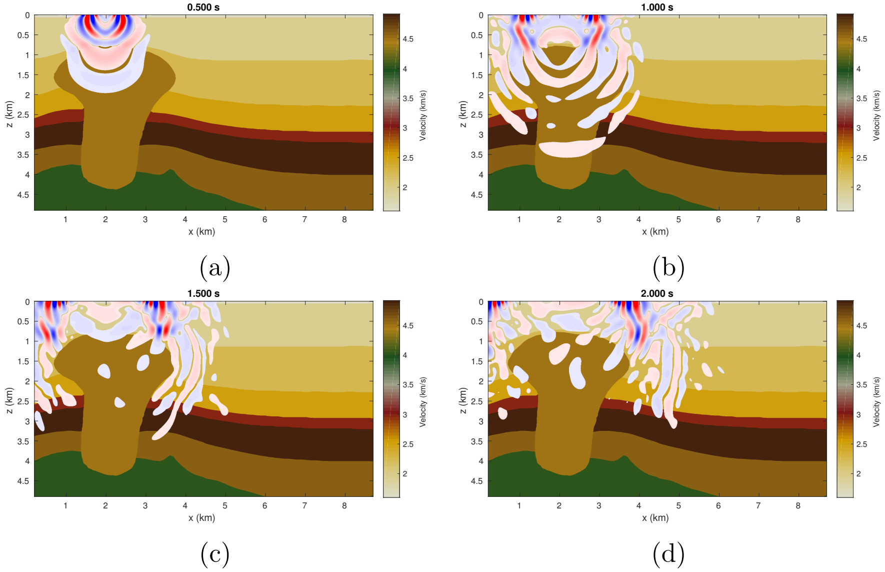


Figure 5.5: Interfaces of the salt model taken from [44].

Simulations were carried out with the same implementation and in the same environment as for the homogeneous test case. The RMS errors are estimated by taking the traces for the 65-node degree-4 method as the "exact" solution. For the RMS error we use the data of receivers between $x = 2.1$ and $x = 4$ km in the time interval $[0, 1.8]$ s. We selected this subset to exclude errors caused by the absorbing boundary layers. To compute the relative RMS error, we divide by the RMS of the data.

An overview of the RMS errors and the wall clock time is given in Table 5.7. The differences in the traces of the different degree-4 methods is of order 10^{-5} and is much smaller than the estimated errors of the lower-degree elements. This indicates that the RMS errors of the degree-4 methods are of order 10^{-5} and supports the idea that the degree-4 method can be used to estimate the accuracy of the lower-degree methods in this case. The table illustrates again that the new degree-2 and degree-3 mass-lumped



(a) (b)

(c) (d)

Figure 5.6: Vertical cross section at $y = 2.2$ km of the P-velocity with a superimposed snapshot of the vertical displacement of the 65-node degree-4 method after 0.5 s (a), 1.0 s (b), 1.5 s (c) and 2 s (d).

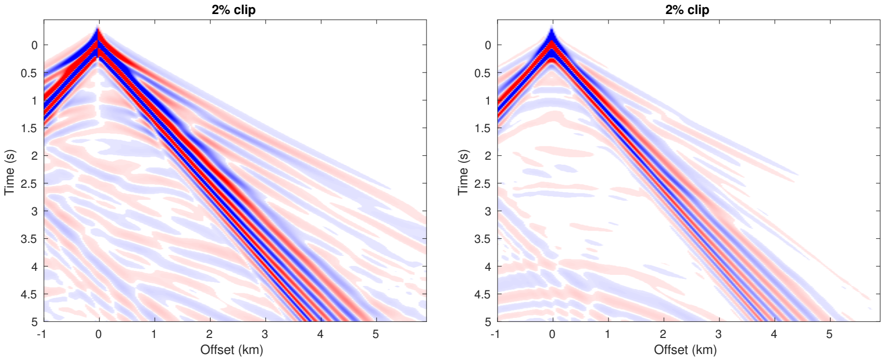


Figure 5.7: Seismogram of the displacement in the x- (left) and z-direction (right) for the 65-node degree-4 method.

tetrahedral elements are more efficient than the current ones. The differences in computation time between the degree-4 methods is mainly due to the difference in the number of time steps. The ML4n65 allows for a larger time step size than the other variants, which makes it slightly more efficient.

Table 5.7: Estimated relative RMS error of the displacement in the x and z-directions and measured wall clock time. The error is estimated by computing the difference with the ML4n65T4 data. ML[p]n[n]T[K] refers to an element of degree p, with n nodes per element, combined with a time-stepping scheme of order K. There are two versions of the ML3n50 element. ML1T2 is also tested on two refined meshes. New mass-lumped methods are marked in bold.

Method	RMS-x	RMS-z	time (s)
ML1T2	0.50	0.42	324
	0.36	0.39	4230
	0.12	0.11	47962
ML2n15T2	0.044	0.048	7358
ML2n23T2	0.058	0.064	28950
ML3n32T4	0.0014	0.0015	41946
ML3n50aT4	0.0016	0.0017	312149
ML3n50bT4	0.0016	0.0017	177312
ML4n60T4	0.000017	0.000019	613945
ML4n61T4	0.000013	0.000014	336362
ML4n65T4	0	0	275933

5.8 Conclusion

We developed a less restrictive accuracy condition for the construction of continuous mass-lumped elements, which enabled us to construct several new tetrahedral elements. The new degree-2 and degree-3 tetrahedral elements require 15 and 32 nodes, while the current versions require 23 and 50 nodes per element, respectively. These new elements require fewer degrees of freedom and allow larger time steps than the current versions. We also developed degree-4 tetrahedral elements with 60, 61, and 65 nodes per element. Mass-lumped tetrahedral elements of this degree had not been found until now.

A dispersion analysis and numerical examples confirm that the new mass-lumped methods maintain an optimal order of accuracy and show that the new elements are significantly more efficient than the existing ones. In particular, the new degree-2 method is shown to be up to one order of magnitude faster than the current method, while the new degree-3 tetrahedral element results in a speed-up of up to a factor 2 for the same accuracy. The new degree-4 elements outperform the lower-degree elements for an accuracy below 10^{-3}. Among these degree-4 elements, the one with 65 nodes is the most efficient, which is mainly due to a larger allowed time step size. The dispersion analysis also shows that the new degree-2 and degree-3 mass-lumped methods require significantly fewer degrees of freedom and a smaller number of time steps than the symmetric interior penalty discontinuous Galerkin methods of the same degree.

We have only considered tetrahedral elements in this chapter, but the accuracy condition might also lead to more efficient triangular or higher-dimensional simplicial elements. Furthermore, although we focused only on linear wave propagation problems, mass lumping is useful for solving any type of evolution problem that requires explicit time-stepping.

5.A Nodal basis functions

Lemma 5.A.1. *Let e be a d-simplex in $\mathbb{R}^d$, with $d \geq 0$, and let $s : \mathbb{R}^d \to \mathbb{R}^d$ be an affine mapping that maps e onto itself. Then s can be represented by a permutation of the barycentric coordinates of e. In particular, there exists a permutation function $P : \mathbb{R}^{d+1} \to \mathbb{R}^{d+1}$ such that*

$$s(\mathbf{x})^* = P(\mathbf{x}^*) \qquad\qquad \textit{for all } \mathbf{x} \in \mathbb{R}^d, \tag{5.54}$$

where $\mathbf{x}^*, s(\mathbf{x})^* \in \mathbb{R}^{d+1}$ denote the barycentric coordinates of $\mathbf{x}$ and $s(\mathbf{x})$, respectively.

Proof. Note that both $\mathbf{x} \to s(\mathbf{x})^*$ and $\mathbf{x} \to P(\mathbf{x}^*)$ are affine mappings from $\mathbb{R}^d$ to $\mathbb{R}^{d+1}$. It therefore suffices to show that (5.54) holds for all vertices of e. To do this, let $\mathbf{v}_i \in \mathbb{R}^d$, for $i = 1, \dots, d+1$, denote the vertices of e, and let p be the permutation of $(1, \dots, d+1)$ such that $s(\mathbf{v}_i) = \mathbf{v}_{p(i)}$. If we define P such that $P(\mathbf{e}_i) = \mathbf{e}_{p(i)}$, with $\mathbf{e}_i \in \mathbb{R}^{d+1}$ the unit vector in direction i, then

$$\begin{aligned} s(\mathbf{v}_i)^* &= \mathbf{v}_{p(i)}^* \\ &= \mathbf{e}_{p(i)} \\ &= P(\mathbf{e}_i) \\ &= P(\mathbf{v}_i^*) \end{aligned}$$

for all $i = 1, \dots, d+1$. □

Lemma 5.A.2. *Let $\mathcal{Q}_h$ be defined as in Section 5.3.1, and let $\mathbf{x} \in \overline{e}$, for some $\mathbf{x} \in \mathcal{Q}_h, e \in \mathcal{T}_h$. If (5.7a) is satisfied, then $\phi_e^{-1}(\mathbf{x}) \in \tilde{\mathcal{Q}}$.*

Proof. By definition of $\mathcal{Q}_h$, there exists an element $e^* \in \mathcal{T}_h$ and node $\tilde{\mathbf{x}}^* \in \tilde{\mathcal{Q}}$ such that $\mathbf{x} = \phi_{e^*}(\tilde{\mathbf{x}}^*)$, so $\phi_{e^*}^{-1}(\mathbf{x}) = \tilde{\mathbf{x}}^* \in \tilde{\mathcal{Q}}$. Now construct a reference-to-reference element mapping $s \in \mathcal{S}$ such that $s|_{\phi_{e^*}^{-1}(\overline{e}^* \cap \overline{e})} = \phi_e^{-1} \circ \phi_{e^*}|_{\phi_{e^*}^{-1}(\overline{e}^* \cap \overline{e})}$. Then, using (5.7a), we can obtain $\phi_e^{-1}(\mathbf{x}) = s \circ \phi_{e^*}^{-1}(\mathbf{x}) = s(\mathbf{x}^*) \in \tilde{\mathcal{Q}}$. □

Theorem 5.A.3. *Let $\mathcal{Q}_h$ be defined as in Section 5.3.1, and let $\mathbf{x} \in \mathcal{Q}_h$. If the conditions in (5.6) and (5.7) are satisfied, then the basis function $w_{\mathbf{x}}$, given in (5.5), is well defined and continuous and satisfies*

$$w_{\mathbf{x}}(\mathbf{y}) = \delta_{\mathbf{x}\mathbf{y}} \qquad \textit{for all } \mathbf{y} \in \mathcal{Q}_h, \tag{5.55}$$

where $\delta_{\mathbf{x}\mathbf{y}}$ denotes the Kronecker delta function.

Proof. The fact that $w_{\mathbf{x}}$ is well defined follows immediately from Lemma 5.A.2.

To prove that $w_{\mathbf{x}}$ is continuous, let $f = \partial e^- \cap \partial e^+$ be any face adjacent to the elements $e^-, e^+ \in \mathcal{Q}_h$. It is sufficient to show that $w_{\mathbf{x}}|_{\partial e^+ \cap f} = w_{\mathbf{x}}|_{\partial e^- \cap f}$, where $w|_{\partial e \cap f}$ denotes the trace of function w restricted to e on face f. Suppose that $\mathbf{x} \notin \overline{e}^- \cup \overline{e}^+$. Then $w_{\mathbf{x}}|_{\partial e^- \cap f} = 0 = w_{\mathbf{x}}|_{\partial e^+ \cap f}$.

Now suppose that $\mathbf{x} \in \overline{e}^+ \setminus f$. Then $\mathbf{x} \notin \overline{e}^-$ and $\tilde{\mathbf{x}}^+ \notin \tilde{f}^+$, where $\tilde{\mathbf{x}}^+ := \phi_{e^+}^{-1}(\mathbf{x})$ and $\tilde{f}^+ := \phi_{e^+}^{-1}(f)$. From (5.6) it then follows that $w_{\mathbf{x}}|_{\partial e^+ \cap f} = \tilde{w}_{\tilde{\mathbf{x}}^+} \circ \phi_{e^+}^{-1}|_f = 0 = w_{\mathbf{x}}|_{\partial e^- \cap f}$.

Finally, suppose that $\mathbf{x} \in f$. Let $s \in \mathcal{S}$ be a reference-to-reference element mapping such that $s|_{\tilde{f}^+} = \phi_{e^-}^{-1} \circ \phi_{e^+}|_{\tilde{f}^+}$. We can then derive

$$\begin{aligned} w_{\mathbf{x}}|_{\partial e^+ \cap f} &= \tilde{w}_{\tilde{\mathbf{x}}^+} \circ \phi_{e^+}^{-1}|_f \\ &= \tilde{w}_{s(\tilde{\mathbf{x}}^+)} \circ s \circ \phi_{e^+}^{-1}|_f \\ &= \tilde{w}_{\phi_{e^-}^{-1}(\mathbf{x})} \circ \phi_{e^-}^{-1}|_f \\ &= w_{\mathbf{x}}|_{\partial e^- \cap f}, \end{aligned}$$

where the second line follows from (5.7b).

Since $w_{\mathbf{x}}$ is continuous, $w_{\mathbf{x}}(\mathbf{y})$ is well defined for any $\mathbf{y} \in \Omega$. To prove property (5.55), suppose there is an element $e \in \mathcal{T}_h$ such that $\mathbf{x}, \mathbf{y} \in \overline{e}$. Define $\tilde{\mathbf{x}} := \phi_e^{-1}(\mathbf{x})$ and $\tilde{\mathbf{y}} := \phi_e^{-1}(\mathbf{y})$. Then $w_{\mathbf{x}}(\mathbf{y}) = \tilde{w}_{\tilde{\mathbf{x}}}(\tilde{\mathbf{y}}) = \delta_{\tilde{\mathbf{x}}\tilde{\mathbf{y}}} = \delta_{\mathbf{x}\mathbf{y}}$.

Now suppose that there is no element e such that $\mathbf{x}, \mathbf{y} \in \overline{e}$. Then $\mathbf{x} \neq \mathbf{y}$ and there exists an element e such that $\mathbf{y} \in \overline{e}$ and $\mathbf{x} \notin \overline{e}$. By definition of $w_{\mathbf{x}}$ it then follows that $w_{\mathbf{x}}(\mathbf{y}) = 0 = \delta_{\mathbf{x}\mathbf{y}}$. □

5.B Variants of the degree-four mass-lumped tetrahedral element

Table 5.8: Degree-4 mass-lumped tetrahedral element with 60 nodes.

Nodes	n	ω	Parameters
$\{(0,0,0)\}$	4	0.000093191469557671 76	-
$\{(a,0,0)\}$	12	0.0004829332376473431	0.1614865833496676
$\{(\frac{1}{2},0,0)\}$	6	0.0002005503792135920	-
$\{(b_1,b_1,0)\}$	12	0.002003104085841525	0.1490219288469598
$\{(b_2,b_2,0)\}$	12	0.001126849366800016	0.3944591972171783
$\{(c_1,c_1,c_1)\}$	4	0.009159244489996298	0.1302058846372564
$\{(d,d,\frac{1}{2}-d)\}$	6	0.006725322654059780	0.06386116838612691
$\{(c_2,c_2,c_2)\}$	4	0.01118676108633598	0.3012179234079087

$$U = \mathcal{P}_4 \oplus \mathcal{B}_f\mathcal{P}_2 \oplus \mathcal{B}_e(\mathcal{P}_2 \oplus \mathcal{B}_f)$$

$$= \{x_1, x_1^2x_2, x_1^2x_2^2, \beta_f x_1, \beta_f x_1x_2, \beta_e x_1, \beta_e x_1x_2, \beta_e\beta_f\}$$

$$U \otimes \mathcal{P}_2 = \{x_1, x_1^2x_2, x_1^3x_2^2, x_1^3x_2^3, \beta_f x_1, \beta_f x_1^2x_2, \beta_f x_1^2x_2^2, \beta_f^2 x_1, \ldots$$

$$\ldots, \beta_e x_1, \beta_e x_1^2x_2, \beta_e x_1^2x_2^2, \beta_e\beta_f x_1, \beta_e\beta_f x_1x_2, \beta_e^2 x_1\}$$

Table 5.9: Degree-4 mass-lumped tetrahedral element with 61 nodes.

Nodes	n	ω	Parameters
$\{(0,0,0)\}$	4	0.0001593069370906064	-
$\{(a,0,0)\}$	12	0.0004461325181676239	0.2001628104707848
$\{(\frac{1}{2},0,0)\}$	6	0.0003715829945705960	-
$\{(b_1,b_1,0)\}$	12	0.001884294964657102	0.1397350972238366
$\{(b_2,b_2,0)\}$	12	0.001545425606069384	0.4319436235177682
$\{(c_1,c_1,c_1)\}$	4	0.008841425190569096	0.1282209316290979
$\{(d,d,\frac{1}{2}-d)\}$	6	0.006891012924401557	0.08742182088664353
$\{(c_2,c_2,c_2)\}$	4	0.007499563520517103	0.3124061452070811
$\{(\frac{1}{4},\frac{1}{4},\frac{1}{4})\}$	1	0.01057967149339721	-

$$U = \mathcal{P}_4 \oplus \mathcal{B}_f \mathcal{P}_2 \oplus \mathcal{B}_e(\mathcal{P}_2 \oplus \mathcal{B}_f \oplus \mathcal{B}_e)$$
$$= \{x_1, x_1^2x_2, x_1^2x_2^2, \beta_f x_1, \beta_f x_1x_2, \beta_e x_1, \beta_e x_1x_2, \beta_e\beta_f, \beta_e^2\}$$

$$U \otimes \mathcal{P}_2 = \{x_1, x_1^2x_2, x_1^3x_2^2, x_1^3x_2^3, \beta_f x_1, \beta_f x_1^2x_2, \beta_f x_1^2x_2^2, \beta_f^2 x_1, \dots$$
$$\dots, \beta_e x_1, \beta_e x_1^2x_2, \beta_e x_1^2x_2^2, \beta_e\beta_f x_1, \beta_e\beta_f x_1x_2, \beta_e^2 x_1, \beta_e^2 x_1x_2\}$$

Chapter 6

Efficient quadrature rules for computing the stiffness matrices of mass-lumped tetrahedral elements for linear wave problems[1]

Abstract

We present new and efficient quadrature rules for computing the stiffness matrices of mass-lumped tetrahedral elements for wave propagation modelling. These quadrature rules allow for a more efficient implementation of the mass-lumped finite element method and can handle materials that are heterogeneous within the element without loss of the convergence rate. The quadrature rules are designed for the specific function spaces of recently developed mass-lumped tetrahedra [Geevers, S., Mulder, W. A., & van der Vegt, J. J. W. (2018). New higher-order mass-lumped tetrahedral elements for wave propagation modelling. Accepted for publication in *SIAM Journal on Scientific Computing*. arXiv:1803.10065], which consist of standard polynomial function spaces enriched with higher-degree bubble functions. For the degree-2 mass-lumped tetrahedron, the most efficient quadrature rule seems to be an existing 14-point quadrature rule, but for tetrahedra of degrees 3 and 4, we construct new quadrature rules that require less integration points than those currently available in literature. Several numerical examples confirm that this approach is more efficient than computing the stiffness matrix exactly and that an optimal order of convergence is maintained, even when material properties vary within the element.

[1] Submitted to *SIAM Journal on Scientific Computing* as: Geevers, S., Mulder, W.A. & van der Vegt, J.J.W. Efficient quadrature rules for computing the stiffness matrices of mass-lumped tetrahedral elements for linear wave problems.

6.1 Introduction

Mass-lumped tetrahedral element methods are efficient methods for solving linear wave equations, such as the acoustic wave equation, the elastic wave equations, or Maxwell's equations, on complex 3D domains with sharp material interfaces [81]. They offer the same convergence rate and geometric flexibility as standard continuous tetrahedral element methods, but also allow for explicit time-stepping due to a diagonal mass matrix.

To obtain mass-lumped elements, Lagrangian basis functions are combined with an inexact quadrature rule for computing the mass matrix, with quadrature points that coincide with the basis function nodes. For hexahedral elements, mass-lumping is achieved using tensor-product basis functions and Gauss–Lobatto points, resulting in the well-known spectral element method [59, 63, 43]. For linear triangular and tetrahedral elements, mass-lumping is achieved with standard Lagrangian basis functions and a Newton–Cotes integration rule. For higher-degree triangular and tetrahedral elements, however, the element space needs to be enriched with higher-degree bubble functions in order to maintain stability and optimal order of convergence [29, 14]. The first higher-degree tetrahedral elements were developed in [51] for degree 2 and [11] for degree 3. Recently, we developed new mass-lumped tetrahedral elements of degrees 2 to 4 [31]. The new degree-2 and degree-3 elements require significantly less nodes than the earlier versions, while mass-lumped tetrahedra of degree 4 had not been found before. Because of the reduced number of nodes, these new mass-lumped elements are also much more efficient than the earlier versions [31] and are therefore more suitable for large-scale 3D simulations.

A question that remains is how to efficiently compute the stiffness matrix for these elements. When the material parameters are piecewise constant, the stiffness matrix can be evaluated exactly [54]. Alternatively, we can use a quadrature rule to approximate the stiffness matrix. This latter approach can significantly reduce the number of computations as we will demonstrate in Section 6.5. Moreover, it also allows us to handle material parameters that vary within the element without loss of convergence rate as we will prove in Section 6.4.

Finding an efficient quadrature rule for the stiffness matrix for mass-lumped tetrahedra is not straightforward. For hexahedral elements, the stiffness matrix can be approximated with the same Gauss–Lobatto quadrature rule that is used for the mass matrix, but for mass-lumped tetrahedra, using the quadrature rule of the mass matrix to also evaluate the stiffness matrix turns out to be inefficient or inaccurate. In this chapter, we therefore present new and efficient quadrature rules for computing the stiffness matrices for mass-lumped tetrahedra.

To obtain these quadrature rules, we show that the quadrature rule only needs to be exact for functions in the space $\mathcal{P}_{p-1} \otimes D\tilde{U}$, where $p \geq 2$ denotes the degree of the element, $\mathcal{P}_{p-1}$ denotes the space of polynomials up to degree $p-1$, and $D\tilde{U}$ denotes the space of partial derivates of functions in the element space. Since the mass-lumped tetrahedra contain higher-degree bubble functions, so does the space $\mathcal{P}_{p-1} \otimes D\tilde{U}$ that needs to be integrated exactly. Most quadrature rules in literature, however, are designed to be exact for spaces of the form $\mathcal{P}_k$. We could choose k equal to the highest polynomial degree that appears in $\mathcal{P}_{p-1} \otimes D\tilde{U}$, but the resulting number of quadrature points may then be suboptimal. Instead, we try to find quadrature rules that are exact for $\mathcal{P}_{p-1} \otimes D\tilde{U}$ with a minimal number of quadrature points.

For the degree-2 tetrahedral element, the most efficient quadrature rule still seems to be the 14-point rule of [38] that is accurate for polynomials up to degree 5. For the degree-3 element and the three degree-4 elements of 60, 61, and 65 nodes, however, we present new quadrature rules that require 21, 51, 60, and 60 points, respectively, while using a quadrature rule from the current literature would require 24, 61, 81, and 81 points [80].

This chapter is organised as follows. In Section 6.2, we introduce the mass-lumped finite element method, and in Section 6.3, we present our new quadrature rules for evaluating the stiffness matrix. We prove in Section 6.4 that the conditions used to obtain our quadrature rules result in optimal convergence rates. In Section 6.5, we show numerical examples demonstrating that using our numerical quadrature rules for evaluating the stiffness matrix is more efficient than evaluating the integrals exactly and that the convergence rate is not lost when material parameters vary within the elements. Finally, we summarise our main conclusions in Section 6.6.

6.2 The mass-lumped finite element method

To present and analyse the mass-lumped finite element method, we consider the scalar wave equation given by

$$\rho\partial_t^2 u = \nabla\cdot c\nabla u + f \qquad \text{in } \Omega\times(0,T), \tag{6.1a}$$

$$u = 0 \qquad \text{on } \partial\Omega\times(0,T), \tag{6.1b}$$

$$u|_{t=0} = u_0 \qquad \text{in } \Omega, \tag{6.1c}$$

$$\partial_t u|_{t=0} = v_0 \qquad \text{in } \Omega, \tag{6.1d}$$

where $\Omega\subset\mathbb{R}^3$ is the spatial domain, $(0,T)$ is the time domain, $u:\Omega\times(0,T)\to\mathbb{R}$ is the scalar field that needs to be solved, ∇ is the gradient operator, $f:\Omega\times(0,T)\to\mathbb{R}$ is the source term, $u_0,v_0:\Omega\to\mathbb{R}$ are the initial values, and $\rho,c:\Omega\to\mathbb{R}^+$ are positive spatial parameters. The spatial domain Ω is assumed to be a bounded open domain with Lipschitz boundary $\partial\Omega$, and the parameters ρ and c are assumed to be bounded by $\rho_0\leq\rho\leq\rho_1$ and $c_0\leq c\leq c_1$, with ρ_0,ρ_1,c_0,c_1 strictly positive constants.

To solve the scalar wave equation with a finite element method, we consider the weak formulation. Let $L^2(\Omega)$ denote the standard Lesbesque space of square-integrable functions on Ω, $H_0^1(\Omega)$ the standard Sobolov space of functions in $L^2(\Omega)$ that vanish on $\partial\Omega$ and have square-integrable weak derivatives, and $L^2(0,T;U)$, with U a Banach space, the Bochner space of functions $f:(0,T)\to U$ such that $\|f\|_U$ is square integrable on $(0,T)$. The weak formulation of (6.1) can then be written as finding $u\in L^2\big(0,T;H_0^1(\Omega)\big)$, with $\partial_t u\in L^2\big(0,T;L^2(\Omega)\big)$ and with $\partial_t(\rho\partial_t u)\in L^2\big(0,T;H^{-1}(\Omega)\big)$, such that

$$\langle\partial_t(\rho\partial_t u),w\rangle + (c\nabla u,\nabla w) = (f,w) \qquad \text{for all } w\in H_0^1(\Omega), \tag{6.2}$$

where $(\cdot,\cdot)$ denotes the standard $L^2(\Omega)$ inner-product and $\langle\cdot,\cdot\rangle$ denotes the pairing between $H^{-1}(\Omega)$ and $H_0^1(\Omega)$.

This weak form of the wave equation can be solved with the mass-lumped finite element method, which consists of the following components:

- a tetrahedral mesh $\mathcal{T}_h$, where h denotes the radius of the smallest sphere that can contain each element,

- a reference tetrahedron $\tilde{e}$ with reference space $\tilde{U}=\mathcal{P}_p\oplus\tilde{U}^+ := \{u \mid u=w+u^+ \text{ for some } w\in\mathcal{P}_p, u^+\in\tilde{U}^+\}$, where $\mathcal{P}_p$ denotes the space of polynomials of degree p or less and $\tilde{U}^+$ a space of higher-degree face and interior bubble functions,

- a set of reference nodes $\tilde{\mathcal{Q}}$ that can be used for both interpolation and quadrature on $\tilde{e}$,
- a set of quadrature weights $\{\tilde{\omega}_{\tilde{\mathbf{x}}}\}_{\tilde{\mathbf{x}}\in\tilde{\mathcal{Q}}}$.

Using these components, a finite element space can be constructed of the form

$$U_h = H_0^1(\Omega) \cap U(\mathcal{T}_h, \tilde{U}),$$

where

$$U(\mathcal{T}_h, \tilde{U}) := \{u \in H^1(\Omega) \mid u \circ \phi_e \in \tilde{U} \text{ for all } e \in \mathcal{T}_h\},$$

with $\phi_e : \tilde{e} \to e$ the reference-to-physical element mapping. The interpolation points are given by $\mathcal{Q}_h = \mathcal{Q}(\mathcal{T}_h, \tilde{\mathcal{Q}})$, where

$$\mathcal{Q}(\mathcal{T}_h, \tilde{\mathcal{Q}}) := \bigcup_{e\in\mathcal{T}_h} \bigcup_{\tilde{\mathbf{x}}\in\tilde{\mathcal{Q}}} \phi_e(\tilde{\mathbf{x}}),$$

and the $L^2(\Omega)$ inner-product is approximated by

$$(u, w) = \sum_{e\in\mathcal{T}_h} \frac{|e|}{|\tilde{e}|} \int_{\tilde{e}} \tilde{u}_e \tilde{w}_e \, d\tilde{x} \approx \sum_{e\in\mathcal{T}_h} \sum_{\tilde{\mathbf{x}}\in\tilde{\mathcal{Q}}} \frac{|e|}{|\tilde{e}|} \omega_{\tilde{\mathbf{x}}} \tilde{u}_e(\tilde{\mathbf{x}}) \tilde{w}_e(\tilde{\mathbf{x}}) =: (u, w)_{\mathcal{Q}_h},$$

with $|e|$ and $|\tilde{e}|$ the volume of e and $\tilde{e}$, respectively, and $\tilde{u}_e := u \circ \phi_e$, $\tilde{w}_e := w \circ \phi_e$.

The finite element method can then be formulated as finding $u_h : [0, T] \to U_h$ such that $u_h|_{t=0} = I_h u_0$, $\partial_t u_h|_{t=0} = I_h v_0$, and

$$(\rho \partial_t^2 u_h, w)_{\mathcal{Q}_h} + (c\nabla u_h, \nabla w) = (f, w)_{\mathcal{Q}_h} \qquad \text{for all } w \in U_h, \tag{6.3}$$

where I_h is the interpolation operator that interpolates a continuous function at the points $\mathcal{Q}_h$ by a function in $U(\mathcal{T}_h, \tilde{U})$.

Now let $\{\mathbf{x}_i\}_{i=1}^N$ be the set of all interpolation points $\mathcal{Q}_h$ that do not lie on the boundary $\partial\Omega$, and define nodal basis functions $\{w_i\}_{i=1}^N$ such that $w_i(\mathbf{x}_j) = \delta_{ij}$ for all $i, j = 1, \ldots, N$, with δ the Kronecker delta. Also define, for any continuous function $u \in \mathcal{C}(\overline{\Omega})$, the interpolation vector $\underline{\mathbf{u}} \in \mathbb{R}^N$ such that $\underline{\mathbf{u}}_i := u(\mathbf{x}_i)$ for all $i = 1, \ldots, N$. The finite element method can then be formulated as finding $\underline{\mathbf{u}}_h : [0, T] \to \mathbb{R}^N$ such that $\underline{\mathbf{u}}_h|_{t=0} = \underline{u_0}$, $\partial_t \underline{\mathbf{u}}_h|_{t=0} = \underline{v_0}$, and

$$\partial_t^2 \underline{\mathbf{u}}_h + M^{-1} A \underline{\mathbf{u}}_h = \underline{\mathbf{f}}. \tag{6.4}$$

Here, $M \in \mathbb{R}^{N\times N}$, with $M_{ij} := (\rho w_i, w_j)_{\mathcal{Q}_h}$, is the mass matrix, and $A \in \mathbb{R}^{N\times N}$, with $A_{ij} := (c\nabla w_i, \nabla w_j)$, is the stiffness matrix.

Since the interpolation points and quadrature points coincide, the mass matrix is diagonal with entries $M_{ii} = (\rho w_i, 1)_{\mathcal{Q}_h}$. Therefore, we can efficiently solve the system of ODE's in (6.4) using an explicit time-stepping scheme. Standard conforming finite element methods do not result in a (block)-diagonal mass matrix and are therefore less suitable for solving wave equations on large three-dimensional meshes.

To remain accurate and stable, the mass-lumped finite element method needs to satisfy the following conditions [31]:

C1 (Unisolvent). The space $\tilde{U}$ is unisolvent on the nodes $\tilde{\mathcal{Q}}$.

C2 (Symmetry). The space $\tilde{U}$ and the set $\tilde{\mathcal{Q}}$ are invariant to affine mappings that map $\tilde{e}$ onto itself.

C3 (Face-conforming). If $\tilde{u} \in \tilde{U}$ is zero at all nodes in $\tilde{\mathcal{Q}} \cap \tilde{f}$, with $\tilde{f}$ a reference face, then $\tilde{u}$ is zero on $\tilde{f}$.

C4 (Positivity). The weights $\{\tilde{\omega}_{\tilde{\mathbf{x}}}\}_{\tilde{\mathbf{x}}\in\tilde{\mathcal{Q}}}$ are all strictly positive.

C5 (Accuracy). The quadrature rule is exact for functions in $\mathcal{P}_{p-2} \otimes \tilde{U}$ when $p \geq 2$.

The first three conditions are necessary to guarantee that the global basis functions are well-defined and continuous. The last two conditions are necessary for stability and for maintaining an optimal order of convergence.

When $p \geq 2$, these conditions can not all be met for standard polynomial spaces $\tilde{U} = \mathcal{P}_p$. Therefore, the element space needs to be enriched with higher-degree bubble functions. We will focus on the mass-lumped tetrahedral elements recently developed in [31]. An overview of these elements is given in Table 6.1. There, n denotes the dimension of $\tilde{U}$, which is equal to the number of nodes per element, $B_f := \{x_1x_2x_3, x_1x_2x_4, x_1x_3x_4, x_2x_3x_4\}$ denotes the four face bubble functions, and $B_e := \{x_1x_2x_3x_4\}$ denotes the element bubble function, with x_1, x_2, x_3, x_4 the four barycentric coordinates. We also used the notation $UV := U \otimes V := \{w \mid w = uv \text{ for some } u \in U, v \in V\}$ for any two function spaces U, V.

To apply these elements more efficiently, we also approximate the L^2 inner-product for the stiffness matrix, $(c\nabla u, \nabla v)$, with a quadrature rule. This also allows us to handle material parameters c that vary within the element. It turns out that it is more efficient and sometimes even necessary to compute the stiffness matrix with a different quadrature rule than

Table 6.1: Overview of mass-lumped tetrahedra.

p	n	$\tilde{U}$
2	15	$\mathcal{P}_2 \oplus B_f \oplus B_e$
3	32	$\mathcal{P}_3 \oplus B_f\mathcal{P}_1 \oplus B_e\mathcal{P}_1$
4	60	$\mathcal{P}_4 \oplus B_f\mathcal{P}_2 \oplus B_e(\mathcal{P}_2 + B_f)$
	61	$\mathcal{P}_4 \oplus B_f\mathcal{P}_2 \oplus B_e(\mathcal{P}_2 + B_f + B_e)$
	65	$\mathcal{P}_4 \oplus B_f(\mathcal{P}_2 + B_f) \oplus B_e(\mathcal{P}_2 + B_f + B_e)$

for the mass matrix. We will denote the quadrature points and weights for the stiffness matrix by $\tilde{\mathcal{Q}}'$ and $\{\tilde{\omega}'_{\tilde{\mathbf{x}}}\}_{\tilde{\mathbf{x}}\in\tilde{\mathcal{Q}}'}$, respectively, and denote the corresponding approximated $L^2(\Omega)$-product by $(\cdot,\cdot)_{\mathcal{Q}'_h}$.

The resulting finite element method remains stable and accurate if the following conditions are also satisfied:

C6 (Positivity). The weights $\{\tilde{\omega}'_{\tilde{\mathbf{x}}}\}_{\tilde{\mathbf{x}}\in\tilde{\mathcal{Q}}'}$ are all strictly positive.

C7 (Spurious-free). There is no function $\tilde{u} \in \tilde{U}$ with zero gradient $\tilde{\nabla}\tilde{u} = \mathbf{0}$ on all quadrature points $\tilde{\mathcal{Q}}'$ except the constant function. In case of linear elasticity, there is no function $\tilde{\mathbf{u}} \in \tilde{U}^3$ with zero strain $\tilde{\nabla}\tilde{\mathbf{u}} + \tilde{\nabla}\tilde{\mathbf{u}}^t = \mathbf{0}$ on all quadrature points $\tilde{\mathcal{Q}}'$ except the six rigid motions.

C8 (Accuracy). If $p \geq 2$, the quadrature rule for the stiffness matrix is exact for functions in $\mathcal{P}_{p-1} \otimes D\tilde{U}$, where $D\tilde{U}$ denotes the space of all partial derivatives of all functions in $\tilde{U}$.

A proof that these two conditions are sufficient to maintain an optimal order of convergence is given in Section 6.4.

We constructed quadrature rules that satisfy these conditions for the specific function spaces of the higher-degree mass-lumped tetrahedra presented in Table 6.1. For the degree-2 element, the most efficient quadrature rule seems to be an existing 14-point rule that is fifth-order accurate, but for the higher-degree elements, we obtained new quadrature rules that require less points than existing rules. An overview of these rules is given in the next section.

6.3 Efficient quadrature rules for the stiffness matrix

To present the quadrature rules for the stiffness matrix, let $\tilde{e}$ to be the reference tetrahedron with vertices at $(0, 0, 0)$, $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$. The barycentric coordinates of this element are given by the three Cartesian coordinates x_1, x_2, x_3, and the fourth coordinate $x_4 := 1 - x_1 - x_2 - x_3$. These coordinates are useful for describing $\mathcal{S}$, the space of affine mappings that map $\tilde{e}$ onto itself, since any function $s \in \mathcal{S}$ can be defined by a permutation of the barycentric coordinates.

Table 6.2: Quadrature rule of 14 points [38] for the stiffness matrix of the degree-2 15-node tetrahedron.

Nodes	#	ω'	Node parameters
$\{(c_1, c_1, c_1)\}$	4	0.01224884051939366	0.09273525031089123
$\{(c_2, c_2, c_2)\}$	4	0.01878132095300264	0.3108859192633006
$\{(d, d, \frac{1}{2} - d)\}$	6	0.007091003462846911	0.04550370412564965
$V = \mathcal{P}_5 = \{x_1, x_1^2 x_2, x_1^3 x_2^2, \beta_f x_1, \beta_f x_1 x_2, \beta_e x_1\}$			

Table 6.3: New quadrature rule of 21 points for the stiffness matrix of the degree-3 32-node tetrahedron.

Nodes	#	ω'	Node parameters
$\{(c_1, c_1, c_1)\}$	4	0.008382813462606309	0.08360982293995379
$\{(c_2, c_2, c_2)\}$	4	0.01062803097330636	0.3195556046935656
$\{(f_1, f_1, f_2)\}$	12	0.005973459577178217	$\begin{bmatrix} 0.06366100187501753 \\ 0.3362519222398494 \end{bmatrix}$
$\{(\frac{1}{4}, \frac{1}{4}, \frac{1}{4})\}$	1	0.01894177399687740	-
$V = \mathcal{P}_5 \oplus B_f \mathcal{P}_3$ $= \{x_1, x_1^2 x_2, x_1^3 x_2^2, \beta_f x_1, \beta_f x_1^2 x_2, \beta_f^2, \beta_e x_1, \beta_e x_1 x_2\}$			

Now let $\{\mathbf{x}\}$ denote point $\mathbf{x}$ and all equivalent points $s(\mathbf{x})$, with $s \in \mathcal{S}$. The quadrature points will consist of several equivalence classes $\{\mathbf{x}\}$ with quadrature weights that are the same for each equivalence class. To give

Table 6.4: New quadrature rule of 51 points for the stiffness matrix of the degree-4 60-node tetrahedron.

Nodes	#	ω'	Node parameters
$\{(c_1, c_1, c_1)\}$	4	0.001076330088382485	0.04010756377220036
$\{(c_2, c_2, c_2)\}$	4	0.006422430307819483	0.1881144601918900
$\{(d, d, \frac{1}{2} - d)\}$	6	0.003859721113202450	0.1124010568611476
$\{(f_{11}, f_{11}, f_{12})\}$	12	0.003162722714222902	$\begin{bmatrix} 0.04781990270450464 \\ 0.2053222493389064 \end{bmatrix}$
$\{(f_{21}, f_{21}, f_{22})\}$	12	0.004715130256124021	$\begin{bmatrix} 0.2347999378738287 \\ 0.03405863749492695 \end{bmatrix}$
$\{(f_{31}, f_{31}, f_{32})\}$	12	0.001320748780834370	$\begin{bmatrix} 0.4614535776221135 \\ 0.06693547308143162 \end{bmatrix}$
$\{(\frac{1}{4}, \frac{1}{4}, \frac{1}{4})\}$	1	0.003130077388468573	-

$$V = \mathcal{P}_7 \oplus B_f(\mathcal{P}_5 \oplus B_f\mathcal{P}_3) \oplus B_e\mathcal{P}_5$$
$$= \{x_1, x_1^2x_2, x_1^3x_2^2, x_1^4x_2^3, \beta_f x_1, \beta_f x_1^2 x_2, \beta_f x_1^3 x_2^2, \beta_f^2 x_1, \beta_f^2 x_1^2 x_2, \beta_f^3, \ldots$$
$$\ldots, \beta_e x_1, \beta_e x_1^2 x_2, \beta_e x_1^3 x_2^2, \beta_e \beta_f x_1, \beta_e \beta_f x_1 x_2, \beta_e^2 x_1\}$$

an example of an equivalence class, consider the point (c_1, c_1, c_1). The barycentric coordinates of this point are given by $c_1, c_1, c_1, 1 - 3c_1$, so the equivalence class $\{(c_1, c_1, c_1)\}$ consists of the four points (c_1, c_1, c_1), $(1 - 3c_1, c_1, c_1)$, $(c_1, 1 - 3c_1, c_1)$, and $(c_1, c_1, 1 - 3c_1)$ when $c_1 \neq \frac{1}{4}$.

To find a set of points and weights that satisfy accuracy condition C8, we construct a linear basis that spans $V \supset \mathcal{P}_{p-1} \otimes D\tilde{U}$. We describe this linear basis using the notation $\{f_1, f_2, \ldots, f_k\}$, which denotes the functions $f_1, \ldots, f_k$ and all equivalent functions $f_i \circ s$, with $i = 1, \ldots, k$ and $s \in \mathcal{S}$. To give an example, all equivalent versions of $x_1^2x_2^2x_3x_4$ are given by the six functions $x_1^2x_2^2x_3x_4$, $x_1^2x_2x_3^2x_4$, $x_1^2x_2x_3x_4^2$, $x_1x_2^2x_3^2x_4$, $x_1x_2^2x_3x_4^2$, and $x_1x_2x_3^2x_4^2$.

After having constructed a basis $\{f_1, f_2, \ldots, f_k\}$ for V, we search for a quadrature rule that has a configuration with k parameters. These parameters consist of location parameters and quadrature weights. Because of the symmetry, a quadrature rule that is exact for a function f is exact for all equivalent functions. Therefore, to satisfy C8, we end up with a

Table 6.5: New quadrature rule of 60 points for the stiffness matrix of the degree-4 61- and 65-node tetrahedron.

Nodes	#	ω'	Node parameters
$\{(c_1, c_1, c_1)\}$	4	0.001137453809249273	0.04091036488546224
$\{(c_2, c_2, c_2)\}$	4	0.006907244220995018	0.1942594527940223
$\{(c_3, c_3, c_3)\}$	4	0.004458749819772567	0.3166409312612929
$\{(d_1, d_1, \frac{1}{2} - d_1)\}$	6	0.001389883779363477	0.02776256108257648
$\{(d_2, d_2, \frac{1}{2} - d_2)\}$	6	0.004236295194116969	0.1022199785693040
$\{(f_{11}, f_{11}, f_{12})\}$	12	0.001788418107829456	$\begin{bmatrix} 0.03511432271187172 \\ 0.2097218125202450 \end{bmatrix}$
$\{(f_{21}, f_{21}, f_{22})\}$	12	0.003642034272731381	$\begin{bmatrix} 0.1790174868402900 \\ 0.03980830656880513 \end{bmatrix}$
$\{(f_{31}, f_{31}, f_{32})\}$	12	0.001477531071582210	$\begin{bmatrix} 0.4192720711456938 \\ 0.00895031787296103 \end{bmatrix}$

$$V = \mathcal{P}_8 \oplus B_f^2 \mathcal{P}_3 \oplus B_e(\mathcal{P}_5 \oplus B_f \mathcal{P}_3)$$
$$= \{x_1, x_1^2 x_2, x_1^3 x_2^2, x_1^4 x_2^3, x_1^4 x_2^4, \beta_f x_1, \beta_f x_1^2 x_2, \beta_f x_1^3 x_2^2, \beta_f^2 x_1, \beta_f^2 x_1^2 x_2, \beta_f^3, \ldots$$
$$\ldots, \beta_e x_1, \beta_e x_1^2 x_2, \beta_e x_1^3 x_2^2, \beta_e \beta_f x_1, \beta_e \beta_f x_1^2 x_2, \beta_e \beta_f^2, \beta_e^2 x_1, \beta_e^2 x_1 x_2\}$$

nonlinear system of k equations:

$$\int_{\tilde{e}} f_i(\tilde{\mathbf{x}})\, d\tilde{x} = \sum_{\tilde{\mathbf{x}} \in \tilde{\mathcal{Q}}'} \omega'_{\tilde{\mathbf{x}}} f_i(\tilde{\mathbf{x}}), \qquad i = 1, \ldots, k.$$

We obtain solutions of this system using Newton's method and check for each solution if it satisfies C6 and C7. When we cannot find an admissible solution for configurations with k parameters, we add an additional basis function f_{k+1} to the set that needs to be integrated exactly, and search for quadrature rules with configurations of $k+1$ parameters. We continue this process until we find a suitable quadrature rule.

The quadrature rules that were obtained in this way are given in Tables 6.2-6.5. There, # denotes the number of nodes in each equivalence class, and $\beta_f := x_1 x_2 x_3$ and $\beta_e = x_1 x_2 x_3 x_4$ denote the face bubble function and interior bubble function, respectively.

The quadrature rule with the least number of points we could find for the degree-2 15-node tetrahedron is the 14-point fifth-order accurate rule of

[38]. We also found an accurate quadrature rule of 10 points with positive weights, but the resulting method was not accurate since condition C7 was not satisfied: it had one non-constant mode with gradient equal to zero at all 10 points. We also considered the 15-point quadrature rule used for the mass matrix, but this significantly increased the condition number of the element matrix and therefore resulted in a considerably smaller time step size.

The quadrature rules for the degree-3 and degree-4 elements are new and require less quadrature points then rules currently available in literature, since most quadrature rules in literature are constructed to be exact for a function space of the form $\mathcal{P}_k$ and not for the specific function spaces $\mathcal{P}_{p-1}\otimes D\tilde{U}$. To give an example, the highest polynomial degree of $D\tilde{U}$ of the degree-4 61- or 65-node tetrahedron is 7, so $\mathcal{P}_3\otimes D\tilde{U}$ contains a polynomial of degree 10. A quadrature rule that is order-10 accurate already requires 81 quadrature points [80], while our quadrature rule for these elements only requires 60 points. Similarly, our quadrature rules for the degree-3 32-node tetrahedron and the degree-4 60-node tetrahedron require 21 and 51 points, respectively, while the quadrature rules currently available in literature for these elements require 24 and 61 points [80].

6.4 Error estimates

In this section, we prove that when conditions C1-C8 are satisfied, the finite element method maintains an optimal order of convergence for a related elliptic problem. Convergence for the wave equation immediately follows from this result in a way analogous to [31].

Throughout this section, we will let p denote the degree of the finite element space, by which we mean the largest degree such that $\tilde{U} \supset \mathcal{P}_p$. We will also let C denote a positive constant that may depend on the domain Ω, the regularity of the mesh, the parameters ρ and c, the reference space $\tilde{U}$, and the reference quadrature rules $(\tilde{\mathcal{Q}}, \{\tilde{\omega}_{\tilde{\mathbf{x}}}\}_{\tilde{\mathbf{x}}\in\tilde{\mathcal{Q}}})$ and $(\tilde{\mathcal{Q}}', \{\tilde{\omega}_{\tilde{\mathbf{x}}}\}_{\tilde{\mathbf{x}}\in\tilde{\mathcal{Q}}'})$, but does not depend on the mesh resolution h and the functions that appear in the inequalities.

6.4.1 Preliminary results

To obtain error bounds, we first define some norms and function spaces and list a few preliminary results. Let $H^k(\Omega)$ denote the Sobolev space of functions on Ω with order-k square-integrable weak derivatives, and equip the space with norm

$$\|u\|_k^2 := \sum_{|\alpha| \leq k} \|D^{\boldsymbol{\alpha}} u\|_0^2,$$

where $\|\cdot\|_0$ denotes the L^2-norm, $D^{\boldsymbol{\alpha}} := \partial_1^{\alpha_1}\partial_2^{\alpha_2}\partial_3^{\alpha_3}$ the partial derivative, and $|\boldsymbol{\alpha}| := \alpha_1 + \alpha_2 + \alpha_3$ the order of the derivative. We also define the broken Sobolev spaces $H^k(\mathcal{T}_h) := \{u \in L^2(\Omega) \mid u|_e \in H^k(e) \text{ for all } e \in \mathcal{T}_h\}$, equipped with norm

$$\|u\|_{\mathcal{T}_h,k}^2 := \sum_{e \in \mathcal{T}_h} \|u|_e\|_k^2.$$

Throughout this section, we will use the fact that $H^2(\Omega) \supset \mathcal{C}^0(\overline{\Omega})$ for any three-dimensional domain Ω.

We also define the semi-norms $|u|_{\mathcal{Q}_h}^2 := (u,u)_{\mathcal{Q}_h}$ and $|\sigma|_{\mathcal{Q}'_h}^2 := (\sigma,\sigma)_{\mathcal{Q}'_h}$ for piecewise continuous functions u and σ, and define $\Pi_{h,q}$ to be the L^2-projection operators on the discontinuous piecewise-polynomial spaces $V(\mathcal{T}_h, \mathcal{P}_q) := \{u \in L^2(\Omega) \mid u \circ \phi_e \in \mathcal{P}_q \text{ for all } e \in \mathcal{T}_h\}$. Several useful properties of these spaces and operators are listed below.

Lemma 6.4.1. *Let $q \geq 0$. Then*

$$|u_h|_{\mathcal{Q}_h} \leq C\|u_h\|_0 \qquad \text{for all } u_h \in V(\mathcal{T}_h, \mathcal{P}_q),$$
$$|\boldsymbol{\sigma}_h|_{\mathcal{Q}'_h} \leq C\|\boldsymbol{\sigma}_h\|_0 \qquad \text{for all } \boldsymbol{\sigma}_h \in V(\mathcal{T}_h, \mathcal{P}_q)^3.$$

Proof. These results follow immediately from the fact that the elements are affine equivalent with the reference element and that the reference space $\mathcal{P}_q$ is finite dimensional. □

Lemma 6.4.2. *If conditions C1-C4 are satisfied, then $|\cdot|_{\mathcal{Q}_h}$ becomes a full norm $\|\cdot\|_{\mathcal{Q}_h}$ on $U(\mathcal{T}_h, \tilde{U})$ and*

$$\|u_h\|_{\mathcal{Q}_h} \geq C\|u_h\|_0 \qquad \text{for all } u_h \in U_h.$$

Furthermore, if conditions C1-C3, C6, and C7 are satisfied, then $|\cdot|_{\mathcal{Q}'_h}$ becomes a full norm $\|\cdot\|_{\mathcal{Q}'_h}$ on $V(\mathcal{T}_h, D\tilde{U})$ and

$$\|\nabla u_h\|_{\mathcal{Q}'_h} \geq C\|\nabla u_h\|_0 \qquad \text{for all } u_h \in U_h.$$

Proof. These inequalities follow immediately from the fact that the elements are affine equivalent with the reference element and that the reference element space $\tilde{U}$ is finite dimensional. □

Lemma 6.4.3. *Let $u \in H^k(\mathcal{T}_h)$ and $\boldsymbol{\sigma} \in H^k(\mathcal{T}_h)^3$, with $k \geq 0$, and let $q \geq 0$. Then*

$$\|u - \Pi_{h,q}u\|_{\mathcal{T}_h,m} \leq Ch^{\min(q+1,k)-m}\|u\|_{\mathcal{T}_h,\min(q+1,k)}, \quad m \leq \min(q+1,k),$$
$$\|\boldsymbol{\sigma} - \Pi_{h,q}\boldsymbol{\sigma}\|_{\mathcal{T}_h,m} \leq Ch^{\min(q+1,k)-m}\|\boldsymbol{\sigma}\|_{\mathcal{T}_h,\min(q+1,k)}, \quad m \leq \min(q+1,k).$$

Furthermore, if $k \geq 2$, then

$$|u - \Pi_{h,q}u|_{\mathcal{Q}_h} \leq Ch^{\min(q+1,k)}\|u\|_{\mathcal{T}_h,\min(q+1,k)},$$
$$|\boldsymbol{\sigma} - \Pi_{h,q}\boldsymbol{\sigma}|_{\mathcal{Q}'_h} \leq Ch^{\min(q+1,k)}\|\boldsymbol{\sigma}\|_{\mathcal{T}_h,\min(q+1,k)}.$$

Finally, if $u \in H^1(\Omega) \cap H^k(\mathcal{T}_h)$ with $k \geq 2$, then

$$\|u - I_hu\|_{\mathcal{T}_h,m} \leq Ch^{\min(p+1,k)-m}\|u\|_{\mathcal{T}_h,\min(p+1,k)}, \quad m \leq \min(p+1,k)$$

with $p \geq 2$ the degree of the finite element space.

Proof. The first, second, and last inequality follow from [12, Chapter 3.1]. For the fourth inequality, let $q^* \geq q$, be a polynomial degree and $\tilde{\mathcal{Q}}^* \supset \tilde{\mathcal{Q}}'$ a set of points such that $\mathcal{P}_{q^*}$ is unisolvent on $\tilde{\mathcal{Q}}^*$ and let I_h^* be the interpolation operator that interpolates a function in $H^2(\mathcal{T}_h)$ at the nodes $\mathcal{Q}(\mathcal{T}_h, \tilde{\mathcal{Q}}^*)$ by a function in $V(\mathcal{T}_h, \mathcal{P}_{q^*})$. We can then obtain the fourth inequality as follows:

$$\begin{aligned}
|\boldsymbol{\sigma} - \Pi_{h,q}\boldsymbol{\sigma}|_{\mathcal{Q}'_h} &= |I_h^*\boldsymbol{\sigma} - \Pi_{h,q}\boldsymbol{\sigma}|_{\mathcal{Q}'_h} \\
&\leq C\|I_h^*\boldsymbol{\sigma} - \Pi_{h,q}\boldsymbol{\sigma}\|_0 \\
&\leq C(\|I_h^*\boldsymbol{\sigma} - \boldsymbol{\sigma}\|_0 + \|\boldsymbol{\sigma} - \Pi_{h,q}\boldsymbol{\sigma}\|_0) \\
&\leq Ch^{\min(q+1,k)}\|\boldsymbol{\sigma}\|_{\mathcal{T}_h,\min(q+1,k)},
\end{aligned}$$

where we used Lemma 6.4.1 in the the second line and the triangle inequality in the third line. The last line follows from [12, Chapter 3.1].

The third inequality can be derived in a way analogous to the fourth inequality. □

6.4.2 Estimates on the integration error

Define integration errors $r_h(u,w) := (u,w) - (u,w)_{\mathcal{Q}_h}$ and $r'_h(\boldsymbol{\sigma},\boldsymbol{\tau}) := (\boldsymbol{\sigma},\boldsymbol{\tau}) - (\boldsymbol{\sigma},\boldsymbol{\tau})_{\mathcal{Q}'_h}$. In [31] we derived the following bounds on r_h.

Lemma 6.4.4. *Let $p \geq 2$ be the degree of the finite element space, $u \in H^k(\Omega)$ with $k \geq 2$, and $w \in U_h$. If conditions C1-C5 are satisfied, then*

$$|r_h(u,w)| \leq Ch^{\min(p,k)}\|u\|_{\min(p,k)}\|w\|_1,$$
$$|r_h(u,w)| \leq Ch^{\min(p+1,k)}\|u\|_{\min(p+1,k)}\|w\|_{\mathcal{T}_h,2}.$$

We also derive bounds on the integration error for the stiffness matrix.

Lemma 6.4.5. *Let $p \geq 2$ be the degree of the finite element space, $\boldsymbol{\sigma} \in H^k(\mathcal{T}_h)^3$ with $k \geq 2$, and $\boldsymbol{\tau} \in V(\mathcal{T}_h, D\tilde{U})^3$. If conditions C1-C3, C6, and C8 are satisfied, then*

$$|r'_h(\boldsymbol{\sigma}, \boldsymbol{\tau})| \leq Ch^{\min(p,k)} \|\boldsymbol{\sigma}\|_{\mathcal{T}_h, \min(p,k)} \|\boldsymbol{\tau}\|_0, \tag{6.5}$$

$$|r'_h(\boldsymbol{\sigma}, \boldsymbol{\tau})| \leq Ch^{\min(p+1,k)} \|\boldsymbol{\sigma}\|_{\mathcal{T}_h, \min(p+1,k)} \|\boldsymbol{\tau}\|_{\mathcal{T}_h,1}. \tag{6.6}$$

Proof. Using C8, we can write

$$r'_h(\boldsymbol{\sigma}, \boldsymbol{\tau}) = r'_h(\boldsymbol{\sigma} - \Pi_{h,p-1}\boldsymbol{\sigma}, \boldsymbol{\tau}).$$

The inequality (6.5) then follows from the Cauchy–Schwarz inequaltiy and Lemma 6.4.3.

Using C8 and the fact that $\mathcal{P}_p \subset \mathcal{P}_{p-1} \otimes D\tilde{U}$ for $p \geq 2$, we can also write

$$\begin{aligned} r'_h(\boldsymbol{\sigma}, \boldsymbol{\tau}) &= r'_h\big((\boldsymbol{\sigma} - \Pi_{h,p}\boldsymbol{\sigma}) + (\Pi_{h,p}\boldsymbol{\sigma} - \Pi_{h,p-1}\boldsymbol{\sigma}) \\ &\qquad + \Pi_{h,p-1}\boldsymbol{\sigma}, (\boldsymbol{\tau} - \Pi_{h,0}\boldsymbol{\tau}) + \Pi_{h,0}\boldsymbol{\tau}\big) \\ &= r'_h(\boldsymbol{\sigma} - \Pi_{h,p}\boldsymbol{\sigma}, \boldsymbol{\tau}) + r'_h(\Pi_{h,p}\boldsymbol{\sigma} - \Pi_{h,p-1}\boldsymbol{\sigma}, \boldsymbol{\tau} - \Pi_{h,0}\boldsymbol{\tau}). \end{aligned}$$

The inequality (6.6) then follows from the Cauchy–Schwarz inequaltiy and Lemma 6.4.3. □

6.4.3 Error estimates for a related elliptic problem

Let $v \in \mathcal{C}^0(\overline{\Omega})$. The elliptic problem corresponding to (6.2) is finding $u \in H_0^1(\Omega)$ such that

$$(c\nabla u, \nabla w) = (v, w) \qquad \text{for all } w \in H_0^1(\Omega). \tag{6.7}$$

The corresponding mass-lumped finite element method is finding $u_h \in U_h$ such that

$$(c\nabla u_h, \nabla w)_{\mathcal{Q}'_h} = (v, w)_{\mathcal{Q}_h} \qquad \text{for all } w \in U_h. \tag{6.8}$$

In the next two theorems we prove optimal convergence in the H^1-norm and L^2-norm.

Theorem 6.4.6 (Optimal convergence in the H^1-norm)**.** *Let u be the solution of (6.7) and u_h the solution of (6.8). Assume $c \in \mathcal{C}^p(\overline{\Omega})$, $u \in H^{k_u}(\Omega)$, and $v \in H^{k_v}(\Omega)$, with $k_u, k_v \geq 2$. If conditions C1-C8 are satisfied, then*

$$\|u - u_h\|_1 \leq Ch^{\min(p,k_u-1,k_v)}(\|u\|_{\min(p+1,k_u)} + \|v\|_{\min(p,k_v)}), \tag{6.9}$$

with p the degree of the finite element method.

Proof. Define $e_h := I_h u - u_h$ and $\epsilon_h := u - I_h u$. Using (6.7), we can write

$$\begin{aligned}(c\nabla I_h u, \nabla e_h)_{\mathcal{Q}'_h} &= - r'_h(c\nabla I_h u, \nabla e_h) - (c\nabla \epsilon_h, \nabla e_h) + (c\nabla u, \nabla e_h)\\ &= - r'_h(c\nabla I_h u, \nabla e_h) - (c\nabla \epsilon_h, \nabla e_h) + (v, e_h),\end{aligned}$$

and using (6.8), we can obtain

$$(c\nabla u_h, \nabla e_h)_{\mathcal{Q}'_h} = (v, e_h)_{\mathcal{Q}_h}.$$

Subtracting these two equalities gives

$$(c\nabla e_h, \nabla e_h)_{\mathcal{Q}'_h} = -r'_h(c\nabla I_h u, \nabla e_h) - (c\nabla \epsilon_h, \nabla e_h) + r_h(v, e_h) \tag{6.10}$$

From Lemma 6.4.2, the positivity of c, and Poincaré's inequality, it follows that

$$\|e_h\|_1^2 \leq C(c\nabla e_h, \nabla e_h)_{\mathcal{Q}'_h}. \tag{6.11}$$

Using Lemma 6.4.5, Lemma 6.4.3, and the regularity of c, we can obtain

$$\begin{aligned}|r'_h(c\nabla I_h u, \nabla e_h)| &\leq Ch^{\min(p,k_u-1)}\|c\nabla I_h u\|_{\mathcal{T}_h,\min(p,k_u-1)}\|e_h\|_1\\ &\leq Ch^{\min(p,k_u-1)}\|u\|_{\min(p+1,k_u)}\|e_h\|_1. \end{aligned}\tag{6.12}$$

Using the Cauchy–Schwarz inequality, the boundedness of c, and Lemma 6.4.3, we can also obtain

$$|(c\nabla \epsilon_h, \nabla e_h)| \leq Ch^{\min(p,k_u-1)}\|u\|_{\min(p+1,k_u)}\|e_h\|_1. \tag{6.13}$$

Finally, using Lemma 6.4.4, we can obtain

$$|r_h(v, e_h)| \leq Ch^{\min(p,k_v)}\|v\|_{\min(p,k_v)}\|e_h\|_1. \tag{6.14}$$

Combining (6.10)-(6.14) gives

$$\|e_h\|_1 \leq Ch^{\min(p,k_u-1,k_v)}(\|u\|_{\min(p+1,k_u)} + \|v\|_{\min(p,k_v)}).$$

Since $u - u_h = e_h + \epsilon_h$, inequality (6.9) then follows from the above and Lemma 6.4.3. □

To prove optimal convergence in the L^2-norm, we make the following regularity assumption: for any $v \in L^2(\Omega)$, the solution of (6.7) is in $H^2(\Omega)$ and satisfies

$$\|u\|_2 \leq C\|v\|_0. \tag{6.15}$$

This is certainly true when $\partial\Omega$ is $\mathcal{C}^2$ and $c \in \mathcal{C}^1(\overline{\Omega})$.

Theorem 6.4.7 (Optimal convergence in the L^2-norm)**.** *Let u be the solution of (6.7) and u_h the solution of (6.8). Assume $c \in \mathcal{C}^{p+1}(\overline{\Omega})$, $u \in H^{k_u}(\Omega)$, and $v \in H^{k_v}(\Omega)$, with $k_u, k_v \geq 2$, and assume the regularity condition (6.15) holds. If conditions C1-C8 are satisfied, then*

$$\|u - u_h\|_0 \leq Ch^{\min(p+1,k_u,k_v)}(\|u\|_{\min(p+1,k_u)} + \|v\|_{\min(p+1,k_v)}), \tag{6.16}$$

with p the degree of the finite element method.

Proof. Define $z_h \in H_0^1(\Omega)$ to be the solution of the elliptic problem

$$(c\nabla z_h, \nabla w) = (u - u_h, w) \qquad \text{for all } w \in H_0^1(\Omega).$$

From the regularity assumption it follows that $z_h \in H^2(\Omega)$ and

$$\|z_h\|_2 \leq C\|u - u_h\|_0.$$

From the definition of z_h, it also follows that

$$\begin{aligned}\|u - u_h\|_0^2 &= (c\nabla[u - u_h], \nabla z_h)\\ &= (c\nabla[u - u_h], \nabla[z_h - I_h z_h]) + (c\nabla[u - u_h], \nabla I_h z_h).\end{aligned} \tag{6.17}$$

We can bound the term $(c\nabla[u - u_h], \nabla[z_h - I_h z_h])$ as follows:

$$\begin{aligned}|(c\nabla[u - u_h],\nabla[z_h - I_h z_h])| &\leq C\|u - u_h\|_1\|z_h - I_h z_h\|_1\\ &\leq Ch^{\min(p,k_u-1,k_v)+1}(\|u\|_{\min(p+1,k_u)} + \|v\|_{\min(p,k_v)})\|z_h\|_2\\ &\leq Ch^{\min(p+1,k_u,k_v+1)}(\|u\|_{\min(p+1,k_u)} + \|v\|_{\min(p,k_v)})\|u - u_h\|_0,\end{aligned} \tag{6.18}$$

where we used the Cauchy–Schwarz inequality and the boundedness of c in the first line, Theorem 6.4.6 and Lemma 6.4.3 in the second line, and the regularity assumption in the last line. It then remains to find a bound for $(c\nabla[u - u_h], \nabla I_h z_h)$.

To do this, use (6.7) to write

$$(c\nabla u, \nabla I_h z_h) = (v, \nabla I_h z_h),$$

and use (6.8) to write

$$\begin{aligned}(c\nabla u_h, \nabla I_h z_h) &= r_h'(c\nabla u_h, \nabla I_h z_h) + (c\nabla u_h, \nabla I_h z_h)_{\mathcal{Q}_h'}\\ &= r_h'(c\nabla u_h, \nabla I_h z_h) + (v, I_h z_h)_{\mathcal{Q}_h}.\end{aligned}$$

Subtracting these two equalities gives

$$(c\nabla[u - u_h], \nabla I_h z_h) = -r_h'(c\nabla u_h, \nabla I_h z_h) + r_h(v, I_h z_h). \tag{6.19}$$

Now, set $q := \min(p-1, k_u - 2)$. We can write

$$\begin{aligned}r_h'(c\nabla u_h, \nabla I_h z_h) &= r_h'(\nabla u_h, c\nabla I_h z_h - \Pi_{h,0} c\nabla I_h z_h)\\ &= r_h'(\nabla u_h - \Pi_{h,q}\nabla u, c\nabla I_h z_h - \Pi_{h,0} c\nabla I_h z_h)\\ &\quad + r_h'(\Pi_{h,q}\nabla u, c\nabla I_h z_h - \Pi_{h,0} c\nabla I_h z_h)\\ &= r_h'(\nabla u_h - \Pi_{h,q}\nabla u, c\nabla I_h z_h - \Pi_{h,0} c\nabla I_h z_h)\\ &\quad + r_h'(c\Pi_{h,q}\nabla u, \nabla I_h z_h)\\ &=: R_1 + R_2,\end{aligned}$$

where we used C8 for the first and third equality. We can bound R_1 as follows:

$$\begin{aligned}|R_1| &\leq \|\nabla u_h - \Pi_{h,q}\nabla u\|_0 \|c\nabla I_h z_h - \Pi_{h,0} c\nabla I_h z_h\|_0\\ &\quad + |\nabla u_h - \Pi_{h,q}\nabla u|_{\mathcal{Q}_h'} |c\nabla I_h z_h - \Pi_{h,0} c\nabla I_h z_h|_{\mathcal{Q}_h'}\\ &\leq Ch\|\nabla u_h - \Pi_{h,q}\nabla u\|_0 \|z_h\|_2\\ &\leq Ch(\|\nabla u_h - \nabla u\|_0 + \|\nabla u - \Pi_{h,q}\nabla u\|_0)\|u - u_h\|_0\\ &\leq Ch^{\min(p,k_u-1,k_v)+1}(\|u\|_{\min(p+1,k_u)} + \|v\|_{\min(p,k_v)})\|u - u_h\|_0,\end{aligned}$$

where we used the Cauchy–Schwarz inequality in the first line, Lemma 6.4.1, the regularity of c, and Lemma 6.4.3 for the second inequality, the triangle inequality and the regularity assumption for the third inequality, and Theorem 6.4.6 and Lemma 6.4.3 for the last inequality. We can also bound R_2 as follows:

$$\begin{aligned}|R_2| &\leq Ch^{p+1}\|c\Pi_{h,q}\nabla u\|_{\mathcal{T}_h,p+1}\|\nabla I_h z_h\|_{\mathcal{T}_h,1}\\ &\leq Ch^{p+1}\|\Pi_{h,q}\nabla u\|_{\mathcal{T}_h,p+1}\|I_h z_h\|_{\mathcal{T}_h,2}\\ &\leq Ch^{p+1}\|\Pi_{h,q}\nabla u\|_{\mathcal{T}_h,p+1}\|z_h\|_2\\ &= Ch^{p+1}\|\Pi_{h,q}\nabla u\|_{\mathcal{T}_h,q}\|z_h\|_2\\ &\leq Ch^{p+1}\|\nabla u\|_{\mathcal{T}_h,q}\|z_h\|_2\\ &\leq Ch^{p+1}\|u\|_{\min(p,k_u-1)}\|u-u_h\|_0.\end{aligned}$$

Here, the first line follows from Lemma 6.4.5 and the fact that $c\Pi_{h,q}\nabla u \in H^{p+1}(\mathcal{T}_h)^3$, the second line follows from the regularity of c, the third line follows from Lemma 6.4.3, the fifth line follows from Lemma 6.4.3, and the last line follows from regularity assumption. The fourth line follows from the fact that $\Pi_{h,q}\nabla u$ is piecewise polynomial of degree q and therefore $\|\Pi_{h,q}\nabla u\|_{\mathcal{T}_h,p+1} = \|\Pi_{h,q}\nabla u\|_{\mathcal{T}_h,q}$. By combining the bounds on R_1 and R_2, we then obtain

$$\begin{aligned}|r_h'(c\nabla u_h, \nabla I_h z_h)| &= |R_1 + R_2| \leq |R_1| + |R_2|\\ \leq Ch^{\min(p+1,k_u,k_v+1)}&(\|u\|_{\min(p+1,k_u)} + \|v\|_{\min(p,k_v)})\|u-u_h\|_0. \qquad (6.20)\end{aligned}$$

From Lemma 6.4.4, Lemma 6.4.3, and the regularity assumption, it also follows that

$$\begin{aligned}|r_h(v, I_h z_h)| &\leq Ch^{\min(p+1,k_v)}\|v\|_{\min(p+1,k_v)}\|I_h z\|_{\mathcal{T}_h,2}\\ &\leq Ch^{\min(p+1,k_v)}\|v\|_{\min(p+1,k_v)}\|z_h\|_2\\ &\leq Ch^{\min(p+1,k_v)}\|v\|_{\min(p+1,k_v)}\|u-u_h\|_0 \qquad (6.21)\end{aligned}$$

Combining (6.19), (6.20), and (6.21) gives

$$\begin{aligned}&|(c\nabla[u-u_h], \nabla I_h z_h)|\\ &\leq Ch^{\min(p+1,k_u,k_v)}(\|u\|_{\min(p+1,k_u)} + \|v\|_{\min(p+1,k_v)})\|u-u_h\|_0.\end{aligned}$$

Combining this with (6.17) and (6.18) then gives (6.16). □

These results can be used to prove optimal convergence for the wave equation in a way analogous to [31] by replacing $a(u,w)$ by $a_h(u,w) := (c\nabla u, \nabla w)_{\mathcal{Q}_h'}$ and by defining the projection operator π_h of [31] such that $a_h(\pi_h u, w) = (\nabla \cdot c\nabla u, w)_{\mathcal{Q}_h}$ for all $w \in U_h$.

6.4.4 Error estimates for the linear elastic case

So far, we only analysed the scalar wave equation, but we can obtain error estimates for the elastic wave equations in an analogous way.

In the linear elastic case, the wave field $\mathbf{u} : \Omega \times (0, T) \to \mathbb{R}^3$ is a vector field and (6.1a) becomes

$$\rho \partial_t^2 \mathbf{u} = \nabla \cdot C : \nabla \mathbf{u} + \mathbf{f} \qquad \text{in } \Omega \times (0, T),$$

with $[\nabla \cdot C : \nabla \mathbf{u}]_i = \sum_{j,k,l=1}^{3} \partial_j C_{jikl} \partial_k u_l$, where $C : \Omega \to \mathbb{R}^{3\times3\times3\times3}$ is the elastic tensor field with symmetries $C_{ijkl} = C_{jikl} = C_{ijlk} = C_{klij}$ and bounds

$$c_0 \|\boldsymbol{\sigma} + \boldsymbol{\sigma}^t\| \leq \|C : \boldsymbol{\sigma}\| \leq c_1 \|\boldsymbol{\sigma}\| \qquad \text{for all } \boldsymbol{\sigma} \in \mathbb{R}^{3\times3},$$

with c_0, c_1 strictly positive constants and $\|\boldsymbol{\sigma}\|^2 := \sum_{i,j=1}^{3} \sigma_{ij}^2$.

The only part of the error analysis that requires some additional work in this case, is the second inequality of Lemma 6.4.2. Instead of $\|\nabla u\|_{\mathcal{Q}'_h} \geq C\|\nabla u\|_0$, we need to show that, if conditions C1-C3, C6, and C7, are satisfied, then

$$\|\nabla \mathbf{u}_h + \nabla \mathbf{u}_h^t\|_{\mathcal{Q}'_h} \geq C\|\nabla \mathbf{u}_h + \nabla \mathbf{u}_h^t\|_0 \qquad \text{for all } \mathbf{u}_h \in U_h^3. \tag{6.22}$$

This result follows from the fact that $\tilde{U}$ is finite dimensional and the relations

$$\|\nabla \mathbf{u}_h + \nabla \mathbf{u}_h^t\|_{\mathcal{Q}'_h}^2 = \sum_{e \in \mathcal{T}_h} \frac{|e|}{|\tilde{e}|} \|\mathbf{J}_e^{-1} \cdot (\tilde{\nabla} \tilde{\mathbf{w}}_e + \tilde{\nabla} \tilde{\mathbf{w}}_e^t) \cdot \mathbf{J}_e^{-t}\|_{\tilde{\mathcal{Q}}'}^2,$$
$$\|\nabla \mathbf{u}_h + \nabla \mathbf{u}_h^t\|_0^2 = \sum_{e \in \mathcal{T}_h} \frac{|e|}{|\tilde{e}|} \|\mathbf{J}_e^{-1} \cdot (\tilde{\nabla} \tilde{\mathbf{w}}_e + \tilde{\nabla} \tilde{\mathbf{w}}_e^t) \cdot \mathbf{J}_e^{-t}\|_{\tilde{e}}^2,$$

where $\mathbf{J}_e := \nabla \phi_e$ is the Jacobian of the element mapping, $\mathbf{J}_e^{-t}$ denotes the transposed of the inverse of the Jacobian $\mathbf{J}_e^{-1}$, $\tilde{\mathbf{w}}_e := \mathbf{J}_e \cdot (\mathbf{u}_h \circ \phi_e) \in \tilde{U}^3$, and $\|\tilde{\boldsymbol{\sigma}}\|_{\tilde{\mathcal{Q}}'}^2 := \sum_{\tilde{\mathbf{x}} \in \tilde{\mathcal{Q}}'} \omega'_{\tilde{\mathbf{x}}} \|\tilde{\boldsymbol{\sigma}}(\tilde{\mathbf{x}})\|^2$.

Using the boundedness of C, (6.22), and Korn's inequality, we can show that the bilinear operator for the elastic case $a_h(\mathbf{u}, \mathbf{w}) := (C : \nabla \mathbf{u}, \nabla \mathbf{w})_{\mathcal{Q}'_h}$ is still coercive. The other parts of the error analysis are analogous to the scalar case.

6.5 Numerical tests

6.5.1 Algorithms for computing the element stiffness matrices

Before we present the numerical tests, we first briefly describe the algorithms for computing the element stiffness matrices. In particular, we show how we efficiently compute the element stiffness matrix-vector products on the fly. We do not store the matrices, since this requires storing and fetching significantly more data, and since it was shown in [54] that an on-the-fly approach is more efficient for higher-degree elements.

To describe the algorithms, let $e \in \mathcal{T}_h$ be an arbitrary element. We introduce the following notation.

- $\{\tilde{\mathbf{x}}_i\}_{i=1}^{n} = \tilde{\mathcal{Q}}$: nodes on reference element $\tilde{e}$. Nodes of the different mass-lumped elements can be found in [31].
- $\tilde{w}_i$: nodal basis function corresponding to $\tilde{\mathbf{x}}_i$.
- $w_i^{(e)} := \tilde{w}_i \circ \phi_e^{-1}$: nodal basis function of the physical element.
- $\{\tilde{\mathbf{x}}_i'\}_{i=1}^{n'} = \tilde{\mathcal{Q}}'$: quadrature points for the stiffness matrix on reference element $\tilde{e}$. Quadrature rules for the different elements are given in Section 6.3.
- $\tilde{\omega}_i'$: quadrature weight corresponding to $\tilde{\mathbf{x}}_i'$.
- $A^{(e)} \in \mathbb{R}^{n\times n}$: the element stiffness matrix.
- $u^{(e)}$: the wave field on e.
- $\underline{\mathbf{u}}^{(e)} \in \mathbb{R}^n$: the wave field at the nodes on e.

When using exact integration, the stiffness matrix-vector product $\underline{\mathbf{v}}^{(e)} := A^{(e)}\underline{\mathbf{u}}^{(e)} \in \mathbb{R}^n$ is given by

$$[A^{(e)}\underline{\mathbf{u}}^{(e)}]_i = \int_e c\nabla w_i^{(e)} \cdot \nabla u^{(e)}\, dx, \tag{6.23}$$

for $i = 1, \ldots, n$. After rewriting the integral as an integral over the reference element, this becomes

$$[A^{(e)}\underline{\mathbf{u}}^{(e)}]_i = \int_{\tilde{e}} \tilde{\nabla}\tilde{w}_i \cdot \tilde{\mathbf{c}}^{(e)} \cdot \tilde{\nabla}\tilde{u}^{(e)}\, d\tilde{x}, \tag{6.24}$$

where $\tilde{u}^{(e)} := u^{(e)} \circ \phi_e$, $\tilde{\nabla}$ is the gradient operator in reference coordinates, and $\tilde{\mathbf{c}}^{(e)} := (c \circ \phi_e)\frac{|e|}{|\tilde{e}|}\mathbf{J}_e^{-t} \cdot \mathbf{J}_e^{-1}$ is a tensor field, with $\mathbf{J}_e := \nabla \phi_e$ the Jacobian of the element mapping and $\mathbf{J}_e^{-t}$ the transposed of $\mathbf{J}_e^{-1}$. When c is constant within each element, then $\tilde{\mathbf{c}}^{(e)}$ is also constant and we can compute (6.24) using the algorithm of [54]:

$$[A^{(e)}\underline{\mathbf{u}}^{(e)}]_i = \sum_{i_D,j_D=1}^{3} \tilde{c}^{(e)}_{i_D,j_D} \left(\sum_{j=1}^{n} B^{(i_D,j_D)}_{ij} \underline{\mathbf{u}}^{(e)}_j \right) \tag{6.25}$$

where $B^{(i_D,j_D)} \in \mathbb{R}^{n \times n}$ are precomputed matrices, given by

$$B^{(i_D,j_D)}_{ij} = \int_{\tilde{e}} (\tilde{\partial}_{i_D} \tilde{w}_i)(\tilde{\partial}_{j_D} \tilde{w}_j)\, d\tilde{x},$$

for $i_D, j_D = 1,2,3$, $i,j = 1,\dots,n$, with $\tilde{\partial}_{i_D}$ the derivative in reference coordinate i_D. We can reduce the number of computations in (6.25) using the fact that $\tilde{\mathbf{c}}^{(e)}$ is symmetric:

$$[A^{(e)}\underline{\mathbf{u}}^{(e)}]_i = \sum_{i_D=1}^{3} \sum_{j_D=1}^{i_D} \tilde{c}^{(e)}_{i_D,j_D} \left(\sum_{j=1}^{n} \hat{B}^{(i_D,j_D)}_{ij} \underline{\mathbf{u}}^{(e)}_j \right) \tag{6.26}$$

where $\hat{B}^{(i_D,j_D)} := B^{(i_D,j_D)} + B^{(j_D,i_D)}$ if $i_D \neq j_D$ and $\hat{B}^{(i_D,j_D)} := B^{(i_D,j_D)}$ when $i_D = j_D$. The complete algorithm can then be described as follows:

A1. Compute $\epsilon^{(i_D,j_D)} \in \mathbb{R}^n$ for $i_D = 1,2,3$, $j_D \leq i_D$:

$$\epsilon^{(i_D,j_D)}_i = \sum_{j=1}^{n} \hat{B}^{(i_D,j_D)}_{ij} \underline{\mathbf{u}}^{(e)}_j.$$

A2. Compute $A^{(e)}\underline{\mathbf{u}}^{(e)} \in \mathbb{R}^n$:

$$[A^{(e)}\underline{\mathbf{u}}^{(e)}]_i = \sum_{i_D=1}^{3} \sum_{j_D=1}^{i_D} \tilde{c}^{(e)}_{i_D,j_D} \epsilon^{(i_D,j_D)}_i.$$

The computational work is dominated by the first step where 6 matrix-vector products with matrices of size $n \times n$ are computed.

Alternatively, we can compute $A^{(e)}\underline{\mathbf{u}}^{(e)}$ by evaluating the integrals with a quadrature rule. Equation (6.24) then becomes

$$[A^{(e)}\underline{\mathbf{u}}^{(e)}]_i = \sum_{k=1}^{n'} \tilde{\nabla} \tilde{w}_i(\tilde{\mathbf{x}}'_k) \cdot \tilde{\mathbf{c}}^{(e,k)} \cdot \tilde{\nabla} \tilde{u}^{(e)}(\tilde{\mathbf{x}}'_k), \tag{6.27}$$

where $\tilde{\mathbf{c}}^{(e,k)} := \tilde{\omega}'_k \tilde{\mathbf{c}}^{(e)}(\tilde{\mathbf{x}}'_k) \in \mathbb{R}^{3\times 3}$. We can compute this as follows:

$$[A^{(e)}\underline{\mathbf{u}}^{(e)}]_i = \sum_{k=1}^{n'} \sum_{i_D=1}^{3} D_{ki}^{(i_D)} \left(\sum_{j_D=1}^{3} \tilde{c}_{i_D,j_D}^{(e,k)} \left(\sum_{j=1}^{n} D_{kj}^{(j_D)} \underline{\mathbf{u}}_j^{(e)} \right) \right), \tag{6.28}$$

where $D^{(i_D)} \in \mathbb{R}^{n'\times n}$ are precomputed matrices, given by

$$D_{ki}^{(i_D)} = (\tilde{\partial}_{i_D} \tilde{w}_i)(\tilde{\mathbf{x}}'_k).$$

The complete algorithm can be described as follows:

B1. Compute $\epsilon^{(j_D)} \in \mathbb{R}^{n'}$ for $j_D = 1, 2, 3$:

$$\epsilon_k^{(j_D)} = \sum_{j=1}^{n} D_{kj}^{(j_D)} \underline{\mathbf{u}}_j^{(e)}.$$

B2. Compute $\sigma^{(i_D)} \in \mathbb{R}^{n'}$ for $i_D = 1, 2, 3$:

$$\sigma_k^{(i_D)} = \sum_{j_D=1}^{3} \tilde{c}_{i_D,j_D}^{(e,k)} \epsilon_k^{(j_D)}.$$

B3. Compute $A^{(e)}\underline{\mathbf{u}}^{(e)} \in \mathbb{R}^{n}$:

$$[A^{(e)}\underline{\mathbf{u}}^{(e)}]_i = \sum_{i_D=1}^{3} \left(\sum_{k=1}^{n'} D_{ki}^{(i_D)} \sigma_k^{(i_D)} \right).$$

The computational work for this algorithm is dominated by the first and third step, which both require 3 matrix-vector products with matrices of size $n' \times n$, so 6 of these matrix-vector products in total. Since $n' < n$ for all the quadrature rules presented in this chapter, this number of computations is smaller then for the previous algorithm, although only slightly. However, as we will show next, this quadrature-based algorithm is significantly more efficient than the exact-integral algorithm in case of linear elasticity. Moreover, this quadrature-based algorithm also works if c varies within the element.

In case of linear elasticity, the wave field $\mathbf{u} : \Omega \times (0, T) \to \mathbb{R}^3$ is a vector field and the term $c\nabla u$ becomes the stress tensor $C : \nabla\mathbf{u}$, with $C \in \mathbb{R}^{3\times3\times3\times3}$ the order-four elasticity tensor with symmetries $C_{ijkl} = C_{jikl} = C_{ijlk} = C_{klij}$ and $[C : \nabla\mathbf{u}]_{ij} := \sum_{k,l=1}^{3} C_{ijkl}\partial_l u_k$. The vector

$\underline{\mathbf{u}}^{(e)} \in \mathbb{R}^{3n}$ can in this case be written as a concatenation of three vectors $\underline{\mathbf{u}}^{(e,1)}, \underline{\mathbf{u}}^{(e,2)}, \underline{\mathbf{u}}^{(e,3)} \in \mathbb{R}^n$, where $\underline{\mathbf{u}}^{(e,i)}$ is the wave field component u_i at the nodes on e. The parameter $\tilde{\mathbf{c}}^{(e)}$ becomes $\tilde{C}^{(e)} := \frac{|e|}{|\tilde{e}|}\mathbf{J}_e^{-t} \cdot (C \circ \phi_e) \cdot \mathbf{J}_e^{-1}$, where $[\mathbf{J}_e^{-t} \cdot C \cdot \mathbf{J}_e^{-1}]_{ijkl} = \sum_{p,q=1}^{3} [\mathbf{J}_e^{-t}]_{ip} C_{pjkq} [\mathbf{J}_e^{-1}]_{ql}$, and $\tilde{\mathbf{c}}^{(e,k)}$ becomes $\tilde{C}^{(e,k)} := \tilde{\omega}'_k \tilde{C}^{(e)}(\tilde{\mathbf{x}}'_k)$. The algorithm for computing the element stiffness matrix-vector product using exact integration then becomes

A1*. Compute $\epsilon^{(i_D,j_D,j_V)} \in \mathbb{R}^n$ for $i_D, j_D, j_V = 1, 2, 3$:

$$\epsilon_i^{(i_D,j_D,j_V)} = \sum_{j=1}^{n} B_{ij}^{(i_D,j_D)} \underline{\mathbf{u}}_j^{(e,j_V)}.$$

A2*. Define $\underline{\mathbf{v}}^{(e)} := A^{(e)} \underline{\mathbf{u}}^{(e)}$. Compute $\underline{\mathbf{v}}^{(e,i_V)} \in \mathbb{R}^n$ for $i_V = 1, 2, 3$:

$$\underline{\mathbf{v}}_i^{(e,i_V)} = \sum_{i_D,j_D,j_V=1}^{3} \tilde{C}^{(e)}_{i_D,i_V,j_V,j_D} \epsilon_i^{(i_D,j_D,j_V)}.$$

The computational work is again dominated by the first step, which now requires 27 matrix-vector products with matrices of size $n \times n$.

When using a quadrature rule, the algorithm becomes

B1*. Compute $\epsilon^{(j_D,j_V)} \in \mathbb{R}^{n'}$ for $j_D, j_V = 1, 2, 3$:

$$\epsilon_k^{(j_D,j_V)} = \sum_{j=1}^{n} D_{kj}^{(j_D)} \underline{\mathbf{u}}_j^{(e,j_V)}.$$

B2*. Compute $\sigma^{(i_D,i_V)} \in \mathbb{R}^{n'}$ for $i_D, i_V = 1, 2, 3$:

$$\sigma_k^{(i_D,i_V)} = \sum_{j_D,j_V=1}^{3} \tilde{C}^{(e,k)}_{i_D,i_V,j_V,j_D} \epsilon_k^{(j_D,j_V)}.$$

B3*. Define $\underline{\mathbf{v}}^{(e)} := A^{(e)} \underline{\mathbf{u}}^{(e)}$. Compute $\underline{\mathbf{v}}^{(e,i_V)} \in \mathbb{R}^n$ for $i_V = 1, 2, 3$:

$$\underline{\mathbf{v}}_i^{(e,i_V)} = \sum_{i_D=1}^{3} \left(\sum_{k=1}^{n'} D_{ki}^{(i_D)} \sigma_k^{(i_D,i_V)} \right).$$

The computational work for this algorithm is dominated again by the first and third step, which now both require 9 matrix-vector products with matrices of size $n' \times n$, so 18 of these matrix-vector products in total. The number of computations is therefore reduced by more than a factor 1.5 when compared to the algorithm based on exact integration. Furthermore, the quadrature-based algorithm can also handle tensor fields C that vary within the element.

Both algorithms can be slightly improved by exploiting the fact that the rows and columns of the matrices $B^{(i_D,j_D)}$ and the columns of matrices $D^{(i_D)}$ sum to zero. Furthermore, in case of isotropic elasticity, steps A2* and B2* can be computed more efficiently by exploiting the simple structure of the elasticity tensor C.

In the next section, we demonstrate the superiority of the quadrature-based algorithm for the case of non-constant parameters and linear elasticity.

6.5.2 Acoustic wave on a heterogeneous domain

We first test the methods for an acoustic wave propagation problem with a heterogeneous domain. The acoustic wave equation is given by

$$\frac{1}{\rho c^2}\partial_t^2 p = \nabla \cdot \frac{1}{\rho}\nabla p, \qquad \text{in } \Omega \times (0,T), \tag{6.29}$$

with spatial domain $\Omega \subset \mathbb{R}^3$, time interval $(0,T)$, pressure field $p : \Omega \times (0,T) \to \mathbb{R}$, mass density $\rho : \Omega \to \mathbb{R}$, and acoustic wave speed $c : \Omega \to \mathbb{R}$. We choose $\Omega := (-L_1, L_1) \times (-L_2, L_2) \times (-L_3, L_3)$ and impose zero Neumann boundary conditions.

To construct an analytic solution, let $X_i := x_i + \frac{a_i}{m_i}\cos(m_i x_i)$, for $i = 1,2,3$, be distorted coordinates, with $m_i := \frac{1}{2}\pi/L_i$ and $a_i \in [0,1)$, and define $g_i := \partial_i X_i = 1 - a_i \sin(m_i x_i)$. Also let $\rho_0 \in \mathbb{R}$ be the average mass density, $c_0 \in \mathbb{R}$ the average wave speed, $\mathbf{k} \in \mathbb{R}^3$ the wave vector, and $\omega := c_0|\mathbf{k}|$ the angular velocity, and let parameters ρ and c be given by

$$\begin{aligned} \rho(\mathbf{x}) &:= \rho_0 g_1(x_1) g_2(x_2) g_3(x_3), \\ c(\mathbf{x}) &:= c_0 \sqrt{\frac{k_1^2 + k_2^2 + k_3^2}{k_1^2 g_1^2(x_1) + k_2^2 g_2^2(x_2) + k_3^2 g_3^2(x_3)}}. \end{aligned}$$

Then the standing wave, given by

$$p(\mathbf{x}, t) = \cos(\omega t)\sin(k_1 X_1)\sin(k_2 X_2)\sin(k_3 X_3),$$

is a solution of (6.29) that satisfies the zero Neumann boundary conditions.

Now, set $L_i = 1$ km, $a_i = 0.2$, $k_i = 3m_i$, for $i = 1, 2, 3$, and $c_0 = 2$ km, $\rho_0 = 2$ g/cm^3. To test the numerical methods, we use $p(\mathbf{x}, 0)$ and $\partial_t p(\mathbf{x}, 0)$ as initial conditions. We test on multiple unstructured meshes and simulate in time using a fourth-order time-stepping scheme [18] with time step sizes based on [55] scaled with a factor 0.9. The root mean square (RMS) error is computed after two time oscillations, so at $T = 4\pi/\omega \approx 0.7698$ s.

Table 6.6: Power-law fits of the left graphs of Figures 6.1 and 6.2. Convergence rates are given in bold font.

	RMS error	
Method	Figure 6.1	Figure 6.2
2,4 15	$(3.1 \times 10^2)N^{(-1/3\times\mathbf{3.3})}$	$(2.4 \times 10^1)N^{(-1/3\times\mathbf{2.6})}$
3,4 32	$(2.2 \times 10^3)N^{(-1/3\times\mathbf{4.1})}$	$(2.9 \times 10^0)N^{(-1/3\times\mathbf{2.1})}$
4,4 60	$(5.5 \times 10^4)N^{(-1/3\times\mathbf{5.3})}$	$(9.4 \times 10^0)N^{(-1/3\times\mathbf{2.2})}$
4,4 61	$(7.8 \times 10^4)N^{(-1/3\times\mathbf{5.3})}$	$(9.6 \times 10^0)N^{(-1/3\times\mathbf{2.2})}$
4,4 65	$(7.2 \times 10^4)N^{(-1/3\times\mathbf{5.3})}$	$(1.0 \times 10^0)N^{(-1/3\times\mathbf{2.2})}$

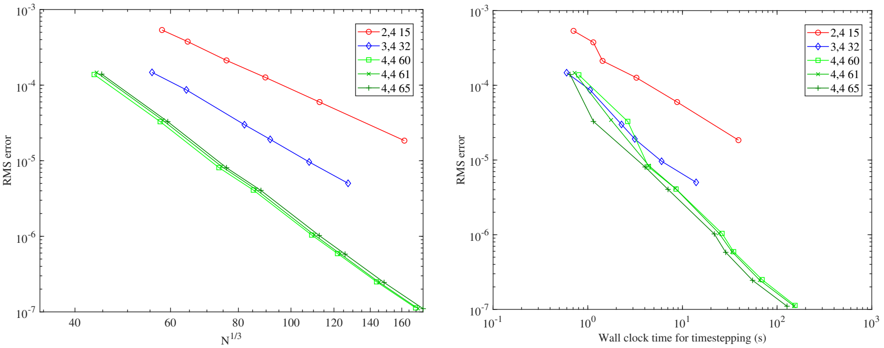


Figure 6.1: RMS errors for the acoustic test case as a function of the cube root of the number of degrees of freedom (left) and as a function of the wall clock time (right). In the legend, $p, K\ n$ refers to the element of degree p with n nodes, combined with an order-K time-stepping scheme. The element stiffness matrices were evaluated using a quadrature rule.

Figure 6.1 shows the RMS error plotted against the cube root of the number of degrees of freedom N and the wall-clock time for the mass-lumped tetrahedral element methods using the quadrature-based algorithm for the stiffness matrix as discussed in the previous subsection. The simulations shown here were performed with an OpenMP implementation on 24 cores of two Intel® Xeon® E5-2680 v3 CPUs running at 2.50GHz. Power-law fits of the left graph are also shown in Table 6.6. This graph shows optimal convergence rates of order $p + 1$ and thereby confirms the error estimates of the previous section. In particular, it confirms that optimal convergence rates are maintained, even though the spatial parameters ρ, c vary within the element.

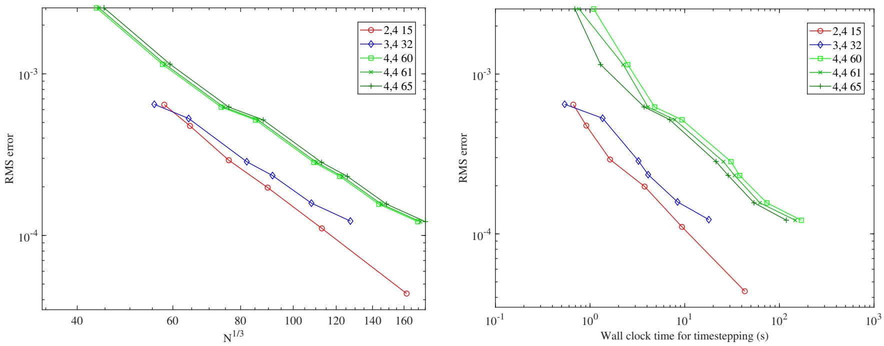


Figure 6.2: Same as Figure 6.1, but using exact integration to evaluate the stiffness matrix and using a piecewise-constant approximation of the mass density ρ. All methods only converge with second order due to the parameter approximation and higher-degree methods only result in more degrees of freedom and computation time.

Figure 6.2 shows the same as Figure 6.1 for the methods using exact integration to evaluate the stiffness matrix and using a piecewise constant approximation of the mass density ρ. Power-law fits of the left graph are again given in Table 6.6. The graph shows that, due to the piecewise constant approximation, only second-order convergence rates are obtained. The higher-degree elements now only result in more computations per element, without any significant gain in accuracy. When comparing with Figure 6.1, it follows that the quadrature-based approach is much more efficient than using exact integration with piecewise-constant parameter approximations.

6.5.3 Elastic wave on a homogeneous domain

We also test the methods for an elastic wave propagation problem on a homogeneous domain. The elastic wave equations are given by

$$\rho\partial_t^2\mathbf{u} = \nabla\cdot C : \nabla\mathbf{u} + \mathbf{f}, \qquad\qquad \text{in } \Omega\times(T_0,T_1),$$

with $\mathbf{u} : \Omega\times(T_0,T_1)\to\mathbb{R}^3$ the displacement field, $\mathbf{f} : \Omega\times(T_0,T_1)\to\mathbb{R}^3$ the force field, $\rho : \Omega\to\mathbb{R}$ the mass density, and $C : \Omega\to\mathbb{R}^{3\times3\times3\times3}$ the elasticity tensor. We consider an isotropic elastic medium, so $C : \nabla\mathbf{u} = \lambda(\nabla\cdot\mathbf{u})\mathbf{I} + \mu(\nabla\mathbf{u}+\nabla\mathbf{u}^t)$, with $\mathbf{I}\in\mathbb{R}^{3\times3}$ the identity tensor, $\nabla\mathbf{u}^t$ the transposed of $\nabla\mathbf{u}$, and $\lambda,\mu : \Omega\to\mathbb{R}$ the Lamé parameters.

We choose a domain $\Omega = [-2,2]\times[-1,1]\times[0,2]$ km^3 with zero Neumann boundary conditions, and set the parameters with a constant mass density $\rho = 2$ g/cm^3, primary wave velocity $v_P := \sqrt{(\lambda+2\mu)/\rho} = 2$ km/s, and secondary/shear wave velocity $w_S := \sqrt{\mu/\rho} = 1.2$ km/s. A unit vertical force source with a 7-Hz Ricker-wavelet is placed at $\mathbf{x}_{src} := (0,0,1000)$ m and receivers are placed between $x_r = -1375$ m and $x_r = 1375$ m with a 50-m interval at $y_r = 200$ m and $z_r = 800$ m. The exact solution can be found in [4]. The simulation time is chosen such that reflections caused by the boundary conditions do not reach the receivers.

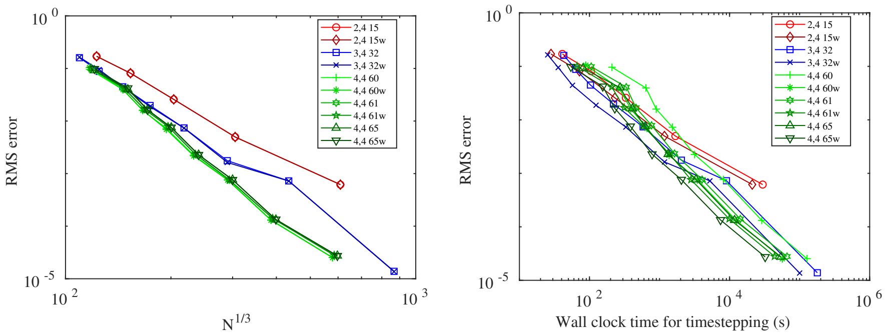


Figure 6.3: RMS errors for the elastic test case as a function of the cube root of the number of degrees of freedom (left) and as a function of the wall clock time (right). In the legend, $p, K\ n$ refers to the element of degree p with n nodes, combined with an order-K time-stepping scheme. Suffix w denotes elements for which the stiffness matrix was evaluated using the quadrature-based approach described in subsection 6.5.1, while for the other elements we used the exact-integral algorithm.

We tested the methods on multiple unstructured meshes and simulated over the time interval $(-0.3, 0.6)$ s, with the time-stepping algorithm as in the previous test case. Simulations were also carried out in the same environment as in the previous test case. The RMS error is based on the errors at all receivers and for all directional components and is plotted against the cube root of the degrees of freedom N and elapsed time in Figure 6.3. The left graph shows that the methods converge with optimal order. The right graph shows that for the degree-3 and degree-4 elements, the quadrature-based procedure reduces the computational cost by a factor of ca. 1.5. For the degree-2 element, this procedure also results in a moderate speed up.

6.6 Conclusion

We presented new and efficient quadrature rules for evaluating the stiffness matrices of mass-lumped tetrahedral elements for wave propagation modelling. These quadrature rules can significantly reduce the number of computations compared to algorithms that evaluate the stiffness matrix using exact integration, and can handle spatial parameters that vary within the element without loss of the optimal convergence rate. Obtaining these quadrature rules is not trivial, since degree-p mass-lumped tetrahedral element spaces contain, apart from polynomials up to degree p, numerous additional higher-degree bubble functions when $p \geq 2$. To obtain efficient quadrature rules, we therefore carefully analysed the stability and accuracy requirements needed to maintain optimal convergence rates. The resulting conditions are presented in this chapter, and we prove that, if these conditions are met, the resulting method can maintain an optimal order of convergence, even when the spatial parameters vary within the element. We found quadrature rules that satisfy these conditions for recently developed mass-lumped tetrahedral elements of degrees two to four.

For the degree-2 element, the quadrature rule with the least number of points we could find was the degree-5 accurate 14-point quadrature rule of [38], but for the degree-3 and degree-4 elements, we found new quadrature rules that require significantly less integration points than existing quadrature rules. Several numerical examples illustrate the accuracy and efficiency of this approach and its superiority to evaluating the integrals for the stiffness matrix exactly. In particular, the quadrature-based approach results in a computational speed-up of around a factor 1.5 in case of elastic waves and maintains an optimal convergence rate when the spatial parameters vary within the element.

Chapter 7

Conclusions and Outlook

In this dissertation, new and existing finite element methods for wave propagation modelling were presented, analysed, and compared. The main conclusions are as follows:

- The new penalty term bound for the discontinuous Galerkin method presented in this dissertation results in a better accuracy and significantly less time steps compared to other bounds available in literature. The computational speed-up for a given accuracy can be an order of magnitude when using this new bound.

- The time step bound for the discontinuous Galerkin method presented in this dissertation can be efficiently computed, guarantees stability of the method, and is always close to the largest allowed time step size. Numerical tests show that the time step bound does not become smaller than a factor 1.4 compared to the largest allowed time step size.

- The theory behind the stability of finite element methods combined with standard explicit time-stepping schemes can not be readily extended to a basic local time-stepping scheme. In particular, for a basic local time-stepping scheme, instabilities can always be present, unless the local time step size is applied everywhere, but this would turn the scheme back into a standard time-stepping scheme.

- The new mass-lumped tetrahedral elements presented in this dissertation are much more efficient than the discontinuous Galerkin and other mass-lumped methods available in literature. The computational speed-up for a given accuracy can be an order of magnitude.

- Based on a dispersion analysis, the degree-2 mass-lumped element method is the most efficient for an accuracy between 0.01 to 0.001, while higher-degree mass-lumped element methods become more efficient for dispersion errors below 0.001. The dispersion analysis presented in this dissertation also provides estimates for the required mesh resolution for a given accuracy.

- The new quadrature rules for evaluating the stiffness matrices of mass-lumped tetrahedral elements presented in this dissertation require up to 25% less quadrature points than other quadrature rules available in literature. Using a quadrature rule to compute the stiffness matrix can significantly speed-up the finite element method compared to algorithms that evaluate the stiffness matrix exactly. In the case of linear elasticity, this speed-up is around a factor 1.5. Furthermore, the quadrature-approach can handle material parameters that vary within the element without loss of accuracy, while approximating the material parameters with piecewise constant parameters completely destroys the efficiency of higher-order methods.

While the work presented in this dissertation greatly improves the efficiency of finite element methods for seismic modelling and wave propagation modelling in general, addressing the following topics in the near-future could even further improve the finite element method:

- **Mesh generation**: generating a high-quality mesh for finite element modelling can take as much time as the simulation itself. This is a major drawback of finite element methods and more efficient and robust mesh generators would therefore greatly improve the efficiency of these methods.

- **Postprocessing**: while the dispersion error converges with order $2p$, with p the degree of the finite element method, the finite element method only converges with order $p+1$ due to interpolation errors. If we can efficiently regain the $2p$ convergence rate by preprocessing and postprocessing the numerical data, this would significantly improve the efficiency of higher-degree finite element methods.

- **Hybrid scheme**: while finite element methods can more efficiently capture the effect of complex topographies, finite difference methods are more efficient in the majority of the domain where the material parameters vary smoothly. An interesting research question is if it is possible to combine these methods and only use the finite element method near the topography.

- **Testing with full waveform inversion**: after having the finite element method fully optimized, it is interesting to test how much efficiency is gained when this method is applied to full waveform inversion.

Bibliography

[1] C. Agut and J. Diaz. Stability analysis of the Interior Penalty Discontinuous Galerkin method for the wave equation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(03):903–932, 2013.

[2] M. Ainsworth. Dispersive and dissipative behaviour of high order discontinuous Galerkin finite element methods. *Journal of Computational Physics*, 198(1):106–130, 2004.

[3] M. Ainsworth, P. Monk, and W. Muniz. Dispersive and dissipative properties of discontinuous Galerkin finite element methods for the second-order wave equation. *Journal of Scientific Computing*, 27(1-3):5–40, 2006.

[4] K. Aki and P. G. Richards. Quantitative Seismology, Theory and Methods, Vol. 1. *New York*, 1980.

[5] P. Antonietti, I. Mazzieri, A. Quarteroni, and F. Rapetti. Nonconforming high order approximations of the elastodynamics equation. *Computer Methods in Applied Mechanics and Engineering*, 209:212–238, 2012.

[6] P. F. Antonietti, C. Marcati, I. Mazzieri, and A. Quarteroni. High order discontinuous Galerkin methods on simplicial elements for the elastodynamics equation. *Numerical Algorithms*, 71(1):181–206, 2016.

[7] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 39(5):1749–1779, 2002.

[8] C. Baldassari, H. Barucq, H. Calandra, and J. Diaz. Numerical performances of a hybrid local-time stepping strategy applied to the reverse time migration. *Geophysical Prospecting*, 59(5):907–919, 2011.

[9] D. Boffi. Finite element approximation of eigenvalue problems. *Acta Numerica*, 19:1–120, 2010.

[10] A. Buffa and I. Perugia. Discontinuous Galerkin approximation of the Maxwell eigenproblem. *SIAM Journal on Numerical Analysis*, 44(5):2198–2226, 2006.

[11] M. J. S. Chin-Joe-Kong, W. A. Mulder, and M. Van Veldhuizen. Higher-order triangular and tetrahedral finite elements with mass lumping for solving the wave equation. *Journal of Engineering Mathematics*, 35(4):405–426, 1999.

[12] P. G. Ciarlet. *The finite element method for elliptic problems.* North-Holland Publishing Company, Amsterdam, New York, Oxford, 1978.

[13] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework. *Mathematics of Computation*, 52(186):411–435, 1989.

[14] G. Cohen. *Higher-order numerical methods for transient wave equations.* Springer, 2002.

[15] G. Cohen, P. Joly, J. E. Roberts, and N. Tordjman. Higher order triangular finite elements with mass lumping for the wave equation. *SIAM Journal on Numerical Analysis*, 38(6):2047–2078, 2001.

[16] G. Cohen, P. Joly, and N. Tordjman. Higher order triangular finite elements with mass lumping for the wave equation. In *Proceedings of the Third International Conference on Mathematical and Numerical Aspects of Wave Propagation*, pages 270–279. SIAM Philadelphia, 1995.

[17] T. Cui, W. Leng, D. Lin, S. Ma, and L. Zhang. High order mass-lumping finite elements on simplexes. *Numerical Mathematics: Theory, Methods and Applications*, 10(2):331–350, 2017.

[18] M. Dablain. The application of high-order differencing to the scalar wave equation. *Geophysics*, 51(1):54–66, 1986.

[19] J. D. De Basabe and M. K. Sen. Grid dispersion and stability criteria of some common finite-element methods for acoustic and elastic wave equations. *Geophysics*, 72(6):T81–T95, 2007.

[20] J. D. De Basabe and M. K. Sen. Stability of the high-order finite elements for acoustic or elastic wave propagation with high-order time stepping. *Geophysical Journal International*, 181(1):577–590, 2010.

[21] J. D. De Basabe, M. K. Sen, and M. F. Wheeler. The interior penalty discontinuous Galerkin method for elastic wave propagation: grid dispersion. *Geophysical Journal International*, 175(1):83–93, 2008.

[22] J. Diaz and M. J. Grote. Energy conserving explicit local time stepping for second-order wave equations. *SIAM Journal on Scientific Computing*, 31(3):1985–2014, 2009.

[23] J. Diaz and M. J. Grote. Multi-level explicit local time-stepping methods for second-order wave equations. *Computer Methods in Applied Mechanics and Engineering*, 291:240–265, 2015.

[24] G. E. Dimock. *The unity of the Odyssey.* The University of Massachusetts Press, 1990.

[25] M. Dumbser, M. Käser, and E. F. Toro. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes-V. Local time stepping and p-adaptivity. *Geophysical Journal International*, 171(2):695–717, 2007.

[26] A. H. Encyclopedia. Namazu. `https://www.ancient.eu/Namazu/`. Retrieved at 2018-03-26.

[27] Y. Epshteyn and B. Rivière. Estimation of penalty parameters for symmetric interior penalty Galerkin methods. *Journal of Computational and Applied Mathematics*, 206(2):843–872, 2007.

[28] J. Etgen, S. H. Gray, and Y. Zhang. An overview of depth imaging in exploration geophysics. *Geophysics*, 74(6):WCA5–WCA17, 2009.

[29] I. Fried and D. S. Malkus. Finite element mass matrix lumping by numerical integration with no convergence rate loss. *International Journal of Solids and Structures*, 11(4):461–466, 1975.

[30] S. Geevers, W. A. Mulder, and J. J. W. van der Vegt. Dispersion properties of explicit finite element methods for wave propagation modelling on tetrahedral meshes. Journal of Scientific Computing (2018). https://doi.org/10.1007/s10915-018-0709-7.

[31] S. Geevers, W. A. Mulder, and J. J. W. van der Vegt. New higher-order mass-lumped tetrahedral elements for wave propagation modelling. Accepted for publication in SIAM Journal on Scientific Computing (2018), arXiv:1803.10065.

[32] S. Geevers and J. J. W. van der Vegt. Sharp penalty term and time step bounds for the interior penalty discontinuous Galerkin method for linear hyperbolic problems. *SIAM Journal on Scientific Computing*, 39(5):A1851–A1878, 2017.

[33] M. J. Grote, M. Mehlin, and T. Mitkova. Runge–Kutta-based explicit local time-stepping methods for wave propagation. *SIAM Journal on Scientific Computing*, 37(2):A747–A775, 2015.

[34] M. J. Grote and T. Mitkova. Explicit local time-stepping methods for Maxwell's equations. *Journal of Computational and Applied Mathematics*, 234(12):3283–3302, 2010.

[35] M. J. Grote and T. Mitkova. High-order explicit local time-stepping methods for damped wave equations. *Journal of Computational and Applied Mathematics*, 239:270–289, 2013.

[36] M. J. Grote, A. Schneebeli, and D. Schötzau. Discontinuous Galerkin finite element method for the wave equation. *SIAM Journal on Numerical Analysis*, 44(6):2408–2431, 2006.

[37] M. J. Grote, A. Schneebeli, and D. Schötzau. Interior penalty discontinuous Galerkin method for Maxwell's equations: optimal L2-norm error estimates. *IMA Journal of Numerical Analysis*, 28(3):440–468, 2008.

[38] A. Grundmann and H. M. Möller. Invariant integration formulas for the n-simplex by combinatorial methods. *SIAM Journal on Numerical Analysis*, 15(2):282–290, 1978.

[39] F. Q. Hu, M. Hussaini, and P. Rasetarinera. An analysis of the discontinuous Galerkin method for wave propagation problems. *Journal of Computational Physics*, 151(2):921–946, 1999.

[40] B. M. Irons and G. Treharne. A bound theorem in eigenvalues and its practical applications. In *Proceedings of the 2nd Conference on Matrix Method in Structural Mechanics, Wright-Patterson AFB, Ohio*, 1971.

[41] M. Käser, V. Hermann, and J. de la Puente. Quantitative accuracy analysis of the discontinuous Galerkin method for seismic wave propagation. *Geophysical Journal International*, 173(3):990–999, 2008.

[42] D. Komatitsch and J. Tromp. Introduction to the spectral element method for three-dimensional seismic wave propagation. *Geophysical Journal International*, 139(3):806–822, 1999.

[43] D. Komatitsch and J.-P. Vilotte. The spectral element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures. *Bulletin of the Seismological Society of America*, 88(2):368–392, 1998.

[44] A. Kononov, S. Minisini, E. Zhebel, and W. A. Mulder. A 3D tetrahedral mesh generator for seismic problems. In *74th EAGE Conference and Exhibition incorporating EUROPEC 2012*, 2012.

[45] P. D. Lax and B. Wendroff. Difference schemes for hyperbolic equations with high order of accuracy. *Communications on Pure and Applied Mathematics*, 17(3):381–398, 1964.

[46] J. L. Lions and E. Magenes. *Non-homogeneous boundary value problems and applications*, volume 1. Springer Verlag, 2012.

[47] V. Lisitsa. Dispersion analysis of discontinuous Galerkin method on triangular mesh for elastic wave equation. *Applied Mathematical Modelling*, 40(7-8):5077–5095, 2016.

[48] T. Liu, M. K. Sen, T. Hu, J. D. De Basabe, and L. Li. Dispersion analysis of the spectral element method using a triangular mesh. *Wave Motion*, 49(4):474–483, 2012.

[49] Y. Liu, J. Teng, T. Xu, and J. Badal. Higher-order triangular spectral element method with optimized cubature points for seismic wavefield modeling. *Journal of Computational Physics*, 336:458–480, 2017.

[50] S. Minjeaud and R. Pasquetti. High Order C^0-Continuous Galerkin Schemes for High Order PDEs, Conservation of Quadratic Invariants and Application to the Korteweg-de Vries Model. *Journal of Scientific Computing*, 74(1):491–518, 2018.

[51] W. A. Mulder. A comparison between higher-order finite elements and finite differences for solving the wave equation. In *Proceedings of the*

Second ECCOMAS Conference on Numerical Methods in Engineering, pages 344–350. John Wiley & Sons, 1996.

[52] W. A. Mulder. Spurious modes in finite-element discretizations of the wave equation may not be all that bad. *Applied Numerical Mathematics*, 30(4):425–445, 1999.

[53] W. A. Mulder. New triangular mass-lumped finite elements of degree six for wave propagation. *Progress in Electromagnetics Research PIER,(141)*, pages 671–692, 2013.

[54] W. A. Mulder and R. Shamasundar. Performance of continuous mass-lumped tetrahedral elements for elastic wave propagation with and without global assembly. *Geophysical Journal International*, 207(1):414–421, 2016.

[55] W. A. Mulder, E. Zhebel, and S. Minisini. Time-stepping stability of continuous and discontinuous finite-element methods for 3-D wave propagation. *Geophysical Journal International*, 196(2):1123–1133, 2014.

[56] NASA, DTAM project team. Quake epicenters. `http://denali.gsfc.nasa.gov/dtam/seismic/`, Public Domain, `https://commons.wikimedia.org/w/index.php?curid=35429`.

[57] M. Ohnaka. *The physics of rock failure and earthquakes.* Cambridge University Press, 2013.

[58] S. J. Owen. A survey of unstructured mesh generation technology. In *IMR*, pages 239–267, 1998.

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

[60] W. H. Reed and T. Hill. Triangular mesh methods for the neutron transport equation. Technical report, Los Alamos Scientific Lab., N. Mex.(USA), 1973.

[61] C. F. Richter. An instrumental earthquake magnitude scale. *Bulletin of the Seismological Society of America*, 25(1):1–32, 1935.

[62] B. Riviere and M. F. Wheeler. Discontinuous finite element methods for acoustic and elastic wave problems. *Contemporary Mathematics*, 329:271–282, 2003.

[63] G. Seriani and E. Priolo. Spectral element method for acoustic wave simulation in heterogeneous media. *Finite elements in analysis and design*, 16(3-4):337–348, 1994.

[64] K. Shahbazi. An explicit expression for the penalty parameter of the interior penalty method. *Journal of Computational Physics*, 205(2):401–407, 2005.

[65] R. E. Sheriff. Encyclopedic dictionary of applied geophysics: Tulsa, Oklahoma. *Society of Exploration Geophysicists*, 2002.

[66] R. E. Sheriff and L. P. Geldart. *Exploration seismology.* Cambridge university press, 1995.

[67] H. Si. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software (TOMS)*, 41(2):11, 2015.

[68] S. Sturluson. Prose Edda. ISBN 1-156-78621-5.

[69] USGS. Common myths about earthquakes. `https://web.archive.org/web/20060925135349/http://earthquake.usgs.gov/learning/faq.php?categoryID=6&faqID=110`. Retrieved at 2018-03-26.

[70] USGS. Earthquake facts. `https://earthquake.usgs.gov/learn/facts.php`. Retrieved at 2018-03-26.

[71] USGS. Earthquakes with 50,000 or more deaths. `https://web.archive.org/web/20130605122458/http://earthquake.usgs.gov/earthquakes/world/most_destructive.php`. Retrieved at 2018-03-26.

[72] USGS. M 5.7 - 6km WSW of Amatrice, Italy. `https://earthquake.usgs.gov/earthquakes/eventpage/us10007twj#executive`. Retrieved at 2018-03-26.

[73] USGS. M 6.2 - 10km SE of Norcia, Italy. `https://earthquake.usgs.gov/earthquakes/eventpage/us10006g7d#executive`. Retrieved at 2018-03-26.

[74] USGS. M 7.8 - 36km E of Khudi, Nepal. `https://earthquake.usgs.gov/earthquakes/eventpage/us20002926#executive`. Retrieved at 2018-03-26.

[75] USGS. Magnitude 8.9 - NEAR THE EAST COAST OF HONSHU, JAPAN. `https://web.archive.org/web/20110313154037/http://earthquake.usgs.gov/earthquakes/eqinthenews/2011/usc0001xgp/`. Retrieved at 2018-03-26.

[76] S. Uyeda, T. Nagao, and M. Kamogawa. Short-term earthquake prediction: Current status of seismo-electromagnetics. *Tectonophysics*, 470(3-4):205–213, 2009.

[77] J. Virieux, H. Calandra, and R.-É. Plessix. A review of the spectral, pseudo-spectral, finite-difference and finite-element modelling techniques for geophysical imaging. *Geophysical Prospecting*, 59(5):794–813, 2011.

[78] J. Virieux and S. Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):WCC1–WCC26, 2009.

[79] T. Warburton and J. S. Hesthaven. On the constants in hp-finite element trace inverse inequalities. *Computer Methods in Applied Mechanics and Engineering*, 192(25):2765–2773, 2003.

[80] L. Zhang, T. Cui, and H. Liu. A set of symmetric quadrature rules on triangles and tetrahedra. *Journal of Computational Mathematics*, pages 89–96, 2009.

[81] E. Zhebel, S. Minisini, A. Kononov, and W. A. Mulder. A comparison of continuous mass-lumped finite elements with finite differences for 3-D wave propagation. *Geophysical Prospecting*, 62(5):1111–1125, 2014.

Acknowledgements

First and foremost I want to thank my supervisor Jaap van der Vegt and my Shell contact person Wim Mulder for all the help they gave me during my research. I am very grateful to Jaap for giving me a lot of academic freedom, for proofreading all my articles and reports in detail, and for discussing my project with me on a weekly basis. I am also very grateful to Wim for all his useful advice and for his substantial contributions to the chapters on mass lumping.

Besides Jaap and Wim, there were several other persons that helped me during my research. I want to thank the secretaries Mariëlle and Linda for helping me with all the administrative tasks during my PhD project, I want to thank my colleagues for all the interesting discussions that we had, and I want to thank the people involved in the hpGEM project for the many things I learned from them. I also like to specifically thank Emile and Kevin for willing to be my paranymphs at my defence.

Finally, I want to thank all my friends and family, housemates, fellow basketball players of DBV Arriba, fellow boxers of ESBV Buitenwesten, and colleagues for the good time in Enschede. This definitely helped me keeping up the spirit.