CHALMERS
UNIVERSITY OF TECHNOLOGY

Localized orthogonal decomposition for the strongly damped wave equation

Master's thesis in Engineering Mathematics and Computational Science

Per Ljung

Department of Mathematical Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

Master's thesis 2018

Localized orthogonal decomposition for the strongly damped wave equation

PER LJUNG

Department of Mathematical Sciences
Chalmers University of Technology
Gothenburg, Sweden 2018

Localized orthogonal decomposition for the strongly damped wave equation
PER LJUNG

Supervisor: Axel Målqvist, Department of Mathematical Sciences
Ass. supervisor: Anna Persson, Department of Mathematical Sciences
Examiner: Stig Larsson, Department of Mathematical Sciences

Master's Thesis 2018
Department of Mathematical Sciences
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by the Department of Mathematical Sciences
Gothenburg, Sweden 2018

Localized orthogonal decomposition for the strongly damped wave equation
PER LJUNG
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

This thesis aims to describe the development of a numerical method for solving a strongly damped wave equation of multiscale type. This multiscale equation is characterized by rapid variations in the data, and may for example arise in the studies of composite materials. For standard finite element methods, a sufficiently refined mesh is required to resolve the variations, implying numerical challenges that might reach the limit of today's technology. Thus, the Localized Orthogonal Decomposition (LOD) method, developed to efficiently solve multiscale type PDEs, is presented. From the LOD framework, a new method based on an orthogonal decomposition with respect to two different multiscale coefficients is derived. With this method, we prove convergence of optimal order for a simplified version of the strongly damped wave equation. The method is then completed by performing both spatial and temporal localization, reducing memory issues as well as improving the efficiency. Furthermore, the implementational considerations, both theoretical as well as practical are discussed. This is followed by numerical examples that illustrate the linear convergence in H^1-norm for the new method, and moreover show similar performance when applying the different localization procedures. Finally, the future research of this method is discussed, including the necessary theory to be developed and techniques that can be utilized to improve the method's efficiency.

Keywords: Strongly damped wave, multiscale equations, finite element methods, Localized Orthogonal Decomposition, convergence of optimal order, localization procedures

Acknowledgements

First of all, I wish to express my gratitude to my supervisors Axel Målqvist and Anna Persson for their fantastic support throughout this entire semester. I would also like to thank my friend Tim Keil for introducing me to this interesting topic, and for always answering my many LOD related questions. Furthermore, my thanks go to Fredrik Hellman for all his help with the implementational aspects. In addition, I want to thank Pontus Larsson and Adam Breitholtz for their great feedback on this thesis. Finally, I wish to thank all the friends I have met during my time at Chalmers, for making these last years truly enjoyable.

Per Ljung, Gothenburg, June 2018

Contents

List of Figures

List of Tables

1
Introduction

The theory of partial differential equations (PDEs) have for a long time been of great importance for mathematics and engineering. Applications range from modelling macroscopic actions such as the fluid flow around a jet's wings, down to particles on an atomic level from a quantum mechanical point of view. For this thesis we consider an initial-boundary value problem that arise in internally damped systems, e.g. in viscoelastic theory, namely

$$\begin{aligned} \ddot{u} - \nabla \cdot (A\nabla\dot{u} + B\nabla u) &= f, \quad \text{in } \Omega \times (0,T], \\ u &= 0, \quad \text{on } \partial\Omega \times (0,T], \end{aligned} \tag{1.1}$$

with initial data $u(\cdot,0) = u_0$ and $\dot{u}(\cdot,0) = v_0$. This equation is often referred to as the *strongly damped wave equation.* The coefficients $A = A(x)$ and $B = B(x)$ represent the system's damping and propagation respectively, $f = f(x,t)$ is the system's source function and $\Omega \subset \mathbb{R}^d$ is a bounded Lipschitz domain.

Along with the evolution of PDE theory, several methods, analytical as well as numerical, have been developed in order to solve the equations and analyse the behaviour of their solutions. Among the numerical methods, most notable is the finite element method (FEM).

FEM is a numerical method based on a discretization of the computational domain. This discretization is represented by a mesh of non-overlapping elements whose shapes are dependent on the domain's dimension (e.g. triangles in 2D, or tetrahedrons in 3D). With this mesh, a Finite Element-space (FE-space) is established with basis functions defined on the elements. By using these basis functions, an approximation of the solution is evaluated on the mesh. In general, FEM works very well for homogeneous media or media that do not vary too much in space. However, when the solutions contain high variations on a microscopic level (which for the PDE in (1.1) may arise from highly varying damping and propagation coefficients), FEM requires a mesh with sufficiently small elements to resolve these variations. This results in a large amount of elements, and thus a large computational complexity.

For this purpose, several finite element-based methods have been developed to deal with these types of problems. They are mainly based on using meshes with high resolution on small subdomains to capture the microscopic features of the solution and incorporate this information into the basis functions of the coarse FE-space. Examples of these methods are the heterogeneous multiscale method (HMM) in [1] and the multiscale finite element method (MsFEM) in [2]. Although they both are applicable, like several other methods, the theoretical foundation of these methods require assumptions like periodicity which limits their usage. There are however finite element-based methods developed to solve multiscale problems efficiently, without the necessity of strong assumptions. An example of this is the *Localized Orthogonal Decomposition method*.

The Localized Orthogonal Decomposition (LOD) method is a numerical method based on the idea of the variational multiscale method presented in [3]. It was first introduced by Målqvist and Peterseim in [4], and has ever since been adapted and analysed for several different equations like Possion's equation in [4] and [5], parabolic equations in [6], etc. The main idea of the method is based on a decomposition of the solution space into a coarse and a fine part. The decomposition is done by defining an interpolation operator that maps functions from an infinite dimensional space into its finite dimensional FE-space. In this way, the interpolant's kernel captures the finescale features that the coarse FE-space misses, and hence defines the finescale space. Subsequently, one may use the finescale space's orthogonal complement with respect to some problem-dependent scalar product as FE-space. This FE-space's basis is computed by solving for finescale corrections that are incorporated into the standard FE-basis, and thus takes the solution's high variations into account. The finescale basis correctors though, unlike the coarse basis functions, have global support. This results in an increased computational complexity, rather than decreasing it as desired. However, it has been proven that a corrector decays exponentially outside its associated node, which in turn justifies the act of simply computing them only on small patches surrounding their nodes. Consequently, this leads to a sparse linear system that can be solved efficiently.

The main focus of this thesis is to apply LOD on the strongly damped wave equation stated in (1.1). Using the standard LOD as a starting point, the goal is to derive a new method that will incorporate the high variations of both the damping as well as the propagation coefficient into the coarse FE-basis. Moreover, the solution to the strongly damped wave equation is time dependent. In particular, in terms of the multiscale coefficients, the damping dominates the solution early in its evolution, while at a steady state the damping term has vanished and solely the propagation impacts the solution. For this purpose, time dependent solution correctors on the microscopic scale is essential for the method.

The thesis is structured as follows: Chapter 2 introduces the mathematical background as well as emphasizes the problems that arise with current methods. Chapter 3 presents the theory behind the standard LOD and demonstrates numerical examples for which cases it is sufficient for solving our equation and not. In Chapter 4, the theory of a new LOD method applied to the strongly damped wave equation is constructed. Chapter 5 is devoted to the implementation of both the standard LOD as well as the new method. Numerical examples illustrating the performance of the new LOD is presented in Chapter 6, and Chapter 7 discusses the conclusions of this thesis.

2
Mathematical background

This chapter is dedicated to presenting the necessary theory that is required for this thesis. It includes studies of weak solutions to the strongly damped wave equation and construction of a finite element setting to numerically find these solutions. The reader is expected to know the basics of functional analysis (see [7]) and the finite element method (see [8]), but is nevertheless introduced to this theory with the strongly damped wave equation in mind. It moreover includes the basic error estimates that characterize the FEM from a convergence point of view. Finally, numerical examples are presented for the standard FEM both with constant coefficients, as well as for rapidly varying coefficients, arguing for the necessity of multiscale methods.

2.1 Weak formulation

Consider the initial-boundary value problem

$$\begin{aligned} \ddot{u} - \nabla \cdot (A\nabla\dot{u} + B\nabla u) &= f, \quad \text{in } \Omega \times (0,T], \\ u &= 0, \quad \text{on } \Gamma \times (0,T], \end{aligned} \tag{2.1}$$

with initial data $u(\cdot,0) = u_0$ and $\dot{u}(\cdot,0) = v_0$, where Ω is a bounded Lipschitz domain in $\mathbb{R}^d, d = 1,2,3$ with polygonal boundary $\Gamma := \partial\Omega$. The source function $f = f(x,t)$ denotes a function in $L^2(\Omega)$, and the coefficients $A = A(x)$ and $B = B(x)$ are independent of time. We furthermore assume a homogeneous boundary condition of Dirichlet type and with sufficient smoothness on A, B, f and Γ, the equation admits a classical solution $u \in \mathcal{C}^2(0,T;\Omega)$. However, as the procedure described in this section involves weakening the regularity of the problem, the solution's regularity assumptions need not be as strict as in its classical formulation.

We wish to apply standard methods to weaken the problem which in (2.1) is written in its strong formulation. For this purpose we introduce the Sobolev space $H_0^1(\Omega)$ to be the space of all functions whose weak partial derivatives of order ≤ 1 belong to $L^2(\Omega)$ and that vanishes on the boundary Γ, i.e.

$$H_0^1(\Omega) := \left\{ u \in L^2(\Omega) : Du \in L^2(\Omega),\ u = 0 \text{ on } \Gamma \right\}.$$

This space is equipped with the inner product

$$(v,w)_{H_0^1(\Omega)} = \int_\Omega vw + DvDw \, \mathrm{d}x$$

and with corresponding norm

$$\|u\|_{H_0^1(\Omega)} = \|u\|_{L^2(\Omega)} + \|Du\|_{L^2(\Omega)}.$$

Furthermore, denote by $L_p(0,T;\mathcal{B})$ the Bochner space with norm

$$\|u\|_{L_p(0,T;\mathcal{B})} = \left(\int_0^T \|u\|_{\mathcal{B}}^p \,\mathrm{d}t\right)^{1/p}, \quad 1 \leq p < \infty,$$

where $\mathcal{B}$ is a Banach space with norm $\|\cdot\|_{\mathcal{B}}$. For convenience, the L^2-norm is abbreviated $\|\cdot\| := \|\cdot\|_{L^2(\Omega)}$ and the Bochner norm $\|\cdot\|_{L^2(0,T;\mathcal{B})} := \|\cdot\|_{L^2(\mathcal{B})}$ throughout the thesis. Consider $H_0^1(\Omega)$ as test space, i.e. let $V := H_0^1(\Omega)$, with corresponding dual space denoted $V^* := H^{-1}(\Omega)$. By standard procedure, (2.1) is multiplied by a test function $v \in V$ and integrated over the domain Ω. Using Green's formula and applying the homogeneous boundary condition, the resulting weak formulation is to find

$$u \in L^2(0,T;H_0^1(\Omega)) \cap H^1(0,T;L^2(\Omega)) \cap H^2(0,T;H^{-1}(\Omega))$$

such that

$$(\ddot{u},v) + a(\dot{u},v) + b(u,v) = (f,v), \quad \forall v \in V, \tag{2.2}$$

where the bilinear forms are defined as

$$a(u,v) := \int_\Omega A\nabla u \cdot \nabla v \,\mathrm{d}x \quad \text{and} \quad b(u,v) := \int_\Omega B\nabla u \cdot \nabla v \,\mathrm{d}x,$$

and $(\cdot,\cdot)$ denotes the usual L^2-scalar product. Note that the solution u now only requires to be weakly differentiable once in spatial sense due to the weakening of the problem. The solution of (2.2) is referred to as the *weak solution*. As the name implies, this solution is weaker than the classical solution in the sense of regularity, but it should be noted that if the problem is smooth enough such that the weak solution achieves spatial $\mathcal{C}^2$-regularity, it is in fact the same as the classical solution. A common discussion of weak solutions is their properties of existence and uniqueness. In many cases, such as for PDEs like the Poission equation, the Lax-Milgram theorem yields satisfying results. For parabolic equations, we find techniques such as semigroup approaches and viscosity techniques [9]. For our equation, we present a discussion regarding existence and uniqueness by construction using Galerkin approximations, a technique commonly used for both parabolic as well as hyperbolic type equations.

2.2 Existence and uniqueness

We wish to show the existence and uniqueness of a weak solution to

$$\begin{aligned} \ddot{u} - \nabla \cdot (A\nabla\dot{u} + B\nabla u) = f,&\ \text{in } \Omega \times (0,T], \\ u = 0,&\ \text{on } \Gamma \times (0,T], \\ u = g,\ \dot{u} = q,&\ \text{on } \Omega \times \{t=0\}, \end{aligned} \tag{2.3}$$

by initially solving a finite dimensional approximation. We assume B to be uniformly elliptic with upper bound β and lower bound α, i.e.

$$0 < \alpha := \operatorname*{ess\,inf}_{x\in\Omega} \inf_{v\in\mathbb{R}^d\backslash\{0\}} \frac{Bv\cdot v}{v\cdot v},$$

$$\infty > \beta := \operatorname*{ess\,sup}_{x\in\Omega} \sup_{v\in\mathbb{R}^d\backslash\{0\}} \frac{Bv\cdot v}{v\cdot v}.$$

Furthermore, we restrict the coefficient $A = cB$ to be the coefficient B multiplied by a constant c. We may then select smooth functions $w_k = w_k(x)$ to be the eigenfunctions satisfying

$$b(w_k, v) = \lambda_k(w_k, v)$$

for $k = 1, 2, ...,$ such that the set $\{w_k\}_{k=1}^{\infty}$ is an orthogonal basis of $H_0^1(\Omega)$ and an orthonormal basis of $L^2(\Omega)$. Fix $m \in \mathbb{N}$ and write

$$u_m(t) := \sum_{k=1}^{m} d_m^k(t) w_k, \tag{2.4}$$

where $d_m^k(t)$ satisfy

$$d_m^k(0) = (g, w_k), \tag{2.5}$$
$$d_m^{k\prime}(0) = (q, w_k), \tag{2.6}$$

for $k = 1, 2, ..., m$, and

$$(u'', w_k) + a(u', w_k) + b(u, w_k) = (f, w_k) \tag{2.7}$$

for $t \in [0, T]$ and $k = 1, 2, ..., m$. Using this setting, we may construct an approximate solution to the weak problem, and derive necessary energy estimates for the solution and its derivatives.

Theorem 2.2.1 (Construction of approximate solution)**.** *For each $m \in \mathbb{N}$, there exists a unique function u_m of the form* (2.4) *satisfying* (2.5)-(2.7)*.*

Proof. Since $\{w_k\}_{k=1}^{\infty}$ is an orthonormal basis of $L^2(\Omega)$, we note that

$$(u_m'', w_k) = \left(\sum_{l=1}^{m} d_m^{l\prime\prime}(t) w_l, w_k\right) = \sum_{l=1}^{m} d_m^{l\prime\prime}(t)(w_l, w_k) = d_m^{k\prime\prime}(t).$$

Furthermore, we note that

$$a(u_m', w_k) = \sum_{l=1}^{m} a(w_l, w_k) d_m^{l\prime}(t) = \sum_{l=1}^{m} c\lambda_l(w_l, w_k) d_m^{l\prime}(t) = c\lambda_k d_m^{k\prime}(t),$$
$$b(u_m, w_k) = \sum_{l=1}^{m} b(w_l, w_k) d_m^{l}(t) = \sum_{l=1}^{m} \lambda_l(w_l, w_k) d_m^{l}(t) = \lambda_k d_m^{k}(t),$$

and denote $f^k(t) = (f(t), w_k)$ for $k = 1, 2, ..., m$. Consequently, (2.7) can be rewritten as a linear system

$$d_m^{k\prime\prime}(t) + c\lambda_k d_m^{k\prime}(t) + \lambda_k d_m^k(t) = f^k(t) \tag{2.8}$$

for $t \in [0, T]$ and $k = 1, 2, ..., m$, with the given initial conditions (2.5) and (2.6). By standard ODE theory, there exists a unique $\mathcal{C}^2$-function

$$\mathbf{d}_m(t) = (d_m^1(t), ..., d_m^m(t))$$

satisfying the initial conditions and solves the ODE (2.8) for $t \in [0, T]$.

□

Theorem 2.2.2 (Energy estimates)**.** *There exists a constant C such that*

$$\max_{t\in[0,T]}\left(\|u_m(t)\|_{H_0^1}\right)+\|u_m'(t)\|_{L^2(H_0^1)}+\|u_m''(t)\|_{L^2(H^{-1})}\leq C(\|f\|_{L^2(H^{-1})}+\|g\|_{H_0^1}+\|q\|),$$

for $m=1,2,...$

Proof. Begin by multiplying (2.7) by $d_m^{k\prime}(t)$ and sum over $k=1,2,...,m$ to get

$$(u_m'',u_m')+a(u_m',u_m')+b(u_m,u_m')=(f,u_m')$$

for a.e. $t\in[0,T]$. Recall the identities

$$(u_m'',u_m')=\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u_m'\|^2,\qquad b(u_m,u_m')=\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}b(u_m,u_m).$$

Using this, we see that

$$\begin{aligned}\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\|u_m'\|^2+b(u_m,u_m)\right)+c\alpha\|u_m'\|_{H_0^1}^2&\leq\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\|u_m'\|^2+b(u_m,u_m)\right)+a(u_m',u_m')\\&=(f,u_m')\\&\leq\|f\|_{H^{-1}}\|u_m'\|_{H_0^1}\\&\leq C\|f\|_{H^{-1}}^2+\frac{c\alpha}{2}\|u_m'\|_{H_0^1}^2\end{aligned}$$

where we have used Young's inequality with weighted coefficients. The inequality can further be rewritten as

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\|u_m'\|^2+b(u_m,u_m)\right)+\frac{c\alpha}{2}\|u_m'\|_{H_0^1}^2\leq C\|f\|_{H^{-1}}^2.$$

Integrate both sides with respect to time, and the inequality becomes

$$\|u_m'\|^2+b(u_m,u_m)+\|u_m'\|_{L^2(H_0^1)}^2\leq C(\|f\|_{L^2(H^{-1})}^2+\|q\|^2+\|g\|_{H_0^1}^2).$$

Since this was done for arbitrary $t\in[0,T]$, we get

$$\max_{t\in[0,T]}\left(\|u_m\|_{H_0^1}^2\right)+\|u_m'\|_{L^2(H_0^1)}^2\leq C(\|g\|_{H_0^1}^2+\|q\|^2+\|f\|_{L^2(H^{-1})}^2).$$

Next, fix $v\in H_0^1(\Omega)$ with $\|v\|_{H_0^1}\leq1$, and decompose $v=v_1+v_2$ where $v_1\in\mathrm{Span}(w_k)$ and $(v_2,w_k)=0$ for $k=1,2,...,m$. Then we have

$$(u_m'',v)=(u_m'',v_1)=(f,v_1)-a(u_m',v_1)-b(u_m,v_1)$$

and furthermore

$$|(u_m'',v)|\leq C(\|f\|_{H^{-1}}+\|u_m'\|_{H_0^1}+\|u_m\|_{H_0^1})$$

since $\|v_1\|_{H_0^1}\leq1$. Recall the $H^{-1}(\Omega)$-norm

$$\|u\|_{H^{-1}}=\sup\{|(u,v)|:v\in H_0^1(\Omega),\|v\|_{H_0^1}\leq1\}.$$

In this norm, we see that

$$\begin{aligned}\|u_m''\|_{H^{-1}}^2 &\leq C(\|f\|_{H^{-1}} + \|u_m'\|_{H_0^1} + \|u_m\|_{H_0^1})^2\\ &\leq C(2(\|f\|_{H^{-1}} + \|u_m'\|_{H_0^1})^2 + 2\|u_m\|_{H_0^1}^2)\\ &\leq C(4\|f\|_{H^{-1}}^2 + 4\|u_m'\|_{H_0^1}^2 + 2\|u_m\|_{H_0^1}^2)\\ &\leq C(\|f\|_{H^{-1}}^2 + \|u_m'\|_{H_0^1}^2 + \|u_m\|_{H_0^1}^2).\end{aligned}$$

Consequently, we arrive at

$$\begin{aligned}\int_0^T \|u_m''\|_{H^{-1}}^2 \,\mathrm{d}t &\leq C\int_0^T \|f\|_{H^{-1}}^2 + \|u_m\|_{H_0^1}^2 + \|u_m'\|_{H_0^1}^2 \,\mathrm{d}t\\ &\leq C(\|g\|_{H_0^1}^2 + \|q\|^2 + \|f\|_{L^2(H^{-1})}^2).\end{aligned}$$

□

Combining the approximate solution with the derived energy estimates, we may now move on to showing the existence of a weak solution.

Theorem 2.2.3. *There exists a weak solution to the problem* (2.3).

Proof. According to the energy estimates in Theorem 2.2.2, we have that $\{u_m\}_{m=1}^\infty$ and $\{u_m'\}_{m=1}^\infty$ are bounded in $L^2(H_0^1(\Omega))$, and $\{u_m''\}_{m=1}^\infty$ is bounded in $L^2(H^{-1}(\Omega))$. Consequently, there exists a subsequence $\{u_{m_l}\}_{l=1}^\infty \subset \{u_m\}_{m=1}^\infty$ and $u \in L^2(H_0^1(\Omega))$, $u' \in L^2(H_0^1(\Omega))$, and $u'' \in L^2(H^{-1}(\Omega))$ such that

$$\begin{aligned}u_{m_l} &\to u \quad \text{weakly in } L^2(H_0^1(\Omega)),\\ u_{m_l}' &\to u' \quad \text{weakly in } L^2(H_0^1(\Omega)),\\ u_{m_l}'' &\to u'' \quad \text{weakly in } L^2(H^{-1}(\Omega)).\end{aligned}$$

Now fix an integer N and choose a function $v \in \mathcal{C}^1([0,T];H_0^1(\Omega))$ of the form

$$v(t) = \sum_{k=1}^N d^k(t)w_k, \tag{2.9}$$

where $\{d^k\}_{k=1}^N$ are smooth functions. Select $m \geq N$, multiply (2.7) by $d^k(t)$, sum over k and integrate with respect to t to get

$$\int_0^T (u_m'', v) + a(u_m', v) + b(u_m, v)\,\mathrm{d}t = \int_0^T (f, v)\,\mathrm{d}t. \tag{2.10}$$

Set $m = m_l$ to find in the limit that

$$\int_0^T (u'', v) + a(u', v) + b(u, v)\,\mathrm{d}t = \int_0^T (f, v)\,\mathrm{d}t.$$

This holds for all functions $v \in L^2(H_0^1(\Omega))$ since the functions (2.9) are dense in this space. From this, it follows that

$$(u'', v) + a(u', v) + b(u, v) = (f, v), \quad \forall v \in H_0^1(\Omega),$$

and a.e. $t \in [0,T]$. It remains to verify that u satisfies the initial conditions, i.e.

$$\begin{aligned} u(0) &= g, \\ u'(0) &= q. \end{aligned}$$

For this purpose, we choose a function $v \in \mathcal{C}^2(0,T;H_0^1(\Omega))$ with $v(T) = v'(T) = 0$. Integration by parts twice with respect to t gives

$$\int_0^T (v'',u) + a(u',v) + b(u,v)\,\mathrm{d}t = \int_0^T (f,v)\,\mathrm{d}t - (u(0),v'(0)) + (u'(0),v(0)).$$

Similarly, from (2.10) we get

$$\int_0^T (v'',u_m) + a(u_m',v) + b(u_m,v)\,\mathrm{d}t = \int_0^T (f,v)\,\mathrm{d}t - (u_m(0),v'(0)) + (u_m'(0),v(0)).$$

Now set $m = m_l$ and go to limit to find that

$$\int_0^T (v'',u) + a(u',v) + b(u,v)\,\mathrm{d}t = \int_0^T (f,v)\,\mathrm{d}t - (g,v'(0)) + (q,v(0)).$$

Compare these given expressions to deduce that u satisfies the initial conditions as $v(0)$ and $v'(0)$ are arbitrary. Thus, u is a weak solution to our original equation. □

It remains to prove the uniqueness, for which purpose we recall Gronwall's inequality on differential form.

Lemma 2.2.4 (Gronwall's inequality, see [10])**.** *Let $\eta(\cdot)$ be a non-negative, absolutely continuous function on $[0,T]$, which satisfies for a.e. t the differential inequality*

$$\eta'(t) \le \phi(t)\eta(t) + \xi(t),$$

where $\phi(t)$ and $\xi(t)$ are non-negative, summable functions on $[0,T]$. Then

$$\eta(t) \le e^{\int_0^t \phi(s)\,\mathrm{d}s}\left[\eta(0) + \int_0^t \xi(s)\,\mathrm{d}s\right]$$

for all $t \in [0,T]$. In particular, if $\eta(0) = 0$ and $\eta \le \phi\eta$ on $[0,T]$, then

$$\eta \equiv 0$$

on $[0,T]$.

We are now set to prove the uniqueness of the weak solution.

Theorem 2.2.5. *The weak solution to (2.3) is unique.*

Proof. It suffices to show that the only weak solution to the original equation with $f \equiv g \equiv q \equiv 0$ is

$$u \equiv 0.$$

To verify this, recall that

$$(u'',u') + b(u,u') = \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}(\|u'\| + b(u,u)) = -a(u',u') \le \|u'\| + b(u,u)$$

and define

$$\eta(t) := \|u'\| + b(u,u).$$

Thus we have an inequality of the form

$$\eta'(t) \leq C\eta(t),$$

and a special case of Gronwall's inequality then gives that

$$\eta(t) \equiv 0.$$

As both terms of $\eta(t)$ are non-negative, this finally results in

$$u \equiv 0 \quad \text{and} \quad u' \equiv 0.$$

□

The remainder of this thesis focuses on developing a numerical method whose purpose is to approximate a solution to (2.2). For this cause, the introduction of the finite element method is of necessity.

2.3 The Finite Element Method

The finite element method is a numerical method that has been developed specifically to approximate solutions to problems on the form (2.2). It is a Galerkin method that defines two separate spaces referred to as the *trial space*, to which u belongs, and a *test space*, to which the test functions v belong (in previous section $H_0^1(\Omega)$). The FEM has been implemented and analysed for most PDEs, including the strongly damped wave equation (see [11]). For PDEs dependent on only spatial variables, like Poisson's equation, it is enough to simply apply a discretization of the spatial domain. However, in our case we consider an equation that evolves with time. Therefore, the notions of *semi-discrete* and *completely discrete* schemes are used.

2.3.1 Spatial discretization

Primarily, let $\mathcal{T}_H$ be a family of coarse, shape regular elements, that forms a partition of the domain Ω. That is,

$$\bigcup_{T\in\mathcal{T}_H} T = \bar{\Omega}$$

and the intersection of two different elements $T_i, T_j \in \mathcal{T}_H$ is either empty, a node, or a common edge. Furthermore, no node is located in the interior of an edge in $\mathcal{T}_H$. This mesh can for instance be represented by, but is not limited to, intervals in one dimension, triangles in two dimensions, and tetrahedrons in three dimensions. For an element $T \in \mathcal{T}_H$, let $H_T := \operatorname{diam}(T)$, and denote the largest diameter of any element in the mesh by $H := \max_{T\in\mathcal{T}_H} \operatorname{diam}(T)$. Furthermore, let $\gamma > 0$ denote the shape regularity parameter of the mesh, defined as

$$\gamma := \max_{T\in\mathcal{T}_H} \gamma_T, \quad \text{where} \quad \gamma_T = \frac{\operatorname{diam}(B_T)}{\operatorname{diam}(T)},$$

where B_T denotes the largest ball inside T. With the mesh and its properties given, we can define an associated finite element space. When using a mesh of triangular elements, we consider the space

$$\mathcal{P}_m(\mathcal{T}_H) := \{v \in \mathcal{C}(\overline{\Omega}) : v|_T \text{ is a polynomial of total degree} \leq m, \forall T \in \mathcal{T}_H\},$$

and in the case of a quadrilateral mesh, we use

$$\mathcal{Q}_m(\mathcal{T}_H) := \{v \in \mathcal{C}(\overline{\Omega}) : v|_T \text{ is a polynomial of partial degree} \leq m, \forall T \in \mathcal{T}_H\},$$

where both spaces consist of piecewise polynomial functions continuous on $\overline{\Omega}$. With this consideration, we set the FE-space to be

$$V_H := V \cap \mathcal{Q}_1(\mathcal{T}_H).$$

Let $\mathcal{N}_H$ denote the set consisting of all interior nodes of the mesh $\mathcal{T}_H$, i.e. $x \cap \Gamma = \emptyset$ for every $x \in \mathcal{N}_H$, and denote its size by $|\mathcal{N}_H| = N_H$. As the FE-space V_H is of finite dimension, there exists a basis $\{\lambda_x\}_{x \in \mathcal{N}_H} \subset \mathcal{Q}_1(\mathcal{T}_H)$ with the property that for two nodes $x, y \in \mathcal{N}_H$

$$\lambda_x(y) = \begin{cases} 1, & \text{if } x = y, \\ 0, & \text{if } x \neq y, \end{cases}$$

and moreover, $V_H = \text{Span}(\{\lambda_x\}_{x \in \mathcal{N}_H})$. Due to the appearance of the basis functions, they are often referred to as *hat functions*, or *tent functions*. An important fact about these basis functions is their compact support, as they vanish outside of a vertex patch surrounding corresponding node. Given the FE-space V_H, we may now define the standard FEM approximation from a semi-discrete point of view.

Definition 2.3.1 (Standard semi-discrete FEM approximation)**.** The semi-discrete Galerkin FEM approximation is to find $u_H \in V_H \subset V$ such that

$$(\ddot{u}_H, v) + a(\dot{u}_H, v) + b(u_H, v) = (f, v), \quad \forall v \in V_H, \tag{2.11}$$

with $u_H(0) = u_{0,H}$ and $\dot{u}_H(0) = v_{0,H}$ where $u_{0,H}$ and $v_{0,H}$ are approximations of u_0 and v_0 onto V_H.

Note that for the standard FEM, the trial and test spaces are chosen to be equal. In general, this must not be the case. One can vary the choices of these spaces in order to experiment with different problems, known as Petrov-Galerkin formulations. With the spatial domain discretized, it remains to deal with the temporal discretization.

2.3.2 Temporal discretization

Similarly as done for the spatial part, a partition of the time domain $[0, T]$ is established. This is done using the uniform discretization

$$0 = t_0 < t_1 < ... < t_N = T,$$

where we denote the time step between two points by $\tau = t_n - t_{n-1}$. Let U^n be the approximation of $u(t)$ at time $t = t_n$, and denote $f^n := f(t_n)$. Furthermore, we use the *backward Euler-Galerkin method* and define the discrete time derivative $\bar{\partial}_t$ as

$$\bar{\partial}_t U^n := \frac{U^n - U^{n-1}}{\tau}.$$

By plugging these expressions into the weak formulation, we are given the completely discrete scheme for the equation.

Definition 2.3.2 (Backward Euler-Galerkin FEM approximation)**.** The completely discrete backward Euler-Galerkin FEM is to find $U^n \in V_H$ such that

$$(\bar{\partial}_t \bar{\partial}_t U^n, v) + a(\bar{\partial}_t U^n, v) + b(U^n, v) = (f^n, v), \quad \forall v \in V_H,$$

for $n = 1, 2, ..., N$ with $U^0 \in V_H$ is some approximation of u_0, and $U^1 \in V_H$ of $u(t_1)$.

The approximations of the initial values can be chosen in different ways. For example, one could choose $U^0 = P_H u_0$ where P_H is the L^2-projection onto V_H. With the completely discrete scheme defined, one can now construct the fully discrete formulation of the finite element method.

2.3.3 Fully discrete formulation

Recall that the FE-space V_H is spanned by the set of basis functions $\{\lambda_x\}_{x \in \mathcal{N}_H}$. Since $U^n \in V_H$, we can write it as a linear combination of these basis functions, i.e.

$$U^n = \sum_{x \in \mathcal{N}_H} \alpha_x^n \lambda_x.$$

Plug this expression into the backward Euler-Galerkin FEM, and we acquire the following numerical equation

$$\Big(\mathcal{M} + \tau\mathcal{K} + \tau^2\mathcal{S}\Big) \cdot \mathcal{U}_H^n = \tau^2\mathcal{F} + \tau\mathcal{K} \cdot \mathcal{U}_H^{n-1} + \mathcal{M} \cdot \mathcal{U}_H^{n-1} + \tau\mathcal{M} \cdot \mathcal{V}_H^{n-1},$$

where $\mathcal{U}_H^n$ is a vector of size N_H consisting of the coefficients α_x^n. The entries of the stiffness matrices $\mathcal{S}$ and $\mathcal{K}$, the mass matrix $\mathcal{M}$ and the load vector $\mathcal{F}$ are determined by

$$\begin{aligned} \mathcal{S}_{x,y} &:= b(\lambda_x, \lambda_y), \\ \mathcal{K}_{x,y} &:= a(\lambda_x, \lambda_y), \\ \mathcal{M}_{x,y} &:= (\lambda_x, \lambda_y), \\ \mathcal{F}_x &:= (f^n, \lambda_x), \end{aligned}$$

and x and y varies over all the nodes in $\mathcal{N}_H$. Thanks to the small support of the basis functions, the resulting matrices are sparse, and are thus easy to compute. This is the great advantage of the finite element method, as it results in a sparse linear system that can be solved easily and efficiently. Also note that the size of the system is dependent on the size of the set $\mathcal{N}_H$. Thus, it implies that the size of the system increases with decreasing element size H, as this results in a larger amount of elements in $\mathcal{T}_H$.

2.4 Error bounds

The finite element method for the strongly damped wave equation have earlier been studied in terms of error analysis. In [11], Larsson, Thomée and Wahlbin analyse the equation in the homogeneous case where the coefficients A and B coincide with respect to some constant. The inhomogeneous case is presented and analysed as a viscoelasticity type equation by Lin, Thomée and Wahlbin in [12], where estimates on the space discretization are established. For these estimates we require the FE-space $V_H \subset H_0^1$ to be the subset as presented in previous sections, with the property to uniformly in H satisfy

$$\min_{\chi \in V_H} \{\|v - \chi\| + H\|v - \chi\|_1\} \leq CH^2 \|v\|_{H^2},$$

for all $v \in H_0^1(\Omega) \cap H^2(\Omega)$. In order to estimate the error for the semi-discrete solution $u_H(t)$ to our strongly damped wave equation, we begin by considering the following lemma.

Lemma 2.4.1 (Lin, Thomée and Wahlbin in [12]). *Let u be the solution to the initial boundary value problem*

$$\begin{aligned} -\nabla \cdot (A\nabla u_t + B\nabla u) &= f, \quad \textit{in } \Omega \times (0, T], \\ u &= 0, \quad \textit{on } \Gamma \times (0, T], \end{aligned}$$

with initial condition $u(\cdot, 0) = v$, and with u smooth enough, and let u_H be the solution to its semi-discrete analogue, namely

$$a(u_{H,t}, \chi) + b(u_H, \chi) = (f, \chi),$$

for all $\chi \in V_H$ and $t \in (0, T]$ with initial condition $u_H(0) = v_H$. Here, v_H denotes an approximation of v in V_H. Let $p \in [2, \infty)$ and assume that

$$|v_H - v|_p + H\|v_H - v\|_{1,p} \leq CH^2.$$

holds. Then, for any non-negative integer k we have

$$|D_t^k (u_H(t) - u(t))|_p \leq C_{p,k} H^2.$$

Proof. See Theorem 5.1 in [12]. □

With this lemma, one may now prove the following result for the error of the spatial discretization. The proof is a detailed version of the one given in [12].

Theorem 2.4.2 (Lin, Thomée and Wahlbin in [12]). *Let u_H and u be the solutions of* (2.11) *and* (2.1) *respectively, with initial conditions $u(0) = v$ and $\dot{u}(0) = w$, and u smooth enough. Assume that*

$$\begin{aligned} \|v_H - v\| + H\|v_H - v\|_1 &\leq CH^2, \\ \|w_H - w\| &\leq CH^2, \end{aligned} \tag{2.12}$$

holds true. Then

$$\|u_H(t) - u(t)\| \leq CH^2$$

for $t \in [0, T]$.

Proof. First, we define the Ritz-Volterra projection $W_H : ([0,T]; H_0^1(\Omega)) \to ([0,T]; V_H)$ such that $W_H u$ solves

$$a((W_H u)_t, \chi) + b(W_H u, \chi) = a(u_t, \chi) + b(u, \chi), \quad \forall \chi \in V_H,$$

with $W_H u(0) = v_h$, and write the error as

$$e = u_H - u = (u_H - W_H u) + (W_H u - u) = \theta + \rho.$$

From Lemma 2.4.1 we have for each non-negative k

$$\|D_t^k \rho(t)\| \leq CH^2 \tag{2.13}$$

for $t \in [0,T]$, so it remains to estimate $\theta(t)$. By using the weak formulation along with the definition of the Ritz-Volterra projection, we observe that

$$\begin{aligned}
(\theta_{tt}, \chi) + a(\theta_t, \chi) + b(\theta, \chi) &= (u_{tt,H}, \chi) + a(u_{t,H}, \chi) + b(u_H, \chi) \\
&\qquad - ((W_H u)_{tt}, \chi) - a((W_H u)_t, \chi) - b(W_H u, \chi) \\
&= (f, \chi) - a((W_H u)_t, \chi) - b(W_H u, \chi) - ((W_H u)_{tt}, \chi) \\
&= (u_{tt}, \chi) - ((W_H u)_{tt}, \chi) \\
&= -(\rho_{tt}, \chi)
\end{aligned}$$

for all $\chi \in V_H$. Recall the inequality

$$\|u\|\|v\| \leq \frac{1}{2}(\|u\|^2 + \|v\|^2)$$

and set $\chi = \theta_t$. Since $\theta(0) = v_H - W_H u(0) = 0$, we find that

$$\begin{aligned}
\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\theta_t\|^2 + \alpha\|\theta_t\|_1^2 &\leq (\theta_{tt}, \theta_t) + a(\theta_t, \theta_t) \\
&\leq -(\rho_{tt}, \theta_t) - b(\theta, \theta_t) \\
&\leq \|\rho_{tt}\|\|\theta_t\| + C\|\theta\|_1\|\theta_t\|_1 \\
&\leq \|\rho_{tt}\|^2 + C_1\|\theta_t\|^2 + C_2\|\theta\|_1^2 + \frac{\alpha}{2}\|\theta_t\|_1^2 \\
&\leq \|\rho_{tt}\|^2 + \frac{\alpha}{2}\|\theta_t\|_1^2 + C'\left\{\|\theta_t\|^2 + \int_0^t \|\theta_t\|_1^2 \,\mathrm{d}\tau\right\}.
\end{aligned}$$

Now define

$$\begin{aligned}
\eta(t) &:= \|\theta_t\|^2 + \int_0^t \|\theta_t\|_1^2 \,\mathrm{d}\tau, \\
\xi(t) &:= \|\rho_{tt}\|^2.
\end{aligned}$$

Then the inequality can be rewritten on the form

$$\eta'(t) \leq C_1\eta(t) + C_2\xi(t),$$

for appropriate choices of constants C_1 and C_2. Gronwall's inequality, in combination with (2.13) yields the inequality

$$\|\theta_t\|^2 + \int_0^t \|\theta_t\|_1^2 \,\mathrm{d}\tau \leq C\|\theta_t(0)\|^2 + CH^4.$$

Furthermore, using (2.12) and (2.13) we observe that

$$\|\theta_t(0)\| \leq \|(W_H u)_t(0) - w\| + \|w_H - w\| \leq CH^2.$$

Thus, we conclude that

$$\|\theta_t\|^2 + \int_0^t \|\theta_t\|_1^2 \, \mathrm{d}\tau \leq CH^4$$

and after integration with respect to time, we arrive at the estimation

$$\|\theta(t)\| \leq CH^2.$$

□

2.5 Numerical examples

The finite element method is a well-known method that efficiently approximates smooth PDEs with great results. We conclude this chapter by illustrating some numerical examples to show the method's performance when evaluating solutions to our strongly damped wave equation. First, an example where the coefficients do not oscillate is presented to confirm the theory introduced in this chapter. Furthermore, a multiscale example where the coefficients vary with a high frequency is considered, emphasizing the necessity of multiscale methods.

2.5.1 Non-oscillating coefficients

Consider the strongly damped wave equation in one dimension with coefficients $A(x) = B(x) = 1$ and source term $f = 2(t+1)$, i.e.

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}u(x,t) - \left(\frac{\mathrm{d}^3}{\mathrm{d}x^2\mathrm{d}t}u(x,t) + \frac{\mathrm{d}^2}{\mathrm{d}x^2}u(x,t)\right) = 2(t+1), \quad \text{in } \Omega \times (0,T]$$

with homogeneous Dirichlet boundary condition and initial values $u_0 = 0$ and $v_0 = x(1-x)$. This equation has the analytical solution $u(x,t) = tx(1-x)$. For the standard FEM approximation of this solution, a uniform time step $\tau = 0.01$ is used, and the number of time steps is $N = 100$, such that $T = 1$. The approximate solutions for different values on $1/H$ can be seen in Figure 2.1 and the L^2-errors between the approximation and the analytical solution in Figure 2.2. It is seen in these figures that the standard FEM yields satisfying results in terms of error convergence, and that the approximation is well aligned with the exact solution already at $1/H = 16$. With these results in mind, subsequent section introduces similar calculations, but when the coefficients contain high variations.

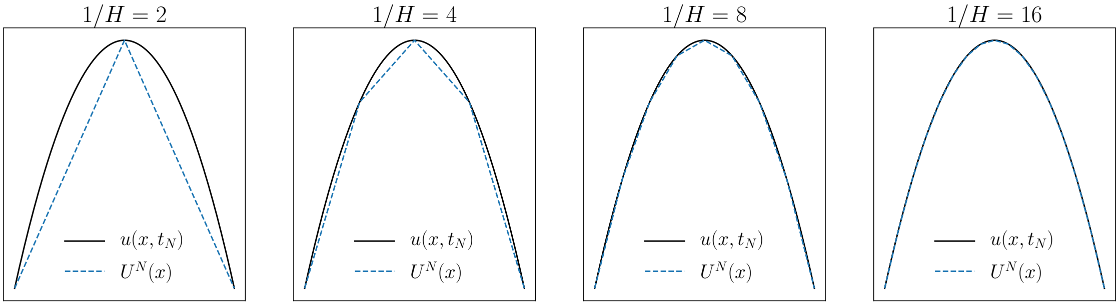


Figure 2.1: The analytical solution (black) plotted with its FEM approximation (blue).

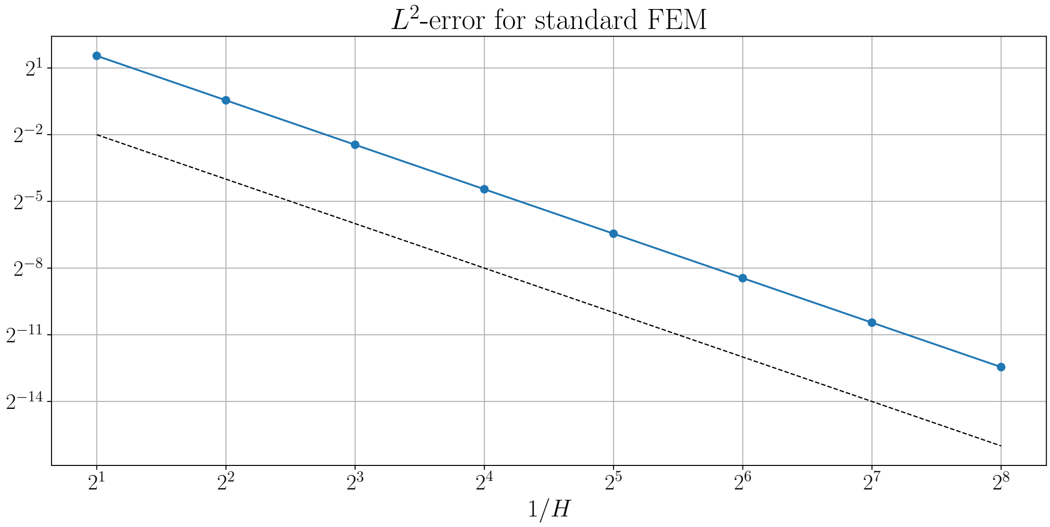


Figure 2.2: L^2-error for standard FEM with non-oscillating coefficients (blue line). The black line indicates the error convergence $\mathcal{O}(H^2)$.

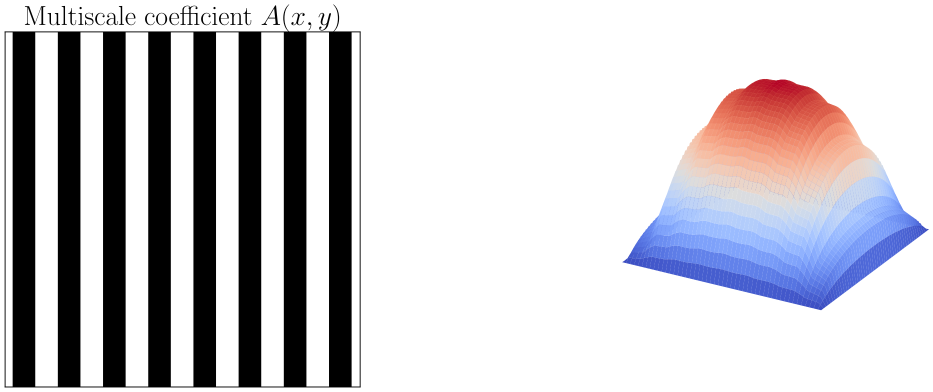
2.5.2 Multiscale coefficients

Earlier it has been seen how the standard FEM works well for the strongly damped wave equation as long as the coefficients vary at a sufficiently low frequency. This section is dedicated to presenting a multiscale example where the FEM lacks in performance, emphasizing the need for multiscale methods. The example considered does not necessarily admit an analytical solution. However, for the sake of comparing with a reference, a finescale solution has been evaluated with standard FEM on a sufficiently refined mesh. The example is an adaptation made for the strongly damped wave equation based on an example from [13].

Consider the equation

$$\begin{aligned} \ddot{u} - \nabla \cdot (A\nabla \dot{u} + B\nabla u) &= 1, \text{ in } \Omega \times (0, T], \\ u &= 0, \text{ on } \Gamma \times (0, T], \end{aligned}$$

with initial values $u_0 = 0$ and $v_0 = 1$. The domain is set to the unit square, i.e. $\Omega = [0,1] \times [0,1]$, the propagation coefficient $B(x,y) = 1$, and the damping coefficient $A(x,y)$ is seen in Figure 2.3(a). For the time discretization, the time step $\tau = 0.1$ is used, and the number of time steps is $N = 2$ such that the time domain is $[0, 0.2]$.



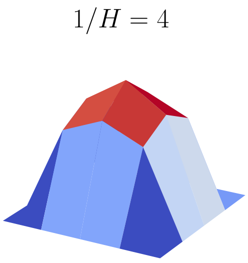
(a) Coefficient $A(x,y)$.

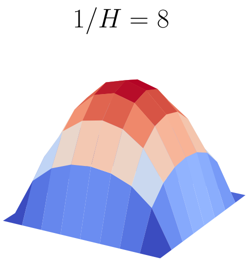
(b) Finescale solution $u(x,y,t_N)$.

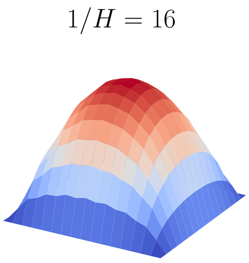
Figure 2.3: The multiscale damping coefficient and finescale solution evaluated on a fine mesh with $1/H = 1024$. Black is 1 and white is 0.05.

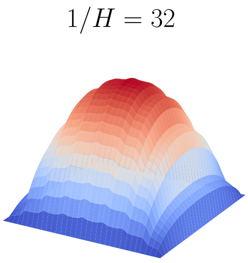
In Figure 2.3(b) one can see the finescale solution to the problem, evaluated on a fine mesh with $1/H = 1024$. As the solution to the strongly damped wave equation is much affected by the damping at an early state, the focus for this example lies in the damping coefficient, which is characterized by several channels. The coefficient's behaviour can be seen reflected in the finescale solution as different layers which its surface follows. The approximate solutions depending on varying mesh sizes can be seen in Figure 2.4. It is seen that, although the damping does not vary at a specifically high frequency, its behaviour is still missing in the solution for small values on $1/H$. It is not until $H = 2^{-5}$ that the layers appear in the numerical solution, reflecting the damping coefficient's form. This is further recognized by Figure 2.5, where the L^2-error between the exact and the approximate solution is evaluated as H gets smaller. In this plot one sees that the error lies at an almost constant level, and once $1/H = 2^4$ we start distinguishing the desired convergence rate.

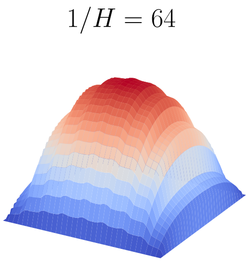
This example affirms that when the solution is highly varying on a microscopic level, the FEM is not a viable method as long as the element sizes are larger than the oscillation's frequency, i.e. when $H \geq \varepsilon_{A,B}$. The expected rate of convergence does not start until the mesh can resolve the variations from the coefficients. This comes from the fact that the error convergence for the standard FEM (based on piecewise continuous polynomials) relies on (at least) spatial H^2-regularity. In particular, using the FE-space $V_H = V \cap \mathcal{Q}_1(\mathcal{T}_H)$, the error depends on $\|u\|_{H^2} \sim \varepsilon^{-1}$, where ε indicates the coefficients' frequency of oscillation. Thus we require an element size $H \ll \varepsilon$ for a satisfying approximation, which is not computationally reasonable.











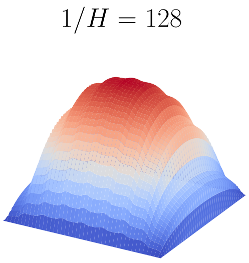


Figure 2.4: Approximate solutions $U^N(x, y)$ of the exact solution $u(x, y, t_N)$ depending on the mesh width.

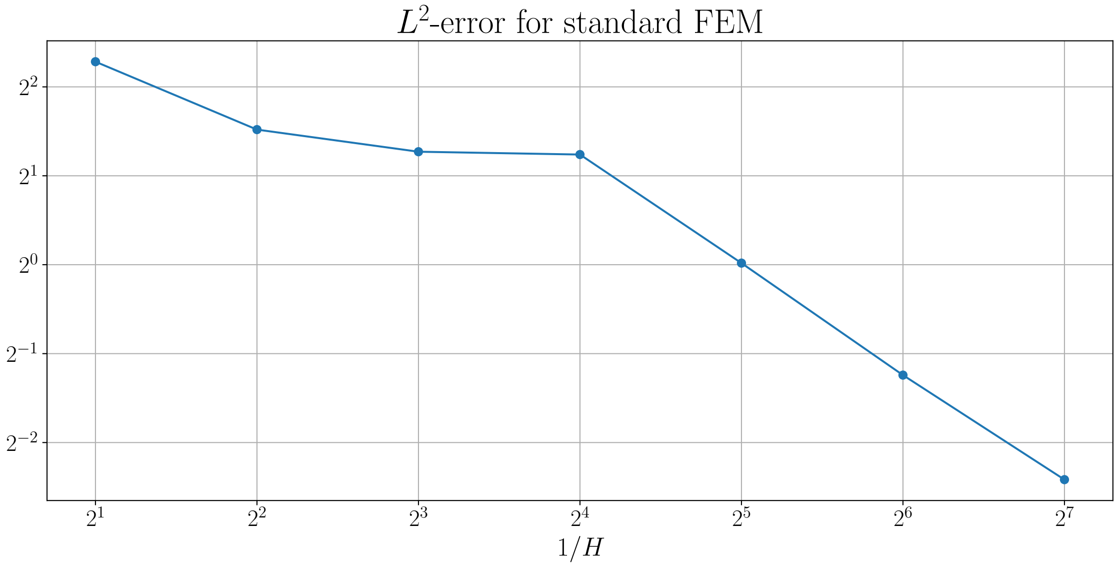


Figure 2.5: The L^2-error between the exact solution $u(x, y, t_N)$ and its approximation $U^N(x, y)$ for different values on $1/H$.

This example argues for the necessity of a multiscale method that can incorporate the finescale features of these solutions, without necessarily computing them on a fine mesh. The goal of the multiscale methods is to achieve the sought convergence rate as soon as the mesh refinement starts. For this purpose, this thesis dedicates itself to one such method, known as the Localized Orthogonal Decomposition method.

3
The Localized Orthogonal Decomposition method

The purpose of this chapter is to present the theory behind the Localized Orthogonal Decomposition (LOD) method, first introduced in [4]. Initially, LOD was presented for the elliptic problem

$$\begin{aligned} -\nabla \cdot (A(x)\nabla u(x)) &= f(x), \quad & x \in \Omega, \\ u(x) &= 0, & x \in \partial\Omega, \end{aligned}$$

so for convenience we set aside our strongly damped wave equation and present LOD with respect to this problem. As earlier shown, the standard FEM is not sufficient to efficiently solve problems in the case when the coefficients are highly varying. For these purposes the LOD may be used to evaluate solutions u_H^{ms} that satisfy the error estimate

$$|||u - u_H^{\mathrm{ms}}||| \leq CH,$$

where $|||\cdot||| := \|A^{1/2}\nabla \cdot \|$ denotes the energy norm and the coefficient C depends on the right hand side f and the global bounds of A, but not on its variations. Thus, the method may be used to solve the multiscale problems for which standard FEM does not suffice. Widely speaking, the goal of LOD is to incorporate the finescale behaviour of the coefficient A into the basis of the coarse FE-space, resulting in a modified vector space. In computational terms, this implies large and complex calculations resulting in a non-efficient method. It was however proven in [4] that the functions used to modify the coarse basis decay exponentially outside of an area of its corresponding node, giving the opportunity to only solve localized problem that are not too heavy to compute. Consequently, LOD turns into an efficient method for solving these multiscale type problems.

At first, we discuss the interpolation operator whose role is to decompose the solution space into a coarse and fine space respectively. This decomposition is further presented, along with the different localization procedures available for the method. Following this, an alternative LOD representation on a Petrov-Galerkin form is presented, along with the numerical setting of the method. Finally, the chapter presents several numerical examples that show the advantages of the LOD, as well as emphasize the necessity of another method to be used for the strongly damped wave equation.

3.1 Interpolation

It has previously been shown how the finite dimensional FE-space $V_H \subset V$ is not able to capture the fine variations that appear on a microscopic level in multiscale problems (unless the mesh is highly refined). For this purpose, we wish to decompose the space V into a coarse part, V_H, and a fine part V^f characterised by a mesh width $h < H$. The main goal is to evaluate the finescale features in the space V^f, and then incorporate them into the coarse basis functions $\{\lambda_x\}_{x\in\mathcal{N}_H}$ in V_H. The key tool to the construction of this decomposition is a bounded, linear, surjective (quasi-) interpolation operator

$$\mathfrak{I}_H : V \to V_H,$$

that maps functions $v \in V$ to functions $v_H \in V_H$. This interpolant will have the task of characterizing the microscopic details of the solution. It should be noted that the choice of this operator is not uniquely defined, but different choices might correspond to different multiscale methods. However, it is desired that the interpolation operator satisfies several properties, such as stability in $L^2(\Omega)$ and $H^1(\Omega)$. In general, the interpolation of a function $v \in V$ has the form

$$\mathfrak{I}_H v = \sum_{x\in\mathcal{N}_H} (\mathfrak{I}_H v)(x)\lambda_x, \tag{3.1}$$

where $\mathfrak{I}_H v(x)$ for $x \in \mathcal{N}_H$ is some operator dependent value, and for $x \notin \mathcal{N}_H$ is zero. Before introducing the decomposition using this operator, we first present several examples to get some perspective on interpolants that can be used.

Example 3.1.1 (Clément interpolant)**.** A suitable interpolation operator that has frequently been used for this purpose (among others [4] and [6]) is the (weighted) Clément interpolant with nodal values, defined by

$$(\mathfrak{I}_H v)(x) := \frac{(v, \lambda_x)_{L^2(\Omega)}}{(1, \lambda_x)_{L^2(\Omega)}},$$

for $x \in \mathcal{N}_H$. These nodal values consist of weighted averages over the domain $\mathrm{supp}(\lambda_x)$. Considering the property of summing over interior vertices, the homogeneous Dirichlet boundary condition is incorporated in the interpolant.

Example 3.1.2 (Localized L^2-projection)**.** Another suitable interpolant that has been used for the LOD (see e.g. [5]) is the L^2-projection. The localized L^2-projection $\mathfrak{I}_H : V \to V_H$ is defined as to satisfy (3.1), where for every node $x \in \mathcal{N}_H$, we have

$$(\mathfrak{I}_H v)(x) := (P_x v)(x),$$

and the projection $P_x v \in V_H\Big|_{\omega_x}$ satisfies

$$\int_{\omega_x} P_x v w_H \,\mathrm{d}x = \int_{\omega_x} v w_H \,\mathrm{d}x,$$

for all $w_H \in V_H\Big|_{\omega_x}$, where $\omega_x = \mathrm{supp}(\lambda_x)$. Since $\mathfrak{I}_H(v_H) = v_H$ for all $v_H \in V_H$, the operator is by definition a projection. Although it is not a requirement for the interpolation operators, they may in fact be projections.

For the sake of completeness, one should mention an example of a *non-suitable* operator. The Lagrange nodal interpolation is an example of such an operator, as it does not satisfy the necessary stability properties. More detailed examples regarding the interpolation operator can be seen in [5]. Throughout this thesis, the L^2-projection will be utilized as the interpolant $\mathfrak{I}_H$. With $\mathfrak{I}_H$ defined, the decomposition of the high resolution space V may now be done.

3.2 Two-scale orthogonal decomposition

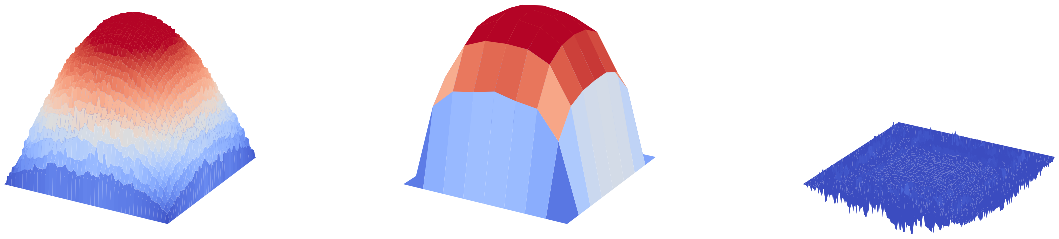
As recently described, the interpolant $\mathfrak{I}_H$ maps high resolution functions $v \in V$ into the coarse FE-space V_H. The microscopic features of V, i.e. the features not captured by V_H, are represented in a finescale space V^f defined by the interpolant's kernel, i.e.

$$V^f := \{v \in V : \mathfrak{I}_H v = 0\}.$$

The high resolution space V can then be decomposed as

$$V = V_H \oplus V^f.$$

Given functions $v_H \in V_H$ and $v^f \in V^f$, the functions $v \in V$ can be written as the sum $v = v_H + v^f$, and $(v_H, v^f) = 0$. The decomposition is illustrated with a numerical example in Figure 3.1.



(a) $v \in V$.

(b) $v_H \in V_H$.

(c) $v^f \in V^f$.

Figure 3.1: The high resolution function $v = v_H + v^f$ decomposed into its coarse and finescale part.

The LOD requires further splittings, which are done with respect to the problem's bilinear form. Originally, LOD was presented for elliptic multiscale problems, where the weak formulation is to find $u \in V$ such that

$$a(u, v) = (f, v), \quad \forall v \in V, \tag{3.2}$$

where the bilinear form $a(\cdot,\cdot)$ is similarly defined as for our weak formulation (but the coefficient $A(x)$ represents diffusion instead of damping). The existence and uniqueness of a solution to this weak formulation is guaranteed by the Lax-Milgram theorem. Since knowledge about the standard LOD is required, the method will be presented with respect to the weak formulation (3.2). This method will throughout the thesis be referred to as the standard LOD, and is not to be confused with future methods.

3.3 The standard method

For the standard LOD, the weak form in (3.2) is considered. The aim is to derive an orthogonal decomposition with respect to the bilinear form $a(\cdot,\cdot)$. For this purpose, for a given $v_H \in V_H$, define the Ritz projection $\mathcal{Q}$ such that $\mathcal{Q}v_H \in V^f$ solves

$$a(\mathcal{Q}v_H, w) = a(v_H, w), \quad \forall w \in V^f. \tag{3.3}$$

That is, the Ritz projection $\mathcal{Q} : V_H \to V^f$ corresponds to a finescale projection onto V^f. Given $\mathcal{Q}$, the multiscale space can be defined as

$$V_H^{\text{ms}} := (V_H - \mathcal{Q}V_H).$$

The dimension of the multiscale space V_H^{ms} still coincides with the one of the coarse FE-space V_H, hence it can be regarded as a modified coarse FE-space. This modification causes the finescale features to be incorporated into the coarse space. Furthermore, the projection $\mathcal{Q}$ yields an orthogonal splitting with respect to the bilinear form $a(\cdot,\cdot)$ as

$$V = V_H^{\text{ms}} \oplus_a V^f.$$

That is, any function $v \in V$ can be decomposed into $v_H^{\text{ms}} \in V_H^{\text{ms}}$ and $v^f \in V^f$ as $v = v_H^{\text{ms}} + v^f$, and

$$a(v_H^{\text{ms}}, v^f) = a(v_H - \mathcal{Q}v_H, v^f) = a(v_H, v^f) - a(\mathcal{Q}v_H, v^f) = 0.$$

With this multiscale space V_H^{ms} defined, we may introduce the LOD Galerkin approximation.

Definition 3.3.1 (Standard LOD approximation)**.** The standard LOD Galerkin approximation (with the elliptic weak form (3.2) in consideration) is to find $u_H^{\text{ms}} \in V_H^{\text{ms}}$ such that

$$a(u_H^{\text{ms}}, v) = (f, v), \quad \forall v \in V_H^{\text{ms}}. \tag{3.4}$$

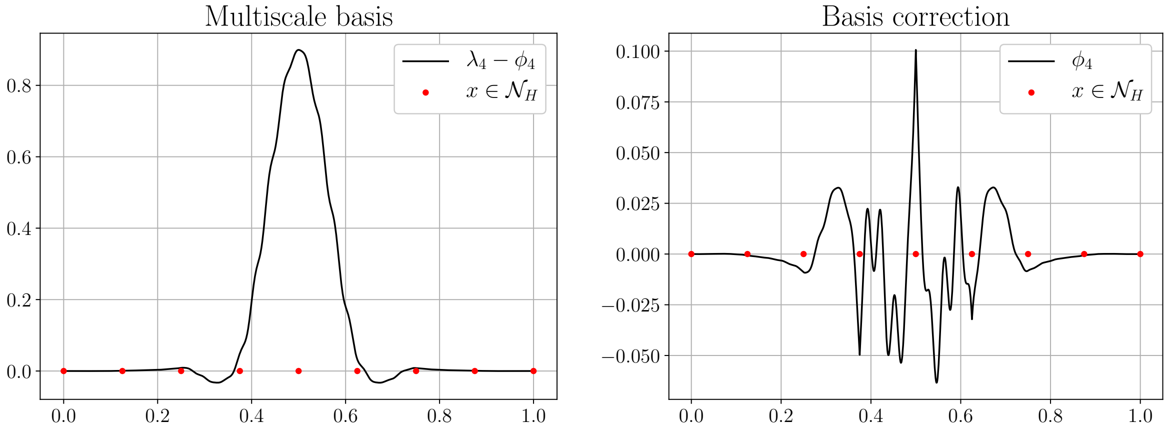
Finally, a basis for the multiscale space V_H^{ms} is to be introduced. Denote by $\phi_x := \mathcal{Q}\lambda_x \in V^f$ the image of the nodal basis λ_x under the finescale operator $\mathcal{Q}$. That is, the basis correction ϕ_x for each node $x \in \mathcal{N}_H$ satisfies the corrector problem

$$a(\phi_x, v^f) = a(\lambda_x, v^f), \quad \forall v^f \in V^f. \tag{3.5}$$

Since the dimensions of V_H and the modified space V_H^{ms} coincide, we may simply incorporate the corrections to each node of the corresponding basis function. From the definition of the multiscale space, the new basis is on the form

$$\{\lambda_x - \phi_x : x \in \mathcal{N}_H\}.$$

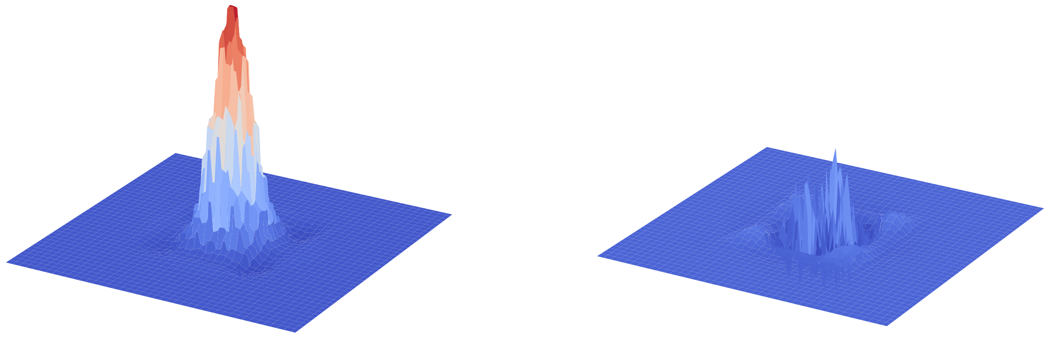
Examples of the correction ϕ_x and the modified basis $\lambda_x - \phi_x$ for a node $x \in \mathcal{N}_H$ can be seen in one and two dimensions in Figures 3.2 and 3.3 respectively.



(a) $\lambda_4 - \phi_4$.

(b) ϕ_4.

Figure 3.2: The correction ϕ_x and the modified basis $\lambda_x - \phi_x$ in one dimension for the node $x = 4$, using a coarse mesh with $1/H = 8$.



(a) $\lambda_{40} - \phi_{40}$.

(b) ϕ_{40}.

Figure 3.3: The correction ϕ_x and the modified basis $\lambda_x - \phi_x$ in two dimensions for the node $x = 40$, using a coarse mesh with $1/H = 64$.

To solve the system (3.5), it is required to use yet another Galerkin method, but this time on the finescale space. However, in comparison to the usual FEM Galerkin method, the basis corrections $\{\phi_x\}_{x \in \mathcal{N}_H}$ do not have local support. Their definition instead yields them a global support. Due to this, the resulting stiffness matrix is not sparse, but instead rather dense. Furthermore, the dimensions of V^f is higher than V_H (and V_H^{ms}), such that computing Galerkin systems on this space is expensive in comparison to the coarse spaces. These facts imply a high computational complexity of the LOD, and its practical uses are thus highly limited.

To work around the issue resulted by the correctors' global support, it was proven in [4] that a corrector ϕ_x decays exponentially outside an area of its corresponding coarse node x. Thus, one may truncate the computational domain to a neighbourhood of a corrector such that it is restricted solely to a patch around its node. By doing this, we restrict the support of the correctors such that the resulting matrices once again are sparse, making the system computable without being too computationally intensive.

3.4 Localization to patches

This section is dedicated to presenting the localization of the basis correction functions that justifies the LOD as a viable method. For this purpose, we introduce the following definition.

Definition 3.4.1 (Coarse grid patches)**.** Let $\omega \subset \Omega$ be either a node $x \in \mathcal{N}_H$ or an element $T \in \mathcal{T}_H$, and let $k \in \mathbb{N} \cup \{0\}$. Then we define the coarse grid patches $U_k(\omega) \subseteq \Omega$ such that

$$\begin{aligned} &U_0(\omega) = \omega, \\ &U_k(\omega) = \bigcup \left\{ \overline{T} \in \mathcal{T}_H : \overline{T} \cup U_{k-1}(\omega) \neq \emptyset \right\}, \quad k = 1, 2, ... \end{aligned}$$

Given these coarse grid patches, we wish to restrict the finescale space V^f to them. This is done by giving the finescale functions $v \in V^f$ compact support on $U_k(\omega)$ for a given k and ω. That is, our restricted finescale space becomes

$$V^f(U_k(\omega)) := \left\{ v \in V^f : \text{supp}(v) \subseteq U_k(\omega) \right\}.$$

With this truncated finescale space, the basis correctors will no longer have global support, but rather a local support restricted to our patches.

The patches can be defined in two different ways. The first way, presented in [4], is to let $\omega = \{x\}$ for $x \in \mathcal{N}_H$. For this method we call $U_k(\omega)$ a k-layer node patch. The other way (used in among others [6], [14] and [15]) is to let $\omega = T \in \mathcal{T}_H$. In this case $U_k(\omega)$ is called a k-layer element patch. Both of these methods work well for the LOD, but each method implies its separate error analysis. However, the element patch localization method has been proven to imply better results in terms of error estimates. This section gives an introduction to both localization techniques.

3.4.1 Node based localization

For the localization method based on the coarse nodes, we let $\omega = \{x\}$ for some $x \in \mathcal{N}_H$. In [4], the patches were then defined as

$$\begin{aligned} \omega_{x,1} &:= U_1(x) = \text{supp}(\lambda_x), \\ \omega_{x,k} &:= U_k(x), \quad k = 2, 3, ... \end{aligned}$$

That is, we get patches that spread across the coarse mesh as according to Figure 3.4.

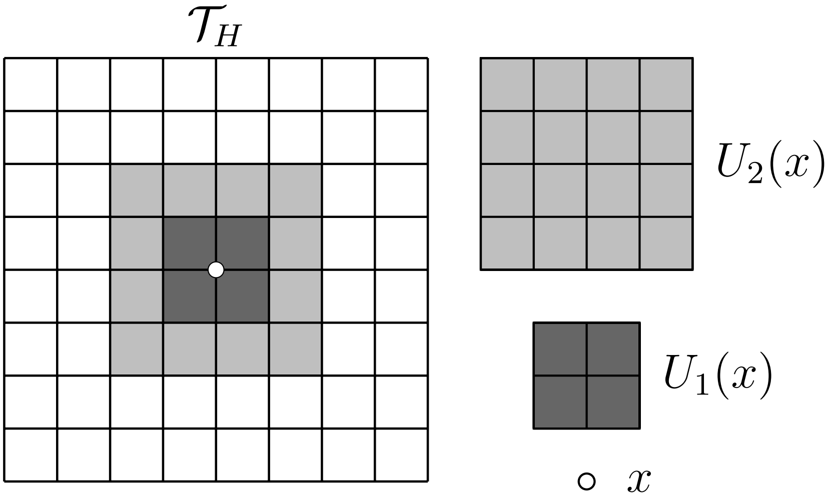


Figure 3.4: Illustration of how the node based patches spread across the mesh $\mathcal{T}_H$ with increasing k.

When using this method, for any given $x \in \mathcal{N}_H$, we denote the corresponding locally supported basis correction by $\phi_{x,k} \in V^f(\omega_{x,k})$, which solves

$$a(\phi_{x,k}, v^f) = a(\lambda_x, v^f), \quad \forall v^f \in V^f(\omega_{x,k}).$$

Incorporating these into the coarse basis functions gives us a node based localized multiscale space $\widetilde{V}^{\mathrm{ms}}_{H,k}$, spanned by the basis

$$\{\lambda_x - \phi_{x,k}\}_{x \in \mathcal{N}_H}.$$

For the node based localization we use an alternative notation so that the different localized spaces are distinguishable. Given this space, the following method is established.

Definition 3.4.2 (Node localized LOD)**.** The node patch localized LOD approximation (with respect to the elliptic weak form (3.2)) is to find $u^{\mathrm{ms}}_{H,k} \in \widetilde{V}^{\mathrm{ms}}_{H,k}$ such that

$$a(u^{\mathrm{ms}}_{H,k}, v) = (f, v), \quad \forall v \in \widetilde{V}^{\mathrm{ms}}_{H,k}.$$

for $k \in \mathbb{N}$.

The node based patches are used for the classical LOD in [4]. Later on, the element based patches have been a more common choice for the localization due to better convergence results in terms of error bounds.

3.4.2 Element based localization

For the element based localization, we set $\omega = T \in \mathcal{T}_H$. By Definition 3.4.1, we then get patches as according to Figure 3.5.

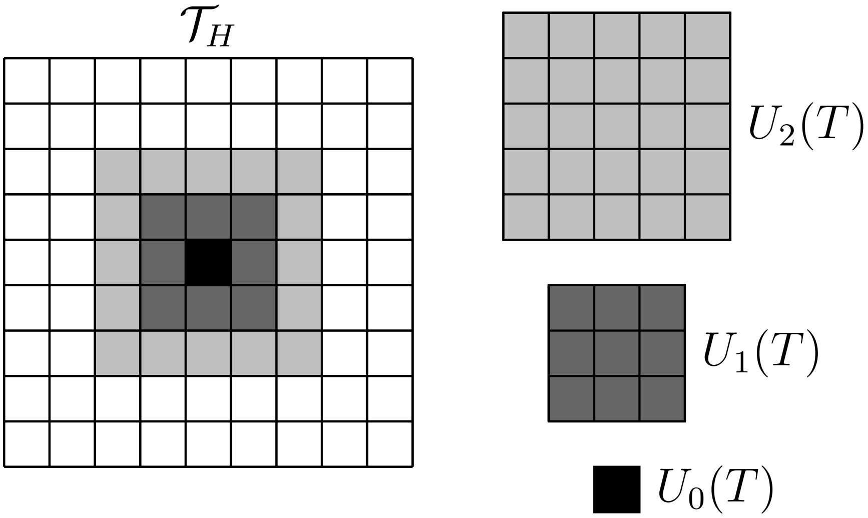


Figure 3.5: Illustration of how the element based patches spread across the mesh $\mathcal{T}_H$ with increasing k.

To begin the localization, we define the element restricted Ritz projection $\mathcal{Q}_T$ such that $\mathcal{Q}_T v \in V^f$ is the solution to the system

$$a(\mathcal{Q}_T v, v^f)_\Omega = a(v, v^f)_T, \quad \forall v^f \in V^f,$$

where we have denoted

$$a(u,v)_\Lambda := \int_\Lambda A\nabla u \cdot \nabla v \,\mathrm{d}x$$

for an arbitrary domain $\Lambda \subseteq \Omega$. Note here that if we sum over all elements $T \in \mathcal{T}_H$ we get

$$a\Big(\sum_T \mathcal{Q}_T v, v^f\Big)_\Omega = \sum_T a(v, v^f)_T = a(v, v^f)_\Omega, \quad \forall v^f \in V^f,$$

from which we see that for any $v \in V$ our global Ritz projection $\mathcal{Q}$ in (3.3) can be decomposed as

$$\mathcal{Q}v = \sum_{T \in \mathcal{T}_H} \mathcal{Q}_T v.$$

Furthermore, for $k \in \mathbb{N} \cup \{0\}$ we define the element patch localized Ritz projection $\mathcal{Q}_{k,T} : V \to V^f(U_k(T))$ such that for any $v \in V$ its projection $\mathcal{Q}_{k,T} v \in V^f(U_k(T))$ solves

$$a(\mathcal{Q}_{k,T} v, v^f)_{U_k(T)} = a(v, v^f)_T, \quad \forall v^f \in V^f(U_k(T)).$$

Similarly as earlier, this is a decomposition of a global localized Ritz projection $\mathcal{Q}_k$, such that for any $v \in V$ it is given by

$$\mathcal{Q}_k v = \sum_{T \in \mathcal{T}_H} \mathcal{Q}_{k,T} v.$$

Using this localized Ritz projection, we define a localized multiscale space

$$V_{H,k}^{\mathrm{ms}} := V_H - \mathcal{Q}_k V_H,$$

which is spanned by $\{\lambda_x - \mathcal{Q}_k \lambda_x\}_{x \in \mathcal{N}_H}$. With this space defined, we arrive at the following localized multiscale method.

Definition 3.4.3 (Element localized LOD)**.** The element patch localized LOD approximation (with the elliptic weak form (3.2) in consideration) is to find $u_{H,k}^{\mathrm{ms}} \in V_{H,k}^{\mathrm{ms}}$ such that

$$a(u_{H,k}^{\mathrm{ms}}, v) = (f, v), \quad \forall v \in V_{H,k}^{\mathrm{ms}}. \tag{3.6}$$

for $k \in \mathbb{N} \cup \{0\}$.

Note that the only difference between this localized approximation and the standard LOD is the choice of test and trial space. Furthermore, another commonly used formulation, in which the trial and test spaces are no longer equal, is known as the Petrov-Galerkin LOD version.

3.5 Petrov-Galerkin formulation

The Petrov-Galerkin version of the LOD (PG-LOD), first presented in [16], is a modification of the standard method. The general difference is the choice of trial and test space. Similarly to the standard LOD, the multiscale space V_H^{ms} is used as trial space. However, we replace the multiscale space with the standard FE-space V_H. More exactly, we end up with the following formulation.

Definition 3.5.1 (PG-LOD approximation)**.** The PG-LOD approximation (with the elliptic weak form (3.2) in consideration) is to find $u_H^{\mathrm{PG}} \in V_H^{\mathrm{ms}}$ such that

$$a(u_H^{\mathrm{PG}}, v) = (f, v), \quad \forall v \in V_H. \tag{3.7}$$

Note that when using this formulation, the trial space is still the same as earlier, and thus the finescale correctors remain the same. This implies that the localization arguments earlier used may be repeated for this formulation, and we get our localized PG-LOD version.

Definition 3.5.2 (Element localized PG-LOD)**.** The element patch localized PG-LOD approximation (with the elliptic weak form (3.2) in consideration) is to find $u_{H,k}^{\mathrm{PG}} \in V_{H,k}^{\mathrm{ms}}$ such that

$$a(u_{H,k}^{\mathrm{PG}}, v) = (f, v), \quad \forall v \in V_H.$$

for $k \in \mathbb{N} \cup \{0\}$.

The localized versions of the LOD and PG-LOD are similar, but separated in terms of test space. This difference implies a decrease in computational complexity for the PG-LOD, as the same communication with the finescale correctors is not necessary. In terms of convergence, it was proven in [16] that the PG-LOD version still satisfies similar results. However, since the trial and test spaces no longer are equal, the coercivity of the bilinear form is no longer guaranteed. Consequently, we can no longer rely on Lax-Milgram for the existence and uniqueness of the problem. Instead, we use the more general Banach-Nečas-Babuška theorem to guarantee well-posedness of the problem.

Theorem 3.5.1 (Banach-Nečas-Babuška, see [17])**.** *Let W be a Banach space, V a reflexive Banach space and $a : W \times V \to \mathbb{R}$ a continuous bilinear form. Then the problem* (3.2) *is well-posed if, and only if,*

$$\exists \alpha > 0 : \inf_{w \in W} \sup_{v \in V} \frac{a(v,w)}{\|w\|_W \|v\|_V} \geq \alpha,$$

and for all $v \in V$ it holds that

$$a(v,w) = 0, \ \forall w \in W \implies v = 0.$$

Once $a(\cdot,\cdot)$ satisfies this inf-sup condition (depends on k and might not always be fulfilled, see e.g. [13, Lemma 6.2.2]), the existence of a unique solution is assured, and thus the usage of the PG-LOD version is justified. In Section 3.7, some numerical examples are presented where the PG-LOD version has been utilized. Before this, the implementational aspects of the LOD and PG-LOD are established.

3.6 Implementation

Just as for the finite element methods, we apply similar arguments to transform the weak formulation into a fully discrete setting for the LOD, the PG-LOD, as well as their localized versions. As previously, we denote the finescale correction of the hat functions by $\phi_x := \mathcal{Q}\lambda_x$. Furthermore, the advantages of the PG-LOD from a computational point of view are briefly discussed.

3.6.1 Standard LOD formulation

Recall that for the standard LOD, we use the multiscale space V_H^{ms} as FE-space. As this space is spanned by the set $\{\lambda_x - \phi_x\}_{x \in \mathcal{N}_H}$, the solution $u_H^{\text{ms}} \in V_H^{\text{ms}}$ can be written on the form

$$u_H^{\text{ms}} = \sum_{x \in \mathcal{N}_H} \alpha_x (\lambda_x - \phi_x).$$

Inserting this expression into the approximation problem (3.4), we acquire the linear system

$$\mathcal{S}^{\text{ms}} \cdot \mathcal{U}_H^{\text{ms}} = \mathcal{F}^{\text{ms}},$$

where $\mathcal{U}_H^{\text{ms}} \in \mathbb{R}^{N_H}$ is the coefficient vector for u_H^{ms} and the entries of $\mathcal{S}^{\text{ms}} \in \mathbb{R}^{N_H \times N_H}$ and $\mathcal{F}^{\text{ms}} \in \mathbb{R}^{N_H}$ are given by

$$\begin{aligned} \mathcal{S}_{x,y}^{\text{ms}} &:= a(\lambda_x - \phi_x, \lambda_y - \phi_y), \\ \mathcal{F}_x^{\text{ms}} &:= (f, \lambda_x - \phi_x), \end{aligned}$$

for all $x, y \in \mathcal{N}_H$. Remember that since the support of $\lambda_x - \phi_x$ covers all of Ω for all $x \in \mathcal{N}_H$, the sparsity of the stiffness matrix no longer exists. Due to this, we introduced its localized version, where the FE-space is represented by $V_{H,k}^{\text{ms}}$ for $k \in \mathbb{N} \cup \{0\}$ instead. With this method, the fine mesh is only utilized to compute local problems, preventing problems such as memory issues and computational complexity. In this space, the solution may be written as a linear combination of the basis functions $\{\lambda_x - \mathcal{Q}_k\lambda_x\}_{x\in\mathcal{N}_H}$ that spans $V_{H,k}^{\mathrm{ms}}$. Similarly as above, we insert the solution's expression into the localized approximation problem (3.6), and end up with a linear system of the form

$$\mathcal{S}^{\mathrm{ms},k} \cdot \mathcal{U}_H^{\mathrm{ms}} = \mathcal{F}^{\mathrm{ms},k},$$

where $\mathcal{S}^{\mathrm{ms,k}} \in \mathbb{R}^{N_H\times N_H}$ and $\mathcal{F}^{\mathrm{ms},k} \in \mathbb{R}^{N_H}$ are determined by

$$\begin{aligned}\mathcal{S}_{x,y}^{\mathrm{ms},k} &:= a(\lambda_x - \mathcal{Q}_k\lambda_x, \lambda_y - \mathcal{Q}_k\lambda_y),\\ \mathcal{F}_x^{\mathrm{ms},k} &:= (f, \lambda_x - \mathcal{Q}_k\lambda_x),\end{aligned}$$

for all $x, y \in \mathcal{N}_H$. Since this formulation requires more communication between the finescale correctors, the alternative formulation of Petrov-Galerkin kind was further introduced, for which we establish a fully discrete setting as well.

3.6.2 The PG-LOD formulation

The only difference between PG-LOD and standard LOD is the choice of test space. As the trial spaces between the methods still are equal, the solution is still a linear combination of the same basis functions, i.e.

$$u_H^{\mathrm{PG}} = \sum_{x\in\mathcal{N}_H} \alpha_x(\lambda_x - \phi_x).$$

In a similar manner as earlier, we plug this into our PG-LOD approximation problem (3.7), and recieve a linear system

$$\mathcal{S}^{\mathrm{PG}} \cdot \mathcal{U}_H^{\mathrm{PG}} = \mathcal{F}^{\mathrm{PG}}.$$

However, due to the different choice of test space, the resulting stiffness matrix is no longer symmetric. Instead $\mathcal{S}^{\mathrm{PG}} \in \mathbb{R}^{N_H\times N_H}$ and $\mathcal{F}^{\mathrm{PG}} \in \mathbb{R}^{N_H}$ are determined by

$$\begin{aligned}\mathcal{S}_{y,x}^{\mathrm{PG}} &:= a(\lambda_x - \phi_x, \lambda_y),\\ \mathcal{F}_x^{\mathrm{PG}} &:= (f, \lambda_x),\end{aligned} \tag{3.8}$$

for all $x, y \in \mathcal{N}_H$. Note the change of index order in the stiffness matrix. By the same localization arguments as earlier, we furthermore require the localized PG-LOD system. The localized system is of the form (3.8), but the stiffness matrix $\mathcal{S}^{\mathrm{PG}}$ is replaced by its localized version $\mathcal{S}^{\mathrm{PG},k}$, whose entries are determined by

$$\mathcal{S}_{y,x}^{\mathrm{PG},k} := a(\lambda_x - \mathcal{Q}_k\lambda_x, \lambda_y),$$

for all $x, y \in \mathcal{N}_H$.

As previously mentioned, the PG-LOD has an advantage over the standard LOD in terms of computational complexity, mainly due to the fact that the standard method requires more communication with the finescale correctors. For example, consider the assembly of the localized standard multiscale stiffness matrix $\mathcal{S}^{\mathrm{ms},k}$. For this process, it requires that we compute entries of the type

$$\int_\Omega A\nabla(\lambda_x - \mathcal{Q}_k\lambda_x) \cdot \nabla(\lambda_y - \mathcal{Q}_k\lambda_y)\,\mathrm{d}x,$$

which in itself involves computing the term

$$\sum_{\substack{T\in\mathcal{T}_H\\T\subset\omega_x}}\sum_{\substack{K\in\mathcal{T}_H\\K\subset\omega_y}}\int_{U_k(T)\cap U_k(K)} A\nabla\mathcal{Q}_{k,T}\lambda_x\cdot\nabla\mathcal{Q}_{k,K}\lambda_y\,\mathrm{d}x,$$

where ω_x denotes the support of the coarse basis function λ_x. With this expression we see that an efficient implementation of this requires knowledge about the patches' intersection. The intersection may in fact be non-zero even if T and K are not particularly close to each other. Moreover, it is impossible to compute these entries without knowing both $\mathcal{Q}_k\lambda_x$ and $\mathcal{Q}_k\lambda_y$ at the same time. The correctors must thus be stored in order to communicate with each other. For the PG-LOD, the structure of an entry is instead given by

$$\int_\Omega A\nabla(\lambda_x-\mathcal{Q}_k\lambda_x)\cdot\nabla\lambda_y\,\mathrm{d}x=\sum_{\substack{T\in\mathcal{T}_H\\T\subset\omega_x}}\int_{U_k(T)} A\nabla(\lambda_x-\mathcal{Q}_{k,T}\lambda_x)\cdot\nabla\lambda_y\,\mathrm{d}x,$$

which can be assembled once $\mathcal{Q}_{k,T}\lambda_x$ is computed. As the test function's support is limited to the one of the coarse basis functions, the resulting stiffness matrix contains far less relevant entries, and it consequently establishes a more sparse linear system than the symmetric matrix. Also, we no longer require the communication between the correctors, which makes it possible to simply delete $\mathcal{Q}_{k,T}\lambda_x$ after it has been used. As a consequence, the storage requirements for the PG-LOD are significantly lower than for the standard LOD. Furthermore, by considering the load vector on the right hand side, we note another disadvantage of the standard method. With the usual coarse FE-space V_H as test space, the complexity to compute the load vector is the same as for the usual FEM on the coarse scale. One should however note that these advantages come with a few trade-offs, as for example matrix symmetry etc. For further details on the Petrov-Galerkin formulation of the LOD and its advantages, see e.g. [16] or [5].

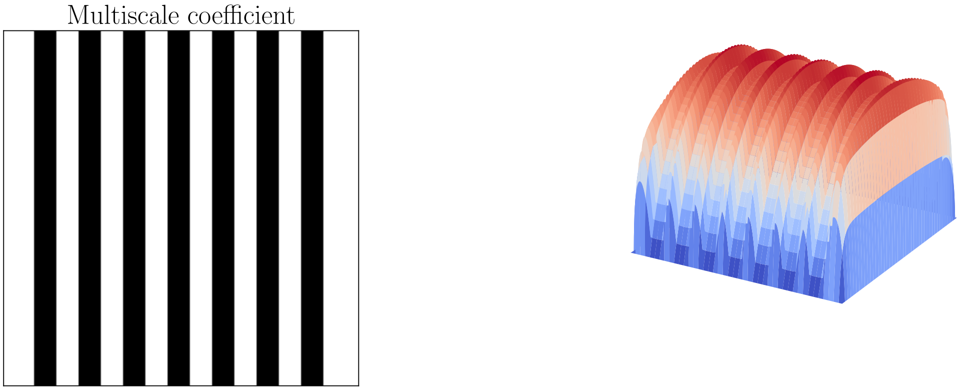
3.7 Numerical examples

In this section, numerical examples are presented with the purpose of showing how the standard LOD method performs when the equation's coefficients are of multiscale type. Although the method has been presented with respect to an elliptic PDE, the strongly damped wave equation is used as a test problem. This is to demonstrate the cases where it suffices to use the standard LOD, and to furthermore emphasize the necessity of a new method. For the completely discretized strongly damped wave equation, we consider the following definition.

Definition 3.7.1 (PG-LOD for the strongly damped wave equation)**.** The Backward-Euler element localized PG-LOD approximation for the strongly damped wave equation is to find $U_{H,k}^{n,\mathrm{PG}}\in V_{H,k}^{\mathrm{ms}}$ such that

$$(\bar\partial_t\bar\partial_t U_{H,k}^{n,\mathrm{PG}},v)+a(\bar\partial_t U_{H,k}^{n,\mathrm{PG}},v)+b(U_{H,k}^{n,\mathrm{PG}},v)=(f^n,v),\quad\forall v\in V_H$$

with initial conditions $U_{H,k}^{0,\mathrm{PG}},U_{H,k}^{1,\mathrm{PG}}\in V_{H,k}^{\mathrm{ms}}$ for $k\in\mathbb{N}\cup\{0\}$ and $n=2,3,...,N$.



(a) $A(x, y)$ and $B(x, y)$.

(b) Finescale solution $u(x, y, t_N)$.

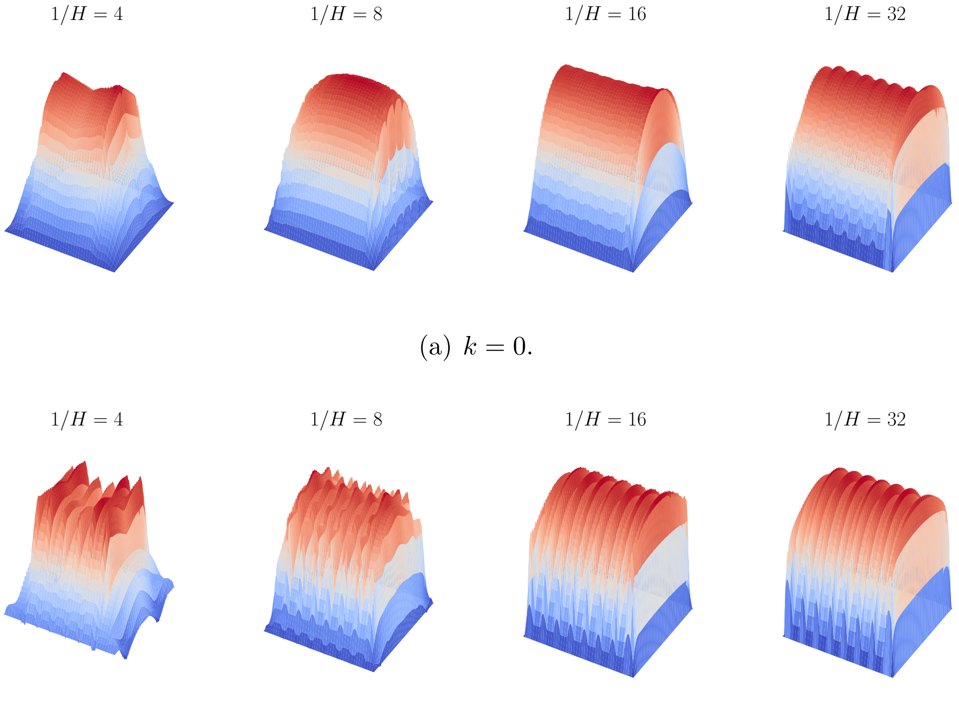
Figure 3.6: The coefficient used for $A(x, y)$ and $B(x, y)$ as well as the solution evaluated on a fine mesh with $1/h = 256$.

Here, the multiscale space $V^{\text{ms}}_{H,k}$ is defined as for the standard LOD, i.e. the correction is only made for the bilinear form $b(\cdot, \cdot)$. At first, an example where the damping and the propagation coefficients coincide are demonstrated. Following this, the coefficients are set as $A \neq B$ and the performance of the PG-LOD is analysed in this case.

3.7.1 Similar multiscale coefficients

In this example, we revisit the multiscale example seen in Section 2.5.2, but let both the damping and the propagation be represented by the same coefficient. The coefficient's behaviour can be seen in Figure 3.6(a). Furthermore, the example uses the unit square as domain and initial data $u_0 = v_0 = 0$. The time step is set to $\tau = 0.1$ and only one time step is performed, i.e. the solution is evaluated at $t_N = 0.1$. The finescale solution evaluated with standard FEM on a mesh with $1/h = 256$ is used as reference solution, and is depicted in Figure 3.6(b).

The approximate solutions are evaluated using element localized PG-LOD. To get some perception of the patch sizes' impact on the solution, it is computed using $k = 0, 1, 2, 3$, and compared with an approximation using standard FEM on the same mesh widths. In Figure 3.7, the solutions using PG-LOD for different k's can be seen. For $k = 0$ the solutions do not approximate the global behaviour very well when $1/H$ is small. Furthermore, as H gets smaller, the solution is supposed to resolve the stripes that appear due to the coefficient's appearance (as can be seen in the reference solution). However, for a small patch size such as $k = 0$, the approximation do not capture this structure well. The stripes appear first at $1/H = 32$, but their depths are not particularly accurate. For larger patch sizes, such as $k = 2$, we see that the macroscopic behaviour is in general better for small values on $1/H$, and the stripes appear already at $1/H = 4$, and at $1/H = 16$ the solution's stripes resembles those of the finescale solution well. In Figure 3.8, the L^2-error between the approximate solutions and the reference solution has been evaluated and plotted for different mesh widths H, and for the different patch sizes k. It is seen here that for $k = 2, 3$ the error converges as wished, while for the other cases the convergence is disrupted due to the highly varying coefficients.



(a) $k = 0$.

(b) $k = 2$.

Figure 3.7: The approximate solutions plotted for different element sizes H and patch sizes $k = 0, 2$.

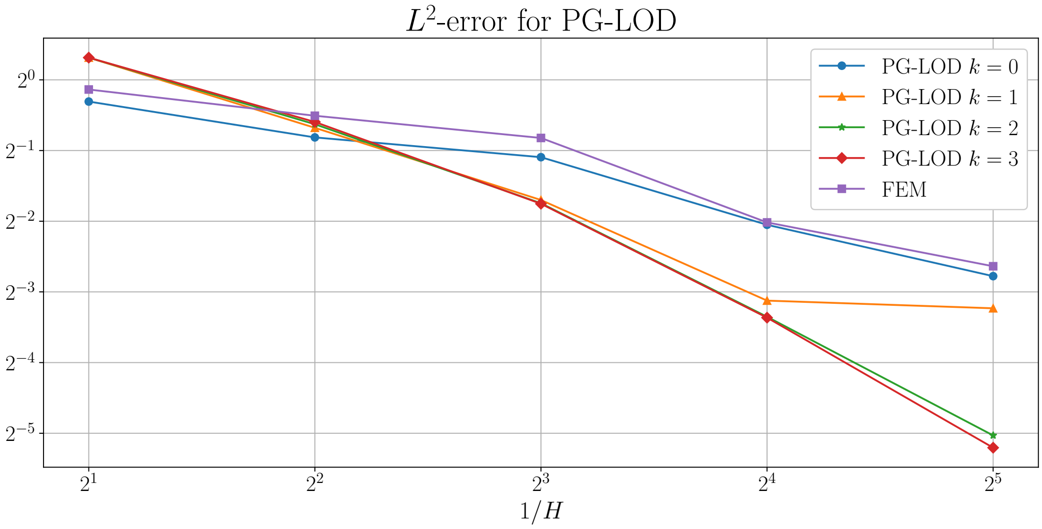
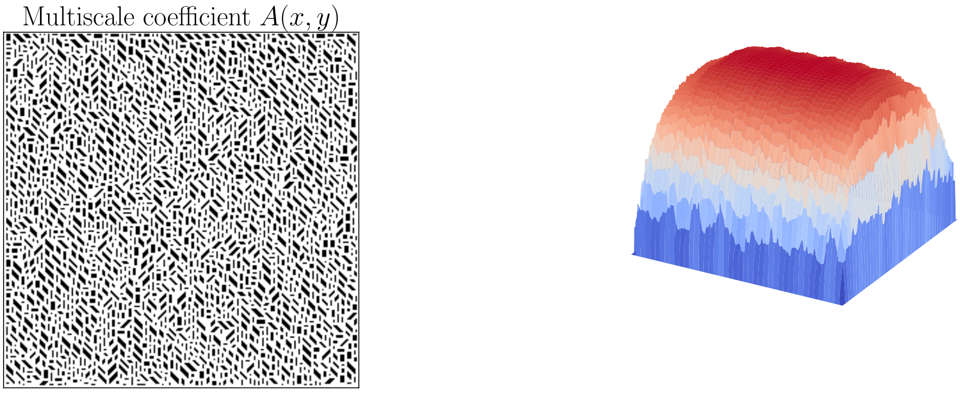


Figure 3.8: The L^2-error between the reference and the approximate solution, evaluated for different element sizes H and different choices of k.



(a) $A(x, y)$.

(b) Finescale solution $u(x, y, t_N)$.

Figure 3.9: The coefficient $A(x, y)$ as well as the solution evaluated on a fine mesh with $1/h = 256$.

3.7.2 Different multiscale coefficients

This example is presented for the purpose of analyzing the PG-LOD performance in the case when $A(x, y)$ and $B(x, y)$ no longer are equal. The numerical settings such as time and spatial discretization, number of time steps, etc., stay the same as in previous example. The propagation coefficient $B(x, y)$ is still described as according to Figure 3.6(a), but in this case we change the damping coefficient $A(x, y)$ to another highly varying coefficient. Figure 3.9(a) depicts the new behaviour of $A(x, y)$, and Figure 3.9(b) shows the corresponding finescale solution.

Once again we use element localized PG-LOD with respect to B to evaluate the approximations. In this case $k = 5$ is used, such that the patch size is large in comparison to previous example. Computations using the standard FEM are also done for comparison reasons. In Figure 3.10, the approximate solutions can be visualized for different mesh widths H. As seen here, the blurry finescale features do first appear at $1/H = 32$, but not nearly in the same manner as the reference solution. Furthermore, the numerical solution fails to approximate the global behaviour of the reference solution, much like in the example presented in Section 2.5.2. In Figure 3.11, the L^2-error between the reference $u(x, y, t_N)$ and the numerical solutions $U^N(x, y)$ are plotted for different mesh widths. In this plot, it is clearly seen how the numerical solutions fail to converge as we expect from the PG-LOD. These results arise from the fact that the standard method only corrects for one of the coefficients, and is not able to incorporate both their behaviours into the coarse basis. Moreover, we see from Figure 3.9 that there is a large impact from the coefficient $A(x, y)$ in the solution. The basis created for $V_{H,k}^{\text{ms}}$ is thus of no use for this problem, causing convergence results as slow as for the standard FEM.

The results given in this section show that the standard PG-LOD method is not sufficient to achieve a well approximated solution to the strongly damped wave equation. However, for the given example, the solution is evaluated at an early

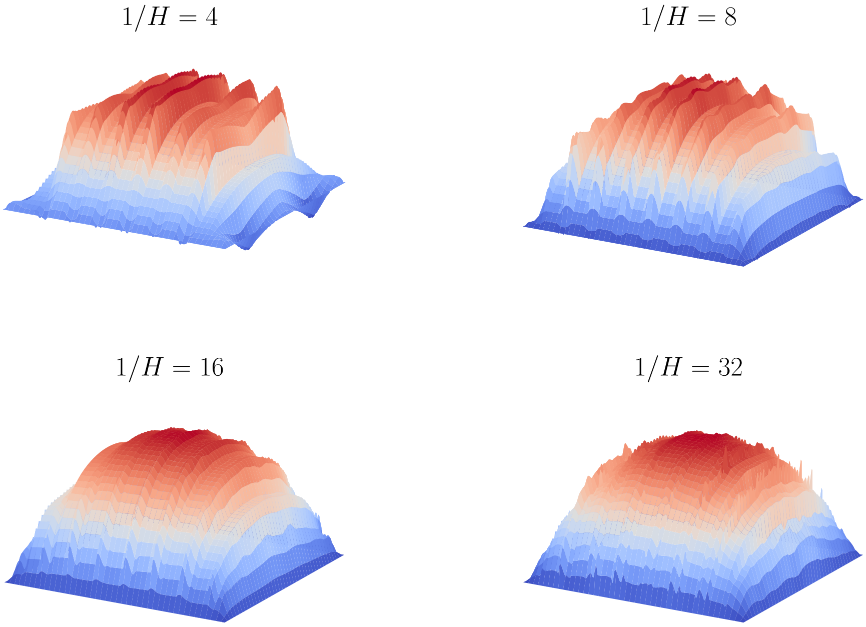


Figure 3.10: The approximate solution for different element sizes H and patch size $k = 5$.

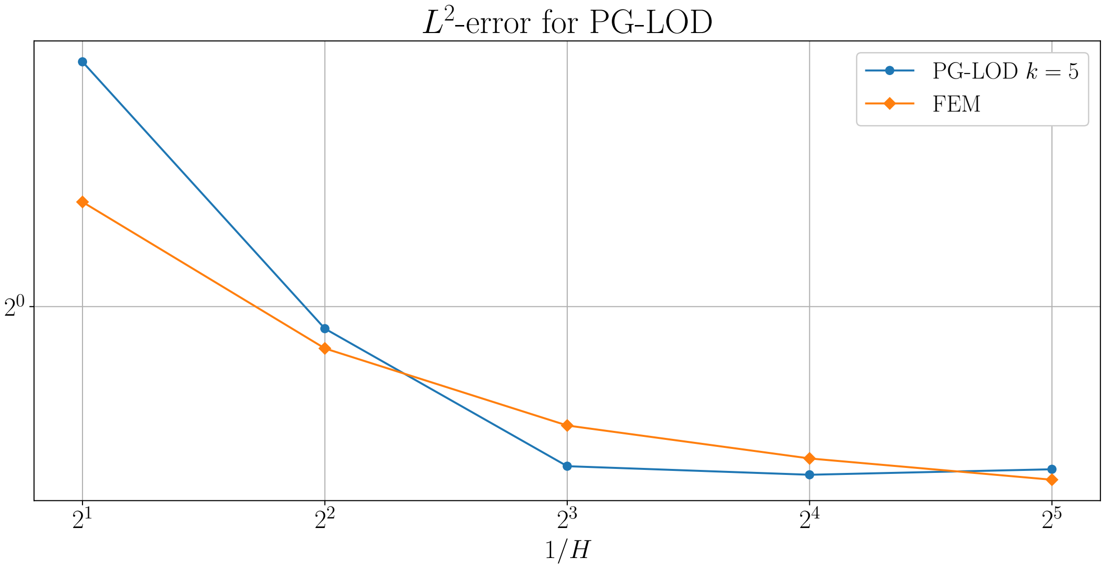


Figure 3.11: The L^2-error between $u(x, y, t_N)$ and $U^N(x, y)$ with respect to $1/H$.

time point $t = 0.1$. One may note that due to the derivative of u that appear in the equation, the solution will be much affected by the damping at an early state, which is seen in the example. As time passes, the solution will eventually reach a steady state, at which the damping term vanishes and no longer affects the solution. That is, for large values on t, the solution's behaviour will instead correspond to the behaviour of the propagation coefficient B. To visualize this effect, the finescale solution for different time points is plotted and seen in Figure 3.12. In these plots it is clearly seen how the behaviour of A vanishes and B becomes the dominant coefficient affecting the solution. Consequently, it suffices to use the PG-LOD for the test problem if one considers the steady state version of it.

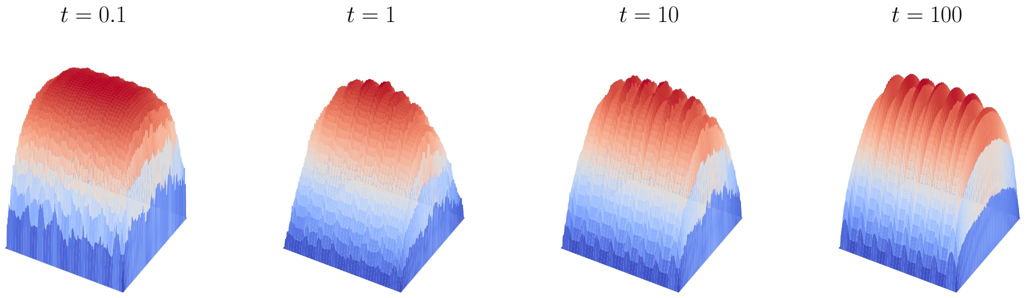


Figure 3.12: The finescale solution $u(x, y, t_n)$ plotted for different time points.

The given examples show how the PG-LOD works well when both coefficients coincide, but lacks in performance as soon as they behave differently. This emphasizes the necessity of a new method, inherited from the standard LOD. For the new method we are looking for coarse solutions that, at an early state, mainly corrects with respect to the damping coefficient A. Furthermore, due to the solution's time dependence, the finescale features will alter as t gets larger. Thus, the method will require further finescale calculations to correct for these changes.

4

LOD for the strongly damped wave equation

The previous chapters have discussed the theoretical settings for methods such as the FEM, LOD and PG-LOD. Furthermore, numerical examples have shown that none of these methods generally manage to estimate a solution to the strongly damped wave equation when the damping and propagation coefficients are characterized by high frequency oscillations. This chapter presents the theoretical approach of a method that is able to resolve the solution's finescale variations already for small values on $1/H$. For simplicity, this is done by first considering a solution decomposition of a simpler problem where neither the source term nor the second derivative is included. Following this, the source term is added, for which equation we present a theorem that proves linear convergence for the approximation error in H^1-norm. At last, the full strongly damped wave equation is considered, as well as a spatial and temporal localization for the necessary finescale calculations.

4.1 Simplified problem formulation

First, we consider the problem where the source term and the solution's second time derivative are neglected. This is done for the purpose of incorporating the correction for both multiscale coefficients on a simpler problem formulation than for the full equation. Thus, consider the equation

$$\begin{aligned} -\nabla \cdot (A\nabla \dot{u} + B\nabla u) &= 0, \text{ in } \Omega \times (0,T], \\ u &= 0, \text{ on } \Gamma \times (0,T], \end{aligned}$$

with initial data $u(\cdot,0) = u_0 \neq 0$. By standard methods, the corresponding weak formulation is to find $u \in V$ such that

$$a(\dot{u}, z) + b(u, z) = 0, \quad \forall z \in V.$$

After a uniform time discretization we arrive at the problem to find $u^n \in V$ such that

$$a(u^n, z) + \tau b(u^n, z) = a(u^{n-1}, z), \quad \forall z \in V,$$

for all $n = 1, 2, ..., N$ with $u^0 = u_0$. In comparison to the elliptic equation used for the standard LOD, we have two different bilinear forms instead of one. Our goal is to construct an orthogonal decomposition of the solution space into a multiscale space V_H^{ms} and a finescale space V^f, but this time with respect to both bilinear forms, and not just $b(\cdot,\cdot)$.

4.2 Orthogonal decomposition

Similarly to the standard LOD, we let $\mathfrak{I}_H$ be an interpolation operator, e.g. the L^2-projection, and let the finescale space be defined as

$$V^f := \ker(\mathfrak{I}_H),$$

such that we yield the usual decomposition $V = V_H \oplus V^f$ between the standard FE-space and the finescale space. Furthermore, assume a scalar product $\langle \cdot, \cdot \rangle_V$ on the high resolution space V to be defined as

$$\langle v, w \rangle_V := a(v, w) + \tau b(v, w). \tag{4.1}$$

Given this scalar product, we define the Ritz projection $\mathfrak{P} : V_H \to V^f$ such that $\mathfrak{P} v_H \in V^f$ solves

$$\langle \mathfrak{P} v_H, z \rangle_V = \langle v_H, z \rangle_V, \quad \forall z \in V^f.$$

The $\mathfrak{P}$-notation is used to distinguish this Ritz projection from the one in the standard LOD. Our new multiscale space can then be defined as

$$V_H^{\text{ms}} := V_H - \mathfrak{P} V_H = \{ v \in V : \langle v, z \rangle_V = 0, \ \forall z \in V^f \},$$

spanned by the basis $\{\lambda_x - \psi_x\}_{x \in \mathcal{N}_H}$, where $\psi_x := \mathfrak{P} \lambda_x \in V^f$ solves

$$a(\psi_x, z) + \tau b(\psi_x, z) = a(\lambda_x, z) + \tau b(\lambda_x, z), \quad \forall z \in V^f.$$

Naturally, this yields an orthogonal decomposition with respect to the scalar product $\langle \cdot, \cdot \rangle_V$, i.e.

$$V = V_H^{\text{ms}} \oplus_V V^f,$$

such that every solution can be written on the form $u^n = v^n + w^n$ with $v^n \in V_H^{\text{ms}}$ and $w^n \in V^f$.

4.3 Compute decomposed solutions

We wish to evaluate the solution u^n for each time step by computing its decomposed parts v^n and w^n. This method results in two systems, a coarse and a fine one. For test functions $z \in V_H^{\text{ms}}$ we note that

$$a(u^n, z) + \tau b(u^n, z) = a(v^n, z) + \tau b(v^n, z) + \underbrace{\langle w^n, z \rangle_V}_{=0},$$

and similarly for test functions $z \in V^f$ we get

$$a(u^n, z) + \tau b(u^n, z) = a(w^n, z) + \tau b(w^n, z) + \underbrace{\langle v^n, z \rangle_V}_{=0}.$$

The two resulting systems become to find $v^n \in V_H^{\text{ms}}$ such that

$$a(v^n, z) + \tau b(v^n, z) = a(u^{n-1}, z), \quad \forall z \in V_H^{\text{ms}},$$

and to find $w^n \in V^f$ such that

$$a(w^n, z) + \tau b(w^n, z) = a(u^{n-1}, z), \quad \forall z \in V^f,$$

respectively. For simplicity, assume the initial data to be a function restricted to the multiscale space, i.e. $u^0 \in V_H^{\text{ms}}$. By the decomposition, this implies that $u^0 = v^0$ and $w^0 = 0$. For $n = 1$, we thus recieve the finescale equation

$$a(w^1, z) + \tau b(w^1, z) = a(u^0, z) = a(v^0, z), \quad \forall z \in V^f. \tag{4.2}$$

Recall that the space V_H^{ms} is spanned by the basis $\{\lambda_x - \psi_x\}_{x \in \mathcal{N}_H}$, so we can write

$$v^0 = \sum_{x \in \mathcal{N}_H} \alpha_x^0 (\lambda_x - \psi_x).$$

We may insert this expression into (4.2), and end up with the system to solve

$$a(w_{x,1}^1, z) + \tau b(w_{x,1}^1, z) = a(\lambda_x - \psi_x, z), \quad \forall z \in V^f, \tag{4.3}$$

for all $x \in \mathcal{N}_H$, such that w^1 can be constructed as

$$w^1 = \sum_{x \in \mathcal{N}_H} \alpha_x^0 w_{x,1}^1.$$

The full solution in the first time step is then constructed as $u^1 = v^1 + w^1$, where v^1 is given by solving the first coarse system that has the form

$$a(v^1, z) + \tau b(v^1, z) = a(u^0, z) = a(v^0, z), \quad \forall z \in V_H^{\text{ms}},$$

since $u^0 \in V_H^{\text{ms}}$. For $n = 2$, we seek to solve

$$a(w^2, z) + \tau b(w^2, z) = a(v^1, z) + a(w^1, z), \quad \forall z \in V^f.$$

Since v^1 and w^1 depend on different coefficients, we decompose the solution as $w^2 = w_1^2 + w_2^2$, such that w_1^2 and w_2^2 are the solutions to

$$\begin{aligned} a(w_1^2, z) + \tau b(w_1^2, z) &= a(v^1, z), \\ a(w_2^2, z) + \tau b(w_2^2, z) &= a(w^1, z), \end{aligned}$$

for all $z \in V^f$. These equations may be divided such that for each $x \in \mathcal{N}_H$ we seek the solutions to

$$\begin{aligned} a(w_{x,1}^2, z) + \tau b(w_{x,1}^2, z) &= a(\lambda_x - \psi_x, z), \\ a(w_{x,2}^2, z) + \tau b(w_{x,2}^2, z) &= a(w_{x,1}^1, z), \end{aligned}$$

for all $z \in V^f$. Note that the first of these equations has already been solved in previous time step, in (4.3). Thus, it is sufficient to only compute the second equation. The decomposed parts w_1^2 and w_2^2 can then be constructed as

$$w_1^2 = \sum_{x \in \mathcal{N}_H} \alpha_x^1 w_{x,1}^2, \qquad w_2^2 = \sum_{x \in \mathcal{N}_H} \alpha_x^0 w_{x,2}^2,$$

and get the finescale solution w^2 by adding these together. With $w^1 \neq 0$, the second coarse system is of the form to find $v^2 \in V_H^{\mathrm{ms}}$ such that

$$\begin{aligned} a(v^2, z) + \tau b(v^2, z) &= a(u^1, z) \\ &= a(v^1, z) + a(w^1, z), \end{aligned}$$

for all $z \in V_H^{\mathrm{ms}}$. Inductively, for each time step n, we have to solve the finescale sytem to find $w^n \in V^f$ such that

$$a(w_{x,n}^n, z) + \tau b(w_{x,n}^n, z) = a(w_{x,n-1}^{n-1}, z), \quad \forall z \in V^f,$$

for all $x \in \mathcal{N}_H$. For arbitrary n, the finescale solution is then written on the form

$$w^n = \sum_{i=1}^{n} \sum_{x \in \mathcal{N}_H} \alpha_x^{n-i} w_{x,i}^n.$$

Since $w_{x,i}^n = w_{x,i}^{n-1}$ for all n, we skip the index of the current time step and denote $w_{x,i}^n = w_{x,i}$ from here on. With the finescale system solved, we may plug this expression into the coarse equation and solve for $v^n \in V_H^{\mathrm{ms}}$ by standard Galerkin methods. The full solution will then be on the form

$$u^n = \sum_{x \in \mathcal{N}_H} \alpha_x^n (\lambda_x - \psi_x) + \sum_{i=1}^{n} \sum_{x \in \mathcal{N}_H} \alpha_x^{n-i} w_{x,i}.$$

Note that no source term is used for this problem. Thus, this technique is merely a decomposition of the original problem, and the result will be exact for each time step. This is done for the purpose of simply analyzing the decomposition of both coefficients before solving more complex problems with the same technique. Next, we add a source term to the right hand side, such that we analyze a problem to which the approximate solution is not exact, but depends on parameters such as the mesh width H.

4.4 Adding a source term

We now extend the equation considered earlier in this chapter by adding a (non-zero) source term $f \in L^2$ to it. That is, the equation becomes

$$\begin{aligned} -\nabla \cdot (A \nabla \dot{u} + B \nabla u) &= f, \text{ in } \Omega \times (0, T], \\ u &= 0, \text{ on } \Gamma \times (0, T], \end{aligned}$$

with initial data $u(\cdot, 0) = u_0$. The corresponding weak form is to find $u \in V$ such that

$$a(\dot{u}, z) + b(u, z) = (f, z), \quad \forall z \in V,$$

and after a backward Euler scheme is applied with uniform time step τ, we seek $u^n \in V$ such that

$$a(u^n, z) + \tau b(u^n, z) = a(u^{n-1}, z) + \tau (f, z), \quad \forall z \in V, \tag{4.4}$$

for all $n = 1, 2, ..., N$, with initial condition $u^0 = u_0$. The same orthogonal decomposition as in Section 4.2 is still used, such that we seek the solution u_H^n of the form $u_H^n = v^n + w^n$, for $v^n \in V_H^{\text{ms}}$ and $w^n \in V^f$. By the same arguments as in previous section, we arrive at a coarse scale equation

$$a(v^n, z) + \tau b(v^n, z) = a(v^{n-1}, z) + a(w^{n-1}, z) + \tau(f, z), \quad \forall z \in V_H^{\text{ms}}, \tag{4.5}$$

and a finescale equation

$$a(w^n, z) + \tau b(w^n, z) = a(v^{n-1}, z) + a(w^{n-1}, z), \quad \forall z \in V^f. \tag{4.6}$$

Note that the source term f is not present in the finescale equation, since $f \in L^2$ and therefore requires no correction. Due to this, the finescale system is identical to the one in previous section, and thus the finescale calculations remain the same as in the homogeneous case. In fact, the only thing that differs from the previous case is that we need to take the term (f, z) into account in the coarse Galerkin method. Since we no longer consider merely a decomposition of the solution, the solution will not be exact in every time step, but will depend on parameters such as the the mesh refinement parameter H instead. The following theorem establishes an error estimate between the exact solution u^n and the approximate solution u_H^n.

Theorem 4.4.1. *Let u^n be the solution to* (4.4)*, and u_H^n the solution to* (4.5)-(4.6)*. Then the error is bounded by*

$$\|u^n - u_H^n\|_{H^1} \leq CH \sum_{j=1}^{n} \tau \|f^j\|,$$

where C does not depend on the variations in A or B.

Proof. Since $u^n \in V$, there are $\bar{v}^n \in V_H^{\text{ms}}$ and $\bar{w}^n \in V^f$ such that $u^n = \bar{v}^n + \bar{w}^n$. Let $e^n = u^n - u_H^n$, and consider

$$\begin{aligned}
\|e^n\|_{a,b}^2 &:= a(e^n, e^n) + \tau b(e^n, e^n) \\
&= a(u^n, e^n) + \tau b(u^n, e^n) - a(u_H^n, e^n) - \tau b(u_H^n, e^n) \\
&= \tau(f^n, e^n) + a(u^{n-1}, e^n) - a(v^n, e^n) - \tau b(v^n, e^n) - a(w^n, e^n) - \tau b(w^n, e^n).
\end{aligned}$$

For $v^n \in V_H^{\text{ms}}$, we have due to the orthogonality and (4.5)

$$\begin{aligned}
a(v^n, e^n) + \tau b(v^n, e^n) &= a(v^n, \bar{v}^n - v^n) + \tau b(v^n, \bar{v}^n - v^n) \\
&= \tau(f^n, \bar{v}^n - v^n) + a(u_H^{n-1}, \bar{v}^n - v^n).
\end{aligned}$$

Similarly, for $w^n \in V^f$, we use the orthogonality and (4.6) to get

$$\begin{aligned}
a(w^n, e^n) + \tau b(w^n, e^n) &= a(w^n, \bar{w}^n - w^n) + \tau b(w^n, \bar{w}^n - w^n) \\
&= a(u_H^{n-1}, \bar{w}^n - w^n).
\end{aligned}$$

Hence,

$$\begin{aligned}
\|e^n\|_{a,b}^2 &= \tau(f^n, e^n) + a(u^{n-1}, e^n) - \tau(f^n, \bar{v}^n - v^n) - a(u_H^{n-1}, \bar{v}^n - v^n) \\
&\quad - a(u_H^{n-1}, \bar{w}^n - w^n) \\
&= \tau(f^n, \bar{w}^n - w^n) + a(u^{n-1} - u_H^{n-1}, e^n).
\end{aligned}$$

The first term can be bounded by using the interpolation operator $\mathfrak{I}_H$

$$\begin{aligned}\tau|(f^n,\bar{w}^n-w^n)| &\leq \tau\|f^n\|\|\bar{w}^n-w^n-\mathfrak{I}_H(\bar{w}^n-w^n)\|\\ &\leq CH\tau\|f^n\|\|\bar{w}^n-w^n\|_{H^1}\leq CH\tau\|f^n\|\|\bar{w}^n-w^n\|_{a,b}\\ &\leq CH\tau\|f^n\|\|e^n\|_{a,b}.\end{aligned}$$

For the second term we note that $u^{n-1}-u_H^{n-1}=e^{n-1}$ so that

$$\|e^n\|_{a,b}\leq CH\tau\|f^n\|+\|e^{n-1}\|_{a,b}.$$

Using this bound repeatedly and $e^0=0$, we get

$$\|e^n\|_{a,b}\leq CH\sum_{j=1}^{n}\tau\|f^j\|+\|e^0\|_{a,b}\leq CH\sum_{j=1}^{n}\tau\|f^j\|.$$

This concludes the proof since $\|e^n\|_{H^1}\leq C\|e^n\|_{a,b}$.

□

Next, a second derivative term is added to the equation, such that the complete non-homogeneous strongly damped wave equation is considered.

4.5 Completing the equation

Now that a promising method has been established for a simplified version of the strongly damped wave equation, we complete the equation by adding the second derivative to it. Thus, we consider the equation

$$\begin{aligned}\ddot{u}-\nabla\cdot(A\nabla\dot{u}+B\nabla u)&=f,\ \text{in } \Omega\times(0,T],\\ u&=0,\ \text{on } \Gamma\times(0,T],\end{aligned}$$

with initial data $u(\cdot,0)=u_0$ and $\dot{u}(\cdot,0)=v_0$. The weak form is to find $u\in V$ such that

$$(\ddot{u},z)+a(\dot{u},z)+b(u,z)=(f,z),\quad \forall z\in V.$$

As earlier, we apply a backward Euler scheme such that we seek $u^n\in V$ satisfying

$$(u^n,z)+\tau a(u^n,z)+\tau^2 b(u^n,z)=\tau^2(f,z)+\tau a(u^{n-1},z)+2(u^{n-1},z)-(u^{n-2},z)$$

for all $z\in V$ and $n=1,2,...,N$, with initial conditions u^{-1} and u^0. The method we present for solving this equation coincides with the one in previous section in the sense that the finescale system stays the same, and we merely alter the coarse system. From here on we let $w_{x,i}:=w_x^i$, such that the future notation will be more convenient. Now, for each coarse node $x\in\mathcal{N}_H$, let w_x^n be the solution to the equation

$$a(w_x^n,z)+\tau b(w_x^n,z)=a(w_x^{n-1},z), \tag{4.7}$$

for all $z\in V^f$, with initial condition $w_x^0=\lambda_x-\psi_x$. Then, if $\{\alpha_x^n\}_{x\in\mathcal{N}_H}$ denotes the set of coefficients for v^n, we have that

$$w^n=\sum_{i=1}^{n}\sum_{x\in\mathcal{N}_H}\alpha_x^{n-i}w_x^i. \tag{4.8}$$

Given w^n on this form, we define the following multiscale method for solving the strongly damped wave equation.

Definition 4.5.1 (LOD for the strongly damped wave equation)**.** The LOD approximation for the strongly damped wave equation is to find $v^n \in V_H^{\mathrm{ms}}$ such that

$$\begin{aligned}(v^n, z) + \tau a(v^n, z) + \tau^2 b(v^n, z) = \tau^2(f, z) + \tau a(v^{n-1}, z) + 2(v^{n-1}, z)\\ - (v^{n-2}, z) + \tau a(w^{n-1}, z)\end{aligned}$$

for all $z \in V_H^{\mathrm{ms}}$ and $n = 1, 2, ..., N$ with initial conditions v^{-1} and $v^0 = u_0 \in V_H^{\mathrm{ms}}$. The full solution is then constructed as $u_H^n = v^n + w^n$, where w^n is given by (4.8).

From here on, the function v^{-1} is used as initial condition. In each method it is chosen such that $(v^0 - v^{-1})/\tau$ is a suitable approximation of the initial condition on the derivative v_0.

There are several computational aspects to consider regarding this method. First of all, we notice that both the construction of w^n and the coarse Galerkin method require storage of all previous coarse solutions, as well as all previous finescale solutions $\{w_x^n\}$ for $x \in \mathcal{N}_H$ and $n = 1, 2, ..., N$. Furthermore, the method as currently defined evaluates the finescale solutions over the entire domain, and not restricted to patches as the standard LOD does. It is thus necessary to consider a spatial localization for the finescale systems, which is discussed in subsequent section.

4.6 Localization

In the method previously described, there are two types of finescale problems to consider. The first type we encounter is the Ritz projection that has to be solved for to construct the multiscale space V_H^{ms}. Following this, there is the requirement to solve for $\{w_x^n\}_{x\in\mathcal{N}_H}$ in every time step. Since these problems differ in some aspects, in specific the support of the right hand side functions, the localization procedure differs as well. For this reason, we denote from here on by $k_0 \in \mathbb{N} \cup \{0\}$ the localization parameter for the basis correction, and $k_1 \in \mathbb{N}$ the parameter for the solution correction.

4.6.1 Basis corrections

Consider the Ritz projection $\psi_x := \mathfrak{P}\lambda_x$ that solves

$$a(\psi_x, z) + \tau b(\psi_x, z) = a(\lambda_x, z) + \tau b(\lambda_x, z), \quad \forall z \in V^f.$$

The system looks much like the Ritz projection used for the standard LOD, but the projection is made with respect to the scalar product $\langle \cdot, \cdot \rangle_V$ instead of solely $a(\cdot, \cdot)$. Thus, analogously to the standard LOD, we define the element restricted Ritz projection $\mathfrak{P}_T$ such that $\mathfrak{P}_T v \in V^f$ solves

$$a(\mathfrak{P}_T v, z)_\Omega + \tau b(\mathfrak{P}_T v, z)_\Omega = a(v, z)_T + \tau b(v, z)_T, \quad \forall z \in V^f,$$

for an element $T \in \mathcal{T}_H$. Using this, we may construct the global correction as the sum

$$\mathfrak{P} v = \sum_{T\in\mathcal{T}_H} \mathfrak{P}_T v.$$

For $k_0 \in \mathbb{N} \cup \{0\}$, we may also restrict the projection to element based patches by letting $\mathfrak{P}_{k_0,T} : V \to V^f(U_{k_0}(T))$ be such that $\mathfrak{P}_{k_0,T}v \in V^f(U_{k_0}(T))$ solves

$$a(\mathfrak{P}_{k_0,T}v, z)_{U_{k_0}(T)} + \tau b(\mathfrak{P}_{k_0,T}v, z)_{U_{k_0}(T)} = a(v, z)_T + \tau b(v, z)_T, \quad \forall z \in V^f(U_{k_0}(T)),$$

and by summation we get the corresponding global projection as

$$\mathfrak{P}_{k_0}v = \sum_{T \in \mathcal{T}_H} \mathfrak{P}_{k_0,T}v.$$

We may now construct our localized multiscale space as

$$V_{H,k}^{\mathrm{ms}} := V_H - \mathfrak{P}_{k_0}V_H,$$

spanned by $\{\lambda_x - \mathfrak{P}_{k_0}\lambda_x\}_{x \in \mathcal{N}_H}$. With this localized space, we arrive the at following partially localized multiscale method.

Definition 4.6.1 (Partially localized LOD for strongly damped wave equation)**.** The partially localized LOD approximation for the strongly damped wave equation is to find $v_k^n \in V_{H,k}^{\mathrm{ms}}$ such that

$$\begin{aligned}(v_k^n, z) + \tau a(v_k^n, z) + \tau^2 b(v_k^n, z) = \tau^2(f, z) + \tau a(v_k^{n-1}, z) + 2(v_k^{n-1}, z)\\ - (v_k^{n-2}, z) + \tau a(w^{n-1}, z)\end{aligned}$$

for all $z \in V_{H,k}^{\mathrm{ms}}$ and $n = 1, 2, ..., N$ with initial conditions v^{-1} and $v^0 = u_0 \in V_{H,k}^{\mathrm{ms}}$. The full solution is then constructed as $u_{H,K}^n = v_k^n + w^n$, where w^n is given by (4.8).

The notation of K is used such that the partially and completely localized methods can be distinguished. It remains to introduce a localization procedure for the calculations of $\{w_x^n\}_{x \in \mathcal{N}_H}$.

4.6.2 Solution corrections

Now consider the problem to find w_x^n such that

$$a(w_x^n, z) + \tau b(w_x^n, z) = a(w_x^{n-1}, z), \quad \forall z \in V^f$$

for every $x \in \mathcal{N}_H$ and $n = 1, 2, ..., N$ with initial condition $w_x^0 = \lambda_x - \mathfrak{P}_{k_0}\lambda_x \in V_{H,k}^{\mathrm{ms}}$. This system differs from the correction problem in the sense that the support of the right hand side function is larger than earlier due to the presence of $\mathfrak{P}_{k_0}\lambda_x$. Consequently, we consider a node localization in this case. That is, for $k_1 \in \mathbb{N}$ and every node $x \in \mathcal{N}_H$, we seek to find $w_{x,k}^n \in V^f(\omega_{x,k_1})$ which solves

$$a(w_{x,k}^n, z) + \tau b(w_{x,k}^n, z) = a(w_{x,k}^{n-1}, z), \quad \forall z \in V^f(\omega_{x,k_1}). \tag{4.9}$$

The localized finescale solution w_k^n is then given by

$$w_k^n = \sum_{i=1}^{n} \sum_{x \in \mathcal{N}_H} \alpha_x^{n-i} w_{x,k}^i. \tag{4.10}$$

We may now define the completely spatially localized multiscale method.

Definition 4.6.2 (Spatially localized LOD for the strongly damped wave equation)**.** The spatially localized LOD approximation for the strongly damped wave equation is to find $v_k^n \in V_{H,k}^{\text{ms}}$ such that

$$\begin{aligned}(v_k^n, z) + \tau a(v_k^n, z) + \tau^2 b(v_k^n, z) = \tau^2(f, z) + \tau a(v_k^{n-1}, z) + 2(v_k^{n-1}, z)\\ - (v_k^{n-2}, z) + \tau a(w_k^{n-1}, z)\end{aligned}$$

for all $z \in V_{H,k}^{\text{ms}}$ and $n = 1, 2, ..., N$ with initial conditions v^{-1} and $v^0 = u_0 \in V_{H,k}^{\text{ms}}$. The full solution is then constructed as $u_{H,k}^n = v_k^n + w_k^n$, where w_k^n is given by (4.10).

4.6.3 Temporal localization

The spatial localization procedure previously introduced is crucial for the method's efficiency. However, the finescale solution corrections still need to be evaluated in each time step, implying large computational complexity of the method. For this purpose, we wish to localize the calculations in time, such that the fine system need not be solved in every time step.

First, consider the solution corrections w_x^n given as the solution to (4.7). The system they solve resembles a parabolic type equation with no source term. That is, the solution will decay exponentially until it is completely vanished. An example of how w_x^n vanish with increasing n can be seen in Figure 4.1.

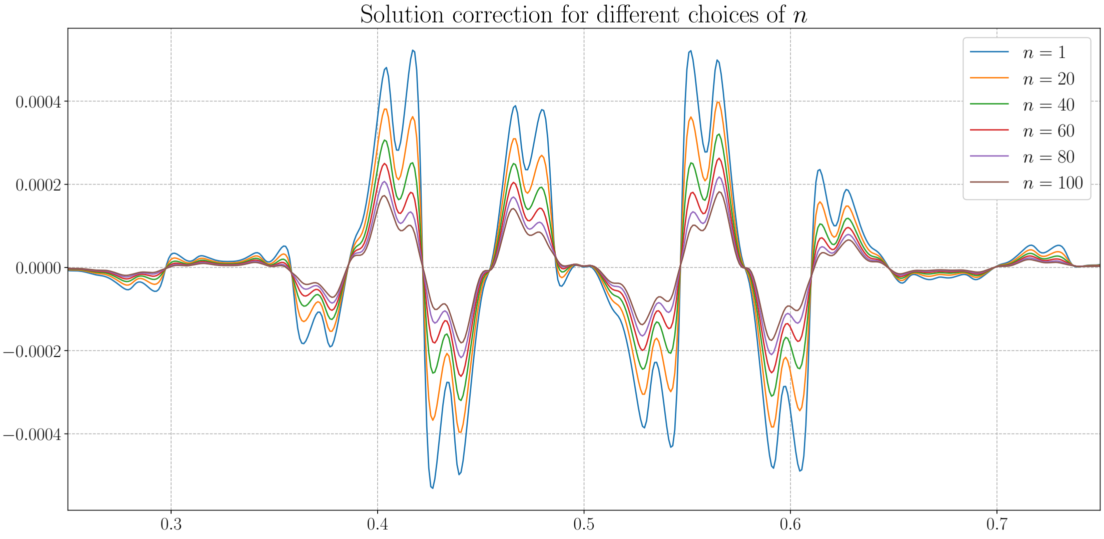


Figure 4.1: The behaviour of the solution corrections w_x^n with increasing n.

Due to the fact that the solution corrections vanish as the time passes, there is no need to evaluate them after a specific t (dependent of H) since they will be smaller than the error due to H. Thus, we consider the system to find $w_{x,k,l}^n$ that solves the spatially localized system (4.9) for all $n = 1, 2, ..., l$ for some $l < N$. Then, the spatially and temporally localized correction will be given by

$$w_{k,l}^n = \begin{cases} \sum_{i=1}^n \sum_{x \in \mathcal{N}_H} \alpha_x^{n-i} w_{x,k,l}^i, & \text{if } n < l, \\ \sum_{i=1}^l \sum_{x \in \mathcal{N}_H} \alpha_x^{n-i} w_{x,k,l}^i, & \text{if } l \leq n. \end{cases} \tag{4.11}$$

We may now introduce a multiscale method that has been localized both in time and space.

Definition 4.6.3 (Spatially and temporally localized LOD for the strongly damped wave equation)**.** The spatially and temporally localized LOD approximation for the strongly damped wave equation is to find $v_{k,l}^n \in V_{H,k}^{\mathrm{ms}}$ such that

$$\begin{aligned}(v_{k,l}^n, z) + \tau a(v_{k,l}^n, z) + \tau^2 b(v_{k,l}^n, z) = \tau^2 (f, z) + \tau a(v_{k,l}^{n-1}, z) + 2(v_{k,l}^{n-1}, z) \\ - (v_{k,l}^{n-2}, z) + \tau a(w_{k,l}^{n-1}, z)\end{aligned}$$

for all $z \in V_{H,k}^{\mathrm{ms}}$ and $n = 1, 2, ..., N$ with initial conditions v^{-1} and $v^0 = u_0 \in V_{H,k}^{\mathrm{ms}}$. The full solution is then constructed as $u_{H,k,l}^n = v_{k,l}^n + w_{k,l}^n$, where $w_{k,l}^n$ is given by (4.11).

5

Implementation

The previous chapters of this thesis have discussed the analytic approaches to methods such as the standard FEM, LOD, PG-LOD, and at last the new LOD adapted for the strongly damped wave equation. In this chapter, we consider the implementational aspects necessary to create and present numerical examples for each method's performance. First, the theoretical considerations are presented, which mainly include the discretization of the domain, as well as the multiscale spaces used for each method. Following this, the implementation details which include a presentation of the code and algorithms that have been used for the numerical examples are discussed. This mainly includes the Python module `gridlod`, as well as the extensive files necessary for the new LOD method.

5.1 Theoretical considerations

The theoretical part needed for the implementation mainly includes the discretizaion. First we consider the discretization of the domain and the necessary spaces, followed by a discussion of how the resulting LOD and PG-LOD methods become. Furthermore, the same discretization procedure is applied to the new LOD method that was presented in Chapter 4.

5.1.1 Discretization

For the implementation of the methods presented in this thesis, we require a discretized setting of the solution space V. For this purpose, consider shape regular meshes $\mathcal{T}_H$ and $\mathcal{T}_h$ ($h < H$), such that the fine mesh $\mathcal{T}_h$ is a refinement of the coarse mesh $\mathcal{T}_H$. That is, an element of $\mathcal{T}_H$ is the union of finitely many elements in $\mathcal{T}_h$. Examples of a coarse and fine mesh respectively can be seen in Figure 5.1. Furthermore, we assume h to be small enough such that the fine mesh may capture the solution's finescale features.

Denote by $\mathcal{N}_H$ and $\mathcal{N}_h$ the set of interior nodes in the coarse and fine mesh respectively, and the sets' sizes by $|\mathcal{N}_H| = N_H$ and $|\mathcal{N}_h| = N_h$. We let $\mathcal{T}_H$ and $\mathcal{T}_h$ be represented by quadrilateral meshes, as in Figure 5.1. For this purpose, in a similar way as in Section 2.3.1, we use the space $\mathcal{Q}_1(\mathcal{T}_h)$ to define the discretized space

$$V_h := V \cap \mathcal{Q}_1(\mathcal{T}_h).$$

Since V_h is sufficiently refined to capture all finescale features, we may use this space to evaluate a reference solution $u_h \in V_h$ with the usual FEM. This procedure

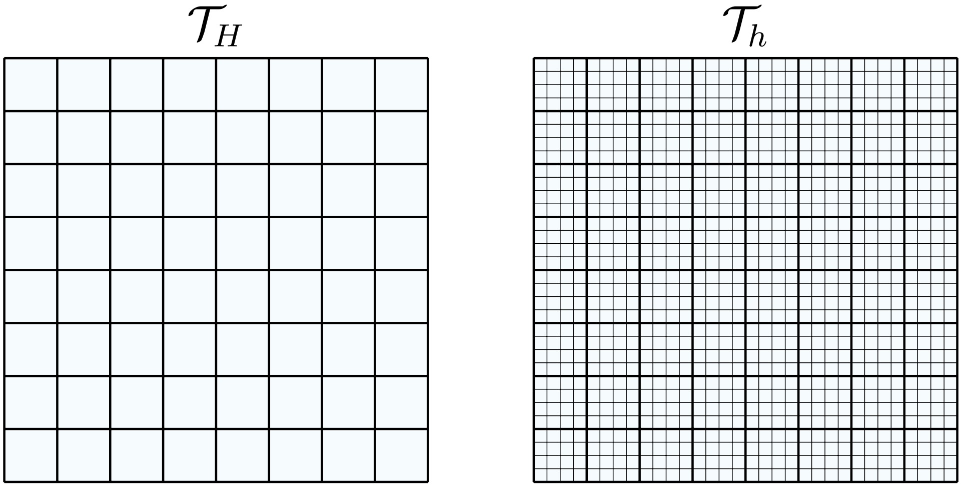


Figure 5.1: The unit square represented by a coarse and a fine quadrilateral mesh respectively, with $1/H = 8$ and $1/h = 32$.

is however computationally expensive and may further imply memory issues. Thus, for methods such as the LOD, the fine mesh is solely used to evaluate the finescale correction for each element $T \in \mathcal{T}_H$ (when using an element based localization). By only using this mesh for patch localized problems, the issue of computational complexity is bypassed. These problems furthermore require a discretized version of the finescale space V^f defined by

$$V_h^f := V^f \cap V_h,$$

and in particular, its element localized counterpart

$$V_h^f(U_k(T)) := V^f(U_k(T)) \cap V_h.$$

Using this discretized setting, the implementational aspects of both the standard LOD and its Petrov-Galerkin analogue may be formulated.

5.1.2 Standard LOD

The main calculations of the standard LOD is much like the standard Galerkin method, with the exception that we require the incorporation of the finescale features into the finite element basis. Using the localized finescale space $V_h^f(U_k(T))$, we may for every function $v_H \in V_H$ and every $T \in \mathcal{T}_H$ compute these finescale corrections with a Galerkin method as the solution $\mathcal{Q}_{k,T}^h v_H \in V_h^f(U_k(T))$ to the problem

$$a(\mathcal{Q}_{k,T}^h v_H, v^f)_{U_k(T)} = a(v_H, v^f)_T, \quad \forall v^f \in V_h^f(U_k(T)).$$

In a similar way as shown in Section 3.4.2, we may then sum these solutions to construct the full correction as

$$\mathcal{Q}_k^h v_H = \sum_{T \in \mathcal{T}_H} \mathcal{Q}_{k,T}^h v_H.$$

Consequently, we may define the localized and discretized multiscale space $V_{H,k}^{\mathrm{ms},h}$ by

$$V_{H,k}^{\mathrm{ms},h} := V_H - \mathcal{Q}_k^h V_H.$$

Thus, the resulting semi-discrete method is to find $u_{H,k}^{\mathrm{ms},h} \in V_{H,k}^{\mathrm{ms},h}$ such that

$$(\ddot{u}_{H,k}^{\mathrm{ms},h}, v) + a(\dot{u}_{H,k}^{\mathrm{ms},h}, v) + b(u_{H,k}^{\mathrm{ms},h}, v) = (f, v), \quad \forall v \in V_{H,k}^{\mathrm{ms},h},$$

for $k \in \mathbb{N} \cup \{0\}$. Furthermore, by applying a backward Euler scheme to this method, we obtain the completely discrete method to find $U_{H,k}^{n,\mathrm{ms},h} \in V_{H,k}^{\mathrm{ms},h}$ such that

$$(\bar{\partial}_t \bar{\partial}_t U_{H,k}^{n,\mathrm{ms},h}, v) + a(\bar{\partial}_t U_{H,k}^{n,\mathrm{ms},h}, v) + b(U_{H,k}^{n,\mathrm{ms},h}, v) = (f^n, v), \quad \forall v \in V_{H,k}^{\mathrm{ms},h}, \tag{5.1}$$

with initial conditions $U_{H,k}^{-1,\mathrm{ms},h}$ and $U_{H,k}^{0,\mathrm{ms},h} \in V_{H,k}^{\mathrm{ms},h}$ for $k \in \mathbb{N} \cup \{0\}$ and $n = 1, ..., N$.

The system in (5.1) can be computed by usual Galerkin method by first constructing corresponding mass and stiffness matrices, as well as the load vector. Let $x_m, x_n \in \mathcal{N}_H$ denote two coarse nodes. After the incorporation of their corresponding finescale correctors $\mathcal{Q}_k^h \lambda_m$ and $\mathcal{Q}_k^h \lambda_n$, the mass matrix $\mathcal{M}_H^{\mathrm{ms},k} \in \mathbb{R}^{N_H \times N_H}$, the two stiffness matrices $\mathcal{S}_H^{\mathrm{ms},k}, \mathcal{K}_H^{\mathrm{ms},k} \in \mathbb{R}^{N_H \times N_H}$ and the load vector $\mathcal{F}_H^{\mathrm{ms},k} \in \mathbb{R}^{N_H}$ have entries of the form

$$\begin{aligned}
\mathcal{M}_H^{\mathrm{ms},k}[m][n] &= (\lambda_n - \mathcal{Q}_k^h \lambda_n, \lambda_m - \mathcal{Q}_k^h \lambda_m),\\
\mathcal{S}_H^{\mathrm{ms},k}[m][n] &= b(\lambda_n - \mathcal{Q}_k^h \lambda_n, \lambda_m - \mathcal{Q}_k^h \lambda_m),\\
\mathcal{K}_H^{\mathrm{ms},k}[m][n] &= a(\lambda_n - \mathcal{Q}_k^h \lambda_n, \lambda_m - \mathcal{Q}_k^h \lambda_m),\\
\mathcal{F}_H^{\mathrm{ms},k}[m] &= (f, \lambda_m - \mathcal{Q}_k^h \lambda_m).
\end{aligned}$$

First of all, note that the entries run over all coarse nodes, such that once these matrices are obtained it is a computationally cheap procedure to solve the final system. Furthermore, in the standard FEM, these matrices are sparse since the basis functions $\{\lambda_x\}_{x \in \mathcal{N}_H}$ each have compact support solely on neighbouring elements. However, since the corrections $\mathcal{Q}_k^h \lambda_x$ are solved for in systems restricted to patches, their resulting support will also be limited to these. Thus, in every case when $\mathrm{supp}(\mathcal{Q}_k^h \lambda_m) \cap \mathrm{supp}(\mathcal{Q}_k^h \lambda_n) = \emptyset$, the corresponding entry in the matrices will be 0. Consequently, the matrices for this method will also be sparse, as long as k is chosen rather small. An example of how the support of the basis correctors may or may not overlap can be seen in Figure 5.2.

With all matrices defined, the system

$$\begin{aligned}
\left(\mathcal{M}_H^{\mathrm{ms},k} + \tau \mathcal{K}_H^{\mathrm{ms},k} + \tau^2 \mathcal{S}_H^{\mathrm{ms},k}\right) \cdot \mathcal{U}_H^{n,\mathrm{ms}} &= \tau^2 \mathcal{F}_H^{\mathrm{ms},k} + \tau \mathcal{K}_H^{\mathrm{ms},k} \cdot \mathcal{U}_H^{n-1,\mathrm{ms}}\\
&\quad + 2\mathcal{M}_H^{\mathrm{ms},k} \cdot \mathcal{U}_H^{n-1,\mathrm{ms}} - \mathcal{M}_H^{\mathrm{ms},k} \cdot \mathcal{U}_H^{n-2,\mathrm{ms}}
\end{aligned}$$

can be solved for, where $\mathcal{U}_H^{n,\mathrm{ms}} \in \mathbb{R}^{N_H}$ is of the form

$$\mathcal{U}_H^{n,\mathrm{ms}} = \begin{bmatrix} \alpha_0^n \\ \alpha_1^n \\ \cdot \\ \cdot \\ \cdot \\ \alpha_{N_H-1}^n \end{bmatrix}.$$

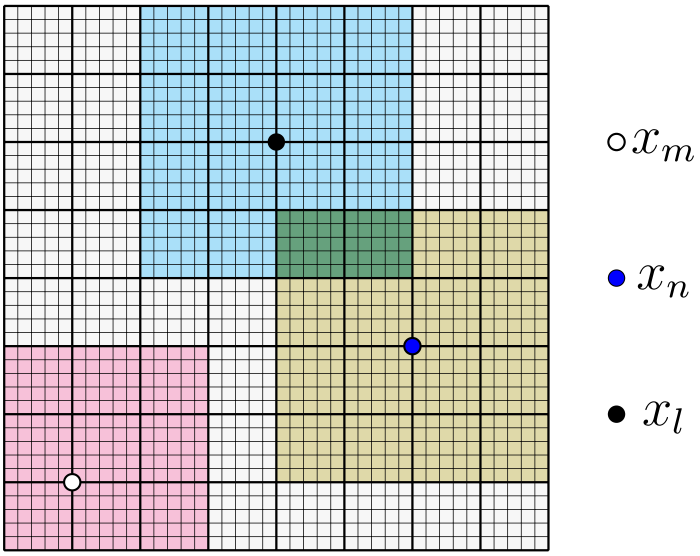


Figure 5.2: Illustration of the support for different basis corrections ϕ_m, ϕ_n and ϕ_l corresponding to the nodes x_m, x_n, and x_l, for $k = 1$. Here it is seen how the support between ϕ_n and ϕ_l overlap, while the intersections $\text{supp}(\phi_n) \cap \text{supp}(\phi_m)$ and $\text{supp}(\phi_l) \cap \text{supp}(\phi_m)$ both result in the empty set.

The final LOD approximation is then constructed as

$$U_{H,k}^{n,\text{ms},h} = \sum_{m=0}^{N_H - 1} \mathcal{U}_H^{n,\text{ms}}[m](\lambda_m - \mathcal{Q}_k^h \lambda_m).$$

Since the LOD uses the multiscale space as both trial and test space, communication between all basis correctors is required. As described in Section 3.6.2, by considering the PG-LOD as alternative method, the same communication is no longer necessary, and thus it gains an advantage in terms of computational complexity.

5.1.3 Petrov-Galerkin LOD

For the PG-LOD method, we consider a similar setting as the one in previous section. The main difference between the two methods is the choice of test space, which in the PG-LOD is chosen as the standard FE-space V_H. Thus, we seek to find $U_{H,k}^{n,\text{PG},h} \in V_{H,k}^{\text{ms},h}$ such that

$$(\bar{\partial}_t \bar{\partial}_t U_{H,k}^{n,\text{PG},h}, v) + a(\bar{\partial}_t U_{H,k}^{n,\text{PG},h}, v) + b(U_{H,k}^{n,\text{PG},h}, v) = (f^n, v), \quad \forall v \in V_H,$$

with initial condition $U_{H,k}^{-1,\text{PG},h}$ and $U_{H,k}^{0,\text{PG},h} \in V_{H,k}^{\text{ms},h}$ for $k \in \mathbb{N} \cup \{0\}$ and $n = 1, ..., N$. Consequently, the entries of the matrices and the load vector instead become

$$\begin{aligned}
\mathcal{M}_H^{\text{PG},k}[m][n] &= (\lambda_n - \mathcal{Q}_k^h \lambda_n, \lambda_m), \\
\mathcal{S}_H^{\text{PG},k}[m][n] &= b(\lambda_n - \mathcal{Q}_k^h \lambda_n, \lambda_m), \\
\mathcal{K}_H^{\text{PG},k}[m][n] &= a(\lambda_n - \mathcal{Q}_k^h \lambda_n, \lambda_m), \\
\mathcal{F}_H^{\text{PG},k}[m] &= (f, \lambda_m),
\end{aligned}$$

where, just as for the LOD, the entries run over the coarse nodes. In these entries, it is clearly seen how the correctors require no communcation with each other, and thus the cost of assembling e.g. $\mathcal{S}_H^{\mathrm{PG},k}$ is cheaper than that of $\mathcal{S}_H^{\mathrm{ms},k}$. With the matrices defined, we get analogously to previous section, the system

$$\begin{aligned}\left(\mathcal{M}_H^{\mathrm{PG},k} + \tau\mathcal{K}_H^{\mathrm{PG},k} + \tau^2\mathcal{S}_H^{\mathrm{PG},k}\right)\cdot \mathcal{U}_H^{n,\mathrm{PG}} &= \tau^2\mathcal{F}_H^{\mathrm{PG},k} + \tau\mathcal{K}_H^{\mathrm{PG},k}\cdot\mathcal{U}_H^{n-1,\mathrm{PG}} \\ &\quad +2\mathcal{M}_H^{\mathrm{PG},k}\cdot\mathcal{U}_H^{n-1,\mathrm{PG}} - \mathcal{M}_H^{\mathrm{PG},k}\cdot\mathcal{U}_H^{n-2,\mathrm{PG}},\end{aligned}$$

with $\mathcal{U}_H^{n,\mathrm{PG}}$ of the same form as $\mathcal{U}_H^{n,\mathrm{ms}}$. The final PG-LOD approximation is then constructed as

$$U_{H,k}^{n,\mathrm{PG},h} = \sum_{m=0}^{N_H-1}\mathcal{U}_H^{n,\mathrm{PG}}[m](\lambda_m - \mathcal{Q}_k^h\lambda_m).$$

5.1.4 Strongly damped wave equation

This chapter has so far presented the theoretical considerations regarding the standard LOD and PG-LOD. A similar procedure is now presented for the new LOD method presented in Chapter 4. First, the decomposition of the solution to the simplified equation is discussed. This is followed by every step until we consider the localized method for the complete strongly damped wave equation.

5.1.4.1 Decomposition of solution

For the decomposition, we consider the Ritz projection $\mathfrak{P}^h : V_H \to V_h^f$ such that $\mathfrak{P}^h v_H \in V_h^f$ solves the system

$$a(\mathfrak{P}^h v_H, z) + \tau b(\mathfrak{P}^h v_H, z) = a(v_H, z) + \tau b(v_H, z), \quad \forall z \in V_h^f.$$

The multiscale space we consider is then given by

$$V_H^{\mathrm{ms},h} := V_H - \mathfrak{P}^h V_H = \mathrm{Span}(\{\lambda_x - \mathfrak{P}^h\lambda_x\}_{x\in\mathcal{N}_H}).$$

Then, with respect to the scalar product $\langle\cdot,\cdot\rangle_V$ defined in (4.1), we have the orthogonal decomposition $V_h = V_H^{\mathrm{ms},h} \oplus_V V_h^f$. By the method described in Section 4.3, we seek to for each n find $v^n \in V_H^{\mathrm{ms},h}$ and $w^n \in V_h^f$, such that we may write $u^n \in V_h$ as $u^n = v^n + w^n$.

Recall that for each n we seek to solve the finescale problem

$$a(w_x^n, z) + \tau b(w_x^n, z) = a(w_x^{n-1}, z), \quad \forall z \in V_h^f, \tag{5.2}$$

for each coarse node $x \in \mathcal{N}_H$, with initial value $w_x^0 = \lambda_x - \mathfrak{P}^h\lambda_x$. Then, if we let $\boldsymbol{\alpha}_H^n$ be the coefficient vector for the coarse solution v^n, i.e.

$$\boldsymbol{\alpha}_H^n = \begin{bmatrix}\alpha_0^n \\ \alpha_1^n \\ \cdot \\ \cdot \\ \cdot \\ \alpha_{N_H-1}^n\end{bmatrix},$$

we can construct w^n by

$$w^n = \sum_{i=1}^{n} \sum_{m=0}^{N_H-1} \boldsymbol{\alpha}_H^{n-i}[m] w_x^i. \tag{5.3}$$

For each n, we also seek the coarse solution v^n to

$$a(v^n, z) + \tau b(v^n, z) = a(v^{n-1}, z) + a(w^{n-1}, z), \quad \forall z \in V_H^{\mathrm{ms},h}. \tag{5.4}$$

To solve this system, we incorporate the corrections into the stiffness matrices used for this problem. The standard LOD setting is considered in the sense that $V_H^{\mathrm{ms},h}$ is used as both trial and test space. Thus, the entries of the matrices are of the form

$$\begin{aligned} \mathcal{S}_H^{\mathrm{ms},h}[m][n] &:= b(\lambda_n - \mathfrak{P}^h \lambda_n, \lambda_m - \mathfrak{P}^h \lambda_m), \\ \mathcal{K}_H^{\mathrm{ms},h}[m][n] &:= a(\lambda_n - \mathfrak{P}^h \lambda_n, \lambda_m - \mathfrak{P}^h \lambda_m). \end{aligned} \tag{5.5}$$

We furthermore introduce the multiscale matrix $\mathcal{R}_{H,i}^{\mathrm{ms},h}$ that appear due to the last term in (5.4). The entries for this matrix are of the form

$$\mathcal{R}_{H,i}^{\mathrm{ms},h}[m][n] := a(w_n^i, \lambda_m - \mathfrak{P}^h \lambda_m). \tag{5.6}$$

Consequently, the system we are to solve is

$$(\mathcal{K}_H^{\mathrm{ms},h} + \tau \mathcal{S}_H^{\mathrm{ms},h}) \cdot \boldsymbol{\alpha}_H^n = \mathcal{K}_H^{\mathrm{ms},h} \cdot \boldsymbol{\alpha}_H^{n-1} + \sum_{i=1}^{n-1} \mathcal{R}_{H,i}^{\mathrm{ms},h} \cdot \boldsymbol{\alpha}_H^{n-i}.$$

The coarse solution v^n is then constructed as

$$v^n = \sum_{m=0}^{N_H-1} \boldsymbol{\alpha}_H^n[m](\lambda_m - \mathfrak{P}^h \lambda_m). \tag{5.7}$$

Finally, the full solution $u^n \in V_h$ is acquired by $u^n = v^n + w^n$. As described in Section 4.3, this procedure is solely a decomposition of the solution, and will thus be exact in each time step. A numerical example of this method is presented in Section 6.1. Next, we add a source term to the problem, such that the solution no longer will be exact, but rather depend on the mesh width H.

5.1.4.2 Adding a source term

It was seen in Section 4.4 that when a source term $f \in L^2$ is added to the equation, the only difference in the method appears in the coarse system. That is, w^n is still given by (5.3). The coarse Galerkin system is altered in the sense that the term (f, z) has to be taken into consideration. Thus, let $\mathcal{S}_H^{\mathrm{ms},h}$, $\mathcal{K}_H^{\mathrm{ms},h}$ and $\mathcal{R}_H^{\mathrm{ms},h}$ be given by (5.5)-(5.6), and define the multiscale load vector with entries

$$\mathcal{F}_H^{\mathrm{ms},h}[m] := (f, \lambda_m - \mathfrak{P}^h \lambda_m).$$

We end up with the linear system

$$(\mathcal{K}_H^{\mathrm{ms},h} + \tau \mathcal{S}_H^{\mathrm{ms},h}) \cdot \boldsymbol{\alpha}_H^n = \tau \mathcal{F}_H^{\mathrm{ms},h} + \mathcal{K}_H^{\mathrm{ms},h} \cdot \boldsymbol{\alpha}_H^{n-1} + \sum_{i=1}^{n-1} \mathcal{R}_{H,i}^{\mathrm{ms},h} \cdot \boldsymbol{\alpha}_H^{n-i}$$

The coarse solution v^n is then constructed as in (5.7), and the full solution by $u_H^n = v^n + w^n$. A numerical example for this method with its results is presented in Section 6.2. Next, we discuss the considerations to be made for the complete strongly damped wave equation.

5.1.4.3 Complete equation

Similarly as when the source term was added, the finescale system remains the same while we adapt the coarse system to work with the full equation. Let $\mathcal{S}_H^{\mathrm{ms},h}$, $\mathcal{K}_H^{\mathrm{ms},h}$, $\mathcal{R}_H^{\mathrm{ms},h}$ and $\mathcal{F}_H^{\mathrm{ms},h}$ be defined as earlier, and introduce the multiscale mass matrix $\mathcal{M}_H^{\mathrm{ms},h}$ with entries

$$\mathcal{M}_H^{\mathrm{ms},h}[m][n] := (\lambda_n - \mathfrak{P}^h\lambda_n, \lambda_m - \mathfrak{P}^h\lambda_m).$$

The linear system to solve becomes

$$\begin{aligned}(\mathcal{M}_H^{\mathrm{ms},h} + \tau\mathcal{K}_H^{\mathrm{ms},h} + \tau^2\mathcal{S}_H^{\mathrm{ms},h}) \cdot \boldsymbol{\alpha}_H^n &= \tau^2\mathcal{F}_H^{\mathrm{ms},h} + \tau\mathcal{K}_H^{\mathrm{ms},h} \cdot \boldsymbol{\alpha}_H^{n-1} + 2\mathcal{M}_H^{\mathrm{ms},h} \cdot \boldsymbol{\alpha}_H^{n-1} \\ &\quad - \mathcal{M}_H^{\mathrm{ms},h} \cdot \boldsymbol{\alpha}_H^{n-2} + \tau\sum_{i=1}^{n-1} \mathcal{R}_{H,i}^{\mathrm{ms},h} \cdot \boldsymbol{\alpha}_H^{n-i},\end{aligned} \tag{5.8}$$

and the coarse solution is given by (5.7). Finally, we have the full solution by $u_H^n = v^n + w^n$. In Section 6.3, numerical examples of this method are presented with corresponding results. Just as with the standard LOD, we further wish to not solve the finescale problems on the full domain, but rather on localized patches. The considerations to be made with these localization procedures are discussed.

5.1.4.4 Spatial localization

For $k_0 \in \mathbb{N} \cup \{0\}$, consider the element based localized Ritz projection $\mathfrak{P}_{k_0,T}^h : V_h \to V_h^f(U_{k_0}(T))$ such that $\mathfrak{P}_{k_0,T}^h v$ solves

$$\langle \mathfrak{P}_{k_0,T}^h v, z\rangle_{V,U_{k_0}(T)} = \langle v, z\rangle_{V,T}, \quad \forall z \in V_h^f(U_{k_0}(T)), \tag{5.9}$$

with corresponding global projection

$$\mathfrak{P}_{k_0}^h v = \sum_{T\in\mathcal{T}_H} \mathfrak{P}_{k_0,T}^h v.$$

We get the localized and discretized multiscale space

$$V_{H,k}^{\mathrm{ms},h} := V_H - \mathfrak{P}_{k_0}^h V_H = \mathrm{Span}(\{\lambda_x - \mathfrak{P}_{k_0}^h \lambda_x\}_{x\in\mathcal{N}_H}).$$

Using this space, the localized multiscale matrices and vector for our method have entries on the form

$$\begin{aligned}\mathcal{S}_{H,k}^{\mathrm{ms},h}[m][n] &:= b(\lambda_n - \mathfrak{P}_{k_0}^h\lambda_n, \lambda_m - \mathfrak{P}_{k_0}^h\lambda_m),\\ \mathcal{K}_{H,k}^{\mathrm{ms},h}[m][n] &:= a(\lambda_n - \mathfrak{P}_{k_0}^h\lambda_n, \lambda_m - \mathfrak{P}_{k_0}^h\lambda_m),\\ \mathcal{M}_{H,k}^{\mathrm{ms},h}[m][n] &:= (\lambda_n - \mathfrak{P}_{k_0}^h\lambda_n, \lambda_m - \mathfrak{P}_{k_0}^h\lambda_m),\\ \mathcal{F}_{H,k}^{\mathrm{ms},h}[m] &:= (f, \lambda_m - \mathfrak{P}_{k_0}^h\lambda_m).\end{aligned} \tag{5.10}$$

Now let $k_1 \in \mathbb{N}$ be the parameter corresponding to the node based localized finescale problem in (4.9). Then the solution w_k^n is given by

$$w_k^n = \sum_{i=1}^{n} \sum_{m=0}^{N_H-1} \boldsymbol{\alpha}_{H,k}^{n-i}[m] w_{m,k_1}^i,$$

where $\boldsymbol{\alpha}^n_{H,k}$ solves the coarse scale equation in the n'th time step. Lastly, define the localized matrix $\mathcal{R}^{\text{ms},h}_{H,i,k}$ by its entries

$$\mathcal{R}^{\text{ms},h}_{H,i,k}[m][n] := a(w^i_{n,k_1}, \lambda_m - \mathfrak{P}^h_{k_0}\lambda_m). \tag{5.11}$$

The localized linear system to solve is similar to the one seen in (5.8), but with the matrices replaced by their localized analogues. The solution is then computed as $u^n_{H,k} = v^n_k + w^n_k$. In Section 6.4, numerical examples that show the performance depending on different choices of k_0 and k_1 are presented.

Once the spatial localization is done, we further wish to improve the method's efficiency in the temporal domain. Consequently, this yields a faster method that is less prone to memory issues.

5.1.4.5 Temporal localization

For this method, let the localized multiscale load vector, mass matrix and stiffness matrices be defined as in (5.10) and (5.11). Furthermore, for $l < N$, let $w^n_{x,k_1,l}$ be the solution to the same system as w^n_{x,k_1}, but solely for all $n \leq l$. The fine solution correction is then constructed as

$$w^n_{k,l} = \begin{cases} \sum_{i=1}^{n} \sum_{m=0}^{N_H-1} \boldsymbol{\alpha}^{n-i}_{H,k}[m] w^i_{m,k_1,l}, & \text{if } n \leq l, \\ \sum_{i=1}^{l} \sum_{m=0}^{N_H-1} \boldsymbol{\alpha}^{n-i}_{H,k}[m] w^i_{m,k_1,l}, & \text{if } l < n. \end{cases}$$

The coarse system thus resembles the one in previous section, but with the sum

$$\sum_{i=1}^{n-1} \mathcal{R}^{\text{ms},h}_{H,i,k} \boldsymbol{\alpha}^{n-i}_{H},$$

replaced such that summation index only sums up until l when $l < n$. Note that the previous requirement to store all coarse solutions is now altered, such that only the l latest solutions are needed. Consequently, the complexity due to memory issues etc., is reduced. Moreover, the method is significantly faster when considering systems where the amount of time steps are larger than l, since no finescale calculations are done for all $n > l$. Numerical examples illustrating the performance of this method can be seen in Section 6.5.

5.2 Implementational details

The last part of this chapter focuses on the practical details of the implementation. For this thesis, all code is written in and run in `Python 2.7`. All simulations are either done in one or two dimensions, with domains $\Omega_1 = [0,1]$ and $\Omega_2 = [0,1] \times [0,1]$ respectively. The code is based on the Python module named `gridlod`, created and used by Hellman in [15]. Furthermore, extensive files written to adapt the code to the case when considering the strongly damped wave equation are presented.

5.2.1 The gridlod module

The `gridlod` module is a Python code created for the purpose of computing element localized correctors used in the LOD for numerical homogenization. The code can be applied for any number of dimensions d, but is restricted to the unit hypercube $[0,1]^d$. On this domain, we use a structured grid as coarse mesh and another structured grid as fine mesh that is a refinement of the coarse mesh. The code is written primarily with memory usage in consideration, and for that purpose contains methods that are specific for the usage of PG-LOD. It should be noted though, that when these methods are not used in particular, a standard LOD method is considered. Furthermore, the code distinguishes between coarse and finescale information, such that the finescale information can be discarded early to reduce the memory usage.

The information of the coarse and fine grid, along with the boundary conditions, are stored in the so called `world`-class. In this class, the variables `NWorldCoarse` and `NWorldFine` describe the coarse and fine grid respectively, and are represented by arrays. For example, in the two dimensional case, by letting `NWorldFine=numpy.array([256,256])`, we create a fine mesh with 256 times 256 square elements in the unit square $[0,1]^2$. The fine mesh is further required to be a refinement of the coarse mesh. This is made sure of by having the elementwise ratio `NCoarseElement:=NWorldFine/NWorldCoarse` be all integers. The code also stores data linearly, with first index corresponding to first node, e.g. if the coefficient is stored in the variable `aFine`, then `aFine[0]` is the coefficient at fine index $(0,0)$, `aFine[1]` at $(1,0)$, etc.

The `gridlod` code contains several files of importance. Their main features can be summarized as:

`fem.py`

Contains the code used to assemble the finite element matrices. For example, to assemble the standard FEM stiffness matrix, one may use `fem.assemblePatchMatrix(NWorldFine,world.ALocFine,aFine)`, where `world.ALocFine` determines that its a stiffness matrix that is evaluated, and `aFine` contains the coefficient in the bilinear form $a(\cdot,\cdot)$. The parameter choices for this may further be altered such that the assembled matrix is restricted to a patch instead of the whole domain, or such that a mass matrix is assembled instead.

`interp.py`

Contains code for the interpolation operators. Here, the localized L^2-projection is used as interpolant. This follows by the usage of the method `interp.L2ProjectionPatchMatrix()`, whose parameter choices determine to which patch the L^2-projection will be localized.

`world.py`

Contains the information regarding the fine and coarse mesh and the boundary conditions. This includes the earlier mentioned `NWorldFine` and `NWorldCoarse`, and furthermore variables such as `NpFine`, `NpCoarse`, `NtFine` and `NtCoarse`, that corresponds to number of fine nodes, coarse nodes, fine elements and coarse elements respectively.

`coef.py`

Contains code for evaluating the coefficients, and furthermore restricting them to patches and elements. The file contains the class `coefficientFine` that creates the coefficient. This class further contains the method `localize()`, whose arguments determine what patch the coefficient shall be localized to.

`femsolver.py`

Contains code used to compute a FEM reference solution. The methods in this file are specific for a certain PDE, and have to be altered to fit the correct equation.

`linalg.py`

Contains methods for solving the linear systems that are established from the Galerkin methods. For example, the method `SchurComplementSolve()` (described in [5]) is used to evaluate the Ritz projection for the basis correctors.

`util.py`

Contains methods for keeping control of grid indices by using mapping functions. These methods can be used to extract the indices of a specific element or patch. For example, by extracting the first fine node in an element T, and adding `util.lowerLeftpIndexMap()` with the element as argument, the set of fine indices that belong to the coarse element T is returned.

`lod.py`

This file contains the code for solving the LOD patch problems. Of specific interest is the method `computeElementCorrector()` that solves the corrector problem for a given element T.

`pg.py`

Contains code to assemble the basis correctors, as well as the PG-LOD multiscale matrices.

These files in the `gridlod` code are the basis for the implementation made for this thesis. These files can solve multiscale elliptic problems, and are primarily written with time dependent diffusion in consideration. Thus, for this thesis we require further files to be written as extension to the `gridlod`. The `gridlod` source code can be found at [18].

5.2.2 Extension to gridlod

For the main implementation of this thesis, we first require methods that deal with the finescale calculations. Since the finescale problems' localization procedures do not coincide, two separate files have been written to deal with each of these systems, named `lod_node.py` and `lod_element.py` respectively. These methods' actions are performed within a central file `lod_wave.py` that the main file uses to control all finescale calculations. The files can be summarized as:

`lod_node.py`

Contains the function `compute_localized_node_correction()`, whose main

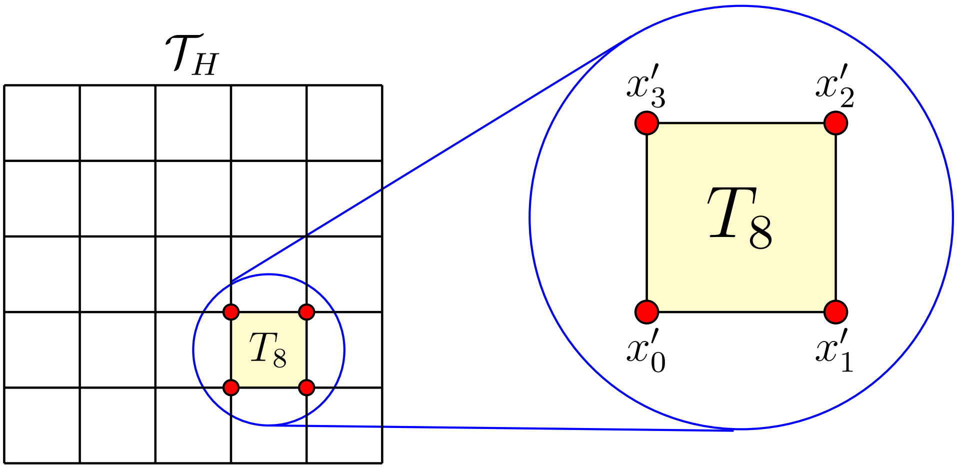


Figure 5.3: Illustration and indexing of nodes whose corresponding basis functions have support on $T_8 \in \mathcal{T}_H$.

purpose is to compute problems on the form (5.2) on node based localized patches. Given a node $x \in \mathcal{N}_H$ with corresponding patch ω_{x,k_1}, it constructs a patch restricted right hand side from the previous time step's solution, then uses the `RitzProjectionToFinePatch()` function from the `gridlod` module to evaluate w^n_{x,k_1}.

`lod_element.py`

Contains the method `compute_element_corrector()` that evaluates finescale problems on the form (5.9). Given an element $T \in \mathcal{T}_H$, it constructs a list `RhsList` that contains the basis functions whose supports overlap the current element T, e.g. `RhsList[0]` represents $\lambda_{x'_0}\big|_T$, where x'_0 is the element's first local node (see Figure 5.3). The `RitzProjectionToFinePatch()` function is then used to evaluate $\mathfrak{P}^h_T \lambda_{x'_i}$ for all T's local nodes x'_i.

`lod_wave.py`

Contains the code for usage of the functions mentioned above. For example, the method `solve_fs_system()` creates a patch for each coarse node, and uses `compute_localized_node_correction()` to evaluate and store these solutions. Furthermore, `compute_basis_correctors()` uses the `compute_element_corrector()` function for every T and stores this information. The global correctors $\mathfrak{P}^h_{k_0}\lambda_x$ can then be assembled by `gridlod`'s function `assemble_basis_correctors()` that performs the summation

$$\mathfrak{P}^h_{k_0}\lambda_x = \sum_{T \in \mathcal{T}_H} \mathfrak{P}^h_{k_0,T}\lambda_x.$$

The main method can be found in the file `loc_damped_wave.py`, which can be described by the pseudo code listed in Algorithm 1.

Several of the lines listed in Algorithm 1 use the files described at the beginning of this section. For example, the actions on lines 5-7 are performed by calling `compute_element_corrector()` followed by `assemble_basis_correctors()`, and the file `lod_node.py` was created for the purpose of computing lines 8-10. Other than that, the remaining lines can easily be computed by the help of functions from the gridlod module. Similarly to the standard gridlod functions, we are restricted to the unit hypercube $[0,1]^d$, with arbitrary number of dimensions d. Thus, the number of dimensions that is used for the simulations can easily be chosen by small modifications in loc_damped_wave.py. The source code used for this thesis can be found at [19].

When multiscale simulations are performed, specifically for $d > 1$, it is convenient to construct a method that easily generates multiscale coefficients. For this purpose a Python file buildcoef2d.py can be used. It works as an extension to gridlod. For example, it was used to create the multiscale coefficients used for the numerical examples in Section 3.7. The file is described in detail in [13], and can be found open source together with files for visualization at [20].

Algorithm 1: Main method in loc_damped_wave.py.

```
Pick $\tau, N, h$
Create fine mesh $1/h$
for all $H$ do
    Initialize coarse mesh $1/H$
    Pick $k_0(H), k_1(H)$
    for all $T \in \mathcal{T}_H$ do
        Compute $\mathfrak{P}^h_{k_0,T}$
    Assemble $\mathfrak{P}^h_{k_0}\lambda_x = \sum_T \mathfrak{P}^h_{k_0,T}\lambda_x$
    for all $n$ do
        Solve finescale system
        Store $\{w^n_{x,k_1}\}_x$
    Set initial values $v^{-1}$ and $u^0 = v^0 \in V^{\mathrm{ms},h}_{H,k}$
    Assemble fine matrices $\mathcal{S}^h, \mathcal{K}^h, \mathcal{M}^h$
    Construct $\mathcal{S}^{\mathrm{ms},h}_{H,k}, \mathcal{K}^{\mathrm{ms},h}_{H,k}, \mathcal{M}^{\mathrm{ms},h}_{H,k}, \mathcal{F}^{\mathrm{ms},h}_{H,k}$
    for all $n$ do
        Construct and store $\mathcal{R}^{\mathrm{ms},h}_{H,k,n}$
        Solve coarse system
        Construct $w^n_k$
        Store $v^n_k, w^n_k$
    Compute reference solution $u^n$
    Evaluate and store $\|u^n - v^n_k - w^n_k\|_{H^1}$
```

6

Numerical examples

The LOD and the PG-LOD have earlier shown great performance for several types of multiscale PDEs, both in efficiency as well as performance. However, they both lack the ability to correct for the variations of two multiscale coefficients when their oscillations do not coincide. In this section numerical examples that show the performance of the method introduced in Chapter 4 are presented and discussed. First, the decomposition of an exact solution is considered for several time steps, followed by every step which finally leads to the spatially and temporally localized method for the complete strongly damped wave equation.

6.1 Solution decomposition

For this numerical example we demonstrate the performance of the decomposition method introduced in Sections 4.1-4.3. Consider the one dimensional equation

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}x}\left(A_\varepsilon \frac{\mathrm{d}^2}{\mathrm{d}x\mathrm{d}t}u_\varepsilon + B_\varepsilon \frac{\mathrm{d}}{\mathrm{d}x}u_\varepsilon\right) &= 0, \text{ in } \Omega \times (0,T], \\ u_\varepsilon &= 0, \text{ on } \Gamma \times (0,T], \end{aligned}$$

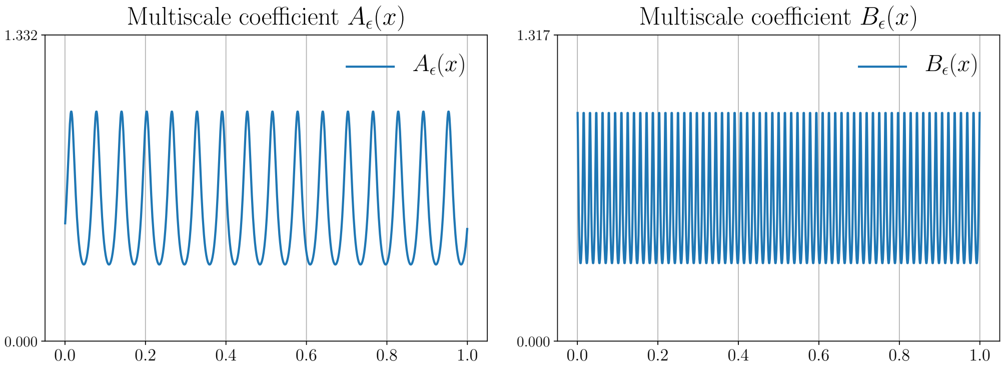
with initial condition $u_\varepsilon(0) = x(1-x)$, where $\Omega = [0,1]$. The damping and propagation coefficients are given respectively by

$$A_\varepsilon(x) := \left(2 - \sin\left(\frac{2\pi x}{\varepsilon_A}\right)\right)^{-1}, \qquad B_\varepsilon(x) := \left(2 - \cos\left(\frac{2\pi x}{\varepsilon_B}\right)\right)^{-1},$$

with $\varepsilon_A = 2^{-4}$ and $\varepsilon_B = 2^{-6}$, see Figure 6.1 for an illustration.

For the spatial discretization a fine mesh with $1/h = 1024$ is used, together with a coarse mesh with $1/H = 4$. The temporal domain is discretized with a uniform time step parameter $\tau = 0.01$ and the amount of time steps N is varied. The L^2-error between the exact solution u_ε^n (computed with standard FEM on the fine mesh) and the decomposed parts v^n and w^n is then evaluated for different values of $n = N$. The results can be seen in Table 6.1.

As seen in the table, the error between the solution and its decomposed parts are not exactly equal, which may seem to be contradictory to the theory. However, the error is so small that it can be interpreted as a small calculation error made by the computer. Note that since the error appears early in the time stepping, it implies a larger error in each sequential time step. This effect can be seen when N increases from 10 to 100. Afterwards, as N gets even larger, the error starts to



(a) Coefficient $A_\varepsilon(x)$.

(b) Coefficient $B_\varepsilon(x)$.

Figure 6.1: The multiscale coefficients used for the one dimensional examples.

decrease instead. This is a consequence from the fact that the system in question contains no source term, and thus the solution vanishes for large N. Consequently, the error will also approach 0 as both the solution as well as its decomposed parts vanish.

Table 6.1: The L^2-error between exact solution and its decomposed parts for different amount of time steps N.

N	$\|u^N - v^N - w^N\|$
10	$1.53 \cdot 10^{-11}$
50	$6.39 \cdot 10^{-11}$
100	$7.91 \cdot 10^{-11}$
500	$1.41 \cdot 10^{-11}$
1000	$1.38 \cdot 10^{-12}$

Next, the non-homogeneous case is considered, where a source term $f \in L^2$ is added to the right hand side. In this case the decomposition is no longer exact, but results in an error that depends on parameters such as H and τ.

6.2 Non-homogeneous case

In this section, a non-zero source term is added to the equation, and the performance of the method in Section 4.4 is presented. By Theorem 4.4.1, the error is expected to converge linearly in H^1-norm with respect to the mesh width H.

Consider the following equation

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}x}\left(A_\varepsilon \frac{\mathrm{d}^2}{\mathrm{d}x\mathrm{d}t}u_\varepsilon + B_\varepsilon \frac{\mathrm{d}}{\mathrm{d}x}u_\varepsilon\right) &= f, \text{ in } \Omega \times (0,T], \\ u_\varepsilon &= 0, \text{ on } \Gamma \times (0,T], \end{aligned}$$

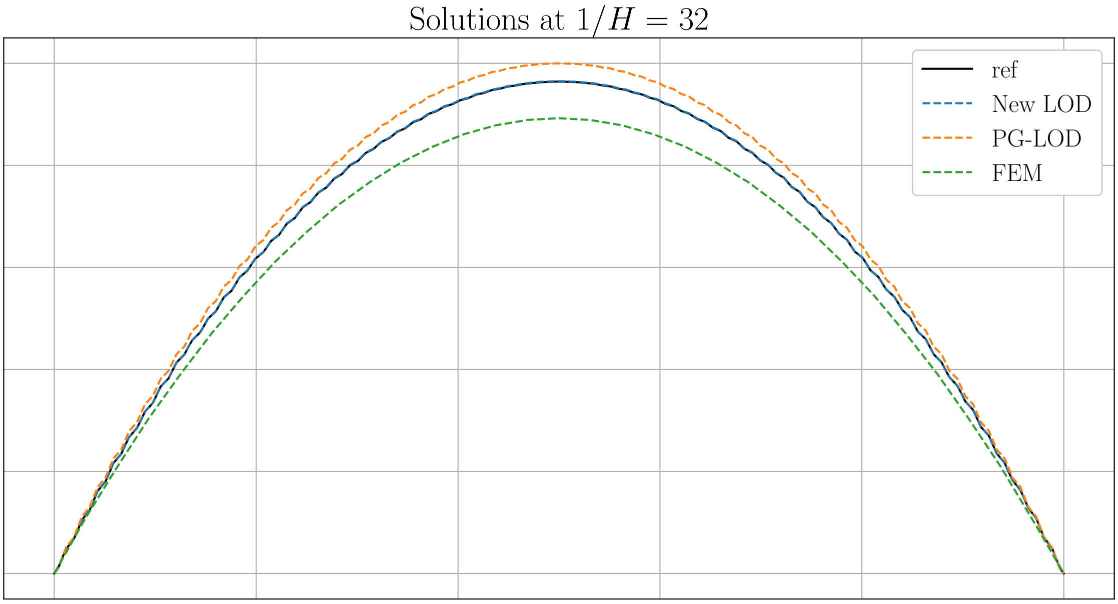


Figure 6.2: Comparison of the different methods' approximations with the reference solution with $1/H = 32$.

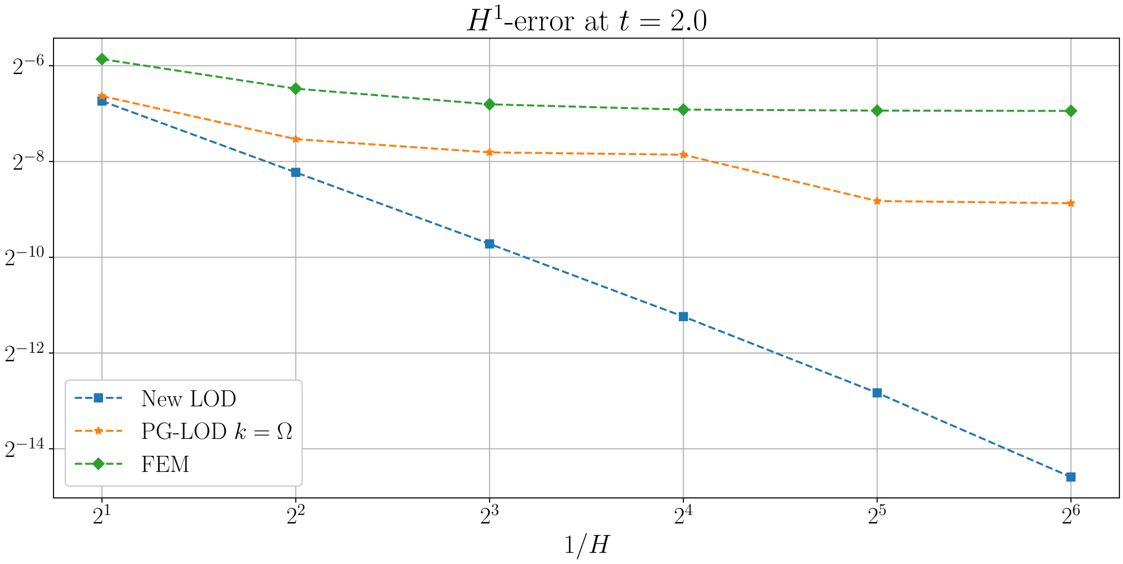


Figure 6.3: The H^1-error between the approximate solution u_H^N and the exact solution u_ε^N for the new LOD method, as well as the standard FEM and PG-LOD.

with initial condition $u_\varepsilon(0) = 0$. Once again, the domain is set to $\Omega = [0, 1]$ and the coefficients remain the same as in Figure 6.1. The fine mesh width is such that $1/h = 256$, while for the coarse mesh $1/H$ varies between 2 and 64 (uniformly refined). The time step $\tau = 0.01$ is used, and the amount of time steps is $N = 200$, such that $t \in [0, 2]$. For each value of $1/H$, the H^1-error between the exact solution u_ε^N and the approximate solution u_H^N is evaluated. Furthermore, the calculations are also performed for the standard FEM, as well as standard PG-LOD (non-localized). In Figure 6.2, the reference solution can be seen plotted next to the approximations for the different methods with $1/H = 32$. The H^1-errors depending on $1/H$ for the different methods are depicted in Figure 6.3.

In Figure 6.2 it is seen how the new LOD method manages to produce a solution that is well aligned with the reference solution. Furthermore it is seen how both the standard FEM and the PG-LOD fail to estimate the solution on a macroscopic level. This is expected since when $1/H = 32$, there is still one of the multiscale coefficients that oscillates with a frequency that cannot be resolved by this mesh, implying the global error. The plot of their respective H^1-errors in Figure 6.3 confirms this. Here it is seen how the error using the new LOD method converges linearly, just as expected from Theorem 4.4.1. Furthermore, it is seen how both the error for the standard FEM as well as the PG-LOD remain basically constant for all considered values of $1/H$. Next, we consider the method's performance on the full strongly damped wave equation.

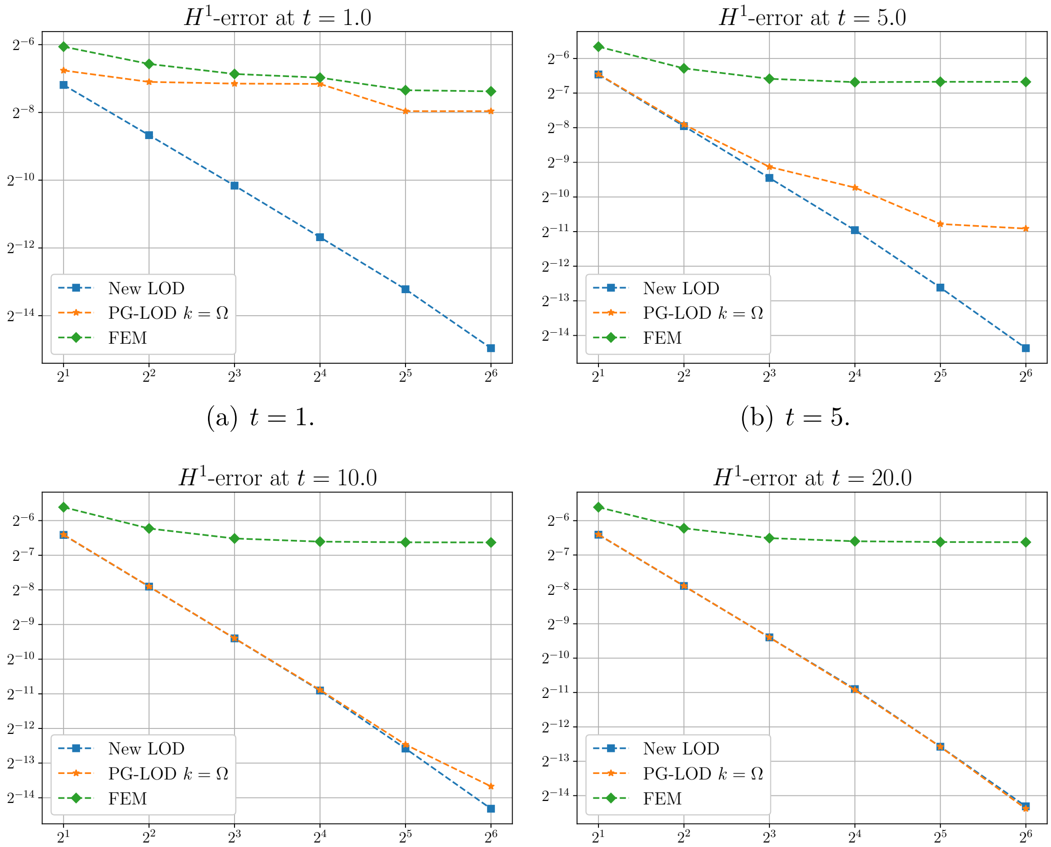
6.3 The complete equation

In this section numerical examples that show the performance of the method introduced in Section 4.5 for the strongly damped wave equation are considered. That is, we consider the equation given in (1.1), with source term $f = 1$ and initial data $u^0 = 0$. For this equation two seperate examples are presented, in one and two dimensions respectively.

6.3.1 One dimension

Let the multiscale coefficients $A_\varepsilon(x)$ and $B_\varepsilon(x)$ remain the same as in previous examples (see Figure 6.1). The fine mesh is such that $1/h = 256$, and for the coarse mesh $1/H$ varies between 2 and 64 (uniformly refined). The time step is set to $\tau = 0.01$, and the number of time steps $N = 2000$ is used such that the final time is $T = 20$. As earlier, the H^1-error between the exact solution u_ε^n (computed with FEM on the fine mesh) and the approximate solution u_H^n is plotted for the different methods (depicted in Figure 6.4). Note that for this example, the new LOD method is not yet localized. This procedure is done for $n = 100, 500, 1000, 2000$ to get some perspective of the performance depending on when the solution is evaluated.

As seen in Figure 6.4, the new LOD method performs well at all points in time. The error converges similarly as in previous example when the second derivative was excluded. For early time states, such for $t = 1$ and $t = 5$, it is seen that neither PG-LOD nor the standard FEM is able to resolve the multiscale variations. However, for



(a) $t = 1$.

(b) $t = 5$.

(c) $t = 10$.

(d) $t = 20$.

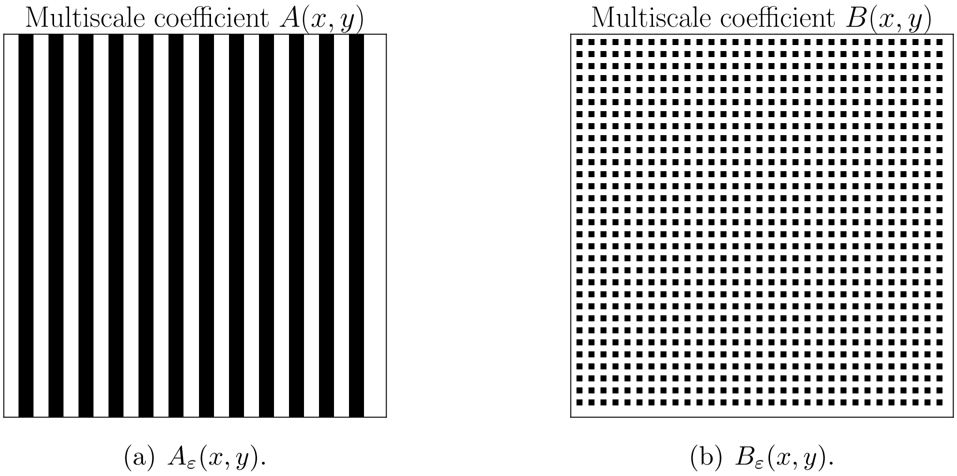
Figure 6.4: The H^1-error between the exact solution u_ε^n and the approximate solution u_H^n for different choices of n.

larger t's, the solution approaches a steady state, at which the damping coefficient $A_\varepsilon(x)$ no longer impacts the solution. This is why the PG-LOD performs better for each new time step, and specifically at $t = 20$ when it performs equally well as the new LOD.

6.3.2 Two dimensions

For the two dimensional example, the fine mesh width is set such that $1/h = 128$, and for the coarse mesh $1/H$ is uniformly refined from 2 to 32. The time step $\tau = 0.1$ is used and the number of time steps is $N = 10$, such that $t \in [0, 1]$. The multiscale coefficients $A_\varepsilon(x, y)$ and $B_\varepsilon(x, y)$ are created using `buildcoef2d.py` and are illustrated in Figure 6.5. The approximate solutions computed with the method from Chapter 4 can be seen for varying coarse mesh widths in Figure 6.6. The H^1-error between the exact solution u_ε^N and the approximate solution u_H^N computed with the new LOD method, as well as standard FEM and PG-LOD can be seen in Figure 6.7.

Just as in the two dimensional example presented in Section 2.5.2, the multiscale coefficient $A_\varepsilon(x, y)$ contains stripes that the solution is characterized by. For this


Multiscale coefficient $A(x, y)$
Multiscale coefficient $B(x, y)$

(a) $A_\varepsilon(x, y)$.

(b) $B_\varepsilon(x, y)$.

Figure 6.5: Multiscale coefficients used for the two dimensional example.

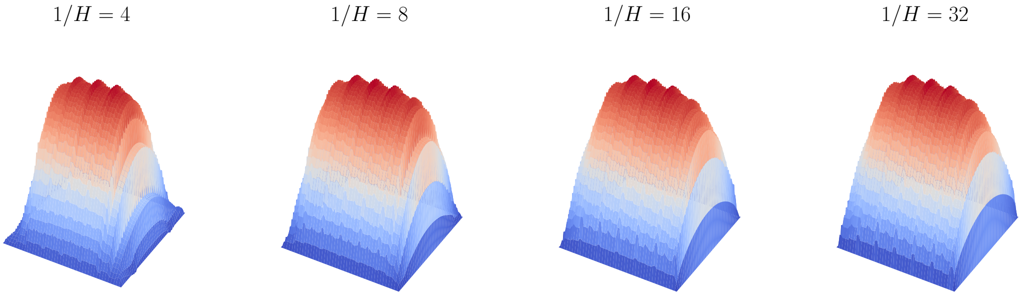


Figure 6.6: The solution evaluated for different choices of $1/H$ using the LOD method for the strongly damped wave equation.

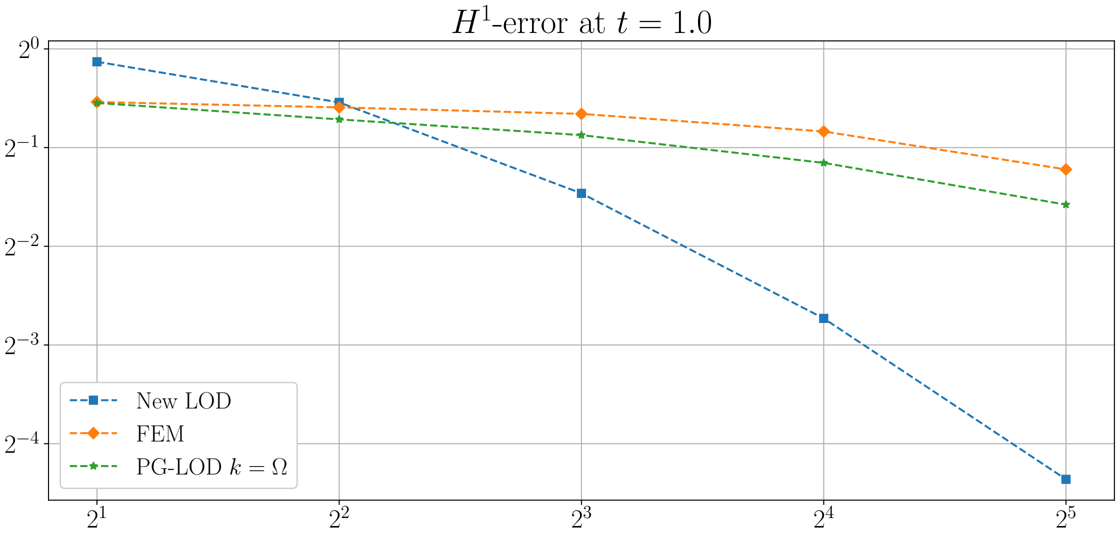
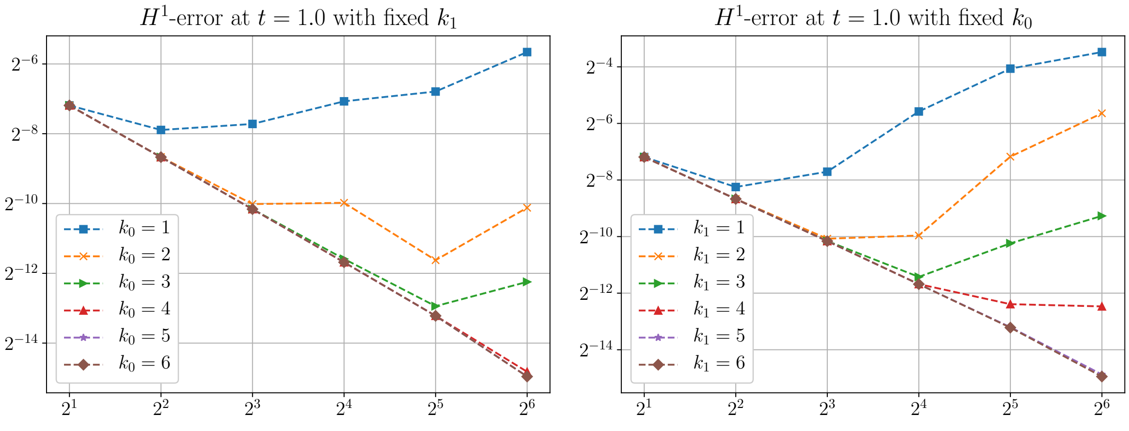


Figure 6.7: The H^1-errors between the exact and approximate solutions for the new LOD, standard FEM and PG-LOD in two dimensions.

example, the stripes can be seen in the solution already for $1/H = 4$. The fact that the method indeed works well for the two dimensional case as well is confirmed in Figure 6.7. This example furthermore shows how both the standard FEM and the PG-LOD fail to resolve the coefficients' variations, and the errors do not converge as the coarse mesh is refined.

6.4 Spatially localized method

Since the new LOD method requires many finescale calculations, it is necessary to restrict these problems to patches such that the computational complexity is reduced. There are two types of finescale problems, implying that we have two localization parameters k_0 and k_1. First, k_0 is set to be large enough such that the whole domain is considered as patch for the basis correction equation, and k_1 is varied. Then, k_1 is set as large enough and the performance depending on k_0 is analyzed. For the last example, both k_0 and k_1 are small such that both problems are localized. In every example, the fine mesh is such that $1/h = 256$ and the coarse mesh width is refined uniformly. The time step is $\tau = 0.01$ and number of time steps is $N = 100$. The calculations are done in one dimension, and the multiscale coefficients remain the same as in previous one dimensional examples (see Figure 6.1). The performances of each localization example can be seen in Figures 6.8 and 6.9.



(a) Fixed k_1.

(b) Fixed k_0.

Figure 6.8: The H^1-error using the new LOD method (spatially localized). In the left image k_1 is fixed large and k_0 is varied, and vice versa for the right image.

In Figure 6.8 it is seen that the error converges well for all considered mesh widths when $k_0 \geq 4$ and k_1 is fixed large such that the whole domain is considered for solution correction problem. That is, once $k_0 > 3$, the essential part of the basis correction's support is included into the basis. As an example, one may note that, e.g., when $1/H = 8$ in Figure 6.8(a), a choice of $k_0 \geq 2$ appears to include the necessary amount of the correction. For this purpose, the basis correction for this problem is plotted and depicted in Figure 6.10(a), where it is seen that outside of

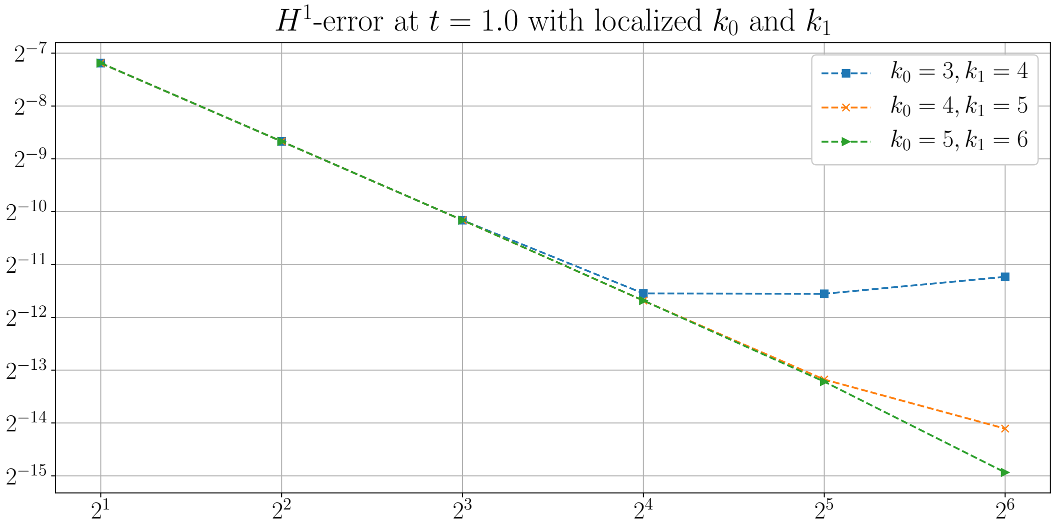
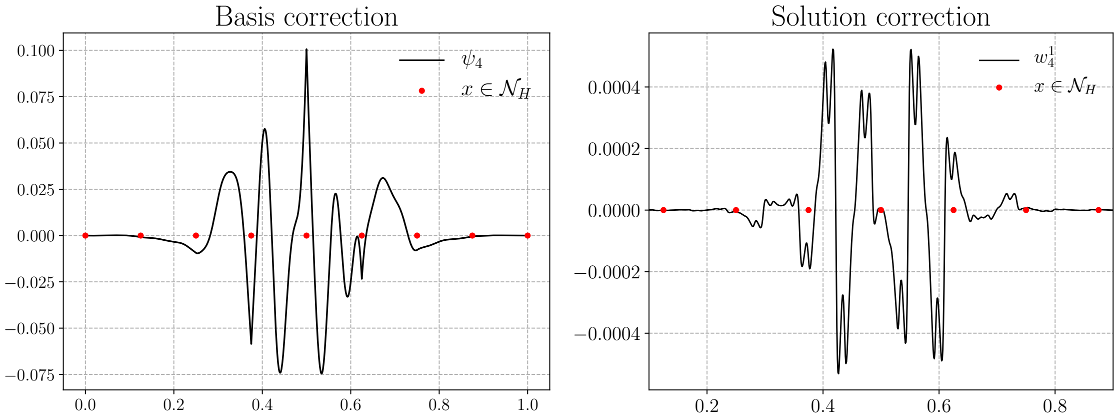


Figure 6.9: The H^1-errors between the exact and approximate solution for the new LOD when using both k_0 and k_1 as localization parameters.

this patch, the correction has basically vanished. Similarly, the finescale solution correction w_x^n for the center node has been plotted (with $n = 1$) in Figure 6.10(b). According to Figure 6.8(b), at this mesh width $k_1 = 2$ should still cover most of the correction. Indeed this is seen in the solution correction plot, since the patch covers the majority of the solution, and everything outside is close to zero.

Furthermore, one may notice in Figure 6.8 that k_0 can be chosen smaller than k_1, since e.g. the error contribution from $k_1 = 4$ is about the same size as $k_0 = 3$, and similarly for larger parameter choices. Thus in Figure 6.9, the parameters was set to $k_0 = k_1 - 1$, and the parameters $k_1 = 4, 5, 6$ were plotted. In the case when $k_1 = 6$, the error converges linearly through all mesh widths, but also the case $k_1 = 5$ works well, with a small exception on the last mesh width $1/H = 64$.



(a) Basis correction.

(b) Solution correction (with $n = 1$).

Figure 6.10: The basis and solution corrections plotted on a coarse mesh with $1/H = 8$.

6.5 Temporally localized method

Due to the decay of the solution correctors, we can pick an $l < N$ such after $n = l$, no more finescale calculations are done. Consequently, only the l previous coarse solution have to be stored, reducing memory issues. The l that may be picked is highly dependent of both the coarse mesh width H and the time step τ. This is due to the fact that the solution corrections can only be neglected once the error by not including them is less than the error induced by the mesh width H. This effect is illustrated by an example where similar calculations as in previous examples are done, but for different choices of l. Furthermore, an example where l is dependent on the mesh width H is presented.

6.5.1 Fixed localization parameters

For this example, the error depending on what choices are made on the localization parameter l and the final time T is illustrated. In the two cases, the final times are set to $T = 1$ and $T = 10$ respectively, such that the behaviour at an early state can be compared to when some time has passed. For both cases, the time step parameter is chosen as $\tau = 0.01$, such that the number of time steps in each case is $N = 100$ respectively $N = 1000$. For the first case, we let $l = 10, 20, ..., 90$ and for the latter $l = 100, 200, ..., 900$, such that equally many finescale calculations are done percentage wise. Similarly to earlier examples, the fine mesh has a mesh width $h = 1/256$, and the coarse mesh width H varies. The one dimensional coefficients used are the same as in Figure 6.1. For $T = 1$, the error in H^1-norm can be seen plotted in Figure 6.11 and for $T = 10$ in Figure 6.12.

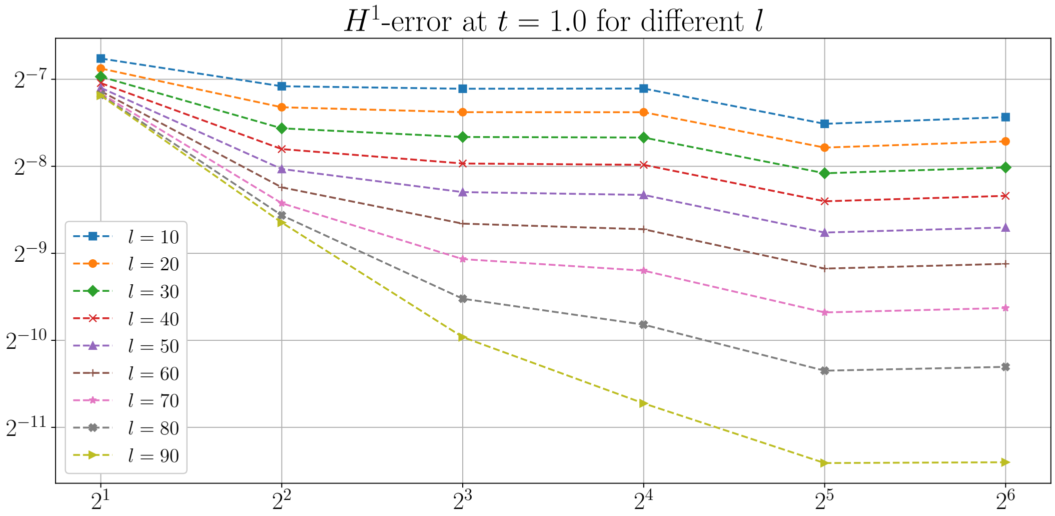


Figure 6.11: The H^1-error for different choices of the time localization parameter l with final time $T = 1$.

Consider the convergence plot in Figure 6.12, where the final time is $T = 10$. Here it is seen that for a choice of $l > 700$, the error converges with the expected linear rate through all considered mesh widths. However, for less refined meshes,

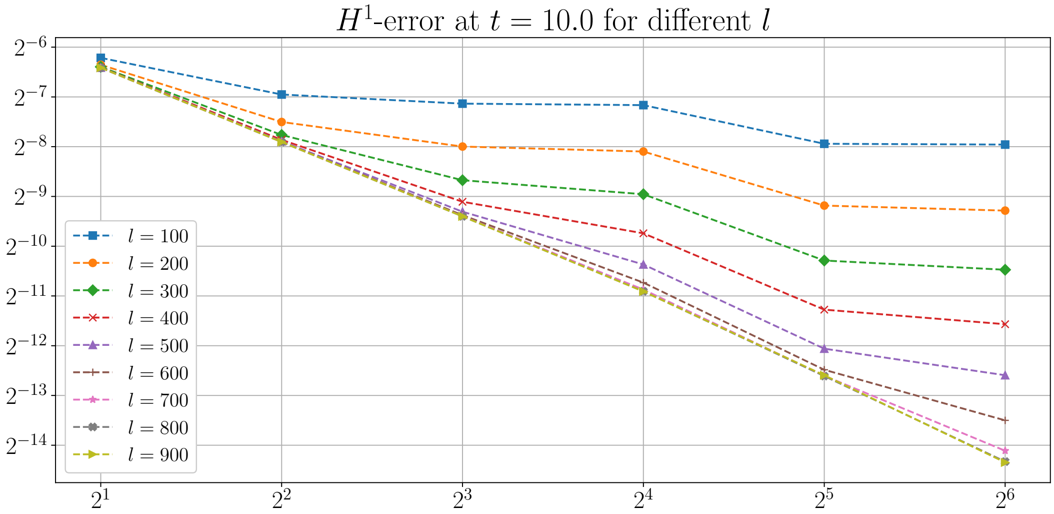


Figure 6.12: The H^1-error for different choices of the time localization parameter l with final time $T = 10$.

such that when $1/H = 8$, a choice of l as small as 500 works fine as well. The reason that this works well is that after $t = 5$ (which corresponds to $l = 500$) the solution corrections have decayed to a level so far below the mesh induced error that excluding them from the method implies a negligible error. When considering the case with $T = 1$ in Figure 6.11, it is clearly seen that not even $l = 90$ is a suitable choice for all mesh widths. While $t < 1$, the corrections are still so large neglecting only a few still contributes to a large error. Thus, for small choices of T, far many more finescale calculations have to be done percentage wise.

For large mesh widths, it is clear that the amount of corrections to include may be far less than for a highly refined mesh. Thus, it is necessary to evaluate l as a function of H. For this purpose, we introduce a tolerance parameter TOL $:= CH$, for some constant C, such that the error contribution from neglected corrections decays linearly with H.

6.5.2 Mesh width dependent localization

This section presents an example where l is varied dependent on what mesh width that is used. Similarly to earlier one dimensional examples, the fine mesh is set to $1/h = 256$ and coarse mesh width is varied. The time step $\tau = 0.01$ is used and the final time is $T = 10$. For each mesh width H, the finescale corrections are evaluated for as long as

$$\|w_x^n\|_{H^1} \geq CH.$$

For this example, the constant is set to $C = 5 \cdot 10^{-4}$. The corresponding value of l for each mesh width can be seen in Table 6.2, and the H^1-error is plotted and visualized in Figure 6.13.

Table 6.2: The localization parameter l for different mesh widths with time step $\tau = 0.01$ and final time $T = 10$.

$1/H$	$1/2$	$1/4$	$1/8$	$1/16$	$1/32$	$1/64$
l	61	166	325	502	539	656
$N - l$	939	834	675	498	461	344

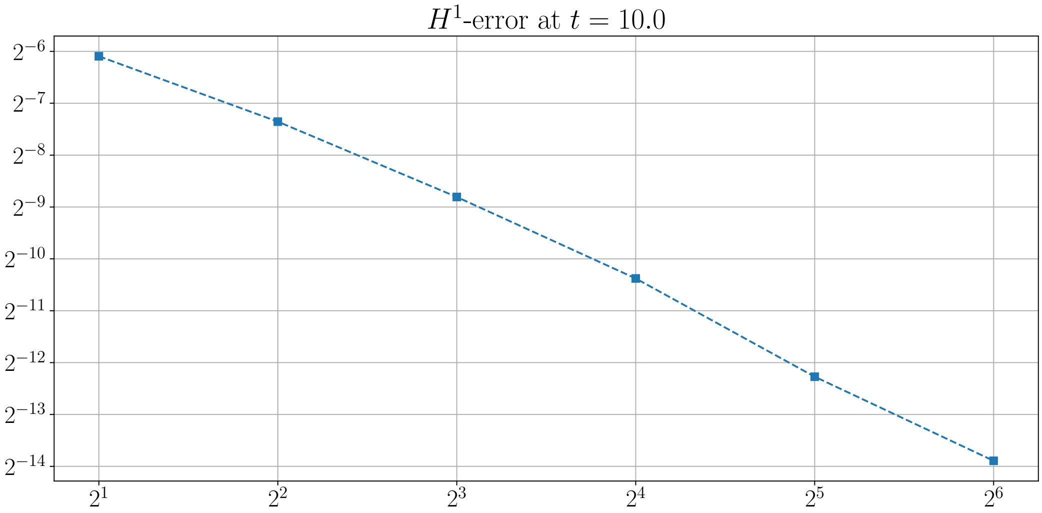


Figure 6.13: The H^1-error when using a localization parameter l dependent on the mesh width H.

In the plot depicted in Figure 6.13, the error converges at similar rate as when no time based localization was performed. The table furthermore shows the number of solution corrections that have been neglected. Here it is seen how for large mesh widths, e.g. when $1/H = 4$, only 166 out of the 1000 calculations have to be done. This implies that only 166 of the coarse solutions have to be stored, reducing the memory issues significantly. Even when $1/H = 64$, 344 calculations are neglected. Despite that so many correction problems can be ignored, Figure 6.13 shows that the error still converges well.

7

Conclusions

This thesis has dedicated itself to the Localized Orthogonal Decomposition method and its adaptation to the strongly damped wave equation. First, the standard FEM theory was presented for the equation. This included multiscale examples where it was shown how the standard FEM fails to resolve the variations on a scale of ε until the mesh width satisfies $H < \varepsilon$. Thus, when ε is small, FEM implies a complexity which is not computationally reasonable, emphasizing the necessity of multiscale methods. For this purpose, the standard LOD theory was presented with respect to an elliptic PDE. This included the orthogonal decomposition of the solution space into a coarse and finescale space. The basis of the coarse multiscale space was constructed by the incorporation of a basis correction evaluated on the finescale space. Due to the basis corrector's global support, it was necessary to truncate the correction problems solely to a neighbourhood of the corrector, implying computational feasibility of the LOD. Two possible methods of localization was introduced, node based as well as element based. Furthermore, the PG-LOD used to reduce the memory issues that arise with the LOD was presented. Finally, numerical examples that illustrated the performance of PG-LOD on the multiscale strongly damped wave equation was shown. Here it was seen how the PG-LOD yields error convergence of optimal order as long as both multiscale coefficients coincide. However, once the coefficients' behaviours were chosen differently, the PG-LOD failed to resolve the solution's variations, even for large values on the localization parameter k. Thereby, a new method that can incorporate both coefficients' dependencies into the solution was deduced from the standard LOD.

In Chapter 4, the construction of the new LOD method was presented. First, a simple problem formulation without the source term and second derivative was considered. For this formulation, an orthogonal decomposition of the solution into a coarse and a fine part was derived. The decomposition was done with respect to a new scalar product $\langle \cdot, \cdot \rangle_V$ that considered the bilinear forms of both A and B. By computing these decomposed parts, it was shown how the solution depended both on finescale corrections from earlier time steps, and moreover on all previous coarse solutions.

After adding a source term to the equation, the approximate solution was no longer merely a decomposition of the exact solution, but instead dependent on the mesh width H. More exactly, it was proven that the error between the approximate and exact solution in H^1-norm satisfies

$$\|u^n - u_H^n\|_{H^1} \leq CH \sum_{j=1}^{n} \tau \|f^j\|, \tag{7.1}$$

where the time step parameter $\tau := t_n/n$ is equal for both u^n and u_H^n. From this estimate, it is seen how the error decays linearly with respect to the mesh width H. Following this, the second derivative was added to the equation, and a similar method was applied to it. Similarly to the standard LOD, the method required spatial truncation around its corrections to be considered an applicable method. However, the new LOD also required to solve for finescale solution corrections in each time step. It was shown how these corrections decay exponentially in time, giving the opportunity to neglect these calculations once they are small enough.

Chapter 5 presented the considerations done for the implementation, both theoretical as well as practical. The Python module `gridlod` was discussed, as well as the files written as extension to it such that it can be applied to the strongly damped wave equation according to the new LOD. In Chapter 6, multiscale examples for the different new methods were presented. Here it was seen how the new LOD method performs well, establishing linear error convergence in H^1-norm, just as expected from (7.1). Furthermore, the spatial localization parameters k_0 and k_1 was varied, where it was seen how for $k_0, k_1 \leq 5$, the error converged at optimal order for $1/H = 2$ to $1/H = 64$. Of course, the choice of k_0 and k_1 are dependent on the mesh width and can be chosen as a function of H. At last, an example showing the performance of the temporal localization was shown. When the final time was set to $T = 10$, it was seen how a large part of the solution correctors could be neglected while the error still converged linearly, implying the possibility for a faster method that requires less memory.

The new LOD method presented in this thesis shows promising performance, both in efficiency as well as in error convergence. However, there are still some aspects that can be improved. Since the solution correctors decay exponentially in time, one may consider the possibility of using an adaptive time step. More precisely, since the correctors will alter less as the time passes, it is possible to skip every second correction, and later every third, etc. Furthermore, there are still some theoretical aspects to consider. The error estimate of the new LOD method for the full strongly damped wave equation, proving convergence of optimal order, is yet not established. Moreover, the spatial decay of both the basis corrections as well as the solution corrections require analytical proofs. These studies will be further discussed in future research.

Bibliography

[1] Weinan E, Bjorn Engquist, and Zhongyi Huang. Heterogeneous multiscale method: A general methodology for multiscale modeling. *Phys. Rev. B*, 67:092101, 03 2003.

[2] Thomas Y Hou and Xiao-Hui Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of computational physics*, 134(1):169–189, 1997.

[3] Thomas JR Hughes, Gonzalo R Feijóo, Luca Mazzei, and Jean-Baptiste Quincy. The variational multiscale method—a paradigm for computational mechanics. *Computer methods in applied mechanics and engineering*, 166(1-2):3–24, 1998.

[4] Axel Målqvist and Daniel Peterseim. Localization of elliptic multiscale problems. *Math. Comp.*, 83(290):2583–2603, 2014.

[5] C. Engwer, P. Henning, A. Målqvist, and D. Peterseim. Efficient implementation of the Localized Orthogonal Decomposition method. *ArXiv e-prints*, February 2016.

[6] Axel Målqvist and Anna Persson. Multiscale techniques for parabolic equations. *Numerische Mathematik*, 138(1):191–217, 2018.

[7] L. Debnath and P. Mikusiński. *Hilbert Spaces with Applications*. Elsevier Academic Press, 2005.

[8] S. Larsson and V. Thomée. *Partial Differential Equations with Numerical Methods*, volume 45. Springer-Verlag Berlin Heidelberg, 2003.

[9] Tomasz Dlotko. Certain methods of proving existence and uniqueness in the pde theory, 01 2001.

[10] Lawrence C. Evans. *Partial differential equations*. American Mathematical Society, Providence, R.I., 2010.

[11] Stig Larsson, Vidar Thomée, and Lars B. Wahlbin. Finite-element methods for a strongly damped wave equation. *IMA Journal of Numerical Analysis*, 11(1):115–142, 1991.

[12] Y. Lin, V. Thomée, and L.B. Wahlbin. *Ritz-Volterra projections to finite element spaces and applications to integro-differential and related equations*. Technical report (Cornell University. Mathematical Sciences Institute). Mathematical Sciences Institute, Cornell University, 1989.

[13] Tim Keil. Variational crimes in the Localized orthogonal decomposition method. Master's thesis, University of Münster, Germany, 2017.

[14] Målqvist, Axel and Persson, Anna. A generalized finite element method for linear thermoelasticity. *ESAIM: M2AN*, 51(4):1145–1171, 2017.

[15] Fredrik Hellman and Axel Målqvist. Numerical homogenization of time-dependent diffusion. *arXiv preprint arXiv:1703.08857*, 03 2017.

[16] Daniel Elfverson, Victor Ginting, and Patrick Henning. On multiscale methods in petrov-galerkin formulation. *Numerische Mathematik*, 131:643–, 11 2015.

[17] Alexandre Ern and J L. Guermond. *Theory Practice of Finite Elements*, volume 159. Springer, 01 1992.

[18] Fredrik Hellman. gridlod. `https://github.com/fredrikhellman/gridlod`.

[19] Per Ljung. Master-thesis. `https://github.com/LjungPer/Master-Thesis`.

[20] Tim Keil. Master's thesis lod. `https://github.com/TiKeil/Master-thesis-LOD`.