University of Basel

University of Basel
Department of Mathematics and Computer Science

Discontinuous Galerkin Finite Element Method for Wave Equations with Time-dependent Coefficients

Master Thesis

Mauro Morini

Supervisor: Prof. Dr. Marcus J. Grote
November 26, 2025

Contents

Abstract

The symmetric interior penalty discontinuous Galerkin finite element method is presented for a second order elliptical problem in 1d which yields a symmetric, positive definite bilinear form for a sufficiently positive penalty parameter. This guarantees existence and uniqueness of the discrete problem. Implementation and theoretical details are discussed. Numerical results confirm the optimal convergence rates of $\mathcal{O}(h^{r+1})$ in the L^2-norm and $\mathcal{O}(h^r)$ in the energy norm for $\mathcal{P}^r$-elements.
The elliptic case is consequently extended to a second order wave equation with time-dependent coefficients. In the spirit of the method of lines the problem is discretized in space using symmetric interior penalty Galerkin finite elements and discretized in time using the explicit second-order leapfrog time integration. The resulting mass matrix is block-diagonal allowing for a fully explicit and inherently parallel time-marching scheme.

0.1 Introduction

The classical continuous Finite Element Method (FEM) is a widely used and very powerful

0.2 Overview

Chapter 1

DG for an Elliptic Problem

First we consider a time-independent elliptic problem. Not only is it useful for initiation to the subject to first consider a simpler elliptic problem, but it is also an essential preparational step in deriving the SIPG bilinear form for the elliptic part of the hyperbolic problem as well.
The goal of this chapter is to build all the necessary theoretical and practical tools to solve a given elliptic problem numerically and then experimentally test the method for different parameters. We will define the necessary notation and derive the SIPG variational formulation as well as in detail describe further implementation steps as for example what basis of the finite element space we chose and how to derive local matrices. Finally this chapter will also include some well established theoretical results in the context of discontinuous Galerkin methods. The derivation of the bilinear form is inspired by Chapter 1 in [7] as well as [2] and [4] for cross reference.

1.1 Problem

We consider the following elliptic model problem:

$$-(c(x)u'(x))' = f(x) \qquad \forall x \in \Omega \tag{1.1a}$$

$$u(0) = g_0, u(1) = g_1 \tag{1.1b}$$

where $\Omega = (0,1)$ is the domain, $g_0, g_1 \in \mathbb{R}$ are Dirichlet boundary conditions, $f \in L^2(\Omega)$ and $c \in C^1(\Omega)$ satisfies:

$$c_{\min} \leq c(x) \leq c_{\max} \qquad \forall x \in \Omega$$

for $0 < c_{\min} \leq c_{\max}$. Multiplying the solution by a test function and integrating by parts over Ω we get the standard weak formulation:
Find $u \in \{v \in H^1(\Omega) \,|\, v(0) = g_0, v(1) = g_1\}$ such that:

$$a(u,v) = (f,v)_{L^2(\Omega)} \qquad \forall v \in C_c^\infty(\Omega) \tag{1.2}$$

Where

$$a : H^1(\Omega) \times H^1(\Omega) \to \mathbb{R}, \qquad (u,v) \mapsto \int_\Omega c(x)u'(x)v'(x)\mathrm{d}x$$

defines the standard elliptic bilinear form on $H^1(\Omega)$ and

$$(u,v)_{L^2(\Omega)} = \int_\Omega uv \,\mathrm{d}x$$

denotes the L^2-inner product.
If we assume $f \in L^2(\Omega)$ and $c \in C^1(\Omega)$ we know by *Lax-Milgram* that there exists a unique (weak) solution of (1.2) and from the *elliptic regularity theory* we can conclude higher regularity of the weak solution, namely $u \in H^2(\Omega)$. For reference see chapter 6.2, 6.3 in [1].

1.2 Discretization

We will now discretize the domain Ω, note that the notation will persist throughout this thesis. Let $0 = x_0 < \cdots < x_{N+1} = 1$ be the mesh faces, $I_n = (x_n, x_{n+1})$ for $n = 0, \dots, N$ be the elements and $\mathcal{T}_h = \{I_n\}_{n=0}^N$ a partition of Ω for some fixed $N \in \mathbb{N}$. Sometimes we will write N_h to emphasize the dependency on h. We denote the element length by $h_n = x_{n+1} - x_n$ for $n = 0, \dots, N$ and the global meshsize by $h = \max_n h_n$. Next we define the discontinuous finite element space

$$V_h^r(\mathcal{T}_h) = \{v \in L^2(\Omega) |\, v|_{I_n} \in \mathcal{P}^r(I_n)\}, \tag{1.3}$$

where $\mathcal{P}^r(I_n)$ denotes the space of polynomials $p : I_n \to \mathbb{R}$ of degree r for $r \in \mathbb{N}$. When the context allows it, we will denote the finite element space with just V_h for simplicity. V_h is our final approximation space in which the numerical solution lays. We observe that in contrast to a continuous finite element approximation space here the resulting solution is a priori discontinuous by construction. Furthermore we have here $V_h \not\subset H^1(\Omega)$. This is especially apparent in 1d due to the Sobolev embedding $H^1(\Omega) \subset C^0(\Omega)$. Any discontinuous element of V_h can therefore not be in $H^1(\Omega)$.
To proceed we will require the following trace operators:

Definition 1.1. *Let $v : \Omega \to \mathbb{R}$ be piecewise continuous and let $n \in \{1, \dots, N\}$*

(i) *We denote*

$$v(x_n^+) := \lim_{x \searrow x_n} v(x), \quad v(x_n^-) := \lim_{x \nearrow x_n} v(x)$$

the limit from above/below.

(ii) *We define the **jump** at x_n as*

$$[\![v(x_n)]\!] := v(x_n^-) - v(x_n^+)$$

*and the **average** at x_n as*

$$\{\!\{v(x_n)\}\!\} := \frac{v(x_n^+) + v(x_n^-)}{2},$$

furthermore by convention we set:

$$[\![v(x_0)]\!] := -v(x_0^+), \quad [\![v(x_{N+1})]\!] := v(x_{N+1}^-), \quad \{\!\{v(x_0)\}\!\} := v(x_0^+), \quad \{\!\{v(x_{N+1})\}\!\} := v(x_{N+1}^-).$$

1.3 Variational Formulation

To derive the SIPG variational formulation, let $v \in V_h$ be a test function. As mentioned in section 1.1 we can assume that the coefficient $c \in C^1(\Omega)$ and the exact solution $u \in H^2(\Omega) \subset C^1(\Omega)$. Due to the discontinuity of the test function in contrast to continuous FEM we multiply u with v on each element I_n and integrate by parts locally

$$\int_{x_n}^{x_{n+1}} fv \,\mathrm{d}x = -\int_{x_n}^{x_{n+1}} (cu')'v \,\mathrm{d}x = \int_{x_n}^{x_{n+1}} cu'v' \,\mathrm{d}x - cu'v\Big|_{x_n}^{x_{n+1}} \qquad \forall n \in \{0, \dots, N\},$$

then sum over all elements

$$(f, v)_{L^2(\Omega)} = \sum_{n=0}^{N} \int_{I_n} cu'v' \,\mathrm{d}x - \sum_{n=0}^{N+1} [\![c(x_n)u'(x_n)v(x_n)]\!], \tag{1.4}$$

where we have used that $\sum_{n=0}^N w\Big|_{x_n}^{x_{n+1}} = w(x_{N+1}^-) - w(x_N^+) + w(x_N^-) - \cdots - w(x_1^+) + w(x_1^-) - w(x_0^+) = \sum_{n=0}^{N+1} [\![w(x_n)]\!]$ for any piece-wise continuous function w.

By our construction c, u' are continuous on Ω, this means

$$[\![c(x_n)u'(x_n)v(x_n)]\!] = c(x_n)u'(x_n)[\![v(x_n)]\!] = \{\!\{c(x_n)u'(x_n)\}\!\}[\![v(x_n)]\!] \qquad \forall n \in \{0,\dots,N+1\}, \tag{1.5}$$

and

$$[\![u(x_n)]\!] = 0 \qquad \forall n \in \{1,\dots,N\}. \tag{1.6}$$

To derive the final variational form we will now have to add two additional terms to (1.4):
Step 1. Firstly we need to symmetrize our currently non-symmetrical right hand side which will correspond to the SIPG bilinear form. To do so we add the symmetry term $-\sum_{n=0}^{N+1}\{\!\{c(x_n)v'(x_n)\}\!\}[\![u(x_n)]\!]$ on both sides of (1.4) so we get

$$\begin{aligned}&(f,v)_{L^2(\Omega)} - g_1 c(x_{N+1}^-)v'(x_{N+1}^-) + g_0 c(x_0^+)v'(x_0^+)\\ =&\sum_{n=0}^{N}\int_{I_n} cu'v'\,\mathrm{d}x - \sum_{n=0}^{N+1}\{\!\{c(x_n)u'(x_n)\}\!\}[\![v(x_n)]\!] + \{\!\{c(x_n)v'(x_n)\}\!\}[\![u(x_n)]\!],\end{aligned}$$

where on the left hand side of the equation we have applied (1.6) for the interior node contributions of the sum (which therefore vanish), and the boundary condition (1.1b) ensuring the left hand side to be soley dependent on v.
Step 2. The bilinear form we seek to create will (for now) be defined on $V_h \times V_h$ meaning it will intake discontinuous functions. In particular the numerical solution will be a discontinuous function wheras the exact solution is continuous. To counterweigh this discrepancy we need to introduce a penalization mechanism, to minimize discontinuous behavior. Technically speaking this penalization term will guarantee coercivity of the bilinear form (see section 1.11).
Let $\sigma > 0$ constant, we define:

$$\mathtt{c}_n := \begin{cases}\max(c(x_n^+), c(x_n^-)), & n = 1,\dots,N\\ c(x_n^+), & n = 0\\ c(x_n^-), & n = N+1\end{cases}, \qquad \mathtt{h}_n := \begin{cases}\min(h_n, h_{n-1}), & n = 1,\dots,N\\ h_n, & n \in \{0, N+1\}\end{cases}$$

with this we define our penalization parameter

$$\mathtt{a}_n := \frac{\sigma \mathtt{c}_n}{\mathtt{h}_n} > 0 \qquad \forall n \in \{0\dots,N+1\}. \tag{1.7}$$

Similarly to Step 1 we can now add the penalty term $\sum_{n=0}^{N+1}\mathtt{a}_n[u(x_n)][v(x_n)]$ on both sides of (1.4) and get the final *discrete* SIPG variational formulation.

Find $u_h \in V_h$ such that:

$$b_h(u_h, v) = \ell_h(v), \qquad \forall v \in V_h, \tag{1.8}$$

where

$$\begin{aligned}b_h(u,v) &= \sum_{n=0}^{N}\int_{I_n} cu'v'\,\mathrm{d}x - \sum_{n=0}^{N+1}\{\!\{c(x_n)u'(x_n)\}\!\}[\![v(x_n)]\!] + \{\!\{c(x_n)v'(x_n)\}\!\}[\![u(x_n)]\!] + \sum_{n=0}^{N+1}\mathtt{a}_n[\![u(x_n)]\!][\![v(x_n)]\!],\\ \ell_h(v) &= (f,v)_{L^2(\Omega)} - g_1 c(x_{N+1}^-)v'(x_{N+1}^-) + g_0 c(x_0^+)v'(x_0^+) + \mathtt{a}_{N+1}g_1 v(x_{N+1}^-) + \mathtt{a}_0 g_0 v(x_0^+)\end{aligned}$$

for $u, v \in V_h$.

Remark 1.2. *In higher dimensions ($d \in \{2,3\}$) there are no point values appearing in the variational formulation, instead each point value evaluation becomes an integral over an element face. Since in practice higher dimensional problems are more interesting we quickly show how the variational formulation would*

look like in arbitrary dimensions.
Let $\mathcal{F}_h$ denote the set of all faces of the partition of a domain $\Omega \in \mathbb{R}^d$ and $\mathcal{T}_h$ denote the set of all elements. The bilinear form and the linear form become

$$b_h(u,v) = \sum_{K\in\mathcal{T}_h}\int_K c\nabla u\cdot\nabla v\,dx - \sum_{F\in\mathcal{F}_h}\int_F \{\!\{c\nabla u\}\!\}\cdot[\![v]\!] + \{\!\{c\nabla v\}\!\}\cdot[\![u]\!]\;dS + \sum_{F\in\mathcal{F}_h}\int_F \mathtt{a}[\![u]\!][\![v]\!]\;dS,$$
$$\ell_h(v) = (f,v)_{L^2(\Omega)} - \sum_{F\in\mathcal{F}_h\cap\partial\Omega}\int_F [\![g]\!]\cdot\{\!\{c\nabla v\}\!\}\;dS + \sum_{F\in\mathcal{F}_h\cap\partial\Omega}\int_F \mathtt{a}[\![g]\!][\![v]\!]\;dS.$$

1.4 Boundary Conditions

By adding the terms $-\sum_{n=0}^{N+1}\{\!\{c(x_n)v'(x_n)\}\!\}[\![u(x_n)]\!], \sum_{n=0}^{N+1}\mathtt{a}_n[\![u(x_n)]\!][\![v(x_n)]\!]$ on both sides of (1.4) we *weakly* imposed the Dirichlet boundary conditions into the variational form. This stands in contrast to how boundary conditions are usually imposed in continuous FEM. Indeed one could also impose them strongly, meaning we could define

$$V_h^r(\mathcal{T}_h) = \{v\in L^2(\Omega)|\, v|_{I_n}\in\mathcal{P}^r(I_n), v(x_0)=g_0, v(x_{N+1})=g_1\},$$

but this soley as a side note, we will continue to work with purely weakly imposed boundary conditions.

One could alternatively desire to impose *Neumann* boundary conditions weakly, this slightly changes the variational formulation. We illustrate the idea on the following example boundary condition. A solution u should satisfy:

$$u(0) = g_0, \quad u'(1)\cdot n_1 = g_1,$$

where again $g_0, g_1 \in \mathbb{R}$ are the boundary values and n_1 denotes the outward normal of the domain at the upper boundary. In 1d we trivially have $n_1 = 1, n_0 = -1$, where n_0 denotes the outward normal at the lower boundary. Note that we need at least one Dirichlet boundary condition, else there exists no unique solution.

Now recall the initial incomplete formulation (1.4). First we take the Neumann boundary contribution $\{\!\{c(x_{N+1})u'(x_{N+1})\}\!\}[\![v(x_{N+1})]\!]$ to the other side of the equation. We get

$$\sum_{n=0}^{N}\int_{I_n} cu'v'\,\mathrm{d}x - \sum_{n=0}^{N}\{\!\{c(x_n)u'(x_n)\}\!\}[\![v(x_n)]\!] = (f,v)_{L^2(\Omega)} + g_1c(x_{N+1}^-)v'(x_{N+1}^-)$$

We have used $\{\!\{c(x_{N+1})u'(x_{N+1})\}\!\}[\![v(x_{N+1})]\!] = c(x_{N+1}^-)u'(x_{N+1}^-)v(x_{N+1}^-)\cdot n_1 = g_1c(x_{N+1}^-)v(x_{N+1}^-)$.

From here on we proceed similarly as in the Dirichlet case. The main difference is that we always ommit the boundary face with the Neumann boundary condition.
We add the symmetry and penalty terms

$$-\sum_{n=0}^{N}\{\!\{c(x_n)v'(x_n)\}\!\}[\![u(x_n)]\!] + \sum_{n=0}^{N}\mathtt{a}_n[\![u(x_n)]\!]\{\!\{v(x_n)\}\!\}$$

to both sides, using again that the real solution has zero jump on the interior faces and applying the boundary conditions we finally derive the variational form

$$\sum_{n=0}^{N}\int_{I_n} cu'v'\,\mathrm{d}x - \sum_{n=0}^{N}\{\!\{c(x_n)u'(x_n)\}\!\}[\![v(x_n)]\!] + \{\!\{c(x_n)v'(x_n)\}\!\}[\![u(x_n)]\!] + \sum_{n=0}^{N}\mathtt{a}_n[\![u(x_n)]\!][\![v(x_n)]\!]$$
$$= (f,v)_{L^2(\Omega)} + g_0c(x_0^+)v'(x_0^+) + \mathtt{a}_0g_0v(x_0^+) + g_1c(x_{N+1}^-)v(x_{N+1}^-)$$

1.5 Matrix-Vector System

We will now transform the variational form (1.8) into Matrix-Vector system form. To do so let $r \in \mathbb{N}$ denote the polynomial degree and consequently the element degree of freedom. Note that in this thesis we will only consider global polynomial degrees, meaning one set polynomial degree for all elements. Next let $\{\Phi_0, \dots, \Phi_M\}$ be a basis of V_h, where $M = \dim(V_h)$. We can represent the sought Galerkin approximation as $u_h = \sum_{j=0}^{M} \alpha_j \Phi_j \in V_h$ for coefficients $\alpha_j \in \mathbb{R}$. Then (1.8) is equivalent to:

$$\sum_{j=0}^{M} \alpha_j b_h(\Phi_j, \Phi_i) = \ell_h(\Phi_i) \qquad \forall i = 0, \dots, M,$$

which corresponds to the system:

$$\mathbf{Bu} = \mathbf{l} \tag{1.9}$$

for $\mathbf{B} \in \mathbb{R}^{M \times M}, [\mathbf{B}]_{i,j} = b_h(\Phi_j, \Phi_i), \mathbf{u} \in \mathbb{R}^M, [\mathbf{u}]_j = \alpha_j, \mathbf{l} \in \mathbb{R}^M, [\mathbf{l}]_j = \ell_h(\Phi_j)$.

1.6 Basis of Finite Element Space

There are many ways of choosing basis functions for finite element spaces. In this thesis we soley focus on elementwise nodal Lagrangian basis functions, although there are alternatives which might be just as valid depending on the goal one pursues, like for example a modal Legendre basis. Having decided to work with a Lagrange nodal basis means we need to decide on the placing of the nodes. In principle for a desired polynomial degree r we could decide any $r+1$ points per elements. Since we restrict ourselves to a single global polynomial degree (i.e. no p-adaptivity) it makes sense to choose the same type of placement for all elements since this will allow us to calculate the actual function values on a single reference element (see below).
Here in contrast to continuous FEM, DG provides additional liberty, since there are no shared nodes over element faces we actually could decide to place the basis nodes purely inside of the element.

In this thesis we will not try to justify the choice of basis function too much and use Gauss-Lobatto quadrature nodes as basis nodes. This is a commonly used nodal basis in continuous FEM and we will also use it for our DG method. In Appendix A.2.3 of [5] the *Fekete points*, which in 1d coincide with the Gauss-Lobatto points, are recommended with the argument that this nodal basis yields a mass matrix with an optimal condition number. For more detailed information on choosing basis functions and alternative approaches see for example Appendix A.2 in [5].

Let $n \in \{1, \dots, N\}$ and $I_n \in \mathcal{T}_h$ be an arbitrary element. We denote $\hat{I} = (-1, 1)$ the *reference element* and $F_n : \hat{I} \to I_n, \xi \mapsto \frac{x_n + x_{n+1}}{2} + \frac{h_n}{2}\xi$ the *element map*. This now allows us to define a basis on the reference element and extend it to all elements using the element map.
For a fixed polynomial degree $r \geq 2$ let $\xi_0, \dots, \xi_r \in [-1, 1]$ be the Gauss-Lobatto nodes.

$r = 2$	$\{-1, 1\}$
$r = 3$	$\{-1, 0, 1\}$
$r = 4$	$\{-1, -\frac{1}{\sqrt{5}}, \frac{1}{\sqrt{5}}, 1\}$
$\vdots$	$\vdots$

The inner nodes are given by the roots of L'_{r-1}, the derivative of the $r-1$-th Legendre polynomial. We define the basis on the reference element as the Lagrangian nodal basis

$$\widehat{\phi}_i(\xi) := \prod_{\substack{j=0 \\ j \neq i}}^{r} \frac{\xi - \xi_j}{\xi_i - \xi_j}, \qquad \text{for } i = 0, \dots, r \tag{1.10}$$

and define the basis functions on the element I_n as

$$\phi_i^n : I_n \to \mathbb{R}, \quad \phi_i^n(x) := \widehat{\phi}_i(F_n^{-1}(x))$$

as a last step we extend the basis functions to the whole domain Ω by zero

$$\Phi_i^n : \Omega \to \mathbb{R}, \quad \Phi_i^n(x) := \begin{cases} \phi_i^n(x), & \text{for } x \in I_n \\ 0, & \text{else} \end{cases} \tag{1.11}$$

for $n = 0, \dots, N$ and $i = 0, \dots, r$. Clearly we have $\text{span}(\widehat{\phi}_0, \dots, \widehat{\phi}_r) = \mathcal{P}^r(\hat{I})$ and by extension $\text{span}(\Phi_0^0, \dots, \Phi_r^N) = V_h^r(\mathcal{T}_h)$. It is essential that our basis has only local support, meaning the basis functions are zero on most of the domain. This is the key property which allows the final matrices to be sparse. Choosing basis functions with global support, would lead to an unfeasible computational cost for small mesh sizes.
By having chosen a Lagrangian nodal basis the mesh nodes exactly coincide with the Gauss-Lobatto nodes on each element. To simplify the notation we introduce a *local-to-global* index map

$$T : \{0, \dots, N\} \times \{0, \dots, r\} \to \{1, \dots, M\} \tag{1.12}$$

where $M = (r+1)(N+1) = \dim(V_h)$. T takes an element index n and a local basis function index i as inputs and returns the globally assigned node index $T(n, i)$. T corresponds to the *connectivity matrix*. In the simplest case we have the global index ordered from left to right and get $T(n, i) = nr + i$.

1.7 Quadrature Rule

To discuss the assembly of the components of the system (1.9) in detail we will require a tool to approximate the integrals appearing in it, i.e. we need to choose a quadrature rule. The quadrature rule should primarily be as accurate as possible for the least amount of nodes necessary. So immediately one thinks of the Gauss-Legendre quadrature rule, which for $r+1$ quadrature nodes, is exact for polynomials of degree $2r+1$. In fact Gauss-Legendre is often the preferred choice due to it's high accuracy. A downside of Gauss-Legendre is that it does not include element boundary nodes. We do require the values of our basis functions at the boundary nodes for the consistency, symmetry and penalty terms in (1.8), as well as the values of c at the element faces. All these values can be preloaded at the desired quadrature nodes, which makes it simpler to use a quadrature rule with nodes at the element boundary. This is why we choose the Gauss-Lobatto quadrature rule despite it's lower accuracy of being exact for for polynomials of degree $2r-1$ with $r+1$ quadrature nodes.
We could still use Gauss-Legendre and interpolate the element boundary values. Suppose we chose $r+1$ Gauss-Legendre quadrature nodes per element, then we require the element boundary values as well, so after all we need to preload the values of the basis functions and the coefficient c at $r+3$ nodes per element (the quadrature nodes plus the boundary nodes). If instead we choose $r+2$ Gauss-Lobatto quadrature nodes we achieve the same accuracy but use a node less.
An additional benefit of using Gauss-Lobatto quadrature nodes is that now the mesh nodes as well as the quadrature nodes are of the same type. This simplifies the program architecture and allows for the possibility of using *mass-lumping*, meaning we can let the basis nodes and the quadrature nodes coincide when assembling the mass matrix

$$[\mathbf{M}]_{i,j} = (\Phi_j, \Phi_i)_{L^2(\Omega)}.$$

When we use $r+1$ Gauss-Lobatto nodes to approximate the integral $\int_\Omega \Phi_j \Phi_i \, \mathrm{d}x$ we introduce an error, since $\Phi_j \Phi_i \in \mathcal{P}^{2r}$, but the resulting mass-lumped mass matrix is diagonal, which is a crucial benefit in explicit time-marching schemes with continuous FEM spacial discretization. We compare the error created by using the inexact $r+1$ node Gauss-Lobatto quadrature rule for $\mathcal{P}^r$-elements with a higher order Gauss-Lobatto quadrature rule experimentally in 1.14.3, but we can already say that due to the decoupled element structure DG provides, the mass matrix is already block-diagonal and mass lumping provides no real benefit.

All of this being said the choice of the Gauss-Lobatto quadrature rule is in our case mainly preferential and not the focus of this thesis.

1.8 Stiffness Matrix Assembly

With the basis functions in (1.11) defined we can now in detail investigate how to assemble the matrix $\mathbf{B}$ in (1.9). To do so we firstly separate the bilinear form b_h into different components

$$\begin{aligned} a_h(u,v) &:= \sum_{n=0}^{N} \int_{I_n} cu'v' \,\mathrm{d}x \\ b_h^{\text{cons}}(u,v) &:= \sum_{n=0}^{N+1} \{\!\{c(x_n)u'(x_n)\}\!\}[\![v(x_n)]\!] + \{\!\{c(x_n)v'(x_n)\}\!\}[\![u(x_n)]\!] \\ b_h^{\text{penal}}(u,v) &:= \sum_{n=0}^{N+1} \mathtt{a}_n [\![u(x_n)]\!][\![v(x_n)]\!] \end{aligned}$$

Let $u_h = \sum_{m=0}^{N}\sum_{j=0}^{r} \alpha_j^m \Phi_j^m \in V_h$ denote the Galerkin approximation, then as discussed in section 1.6 the discrete variational formulation (1.8) is equivalent to

$$\sum_{m=0}^{N}\sum_{j=0}^{r} \alpha_j^m \Big(a_h(\Phi_j^m,\Phi_i^n) - b_h^{\text{cons}}(\Phi_j^m,\Phi_i^n) + b_h^{\text{penal}}(\Phi_j^m,\Phi_i^n) \Big) = \ell_h(\Phi_i^n), \qquad \forall n = 0,\dots,N, i = 0,\dots,r \quad (1.13)$$

which corresponds to the matrix vector system (1.9) where we can write

$$\mathbf{B} = \mathbf{A} - \mathbf{B}_{\text{cons}} + \mathbf{B}_{\text{penal}}$$

we will assemble the three (symmetric) matrices separately.

$$[\mathbf{B}]_{T(n,i),T(m,j)} = b_h(\Phi_j^m,\Phi_i^n), \qquad [\mathbf{A}]_{T(n,i),T(m,j)} = a_h(\Phi_j^m,\Phi_i^n)$$
$$[\mathbf{B}_{\text{cons}}]_{T(n,i),T(m,j)} = b_h^{\text{cons}}(\Phi_j^m,\Phi_i^n) \qquad [\mathbf{B}_{\text{penal}}]_{T(n,i),T(m,j)} = b_h^{\text{penal}}(\Phi_j^m,\Phi_i^n)$$

where T is given by (1.12)

1.8.1 Assembly of A

$\mathbf{A}$ is assembled similarly to the standard stiffness matrix in continuous finite element. The main difference is that there is no overlap in the elementwise contributions. Each set of local (element) basis functions only contributes to the integrals over said element. We can rewrite $\mathbf{A} = \sum_{s=0}^{N} \mathbf{A}^{(s)}$, where

$$[\mathbf{A}^{(s)}]_{T(n,i),T(m,j)} = \int_{I_s} c\,(\Phi_j^m)'(\Phi_i^n)' \,\mathrm{d}x. \quad (1.14)$$

Now since we have $\operatorname{supp}(\Phi_i^n) \subset I_n$ the only non-zero entries of $\mathbf{A}^{(s)}$ are the ones where both $n = m = s$. Pulling back the integral to the reference element using the chain rule and the substitution $F_s^{-1}(x) = \xi$ we find

$$\int_{I_s} c(x)\,(\Phi_j^s)'(x)(\Phi_i^s)'(x)\,\mathrm{d}x = \frac{2}{h_s}\int_{\hat{I}} c(F_s(\xi))\,\widehat{\phi}_j'(\xi)\widehat{\phi}_i'(\xi)\,\mathrm{d}\xi$$

This integral now only depends on the reference shape functions, the element length h_s and the values of the coefficient c. Using a higher order Gauss-Lobatto quadrature rule we can approximate the integral only requiring the values of c at the quadrature nodes, whilst having the values of ϕ, ϕ' preloaded. The total assembly of $\mathbf{A}$ can therefore be achieved by calculating a local contribution matrix $\widehat{\mathbf{A}}^{(s)} \in \mathbb{R}^{(r+1)\times(r+1)}$ for each element I_s and adding it into $\mathbf{A}$.

Example 1.3. *For $c \equiv 1$ with $\mathcal{P}^1$-elements ($r = 1$) we have*

$$\widehat{\boldsymbol{A}}^{(s)} = \frac{1}{h_s}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

1.8.2 Assembly of B consistency part

As before we rewrite $\mathbf{B}_{\text{cons}} = \sum_{s=0}^{N+1} \mathbf{B}_{\text{cons}}^{(s)}$, where

$$[\mathbf{B}_{\text{cons}}^{(s)}]_{T(n,i),T(m,j)} = \{\!\{c(x_s)\Phi_j^{m\,\prime}(x_s)\}\!\}[\![\Phi_i^n(x_s)]\!] + \{\!\{c(x_s)\Phi_i^{n\,\prime}(x_s)\}\!\}[\![\Phi_j^m(x_s)]\!] \tag{1.15}$$

Interior Faces

First let $s \in \{1,\dots,N\}$ denote an interior face, we observe again, that the entries of $\mathbf{B}_{\text{cons}}^{(s)}$ can only be non-zero for an index tuple $(T(n,i),T(m,j))$ if $n,m \in \{s,s-1\}$ by the local support of the basis functions. Therefore assembling $\mathbf{B}_{\text{cons}}$ again comes down to calculating a local contribution matrix

$$\widehat{\mathbf{B}}_{\text{cons}}^{(s)} = \begin{bmatrix} \mathbf{C}_{\text{cons}}^{(s-1,s-1)} & \mathbf{C}_{\text{cons}}^{(s-1,s)} \\ \mathbf{C}_{\text{cons}}^{(s,s-1)} & \mathbf{C}_{\text{cons}}^{(s,s)} \end{bmatrix} \in \mathbb{R}^{2(r+1)\times 2(r+1)}$$

for each interior face x_s consisting of four blocks we will now lay out in more detail. We discuss the boundary case separately. Using again the local support of the basis functions we find

$$\begin{aligned}
[\mathbf{C}_{\text{cons}}^{(s-1,s-1)}]_{i,j} &= \frac{c(x_s^-)}{2}\Phi_j^{s-1\,\prime}(x_s^-)\Phi_i^{s-1}(x_s^-) + \frac{c(x_s^-)}{2}\Phi_i^{s-1\,\prime}(x_s^-)\Phi_j^{s-1}(x_s^-) \\
[\mathbf{C}_{\text{cons}}^{(s-1,s)}]_{i,j} &= \frac{c(x_s^+)}{2}\Phi_j^{s\,\prime}(x_s^+)\Phi_i^{s-1}(x_s^-) - \frac{c(x_s^-)}{2}\Phi_i^{s-1\,\prime}(x_s^-)\Phi_j^{s}(x_s^+) \\
[\mathbf{C}_{\text{cons}}^{(s,s)}]_{i,j} &= -\frac{c(x_s^+)}{2}\Phi_j^{s\,\prime}(x_s^+)\Phi_i^{s}(x_s^+) - \frac{c(x_s^+)}{2}\Phi_i^{s\,\prime}(x_s^+)\Phi_j^{s}(x_s^+)
\end{aligned}$$

where we have used the definitions of jump and average in (1.1). Note that $\mathbf{C}_{\text{cons}}^{(s-1,s)} = (\mathbf{C}_{\text{cons}}^{(s,s-1)})^T$ by the symmetry of the bilinear form b_h^{cons}. Next we represent the values of the basis functions Φ at the element boundary by the values of the reference shape functions $\widehat{\phi}$

$$\begin{aligned}
\Phi_i^s(x_s^+) &= \widehat{\phi}_i(-1), & \Phi_i^{s-1}(x_s^-) &= \widehat{\phi}_i(1) \\
\Phi_i^{s\,\prime}(x_s^+) &= \frac{2}{h_s}\widehat{\phi}_i'(-1), & \Phi_i^{s-1\,\prime}(x_s^-) &= \frac{2}{h_{s-1}}\widehat{\phi}_i'(1)
\end{aligned} \tag{1.16}$$

which finally yields

$$\begin{aligned}
[\mathbf{C}_{\text{cons}}^{(s-1,s-1)}]_{i,j} &= \frac{c(x_s^-)}{h_{s-1}}\widehat{\phi}_j'(1)\widehat{\phi}_i(1) + \frac{c(x_s^-)}{h_{s-1}}\widehat{\phi}_i'(1)\widehat{\phi}_j(1) \\
[\mathbf{C}_{\text{cons}}^{(s-1,s)}]_{i,j} &= \frac{c(x_s^+)}{h_s}\widehat{\phi}_j'(-1)\widehat{\phi}_i(1) - \frac{c(x_s^-)}{h_{s-1}}\widehat{\phi}_i'(1)\widehat{\phi}_j(-1) \\
[\mathbf{C}_{\text{cons}}^{(s,s)}]_{i,j} &= -\frac{c(x_s^+)}{h_s}\widehat{\phi}_j'(-1)\widehat{\phi}_i(-1) - \frac{c(x_s^+)}{h_s}\widehat{\phi}_i'(-1)\widehat{\phi}_j(-1)
\end{aligned}$$

Example 1.4. *Consider $c \equiv 1$ for $\mathcal{P}^1$-elements ($r=1$) with an equidistant mesh with meshsize h we have*

$$\widehat{\boldsymbol{B}}_{cons}^{(s)} = \frac{1}{h}\begin{bmatrix} 0 & -1/2 & 1/2 & 0 \\ -1/2 & 1 & -1 & 1/2 \\ 1/2 & -1 & 1 & -1/2 \\ 0 & 1/2 & -1/2 & 0 \end{bmatrix}$$

Boundary Faces

For $s \in \{0, N+1\}$ and x_s a boundary face we now have a smaller local contribution matrix since the contribution can only come from the one element to which x_s belongs. Meaning we have $\widehat{\mathbf{B}}_{\text{cons}}^{(0)}, \widehat{\mathbf{B}}_{\text{cons}}^{(N+1)} \in$

$\mathbb{R}^{(r+1)\times(r+1)}$ with

$$[\widehat{\mathbf{B}}^{(0)}_{\text{cons}}]_{i,j} = -\frac{2c(x_0^+)}{h_0}\widehat{\phi}_j'(-1)\widehat{\phi}_i(-1) - \frac{2c(x_0^+)}{h_0}\widehat{\phi}_i'(-1)\widehat{\phi}_j(-1)$$
$$[\widehat{\mathbf{B}}^{(N+1)}_{\text{cons}}]_{i,j} = \frac{2c(x_{N+1}^-)}{h_{N+1}}\widehat{\phi}_j'(1)\widehat{\phi}_i(1) + \frac{2c(x_{N+1}^-)}{h_{N+1}}\widehat{\phi}_i'(1)\widehat{\phi}_j(1)$$

Example 1.5. *For $c \equiv 1$ with $\mathcal{P}^1$-elements ($r = 1$) we have*

$$\widehat{\boldsymbol{B}}^{(0)}_{cons} = \frac{1}{h_0}\begin{bmatrix} 2 & -1 \\ -1 & 0 \end{bmatrix}, \qquad \widehat{\boldsymbol{B}}^{(N+1)}_{cons} = \frac{1}{h_{N+1}}\begin{bmatrix} 0 & -1 \\ -1 & 2 \end{bmatrix}$$

1.8.3 Assembly of B penalty part

Again we rewrite $\mathbf{B}_{\text{penal}} = \sum_{s=0}^{N+1} \mathbf{B}^{(s)}_{\text{penal}}$, where

$$[\mathbf{B}^{(s)}_{\text{penal}}]_{T(n,i),T(m,j)} = \mathtt{a}_s[\![\Phi_j^m(x_s)]\!][\![\Phi_i^n(x_s)]\!] \tag{1.17}$$

We proceed analogously to 1.8.2.

Interior Faces

Let $s \in \{1, \dots, N\}$ and x_s denote an interior face. As before we have that the entries of $\mathbf{B}^{(s)}_{\text{penal}}$ can only be nonzero for an index tuple $(T(n,i), T(m,j))$ if $n, m \in \{s, s-1\}$, similar to 1.8.2 we find that the assembly boils down to adding up local contributions represented in a local contribution matrix

$$\widehat{\mathbf{B}}^{(s)}_{\text{penal}} = \begin{bmatrix} \mathbf{C}^{(s-1,s-1)}_{\text{penal}} & \mathbf{C}^{(s-1,s)}_{\text{penal}} \\ \mathbf{C}^{(s,s-1)}_{\text{penal}} & \mathbf{C}^{(s,s)}_{\text{penal}} \end{bmatrix} \in \mathbb{R}^{2(r+1)\times 2(r+1)}$$

and using (1.16), and the definition of the penalization parameter (1.7) we specifically find

$$[\mathbf{C}^{(s-1,s-1)}_{\text{penal}}]_{i,j} = \mathtt{a}_s\widehat{\phi}_j(1)\widehat{\phi}_i(1)$$
$$[\mathbf{C}^{(s-1,s)}_{\text{penal}}]_{i,j} = -\mathtt{a}_s\widehat{\phi}_j(-1)\widehat{\phi}_i(1)$$
$$[\mathbf{C}^{(s,s)}_{\text{penal}}]_{i,j} = \mathtt{a}_s\widehat{\phi}_j(-1)\widehat{\phi}_i(-1)$$

where again by symmetry of the penalty term we have $\mathbf{C}^{(s-1,s)}_{\text{penal}} = (\mathbf{C}^{(s,s-1)}_{\text{penal}})^T$ and $\mathtt{a}_s$ only depends on the two adjacent elements I_{s-1}, I_s.

Example 1.6. *Consider $c \equiv 1$ for $\mathcal{P}^1$-elements ($r = 1$) with an equidistant mesh with meshsize h we have*

$$\widehat{\boldsymbol{B}}^{(s)}_{penal} = \frac{\sigma}{h}\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Boundary Faces

For $s \in \{0, N+1\}$ we again have only the respective boundary element contributing. So the local contribution matrices $\widehat{\mathbf{B}}^{(s)}_{\text{penal}} \in \mathbb{R}^{(r+1)\times(r+1)}$ satisfy

$$[\widehat{\mathbf{B}}^{(0)}_{\text{penal}}]_{i,j} = \mathtt{a}_0\widehat{\phi}_j(-1)\widehat{\phi}_i(-1), \qquad [\widehat{\mathbf{B}}^{(N+1)}_{\text{penal}}]_{i,j} = \mathtt{a}_{N+1}\widehat{\phi}_j(1)\widehat{\phi}_i(1)$$

Example 1.7. *For $c \equiv 1$ with $\mathcal{P}^1$-elements ($r = 1$) we have*

$$\widehat{\boldsymbol{B}}^{(0)}_{penal} = \frac{\sigma}{h_0}\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad \widehat{\boldsymbol{B}}^{(N+1)}_{penal} = \frac{\sigma}{h_{N+1}}\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

1.9 System Vector Assembly

We divide assembling the vector $\mathbf{l}$ in (1.9) into two parts.

$$\mathbf{l} = \mathbf{l}_{\text{load}} + \mathbf{l}_{\text{bc}}$$

First we recall the assembly of the load vector $\mathbf{l}_{\text{load}}$, i.e. the vector containing the contributions of the forcing term f and secondly we will describe how to add the Dirichlet boundary condition contributions ($\mathbf{l}_{\text{bc}}$).

1.9.1 Load Vector

The assembly of the load vector is completely analogous to the continuous finite element case. Using the local support of Φ^n_i we can rewrite

$$\int_\Omega f\Phi^n_i \mathrm{d}x = \sum_{s=0}^{N}\int_{I_s} f\Phi^n_i \mathrm{d}x = \int_{I_n} f(x)\Phi^n_i(x)\mathrm{d}x = \frac{h_n}{2}\int_{-1}^{1} f(F_n(\xi))\widehat{\phi}_i(\xi)\mathrm{d}\xi$$

meaning as before we can assemble $\mathbf{l}_{\text{load}} = \sum_{s=0}^{N}\mathbf{l}^{(s)}_{\text{load}}$ where

$$[\mathbf{l}^{(s)}_{\text{load}}]_{T(n,i)} = \int_{I_s} f\Phi^n_i \mathrm{d}x = \delta_{n,s}\frac{h_n}{2}\int_{-1}^{1} f(F_n(\xi))\widehat{\phi}_i(\xi)\mathrm{d}\xi$$

which can be characterized by the local contribution vector $\widehat{\mathbf{l}}^{(s)}_{\text{load}} \in \mathbb{R}^{r+1}$ defined as

$$[\widehat{\mathbf{l}}^{(s)}_{\text{load}}]_i = \frac{h_s}{2}\int_{-1}^{1} f(F_s(\xi))\widehat{\phi}_i(\xi)\mathrm{d}\xi$$

In practice we approximate the integral using a higher order Gauss-Lobatto quadrature rule.

Example 1.8. *For f piecewise constant (i.e. $f|_{I_s} \equiv f_s \in \mathbb{R} \quad \forall s = 0, \ldots, N$) with $\mathcal{P}^1$-elements ($r = 1$) we have*

$$\widehat{\mathbf{l}}^{(s)}_{\text{load}} = \frac{f_s h_s}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

1.9.2 Dirichlet Boundary Condition Vector

We have

$$[\mathbf{l}_{\text{bc}}]_{T(n,i)} = -g_1 c(x^-_{N+1})\Phi^{n\,\prime}_i(x^-_{N+1}) + g_0 c(x^+_0)\Phi^{n\,\prime}_i(x^+_0) + \mathtt{a}_{N+1}g_1\Phi^n_i(x^-_{N+1}) + \mathtt{a}_0 g_0\Phi^n_i(x^+_0)$$

where the entries can clearly only be non-zero at indices corresponding to boundary elements. We characterize the assembly using the local contribution vectors $\widehat{\mathbf{l}}^{(N+1)}_{\text{bc}}, \widehat{\mathbf{l}}^{(0)}_{\text{bc}} \in \mathbb{R}^{r+1}$ where

$$[\widehat{\mathbf{l}}^{(0)}_{\text{bc}}]_i = \frac{g_0}{h_0}c(x^-_0)\widehat{\phi}_i(-1) + \mathtt{a}_0 g_0\widehat{\phi}_i(-1), \qquad [\widehat{\mathbf{l}}^{(N+1)}_{\text{bc}}]_i = -\frac{g_1}{h_N}c(x^-_{N+1})\widehat{\phi}_i(1) + \mathtt{a}_{N+1}g_1\widehat{\phi}_i(1)$$

1.10 Mass Matrix Assembly

Although the mass matrix has not yet appeared in the discrete formulation as presented in (1.9) due to the absence of a mass term in the elliptic pde as presented in (1.1a), we will require it soon and hence quickly review it's assembly here.
The mass matrix is defined as

$$[\mathbf{M}]_{T(n,i),T(m,j)} = (\Phi_j^m, \Phi_i^n)_{L^2(\Omega)} = \sum_{s=0}^{N} \int_{I_s} \Phi_j^m \Phi_i^n \, \mathrm{d}x.$$

So we can rewrite $\mathbf{M} = \sum_{s=0}^{N} \mathbf{M}^{(s)}$, where

$$[\mathbf{M}^{(s)}]_{T(n,i),T(m,j)} = \int_{I_s} \Phi_j^m \Phi_i^n \, \mathrm{d}x.$$

Similarly to before the entries of $\mathbf{M}^{(s)}$ are only non-zero if $m = n = s$. In that case we pull the integral back to the reference element aby substitution and find

$$\int_{I_s} \Phi_j^m \Phi_i^n \, \mathrm{d}x = \frac{h_s}{2} \int_{-1}^{1} \widehat{\phi}_j \widehat{\phi}_i \, \mathrm{d}x.$$

The assembly of $\mathbf{M}$ corresponds therefore to summing up local contribution matrices $\widehat{\mathbf{M}}^{(s)} \in \mathbb{R}^{(r+1)\times(r+1)}$. In fact since the elements are decoupled due to the discontinuity of the basis functions, the resulting matrix is a block-diagonal matrix where each block corresponds to the local contribution matrix $\widehat{\mathbf{M}}^{(s)}$

Example 1.9. *For $\mathcal{P}^1$-elements ($r = 1$) we have*

$$\widehat{\boldsymbol{M}}^{(s)} = \frac{h_s}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

1.11 Existence of Discrete Solution

Firstly we will recall some basic definitions:

Definition 1.10. *Let V be a normed vector space and $b : V \times V \to \mathbb{R}$ be a bilinear form.*

(i) *We say b is **continuous** if $\exists C_{cont} > 0$, such that*

$$|b(u,v)| \leq C_{cont} \|u\| \, \|v\| \qquad \forall u, v \in V$$

(ii) *We say b is **symmetric** if*

$$b(u,v) = b(v,u) \qquad \forall u, v \in V$$

(iii) *We say b is **coercive** if $\exists C_{coer} > 0$, such that*

$$b(u,u) \geq C_{coer} \|u\|^2 \qquad \forall u \in V$$

Since (1.8) corresponds to the finite dimensional system (1.9) uniqueness and existence of a solution are equivalent. The bilinear form b_h is *symmetric* by construction the goal of this section is to show that b_h is also *coercive* for $\sigma > 0$ big enough. From the coercivity of b_h it will follow that the matrix $\mathbf{B}$ in (1.9) is positive definite and hence invertible, which means there exists a (unique) solution of (1.8).

Lemma 1.11. *Let $V = span(\varphi_1, \dots, \varphi_M)$ be a finite dimensional normed vector space with $\dim(V) = M \in \mathbb{N}$ and let $b : V \times V \to \mathbb{R}$ be a symmetric, coercive bilinear form, then the matrix $[\boldsymbol{B}]_{i,j} = [b(\varphi_j, \varphi_i)]_{i,j} \in \mathbb{R}^{N \times N}$ is symmetric positive definite.*

Proof. Clearly $\mathbf{B}$ is symmetric.
Let $\mathbf{v} = (v_1, \dots, v_M) \in \mathbb{R}^M$ then $v = \sum_{i=1}^M v_i \varphi_i \in V$ and we have:

$$\mathbf{v}^T \mathbf{B} \mathbf{v} = \sum_{i,j=1}^M v_i v_j b(\varphi_j, \varphi_i) = b(v, v) \geq C_{\text{coer}} \|w\|^2$$

where we have used the biliniearity and the coercivity of b. □

1.11.1 Trace Inequalities

We shall now recall inequalities bounding a boundary norm with a norm over the whole domain, commonly called *trace inequalities*. In 1d this means estimating the absolute value of a function at a boundary node, by a norm over the whole interval, which simplifies the proofs a lot. Still let it be said here, that these kind of inequalities hold in higher dimensions as well. General results can for example be found in Ern [5].
This subsection will contain two results, one for general H^1-functions and one for polynomials.

We start by considering the finite dimensional case, inspired by [8].

Lemma 1.12 (Discrete trace inequality)**.** *Let $r \geq 1$ be the polynomial degree, $a, b \in \mathbb{R}$ with $a < b$ and let $\mathcal{P}^r([a,b])$ denote the space of polynomials of degree r defined on $[a,b]$.*
For any $v \in \mathcal{P}^r([a,b])$ we have:

(i) $|v(a)|^2 \leq \dfrac{(r+1)^2}{|b-a|} \|v\|^2_{L^2([a,b])}$

(ii) $|v(b)|^2 \leq \dfrac{(r+1)^2}{|b-a|} \|v\|^2_{L^2([a,b])}$

Proof. We will prove the statements first for the reference element $\hat{I} = [-1, 1]$ and then use a scaling argument to show the general case by applying a simple substitution.

Step 1 (*Setup*)**.**
We will make use of the Legendre orthonormal basis of $\mathcal{P}^r(\hat{I})$: Let $L_0, \dots, L_r$ denote the Legegndre polynomials on $\mathcal{P}^r(\hat{I})$. Recall the following well known facts (see for example [6]):

1. $\{L_0, \dots, L_r\}$ form an orthogonal basis of $\mathcal{P}^r(\hat{I})$ under the $L^2(\hat{I})$ inner product. Meaning:

$$\text{span}(L_0, \dots, L_r) = \mathcal{P}^r(\hat{I}), \quad \int_{-1}^1 L_i L_j \, \mathrm{d}\xi = \begin{cases} \frac{2}{2i+1}, & \text{for } i = j \\ 0, & \text{for } i \neq j \end{cases}$$

2. $L_i(1) = 1, L_i(-1) = (-1)^i, \qquad \forall i = 0, \dots, r$

Let $\psi_i = \sqrt{\frac{(2i+1)}{2}} L_i$ for $i = 0, \dots, r$ denote the normed basis function. Clearly we now have

$$\psi_i(-1) = (-1)^i \sqrt{\frac{2i+1}{2}}, \qquad \psi_i(1) = \sqrt{\frac{2i+1}{2}}, \qquad \int_{-1}^1 \psi_i \psi_j \mathrm{d}\xi = \delta_{i,j}, \qquad \forall i = 0, \dots, r$$

where $\delta_{i,j} = \begin{cases} 1, & \text{for } i = j \\ 0, & \text{for } i \neq j \end{cases}$, and hence $\{\psi_0, \dots, \psi_r\}$ form an orthonomal basis.

Step 2 (*Proof on reference element*)**.**

For any $v \in \mathcal{P}^r(\hat{I})$ there exist coefficients $v_0, \dots, v_r \in \mathbb{R}$, such that $v = \sum_{i=0}^r v_i \psi_i$. By applying Cauchy-Schwarz we find

$$|v(-1)|^2 = \Big|\sum_{i=0}^r v_i \psi_i(-1)\Big|^2 \leq \Big(\sum_{i=0}^r v_i^2\Big)\Big(\sum_{i=0}^r \psi_i(-1)^2\Big) = \Big(\sum_{i=0}^r v_i^2\Big)\Big(\sum_{i=0}^r \frac{2i+1}{2}\Big) = \Big(\sum_{i=0}^r v_i^2\Big)\frac{(r+1)^2}{2}$$

and finally the orthonormality of the ψ_i yields

$$\frac{(r+1)^2}{2}\sum_{i=0}^r v_i^2 = \frac{(r+1)^2}{2}\sum_{i,j=0}^r v_i v_j \delta_{i,j} = \frac{(r+1)^2}{2}\|v\|^2_{L^2(\hat{I})}$$

This yields the first inequality for the reference element. The second inequality can be proven analogously.

Step 3 (*Scaling argument*)**.**

Now we assume that $v \in \mathcal{P}^r([a,b])$. Using the affine (element) map

$$F : [-1,1] \to [a,b], \xi \mapsto \frac{a+b}{2} + \frac{|b-a|}{2}\xi$$

we can pull v back to the reference element by defining $\widehat{v}(\xi) := v(F(\xi))$ for all $\xi \in \hat{I}$. Clearly $\widehat{v} \in \mathcal{P}^r(\hat{I})$ hence, by Step 2 we obtain

$$|v(a)|^2 = |\widehat{v}(F^{-1}(a))|^2 = |\widehat{v}(-1)|^2 \leq \frac{(r+1)^2}{2}\int_{-1}^1 \widehat{v}(\xi)^2 \mathrm{d}\xi = \frac{(r+1)^2}{2}\frac{2}{|b-a|}\|v\|^2_{L^2([a,b])}$$

where in the last equality we have applied a change of variable $x = F(\xi)$ to the integral. Applying the same line of reasoning to $|v(b)|^2$ proves both inequalities and so we are done.

□

Next we consider the continuous case.

Lemma 1.13 (Continuous trace inequality)**.** *Let $I = (a,b) \subset \mathbb{R}$ be an element of length $h := b-a$, then for any $v \in H^1(\Omega)$*

(i) $|v(a)| \leq h^{-1/2}\|v\|_{L^2(I)} + h^{1/2}\|v'\|_{L^2(I)}$;

(ii) $|v(b)| \leq h^{-1/2}\|v\|_{L^2(I)} + h^{1/2}\|v'\|_{L^2(I)}$.

Proof. We show (i), the proof of (ii) is analogous.
Define $\varphi(x) := 1 - \frac{(x-a)}{h}$, where $\varphi(a) = 1, \varphi(b) = 0$ and let $v \in H^1(I)$. By applying the fundamental theorem of calculus we can estimate

$$\begin{aligned}|v(a)| = |v(a)\varphi(a) - v(b)\varphi(b)| &= |\int_I (v\,\varphi)'\,\mathrm{d}x| = |\int_I v'(x)\varphi(x)\,\mathrm{d}x + \int_I v(x)\varphi'(x)\,\mathrm{d}x| \\ &\leq \int_I |v'(x)|\,\underbrace{|\varphi(x)|}_{\leq 1}\,\mathrm{d}x + \frac{1}{h}\int_I |v(x)|\,\mathrm{d}x,\end{aligned}$$

where in the last step we have used that $|\varphi'(x)| = h^{-1}$. Using Cauchy-Schwarz we finally estimate

$$\begin{aligned}|v(a)| &\leq \Big(\int_I 1\,\mathrm{d}x\Big)^{1/2}\Big(\int_I |v'(x)|^2\,\mathrm{d}x\Big)^{1/2} + \frac{1}{h}\Big(\int_I 1\,\mathrm{d}x\Big)^{1/2}\Big(\int_I |v(x)|^2\,\mathrm{d}x\Big)^{1/2} \\ &\leq h^{1/2}\|v'\|_{L^2(I)} + h^{-1/2}\|v\|_{L^2(I)},\end{aligned}$$

which proves (i).

□

Recall the in previous sections established notations, let $r \in \mathbb{N}$ denote the polynomial degree and $V_h^r(\mathcal{T}_h)$ be the discrete subspace.

Definition 1.14. *We define the* ***energy norm*** *on V_h by*

$$\|v\|_\epsilon^2 := \sum_{n=0}^{N} \int_{I_n} c(x)v'(x)^2 dx + \sum_{n=0}^{N+1} \mathtt{a}_n [\![v(x_n)]\!]^2 \tag{1.18}$$

where $\mathtt{a}$ *denotes the penalization term in (1.7).*

Lemma 1.15. *$\|\cdot\|_\epsilon$ defines a norm on V_h.*

Proof. Clearly we have $\|\lambda v\|_\epsilon = |\lambda| \, \|v\|_\epsilon$ for all $\lambda \in \mathbb{R}, v \in V_h$.

By definition we have $\mathtt{a}, c > 0$ and by extension $\|v\|_\epsilon \geq 0$ for all $v \in V_h$. Suppose now that $\|v\|_\epsilon = 0$ for some $v \in V_h$, then we must have $v|_{I_n} \equiv$ const and $[\![v(x_n)]\!] = 0$ for all n. So v must be constant on all elements and have a jump of zero at the element boundaries. These two facts combined imply that v is constant on all of Ω. By the definition of the jump at the boundary nodes of Ω it immediately follows that $v = 0$. Clearly $\|0\|_\epsilon = 0$, therefore $\|\cdot\|_\epsilon$ is positive definite.

Using $[\![v(x_n) + w(x_n)]\!] = [\![v(x_n)]\!] + [\![w(x_n)]\!] \quad \forall v, w \in V_h, n = 0, \ldots, N+1$ we find

$$\begin{aligned}\|v + w\|_\epsilon &\leq \Big(\sum_{n=0}^{N} (\|\sqrt{c}v'\|_{L^2(I_n)} + \|\sqrt{c}w'\|_{L^2(I_n)})^2 + \sum_{n=0}^{N+1} (\sqrt{\mathtt{a}_n}([\![v(x_n)]\!] + [\![w(x_n)]\!]))^2 \Big)^{1/2} \\ &\leq \|v\|_\epsilon + \|w\|_\epsilon\end{aligned}$$

where in the last inequality we have used the triangle inequality of the euclidian vector norm on $\mathbb{R}^{2N+3}$, with the vector given as

$$\mathbf{v} = \big[\|\sqrt{c}v'\|_{L^2(I_0)}, \ldots, \|\sqrt{c}v'\|_{L^2(I_N)}, \sqrt{\mathtt{a}_0}[\![v(x_0)]\!], \ldots, \sqrt{\mathtt{a}_{N+1}}[\![v(x_{N+1})]\!]\big]^T$$

this shows the triangle inequality for $\|\cdot\|_\epsilon$ and hence it is a norm. □

Theorem 1.16. *Let $r \in \mathbb{N}$, the bilinear form b_h in (1.8) is continuous on $V_h^r(\mathcal{T}_h)$ and if furthermore $\sigma \geq \frac{6(r+1)^2 c_{\max}}{c_{\min}}$, b_h is also coercive on $V_h^r(\mathcal{T}_h)$. The coercivity and continuity constants are given by*

$$C_{coer} = \frac{1}{2}, \qquad C_{cont} = (3 + \frac{5}{4}C_\sigma)$$

where $C_\sigma = \frac{(r+1)^2 c_{\max}}{\sigma c_{\min}}$

Proof. **Step 1** (*Coercivity*)**.**
Let $w \in V_h$. Note that

$$b_h(w, w) = \|w\|_\epsilon^2 - 2 \sum_{n=0}^{N+1} \{\!\{c(x_n)w'(x_n)\}\!\}[\![w(x_n)]\!] \tag{1.19}$$

To derive the coercivity of b_h we will estimate the term $2\sum_{n=0}^{N+1} \{\!\{c(x_n)w'(x_n)\}\!\}[\![w(x_n)]\!]$ from above applying Lemma 1.12 and additional smaller tools:

Using Young's inequality: $2ab \le a^2 + b^2, \forall a, b \in \mathbb{R}$, we estimate

$$\begin{aligned} 2\sum_{n=0}^{N+1} \{\!\{c(x_n)w'(x_n)\}\!\}[\![w(x_n)]\!] &= 2\sum_{n=0}^{N+1} \{\!\{c(x_n)w'(x_n)\}\!\}\Big(\frac{\mathtt{a}_n}{2}\Big)^{-1/2}\Big(\frac{\mathtt{a}_n}{2}\Big)^{1/2}[\![w(x_n)]\!] \\ &\le 2\sum_{n=0}^{N+1} \frac{\{\!\{c(x_n)w'(x_n)\}\!\}^2}{\mathtt{a}_n} + \frac{1}{2}\sum_{n=0}^{N+1} \mathtt{a}_n[\![w(x_n)]\!]^2 \end{aligned} \tag{1.20}$$

Recalling $\mathtt{a}_n = \sigma \mathtt{c}_n \mathtt{h}_n^{-1}$ from (1.7) and noting the relations $\mathtt{h}_n \le h_n, \mathtt{c}_n^{-1} \le c(x_n^-)^{-1}, c(x_n^+)^{-1}$ we find

$$\begin{aligned} \mathtt{a}_n^{-1} c(x_n^+) \le \frac{h_n}{\sigma}, \quad \mathtt{a}_n^{-1} c(x_n^-) \le \frac{h_{n-1}}{\sigma}, \quad \forall n = 1, \dots, N \\ \mathtt{a}_0^{-1} c(x_0^+) = \frac{h_0}{\sigma}, \quad \mathtt{a}_{N+1}^{-1} c(x_{N+1}^-) = \frac{h_N}{\sigma} \end{aligned}$$

applying this and the usefull inequality $(a+b)^2 \le 2a^2 + 2b^2$ yields

$$\begin{aligned} &2\sum_{n=0}^{N+1} \frac{\{\!\{c(x_n)w'(x_n)\}\!\}^2}{\mathtt{a}_n} \\ &= 2\sum_{n=1}^{N} \frac{1}{4\mathtt{a}_n}\Big(c(x_n^-)w'(x_n^-) + c(x_n^+)w'(x_n^+)\Big)^2 + \frac{2}{\mathtt{a}_0}\Big(c(x_0^+)w'(x_0^+)\Big)^2 + \frac{2}{\mathtt{a}_{N+1}}\Big(c(x_{N+1}^-)w'(x_{N+1}^-)\Big)^2 \\ &\le 2\sum_{n=1}^{N} \frac{1}{2\sigma}\Big(h_{n-1}c(x_n^-)w'(x_n^-)^2 + h_n c(x_n^+)w'(x_n^+)^2\Big) + \frac{2h_0}{\sigma}c(x_0^+)w'(x_0^+)^2 + \frac{2h_N}{\sigma}c(x_{N+1}^-)w'(x_{N+1}^-)^2 \\ &\le \frac{c_{\max}}{\sigma}\sum_{n=1}^{N}\Big(h_{n-1}w'(x_n^-)^2 + h_n w'(x_n^+)^2\Big) + \frac{2c_{\max}h_0}{\sigma}w'(x_0^+)^2 + \frac{2c_{\max}h_N}{\sigma}w'(x_{N+1}^-)^2 \end{aligned} \tag{1.21}$$

Since $w \in V_h$ is a (broken) polynomial, we can apply Lemma 1.12 elementwise and find

$$w'(x_n^+)^2, w'(x_{n+1}^-)^2 \le \frac{(r+1)^2}{h_n}\|w'\|^2_{L^2(I_n)} \quad \forall n = 0, \dots, N \tag{1.22}$$

By combining (1.21), (1.22) and inserting $1 = c_{\min}c_{\min}^{-1} \le c(x)c_{\min}^{-1} \quad \forall x \in \Omega$ we find

$$2\sum_{n=0}^{N+1} \frac{\{\!\{c(x_n)w'(x_n)\}\!\}^2}{\mathtt{a}_n} \le 3C_\sigma \sum_{n=0}^{N} \|\sqrt{c}w'\|^2_{L^2(I_n)} \tag{1.23}$$

for $C_\sigma := \dfrac{(r+1)^2 c_{\max}}{\sigma c_{\min}} > 0$.

Finally putting together (1.19), (1.20) and (1.23) yields

$$\begin{aligned} b_h(w,w) &\ge \|w\|_\epsilon^2 - 3C_\sigma \sum_{n=0}^{N} \|\sqrt{c}w'\|^2_{L^2(I_n)} - \frac{1}{2}\sum_{n=0}^{N+1} \mathtt{a}_n[\![w(x_n)]\!]^2 \\ &= (1 - 3C_\sigma)\sum_{n=0}^{N} \|\sqrt{c}w'\|^2_{L^2(I_n)} + \frac{1}{2}\sum_{n=0}^{N+1} \mathtt{a}_n[\![w(x_n)]\!]^2 \\ &\ge \frac{1}{2}\|w\|_\epsilon^2 \end{aligned}$$

for $\sigma \ge \frac{6(r+1)^2 c_{\max}}{c_{\min}}$, which proves the coercivity of b_h on V_h.

Step 2 (*Continuity*)**.**

The proof the continuity of b_h uses similar ideas as the coercivity proof. Let $u, v \in V_h$, by using Cauchy-Schwarz we immediately get

$$\begin{aligned}|b_h(u,v)| &\leq \sum_{n=0}^{N} \|\sqrt{c}u'\|_{L^2(I_n)} \|\sqrt{c}v'\|_{L^2(I_n)} + \sum_{n=0}^{N+1} \left|\{\!\{c(x_n)u'(x_n)\}\!\}[\![v(x_n)]\!]\right| \\ &+ \sum_{n=0}^{N+1} \left|\{\!\{c(x_n)v'(x_n)\}\!\}[\![u(x_n)]\!]\right| + \sum_{n=0}^{N+1} \mathtt{a}_n \left|[\![u(x_n)]\!][\![v(x_n)]\!]\right| \\ &=: T_{\text{ell}} + T_{\text{cons}}^{(u)} + T_{\text{cons}}^{(v)} + T_{\text{penal}} \end{aligned} \tag{1.24}$$

The goal is now to estimate the consistency terms T_{cons} from above by something of the form $\sum_{n=0}^{N+1} t_n(u)s_n(v) + \sum_{n=0}^{N+1} t_n(v)s_n(u)$, such that together with the terms $T_{\text{ell}}, T_{\text{penal}}$ we can use discrete Cauchy-Schwarz on the sums and hence separate them into a product of the two energy norms $C_{\text{cont}}\|u\|_h\|v\|_h$ scaled by a positive constant.

We will show the estimate of $T_{\text{cons}}^{(u)}$, the procedure to estimate $T_{\text{cons}}^{(v)}$ is analogous.
First rewrite

$$T_{\text{cons}}^{(u)} = \sum_{n=0}^{N+1} \left|\{\!\{c(x_n)u'(x_n)\}\!\}\mathtt{a}_n^{-1/2}\mathtt{a}_n^{1/2}[\![v(x_n)]\!]\right| \tag{1.25}$$

Next again using the definition of $\mathtt{a}$ and estimates as in Step 1 we find for interior faces $n = 1, \dots, N$

$$\left|\{\!\{c(x_n)u'(x_n)\}\!\}\right|\mathtt{a}_n^{-1/2} \leq \frac{1}{2}\sqrt{\frac{\mathtt{h}_n}{\sigma}}\sqrt{c_{\max}}\left(|u'(x_n^-)| + |u'(x_n^+)|\right)$$

and for the boundary faces

$$\left|\{\!\{c(x_0)u'(x_0)\}\!\}\right|\mathtt{a}_0^{-1/2} \leq \sqrt{\frac{\mathtt{h}_0}{\sigma}}\sqrt{c_{\max}}|u'(x_0^+)|, \quad \left|\{\!\{c(x_{N+1})u'(x_{N+1})\}\!\}\right|\mathtt{a}_{N+1}^{-1/2} \leq \sqrt{\frac{\mathtt{h}_N}{\sigma}}\sqrt{c_{\max}}|u'(x_{N+1}^-)|$$

Applying Lemma (1.12) yields for $\beta_n(u) := \sqrt{C_\sigma}\|\sqrt{c}u'\|_{L^2(I_n)}, n = 0, \dots, N$

$$\begin{aligned}&\left|\{\!\{c(x_n)u'(x_n)\}\!\}\right|\mathtt{a}_n^{-1/2} \leq \frac{\beta_{n-1}(u)}{2} + \frac{\beta_n(u)}{2} \quad \text{for } n = 1, \dots, N \\ &\left|\{\!\{c(x_0)u'(x_0)\}\!\}\right|\mathtt{a}_0^{-1/2} \leq \beta_0(u) \\ &\left|\{\!\{c(x_{N+1})u'(x_{N+1})\}\!\}\right|\mathtt{a}_{N+1}^{-1/2} \leq \beta_N(u)\end{aligned}$$

which we can now plug back into (1.25) to get

$$T_{\text{cons}}^{(u)} \leq \beta_0(u)\gamma_0(v) + \beta_N(u)\gamma_{N+1}(v) + \sum_{n=1}^{N} \frac{\beta_{n-1}(u)}{2}\gamma_n(v) + \sum_{n=1}^{N} \frac{\beta_n(u)}{2}\gamma_n(v) \tag{1.26}$$

for $\gamma_n(v) := \sqrt{\mathtt{a}_n}\left|[\![v(x_n)]\!]\right| \forall n = 0, \dots, N+1$. By furthermore denoting $\alpha_n(u) := \|\sqrt{c}u'\|_{L^2(I_n)}$ we can represent

$$T_{\text{ell}} = \sum_{n=0}^{N} \alpha_n(u)\alpha_n(v), \qquad T_{\text{penal}} = \sum_{n=0}^{N+1} \gamma_n(u)\gamma_n(v)$$

and in total for

$$\begin{aligned}\mathbf{u} &:= [\alpha_0(u),\dots,\alpha_N(u),\beta_0(u),\beta_N(u),\frac{\beta_0(u)}{2},\dots,\frac{\beta_{N-1}(u)}{2},\frac{\beta_1(u)}{2},\dots,\frac{\beta_N(u)}{2},\\ &\qquad \gamma_0(u),\gamma_{N+1}(u),\gamma_1(u),\dots,\gamma_N(u),\gamma_1(u),\dots,\gamma_N(u),\gamma_0(u),\dots,\gamma_{N+1}(u)]^T \in \mathbb{R}^{6N+7}\\ \mathbf{v} &:= [\alpha_0(v),\dots,\alpha_N(v),\gamma_0(v),\gamma_{N+1}(v),\gamma_1(v),\dots,\gamma_N(v),\gamma_1(v),\dots,\gamma_N(v),\\ &\qquad \beta_0(v),\beta_N(v),\frac{\beta_0(v)}{2},\dots,\frac{\beta_{N-1}(v)}{2},\frac{\beta_1(v)}{2},\dots,\frac{\beta_N(v)}{2},\gamma_0(v),\dots,\gamma_{N+1}(v)]^T \in \mathbb{R}^{6N+7}\end{aligned}$$

we get

$$\begin{aligned}&T_{\text{ell}} + T_{\text{cons}}^{(u)} + T_{\text{cons}}^{(v)} + T_{\text{penal}} \le \mathbf{u}^T\mathbf{v} \le |\mathbf{u}|\,|\mathbf{v}|\\ &\le \Big(\sum_{n=0}^{N}(1+\frac{5}{4}C_\sigma)\|\sqrt{c}u'\|^2_{L^2(I_n)} + 3\sum_{n=0}^{N+1}\mathtt{a}_n[\![u(x_n)]\!]^2\Big)^{1/2}\Big(\sum_{n=0}^{N}(1+\frac{5}{4}C_\sigma)\|\sqrt{c}v'\|^2_{L^2(I_n)} + 3\sum_{n=0}^{N+1}\mathtt{a}_n[\![v(x_n)]\!]^2\Big)^{1/2}\\ &\le C_{\text{cont}}\|u\|_\epsilon\|v\|_\epsilon\end{aligned}$$

where $C_{\text{cont}} := (3+\frac{5}{4}C_\sigma)$. This last estimate together with 1.24 proves the continuity of b_h. □

1.12 Extension of the Bilinear Form

To simplify the theory we will continue to assume the exact solution to be in $H^2(\Omega)$. When we intend to extend the bilinear form from V_h to a bigger space, which contains the exact solution, we run into the problem, that $V_h \not\subset H^1(\Omega)$, by extension we also have $V_h \not\subset H^2(\Omega)$. So in contrast to continuous FEM we have to use a bigger space allowing for discontinuities and including both V_h and $H^2(\Omega)$.
A very common approach (see for example [2]) is to choose a vector space sum

$$V := V_h + H^2(\Omega) = \{v \in L^2(\Omega)\,|\,v = v_h + \tilde{v},\ \text{for } v_h \in V_h, \tilde{v} \in H^2(\Omega)\} \tag{1.27}$$

Alternatively Rivière proposes the usage of *broken Sobolev spaces* in [7]. That is, for a given partition $\mathcal{T}_h$ of Ω we can define

$$H^k(\mathcal{T}_h) := \{v \in L^2(\Omega)\,|\,v|_I \in H^k(I), \forall I \in \mathcal{T}_h\}$$

for some $k \in \mathbb{N}_0$, this allows for discontinuities and we have $V_h \in H^2(\mathcal{T}_h)$. We will use V as defined in (1.27). Now we extend the bilinear form b_h and the linear functional ℓ_h from V_h to V and formally get

$$b : V \times V \to \mathbb{R}, \qquad \ell : V \to \mathbb{R} \tag{1.28}$$

where

$$\begin{aligned}b(u,v) &= \sum_{n=0}^{N}\int_{I_n} cu'v'\,\mathrm{d}x - \sum_{n=0}^{N+1}\{\!\{c(x_n)u'(x_n)\}\!\}[\![v(x_n)]\!] + \{\!\{c(x_n)v'(x_n)\}\!\}[\![u(x_n)]\!] + \sum_{n=0}^{N+1}\mathtt{a}_n[\![u(x_n)]\!][\![v(x_n)]\!]\\ \ell(v) &= (f,v)_{L^2(\Omega)} - g_1c(x_{N+1}^-)v'(x_{N+1}^-) + g_0c(x_0^+)v'(x_0^+) + \mathtt{a}_{N+1}g_1v(x_{N+1}^-) + \mathtt{a}_0g_0v(x_0^+)\end{aligned}$$

for $u, v \in V$.

This yields the SIPG variational formulation
Find $u \in V$ such that:

$$b(u,v) = \ell(v), \qquad \forall v \in V \tag{1.29}$$

Furthermore we extend the energy norm $\|\cdot\|_\epsilon$ as presented in (1.18) to V. Note that this still defines a norm, since in lemma 1.15 we never explicitly need the finite dimensionality of V_h.

Lower Regularity Solutions

The H^2-regularity assumption on the solution is a common one in the context of convergence theory, but still a rather strict one in general. If we assume only H^1-regularity, which is a reasonable assumption to make, we encounter the problem of undefined trace values. Recall that from the trace theorem it is known that L^2-regularity is not enough to yield uniquely defined trace values. In 1d this becomes apparent, since point values of L^2-functions are not well defined. In particular for a solution $u \in H^1(\Omega)$, we have that $u'(x_n^+), u'(x_n^-)$ are not well defined. Extending the bilinear form b_h to V requires some additional work, which is often done in one of two ways:

1. Lifting operators (see [4])
2. L^2-projection (see [2])

We will completely circumvent this issue by assuming H^2-regularity.

Recall that in the proof of Theorem 1.16 for coercivity and continuity of the bilinear form we have used the finite dimensionality of the polynomial spaces over each element in $\mathcal{T}_h$. This means that the coercivity and continuity properties cannot be simply applied to our extended form b. For this we would have to introduce either lifting or the L^2-projection as mentioned before.

1.12.1 Consistency of the SIPG Variational Formulation

The final matrix-vector system we solve (yielding our numerical solution) is equivalent to the fully discrete variational formulation (1.8), which in turn is a discretization of not the original weak problem (1.2), but rather the SIPG variational formulation (1.29). We now have to show, that any solution of the SIPG variational formulation is also a solution of the original weak problem and vice versa to ensure our Galerkin approximation approximates the wanted exact solution, i.e. that the SIPG variational formulation is consistent with the weak formulation of the problem.

Theorem 1.17. *The SIPG variational formulation is consistent. That is, for $u \in H^2(\Omega)$ we have: u is a solution of (1.2) if and only if u solves (1.29)*

Proof. **Step 1** (*"⇐"*)**.**
First suppose $u \in H^2(\Omega)$ is a solution of the SIPG variational formulation (1.29), then clearly

$$b(u,v) = \ell(v) \qquad \forall v \in C_c^\infty(\Omega) \subset H^2(\Omega) \tag{1.30}$$

since u, v are continuous we have

$$[\![u(x_n)]\!] = [\![v(x_n)]\!] = 0 \qquad \forall n \in \{1,\dots,N\}$$

and $v \in C_c^\infty(\Omega)$ implies

$$[\![v(x_0)]\!] = [\![v(x_{N+1})]\!] = \{\!\{c(x_0)v'(x_0)\}\!\} = \{\!\{c(x_{N+1})v'(x_{N+1})\}\!\} = 0$$

therefore (1.30) becomes

$$a(u,v) = (f,v)_{L^2(\Omega)} \qquad \forall v \in C_c^\infty(\Omega) \tag{1.31}$$

where a is the elliptic bilinear form of the weak formulation in (1.2).
Now to recover the boundary conditions first note that since $u \in H^2(\Omega)$ and u satisfies (1.31) we have

$$-(c\,u')' = f \qquad \text{a.e. in } \Omega \tag{1.32}$$

From here we follow exactly the derivation of the discrete SIPG variational formulation in section 1.3. We multiply (1.32) by a test function $v \in V$, integrate over the elements $I \in \mathcal{T}_h$ by parts, sum up over all elements and add the symmetry and penalty terms corresponding to the SIPG bilinear form, hence we find

$$b(u,v) = (f,v)_{L^2(\Omega)} - c(x_{N+1}^-)u(x_{N+1}^-)v'(x_{N+1}^-) + c(x_0^+)u(x_0^+)v'(x_0^+) + \mathtt{a}_{N+1}u(x_{N+1}^-)v(x_{N+1}^-) + \mathtt{a}_0 u(x_0^+)v(x_0^+) \tag{1.33}$$

Subtracting (1.33) from (1.29) yields

$$\Big(- c(x_{N+1}^-)v'(x_{N+1}^-) + \mathtt{a}_{N+1}v(x_{N+1}^-)\Big)\Big(u(x_{N+1}^-) - g_1\Big) + \Big(c(x_0^+)v'(x_0^+) + \mathtt{a}_0 v(x_0^+)\Big)\Big(u(x_0^+) - g_0\Big) = 0$$

Since the above equality holds for any $v \in V$ we can choose fitting test functions v to show that u satisfies the boundary conditions.

Step 2.
Now suppose $u \in H^2(\Omega)$ is a solution of the weak problem (1.2), as already argued in Step 1 we have

$$-(c\,u')' = f \qquad \text{a.e. in } \Omega$$

this yields

$$b(u,v) = (f,v)_{L^2(\Omega)} - c(x_{N+1}^-)u(x_{N+1}^-)v'(x_{N+1}^-) + c(x_0^+)u(x_0^+)v'(x_0^+) + \mathtt{a}_{N+1}u(x_{N+1}^-)v(x_{N+1}^-) + \mathtt{a}_0 u(x_0^+)v(x_0^+)$$

by the derivation steps recalled in Step 1. Applying the boundary conditions, which are fixed by the weak formulation (1.2), shows that u indeed satisfies the SIPG variational formulation (1.29) and we are done.

□

1.13 Error Analysis

In this section we analyze the error between the Galerkin approximation resulting from the discrete SIPG formulation and an exact solution. We will focus on the error in the energy norm $\|\cdot\|_h$ but also mention the resulting error measured in the L^2-norm.

We start by recalling an important polynomial approximation result for finite elements.
Fix a polynomial degree $r \geq 1$ and let $I = (a,b) \subset \mathbb{R}$ be an element of length $h = b - a$. We denote the linear interpolation operator by

$$\mathcal{I}_h^r : C(\bar{I}) \to \mathcal{P}^r(I), \tag{1.34}$$

where the distinct interpolation nodes are $\xi_0, \dots, \xi_r \in I$. For our purpose the exact positioning of these nodes can be arbitrary. For readability if the context allows it we will denote the i-th derivative of a function v by $\frac{\mathrm{d}^i}{\mathrm{d}x^i}v = v^{(i)}$.

Lemma 1.18 (Interpolant estimate)**.** *Let $v \in H^{r+1}(I)$, it holds that*

$$|v - \mathcal{I}_h^r v|_{H^m(I)} \leq h^{r+1-m}|v|_{H^{r+1}(I)} \qquad \forall m \in \{0, \dots, r+1\}$$

where $|v|_{H^{r+1}(I)} = \|v^{(r+1)}\|_{L^2(I)}$ denotes the H^{r+1}-seminorm.

Proof. Fix $m \in \{0, \dots, r\}$, we start by first proving the H^m-seminorm

$$|v - \mathcal{I}_h^r v|_{H^m(I)} \leq h^{r+1-m}|v|_{H^{r+1}(I)} \tag{1.35}$$

Define $e := v - \mathcal{I}_h^r v \in H^{r+1}(I) \subset C^r(I)$, by construction of the interpolation operator we have $e(\xi_i) = 0$ for all $i \in \{0, \dots, r\}$, where the ξ_i are the interpolation nodes.
By applying Rolle's theorem in succession we can find r zeroes of e', $r-1$ zeroes of e'', …, and one zero of $e^{(r)}$. Clearly in total there exist some points $y_0, \dots, y_r \in I$ such that

$$e^{(i)}(y_i) = 0 \qquad \forall i \in \{0, \dots, r\}.$$

Next we use the fundamental theorem of calculus inductively, let $x_m \in I$ arbitrary and write

$$\begin{aligned}
e^{(m)}(x_m) &= \underbrace{e^{(m)}(y_m)}_{=0} + \int_{y_m}^{x_m} e^{(m+1)}(x_{m+1})\ \mathrm{d}x_{m+1} \\
&= \underbrace{\int_{y_m}^{x_m} e^{(m+1)}(y_{m+1})\ \mathrm{d}x_{m+1}}_{=0} + \int_{y_m}^{x_m}\int_{y_{m+1}}^{x_{m+1}} e^{(m+2)}(x_{m+2})\ \mathrm{d}x_{m+1}\ \mathrm{d}x_{m+2} \\
&= \cdots = \underbrace{\int_{y_m}^{x_m}\cdots\int_{y_{r-1}}^{x_{r-1}} e^{(r)}(y_r)\ \mathrm{d}x_r\cdots\ \mathrm{d}x_{m+1}}_{=0} + \int_{y_m}^{x_m}\cdots\int_{y_r}^{x_r} e^{(r+1)}(x_{r+1})\ \mathrm{d}x_{r+1}\cdots\ \mathrm{d}x_{m+1}.
\end{aligned}$$

Squaring both sides of this equation and integrating over I yields

$$\begin{aligned}
\|e^{(m)}\|_{L^2(I)}^2 &= \int_I \Big(\int_{y_m}^{x_m} 1\cdot\int_{y_{m+1}}^{x_{m+1}}\cdots\int_{y_r}^{x_r} e^{(r+1)}(x_{r+1})\ \mathrm{d}x_{r+1}\cdots\ \mathrm{d}x_{m+2}\ \mathrm{d}x_{m+1}\Big)^2\ \mathrm{d}x_m \\
&\le \int_I \underbrace{|x_m - y_m|}_{\le h}\int_{y_m}^{x_m}\Big(\int_{y_{m+1}}^{x_{m+1}}\cdots\int_{y_r}^{x_r} e^{(r+1)}(x_{r+1})\ \mathrm{d}x_{r+1}\cdots\ \mathrm{d}x_{m+2}\Big)^2\ \mathrm{d}x_{m+1}\ \mathrm{d}x_m \\
&\le h\int_I\int_I\Big(\int_{y_{m+1}}^{x_{m+1}}\cdots\int_{y_r}^{x_r} e^{(r+1)}(x_{r+1})\ \mathrm{d}x_{r+}\cdots\ \mathrm{d}x_{m+2}\Big)^2\ \mathrm{d}x_{m+1}\ \mathrm{d}x_m \\
&\le h^2\int_I\Big(\int_{y_{m+1}}^{x_{m+1}}\cdots\int_{y_r}^{x_r} e^{(r+1)}(x_{r+1})\ \mathrm{d}x_{r+1}\cdots\ \mathrm{d}x_{m+2}\Big)^2\ \mathrm{d}x_{m+1} \\
&\le \cdots \le h^{2(r+1-m)}\int_I |e^{(r+1)}(x_{r+1})|^2\ \mathrm{d}x_{r+1} = h^{2(r+1-m)}|e|^2_{H^{r+1}(I)},
\end{aligned}$$

where we have applied Cauchy-Schwarz inductively. But now since $\mathcal{I}_h^r v \in \mathcal{P}^r(I)$ it holds that $(\mathcal{I}_h^r v)^{(r+1)} \equiv 0$ in I, which implies that $e^{(r+1)} = v^{(r+1)}$ and therefore $|e|^2_{H^{r+1}} = |v|^2_{H^{r+1}}$. This proves (1.35).
Since we have proven (1.35) for an arbitrary $m \in \{0,\dots,r\}$ we can conclude that

$$|v - \mathcal{I}_h^r v|_{H^m(I)} \le h^{r+1-m}|v|_{H^{r+1}(I)} \qquad \forall m \in \{0,\dots,r\}.$$

If $m = r+1$ then we immediately have $|v - \mathcal{I}_h^r v|_{H^m(I)} = |v|_{H^{r+1}(I)}$.

□

Suppose $u \in V$ satisfies the SIPG variational formulation (1.29) and $u_h \in V_h$ satisfies the discrete SIPG variational formulation (1.8) (here the polynomial degree does not matter), then since $V_h \subset V$ it holds that

$$b(u - u_h, v) = 0 \qquad \forall v \in V_h. \tag{1.36}$$

This is commonly known as *Galerkin orthogonality*.
We will now estimate the error of the Galerkin approximation in reference to the continuous solution. To do so recall the following setup.

Let $r \ge 1$ be a polynomial degree, $\mathcal{T}_h = \{I_n\}_{n=0}^{N_h}$, be a partition of Ω with meshsize $h = \max_{n\in\{0,\dots,N_h\}} h_n > 0$, where $h_n = |x_{n+1} - x_n|$ and $I_n = (x_n, x_{n+1})$ for $n = 0,\dots,N_h$.

Definition 1.19. *We define the linear interpolation operator*

$$\mathcal{I}_h^r : C^0(\Omega) \to V_h^r(\mathcal{T}_h) \cap C^0(\Omega),$$

such that the face nodes x_n are interpolated for all $n \in \{0,\dots,N_h\}$, i.e. for $v \in C^0(\Omega)$ we have

$$\begin{aligned}
\mathcal{I}_h^r v(x_n^-) &= v(x_n) \qquad \forall n \in \{1,\dots,N_h+1\}, \\
\mathcal{I}_h^r v(x_n^+) &= v(x_n) \qquad \forall n \in \{0,\dots,N_h\}.
\end{aligned}$$

Note that for $r > 2$ we require $r-1$ additional interpolation nodes in each element $I \in \mathcal{T}_h$ besides the two element boundary nodes, for our purpose these can be arbitrary as long as they are distinct.

Theorem 1.20 (Convergence in energy norm)**.** *Let $\sigma > 0$ big enough, such that the SIPG bilinear form is coercive on $V_h^r(\mathcal{T}_h)$. Let $u \in H^{r+1}(\Omega)$ be a solution of the SIPG variational formulation (1.29) and $u_h \in V_h^r(\mathcal{T}_h)$ be its Galerkin approximation, i.e. the solution of (1.8).*
Then there exists a constant $C_\epsilon(C_{coer}, c_{\max}, \sigma) > 0$ independent of the meshsize h and the solution u, such that

$$\|u - u_h\|_h \leq C_\epsilon h^r \, |u|_{H^{r+1}(\Omega)}. \tag{1.37}$$

Remark 1.21. *1. The regularity assumption on the solution u is essential, if u is of lower regularity than the polynomial degree, then the optimal convergence rate is not achieved.*

2. Recall $c_{\max}$ is the maximum of the coefficient c in the pde (1.1a) and C_{coer} is the coercivity constant from theorem 1.16.

Proof. **Step 1.**
We interpolate the solution u and shall for readability write $\mathcal{I}_h^r u = \mathcal{I}_h u \in V_h$. By the triangle inequality we can estimate

$$\|u - u_h\|_\epsilon \leq \|u - \mathcal{I}_h u\|_\epsilon + \|\mathcal{I}_h u - u_h\|_\epsilon. \tag{1.38}$$

By the construction of the interpolant we know that $[\![(u - \mathcal{I}_h u)(x_n)]\!] = 0$ for all $n \in \{0, \dots, N_h + 1\}$, so the first term of (1.38) becomes

$$\begin{aligned} \|u - \mathcal{I}_h u\|_\epsilon \leq \|\sqrt{c}(u - \mathcal{I}_h u)'\|_{L^2(\Omega)} &\leq \sqrt{c_{\max}} \Big(\sum_{n=0}^{N_h} |u - \mathcal{I}_h u|^2_{H^1(I_n)} \Big)^{1/2} \\ &\leq \sqrt{c_{\max}}\, h^r \Big(\sum_{n=0}^{N_h} |u|^2_{H^{r+1}(I_n)} \Big)^{1/2} = \sqrt{c_{\max}}\, h^r \, |u|_{H^{r+1}(\Omega)}, \end{aligned} \tag{1.39}$$

where in the last inequality we have used the interpolant estimate (lemma 1.18).

Step 2.
Only the second term of (1.38) remains to be estimated. Define $e_h := u_h - \mathcal{I}_h u \in V_h$ and note that

$$b(e_h, e_h) = \underbrace{b(u_h - u, e_h)}_{=0} + b(u - \mathcal{I}_h u, e_h),$$

by Galerkin orthogonality (1.36). Using the coercivity of the bilinear form b on V_h we can estimate

$$\begin{aligned} C_{\text{coer}} \|e_h\|_h^2 \leq b(e_h, e_h) &= b(u - \mathcal{I}_h u, e_h) \\ &= \Big| \underbrace{\sum_{n=0}^{N_h} \int_{I_n} c\,(u - \mathcal{I}_h u)'\, e_h' \, \mathrm{d}x}_{=:T_1} - \underbrace{\sum_{n=0}^{N_h+1} \{\!\{c(x_n)(u - \mathcal{I}_h u)'(x_n)\}\!\} [\![e_h(x_n)]\!]}_{=:T_2} \\ &\quad - \sum_{n=0}^{N_h+1} \{\!\{c(x_n) e_h'(x_n)\}\!\} \underbrace{[\![(u - \mathcal{I}_h u)(x_n)]\!]}_{=0} + \sum_{n=0}^{N_h+1} \mathsf{a}_n \underbrace{[\![(u - \mathcal{I}_h u)(x_n)]\!]}_{=0} [\![e_h(x_n)]\!] \Big| \\ &= |T_1 - T_2|. \end{aligned}$$

We first estimate $|T_1|$, using Cauchy-Schwarz and Young's inequality: $ab \leq \frac{a^2}{2\varepsilon} + \frac{b^2 \varepsilon}{2}$, for $a, b \in \mathbb{R}, \varepsilon > 0$, we

find

$$|T_1| \le \sqrt{c_{\max}} \sum_{n=0}^{N_h} |u - \mathcal{I}_h u|_{H^1(I_n)} \|\sqrt{c}\, e_h'\|_{L^2(I_n)} \quad \le \sqrt{c_{\max}} \Big(\sum_{n=0}^{N_h} |u - \mathcal{I}_h u|^2_{H^1(I_n)} \Big)^{1/2} \underbrace{\Big(\sum_{n=0}^{N_h} \|\sqrt{c}\, e_h'\|^2_{L^2(I_n)} \Big)^{1/2}}_{\le \|e_h\|_\epsilon}$$

$$\le \frac{c_{\max}}{C_{\text{coer}}} \sum_{n=0}^{N_h} |u - \mathcal{I}_h u|^2_{H^1(I_n)} + \frac{C_{\text{coer}}}{4} \|e_h\|_h^2,$$

having used $\varepsilon = \frac{C_{\text{coer}}}{2}$. Finally applying lemma 1.18 again yields

$$|T_1| \le \frac{c_{\max}}{C_{\text{coer}}} h^{2r} \, |u|^2_{H^{r+1}(\Omega)} + \frac{C_{\text{coer}}}{4} \|e_h\|_h^2. \tag{1.40}$$

To estimate T_2 we make again use of Young's inequality with $\varepsilon = \dfrac{\mathtt{a}_n}{2} C_{\text{coer}}$ and find

$$|T_2| = \Big| \sum_{n=0}^{N_h+1} \{\!\{ c(x_n)(u - \mathcal{I}_n u)'(x_n) \}\!\} [\![e_h(x_n)]\!] \Big| \le \Big| \sum_{n=0}^{N_h+1} \frac{\{\!\{ c(x_n)(u - \mathcal{I}_n u)'(x_n) \}\!\}^2}{\mathtt{a}_n \, C_{\text{coer}}} \Big| + \frac{C_{\text{coer}}}{4} \sum_{n=0}^{N_h+1} \mathtt{a}_n [\![e_h(x_n)]\!]^2$$

$$\le \Big| \sum_{n=0}^{N_h+1} \frac{\{\!\{ c(x_n)(u - \mathcal{I}_n u)'(x_n) \}\!\}^2}{\mathtt{a}_n \, C_{\text{coer}}} \Big| + \frac{C_{\text{coer}}}{4} \|e_h\|_\epsilon^2.$$

Fix $n \in \{1, \dots, N_h\}$, then x_n is an interior face node and shared boundary of the elements I_{n-1}, I_n. Using the continuous trace inequality (lemma 1.13) and consequently the interpolant estimate (lemma 1.18) we estimate

$$\begin{aligned} |(u - \mathcal{I}_h u)'(x_n^+)|^2 &\le 2h_n^{-1} |u - \mathcal{I}_h u|^2_{H^1(I_n)} + 2h_n |u - \mathcal{I}_h u|^2_{H^2(I_n)} \\ &\le 2h_n^{-1} \, h_n^{2r} |u|^2_{H^{r+1}(I_n)} + 2h_n \, h_n^{2(r-1)} |u|^2_{H^{r+1}(I_n)} = 4h_n^{2r-1} \, |u|^2_{H^{r+1}(I_n)}, \end{aligned}$$

and analogously we find

$$|(u - \mathcal{I}_h u)'(x_n^-)|^2 \le 4h_{n-1}^{2r-1} \, |u|^2_{H^{r+1}(I_{n-1})}$$

for the lower element. Recall that $\mathtt{a}_n = \dfrac{\sigma \, \max(c(x_n^+), c(x_n^-))}{\min(h_{n-1}, h_n)}$ (see 1.7), using this and the estimates above we find

$$\begin{aligned} \frac{\{\!\{ c(x_n)(u - \mathcal{I}_h u)'(x_n) \}\!\}^2}{\mathtt{a}_n C_{\text{coer}}} &= \frac{c(x_n)^2 \min(h_{n-1}, h_n)}{\sigma C_{\text{coer}} \, \max(c(x_n^+), c(x_n^-))} \frac{|(u - \mathcal{I}_h u)'(x_n^-) + (u - \mathcal{I}_h u)'(x_n^+)|^2}{4} \\ &\le \frac{c(x_n)}{\sigma C_{\text{coer}}} \frac{\min(h_{n-1}, h_n)}{2} \Big(|(u - \mathcal{I}_h u)'(x_n^-)|^2 + |(u - \mathcal{I}_h u)'(x_n^+)|^2 \Big) \\ &\le \frac{c(x_n)}{\sigma C_{\text{coer}}} \frac{\min(h_{n-1}, h_n)}{2} \Big(4h_{n-1}^{2r-1} \, |u|^2_{H^{r+1}(I_{n-1})} + 4h_n^{2r-1} \, |u|^2_{H^{r+1}(I_n)} \Big) \\ &\le \frac{2c_{\max}}{\sigma C_{\text{coer}}} h^{2r} \Big(|u|^2_{H^{r+1}(I_{n-1})} + |u|^2_{H^{r+1}(I_n)} \Big). \end{aligned}$$

Similarly for the boundary nodes, i.e. $n \in \{0, N_h + 1\}$ we find

$$\frac{\{\!\{ c(x_n)(u - \mathcal{I}_h u)'(x_n) \}\!\}^2}{\mathtt{a}_n C_{\text{coer}}} \le \frac{4c_{\max}}{\sigma C_{\text{coer}}} h^{2r} |u|^2_{H^{r+1}(I_n)}.$$

This gives us now the tools to complete the estimate of $|T_2|$, we find

$$|T_2| \le \frac{6c_{\max}}{\sigma C_{\text{coer}}} h^{2r} |u|^2_{H^{r+1}(\Omega)} + \frac{C_{\text{coer}}}{4} \|e_h\|_\epsilon^2. \tag{1.41}$$

Combining the estimates (1.40), (1.41) yields

$$C_{\text{coer}}\|e_h\|_\epsilon^2 \leq \frac{C_{\text{coer}}}{2}\|e_h\|_\epsilon^2 + \frac{c_{\max}}{C_{\text{coer}}}(1+\frac{6}{\sigma})h^{2r}|u|^2_{H^{r+1}(\Omega)}.$$

By subtracting $\frac{C_{\text{coer}}}{2}\|e_h\|_\epsilon^2$ from both sides and then multiplying by $\frac{2}{C_{\text{coer}}}$ we find

$$\|e_h\|_\epsilon^2 \leq \frac{2c_{\max}}{(C_{\text{coer}})^2}\frac{6+\sigma}{\sigma}h^{2r}|u|^2_{H^{r+1}(\Omega)}.$$

This, together with the estimate (1.39), we can now finally plug back into (1.38) and get

$$\|u-u_h\|_\epsilon \leq \sqrt{c_{\max}}\Big(1+\frac{\sqrt{2(\sigma+6)}}{C_{\text{coer}}\sqrt{\sigma}}\Big)\,h^r|u|_{H^{r+1}(\Omega)} = C_\epsilon h^r|u|_{H^{r+1}(\Omega)},$$

for $C_\epsilon := \sqrt{c_{\max}}\Big(1+\frac{\sqrt{2(\sigma+6)}}{C_{\text{coer}}\sqrt{\sigma}}\Big)$, which proofs (1.37).

□

Next we show that measuring the error in the L^2-norm gives us a power of h more than in the energy norm. This coincides with the convergence rates known from continuous FEM.

Theorem 1.22 (L^2-convergence)**.** *Under the same assumptions as in theorem 1.20, there exists a constant $C_L(C_\epsilon,\Omega,c,\sigma)>0$ independent of the meshsize h and the solution u, such that*

$$\|u-u_h\|_{L^2(\Omega)} \leq C_L h^{r+1}|u|_{H^{r+1}(\Omega)}. \tag{1.42}$$

Proof. We will prove the theorem using a duality argument often referred to as *Aubin-Nitsche trick*.
Step 1.
Define the error $e := u-u_h \in V$ and consider the dual problem

$$-(c\,\varphi')' = e \qquad \text{in } \Omega, \tag{1.43}$$
$$\varphi(0)=\varphi(1)=0. \tag{1.44}$$

By Lax-Milgram there exists a unique weak solution $\varphi \in H_0^1(\Omega)$ of (1.43)-(1.44). Furthermore by elliptic regularity it holds that $\varphi \in H^2(\Omega)$ with

$$\|\varphi\|_{H^2(\Omega)} \leq \tilde{C}\|e\|_{L^2(\Omega)}, \tag{1.45}$$

where $\tilde{C}>0$ is a constant only dependent on Ω and the coefficient $c \in C^1(\Omega)$(see for example Evans chapter 6.3 [1]). By the consistency of the SIPG variational formulation (see theorem 1.17) φ satisfies

$$b(\varphi,v) = (e,v)_{L^2(\Omega)} \qquad \forall v \in V.$$

Note that the right-hand side contains only the L^2-term since the dual problem is posed with homogeneous Dirichlet boundary conditions (1.44). In particular since $e \in V$ we have $b(\varphi,e) = \|e\|^2_{L^2(\Omega)}$.
Next we interpolate φ using the $\mathcal{P}^1$-interpolant $\mathcal{I}_h^1$ (not $\mathcal{P}^r$), this means $\mathcal{I}_h^1\varphi \in V_h^r(\mathcal{T}_h)$ since $r \geq 1$. Now using Galerkin orthogonality (1.36) and the symmetry of the bilinear form b we find

$$\|e\|^2_{L^2(\Omega)} = b(\varphi,e) = b(e,\varphi-\mathcal{I}_h^1\varphi),$$

which in turn we can estimate using Cauchy-Schwarz as follows:

$$\begin{aligned}
b(e,\varphi-\mathcal{I}_h^1\varphi) &= \sum_{n=0}^{N_h}\int_{I_n} c\,e'(\varphi-\mathcal{I}_h^1\varphi)'\,\mathrm{d}x - \sum_{n=0}^{N_h+1}\{\!\{c(x_n)(\varphi-\mathcal{I}_h^1\varphi)'(x_n)\}\!\}[\![e(x_n)]\!] \\
&\quad - \sum_{n=0}^{N_h+1}\{\!\{c(x_n)e'(x_n)\}\!\}\underbrace{[\![(\varphi-\mathcal{I}_h^1\varphi)(x_n)]\!]}_{=0} + \sum_{n=0}^{N_h+1}\mathtt{a}_n[\![e(x_n)]\!]\underbrace{[\![(\varphi-\mathcal{I}_h^1\varphi)(x_n)]\!]}_{=0} \\
&\le \sqrt{c_{\max}}\Big(\sum_{n=0}^{N_h}\int_{I_n}\left|(\varphi-\mathcal{I}_h^1\varphi)'\right|^2\mathrm{d}x\Big)^{1/2}\underbrace{\Big(\sum_{n=0}^{N_h}\int_{I_n}c\left|e'\right|^2\mathrm{d}x\Big)^{1/2}}_{\le\|e\|_\epsilon} \\
&\quad + \Big(\sum_{n=0}^{N_h+1}\frac{\{\!\{c(x_n)(\varphi-\mathcal{I}_h^1\varphi)'(x_n)\}\!\}^2}{\mathtt{a}_n}\Big)^{1/2}\underbrace{\Big(\sum_{n=0}^{N_h+1}\mathtt{a}_n[\![e(x_n)]\!]^2\Big)^{1/2}}_{\le\|e\|_\epsilon}.
\end{aligned}$$

We know from theorem 1.20 that $\|e\|_\epsilon \le C_\epsilon h^r |u|_{H^{r+1}(\Omega)}$, and from the proof that

1. $\left|\varphi-\mathcal{I}_h^1\varphi\right|_{H^1(\Omega)}^2 \le h^2\left|\varphi\right|_{H^2(\Omega)}^2 \qquad \forall n\in\{0,\dots,N_h\}$,
2. $\dfrac{\{\!\{c(x_n)(\varphi-\mathcal{I}_h^1\varphi)'(x_n)\}\!\}^2}{\mathtt{a}_n} \le \gamma_n\dfrac{c_{\max}}{\sigma}h^2\left(|\varphi|_{H^2(I_{n-1})}^2+|\varphi|_{H^2(I_n)}^2\right) \qquad \forall n\in\{0,\dots,N_h+1\}$,

where $\gamma_n = 2$ for $n\in\{1,\dots,N_h\}$ and $\gamma_n = 4$ for $n\in\{0,N_h+1\}$. Putting all of the above together we find

$$\|e\|_{L^2(\Omega)}^2 \le \Big(\sqrt{c_{\max}}+\sqrt{\frac{6c_{\max}}{\sigma}}\Big)C_\epsilon h^{r+1}\left|\varphi\right|_{H^2(\Omega)}|u|_{H^{r+1}(\Omega)} \le \Big(1+\sqrt{\frac{6}{\sigma}}\Big)\sqrt{c_{\max}}C_\epsilon\tilde{C}\,\|e\|_{L^2(\Omega)}\,|u|_{H^{r+1}(\Omega)},$$

where we have used the estimate (1.45). Dividing by $\|e\|_{L^2(\Omega)}$ on both sides finally yields

$$\|e\|_{L^2(\Omega)} \le C_L h^{r+1}\,|u|_{H^{r+1}(\Omega)},$$

where $C_L = \Big(1+\sqrt{\frac{6}{\sigma}}\Big)\sqrt{c_{\max}}C_\epsilon\tilde{C}$.

□

1.14 Numerical Results

1.14.1 Rate of Convergence

First we would like to test the code we have written in `MATLAB` and try to replicate the convergence rates provided in section 1.13. The procedure of doing so can be listed in the following steps:

Step 1 (*Domain*)**.**
We fix a domain Ω. In our case $\Omega = (0,1)$ stays unchanged for all the experiments of this chapter.

Step 2 (*Exact Solution*)**.**
We decide a priori which exact solution we intend to approximate. To simplify calculations we choose one of the simple smooth solutions:

$$u_1(x) = x^r, \qquad u_2(x) = e^{-x}\sin(5x),$$

where r corresponds to the polynomial degree of the finite element space.
u_1 is chosen to ensure the exactness of the method (since we approximate a polynomial of degree r by

a piecewise polynomial of degree r the finite element approximation should be exact up to floating-point errors).

Step 3 (*Coefficient*)**.**
We fix a smooth coefficient, one of

$$c_1(x) = 1, \qquad c_2(x) = \sin(x) + 2,$$

where having these two options allows us to first check if the method works for a homogeneous background material and then for the inhomogeneous case. To ensure our exact solution still satisfies the original pde, we calculate the forcing term f in (1.1a) symbolically, i.e. $f_{i,j} = -(c_j\, u_i')'$.

Step 4 (*Boundary Conditions*)**.**
We choose the type of boundary condition we would like to impose on each boundary. We have mainly focused on Dirichlet boundary conditions on both sides so far, so this is what we will consider. Let it be said, that the results for one Neumann and one Dirichlet (as mentioned in section 1.4) are very similar.
To actually impose the boundary condition we directly take the values of the previously fixed exact solution u at the respective boundary points, meaning

$$g_{0,i} = u_i(0), \qquad g_{1,i} = u_i(1).$$

Step 5 (*Mesh*)**.**
We choose an initial mesh, i.e. an initial partition of the domain Ω. For our purpose we only present the equidistant mesh

$$\mathcal{T}_h^{(0)} = \{(0, 0.25), (0.25, 0.5), (0.5, 0.75), (0.75, 1)\},$$

with initial meshsize $h = \frac{1}{4}$, but the results do not vary for any kind of initial mesh. In addition to the initial mesh we decide on a number of refinement cycles. Refining each element of the initial mesh by dividing it into two new elements and repeating this process for the number of refinement cycles fixed yields a sequence of nested meshes.

Step 6 (*Numerical Approximation*)**.**
Here we only present the case for $r \in \{1, 2\}$, meaning for $\mathcal{P}^1$-, $\mathcal{P}^2$-elements, but the expected convergence rates can be recovered for an arbitrary polynomial degree r. For $r > 6$ there is barely a convergence rate to be made out anymore since after only a couple of refinements the error flattens out and even starts to grow slightly. This happens due to the high condition number of the resulting stiffness matrix $\mathbf{B}$ and the high number of global degrees of freedom (basis nodes).
We also fix a big enough penalization parameter $\sigma = 10(r + 1)^2$ to ensure coercivity.

On each successive mesh $\mathcal{T}_h^{(l)}$ we find the numerical solution $u_h^{(l)}$ by solving the system (1.9).

Step 7 (*Error Computation*)**.**
We are interested in

1. the L^2-error

$$\|u - u_h\|_{L^2(\Omega)} = \Big(\int_\Omega |u(x) - u_h(x)|^2 \mathrm{d}x \Big)^{1/2} = \Big(\sum_{n=0}^{N} \int_{I_n} |u(x) - u_h(x)|^2 \mathrm{d}x \Big)^{1/2};$$

2. the broken H^1-error

$$\|u - u_h\|_{H^1(\mathcal{T}_h)} = \Big(\sum_{n=0}^{N} \int_{I_n} |u'(x) - u_h'(x)|^2 \mathrm{d}x \Big)^{1/2};$$

3. and finally the error in the energy norm

$$\|u - u_h\|_\epsilon = \Big(\sum_{n=0}^{N} \int_{I_n} c(x) \left| u'(x) - u_h'(x) \right|^2 \mathrm{d}x + \sum_{n=0}^{N+1} \mathtt{a}_n [\![u(x_n) - u_h(x_n)]\!]^2 \Big)^{1/2}.$$

We compute for each numerical solution $u_h^{(l)}$ all three named errors and plot them in a plot where both axes are logarithmically scaled.

First to ensure exactness we consider the polynomial exact solution u_1 with the constant coefficient c_1, as expected all three errors display minimal floating-point values of around 10^{-15}, slightly increasing with each further refinement due to the growing condition number of the system matrix. When considering u_1 with the non-trivial coefficient c_2 we do now get real decreasing error rates due to the introduced quadrature error when assembling the stiffness matrix and the load vector.

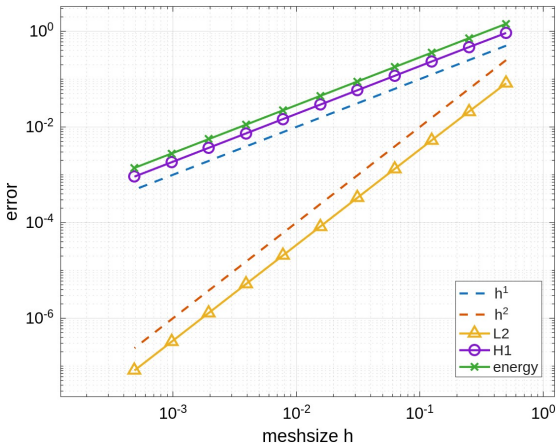

Figure 1.1: Errors of SIPG for P^1-elements

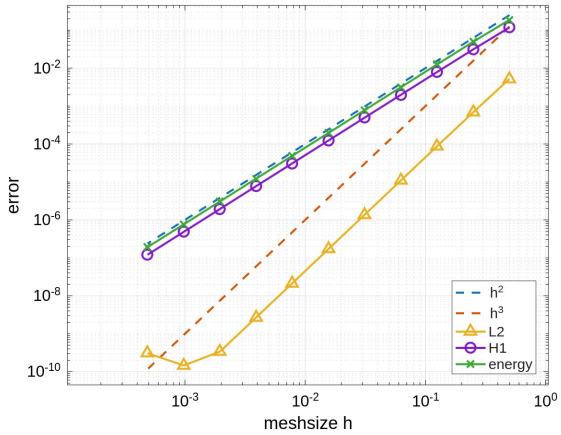

Figure 1.2: Errors of SIPG for P^2-elements

In the Figures 1.1, 1.2 we observe the expected convergence rates of $\mathcal{O}(h^r)$ in the broken H^1- and energy norm as well as $\mathcal{O}(h^{r+1})$ in the L^2-norm for the exact solution $u_2(x) = e^{-x}\sin(5x)$ and the coefficient $c_2(x) = \sin(x) + 2$. We note that the error in the energy norm and the error in the broken H^1-norm behave equivalently. Furthermore we observe that the error in the case of P^2-elements flattens out and starts to grow again after crossing the threshold of 10^{-10}, which as mentioned before is to be expected. In fact an error of 10^{-10} is way smaller than one hopes to achieve in reality using standard computational tools.
If instead we choose the constant coefficient $c_1(x) = 1$, the picture is the same.

1.14.2 Visualization of the SIPG Solution

Now we visualize the numerical solution approximating u_2 on the uniform mesh $\mathcal{T}_h^{(1)}$ (after one global refinement cycle) and play with the penalization parameter σ. If we penalize the jumps heavily by using the generously chosen $\sigma = 10(r+1)^2$ we can not observe any discontinuity in the numerical solution by eye (see figures 1.3 and 1.4). Also the boundary conditions seem to be exact, as they would be in the case of continuous FEM, of course this is not the case as we will see shortly. The circles in the figures 1.3 and 1.4 denote the face nodes of the elements in $\mathcal{T}_h^{(1)}$ (meaning the element boudnary points), in between each of those face nodes the numerical solution is a polynomial of degree r, which when comparing $r = 1$ and $r = 2$ as done here, becomes apparent.

If we relax the penalty by reducing the penalization parameter enough such that the bilinear form is no longer coercive, we can observe the jumps growing. If we look at the figures 1.5, 1.6, where we have

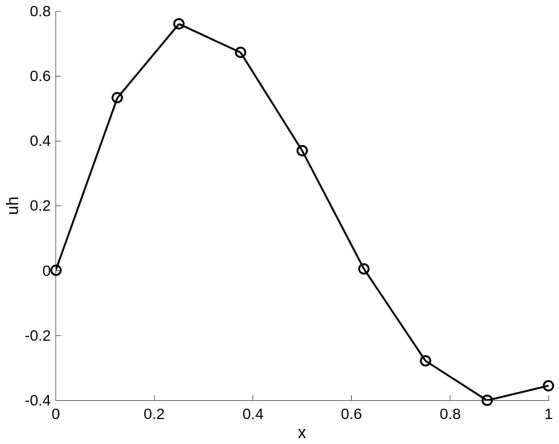

Figure 1.3: high penalty SIPG-approximation for P^1-elements

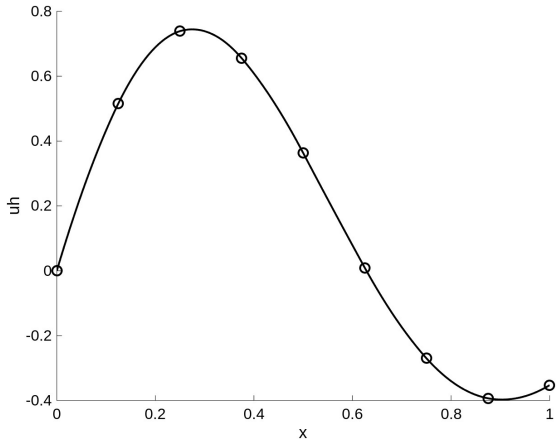

Figure 1.4: high penalty SIPG-approximation for P^2-elements

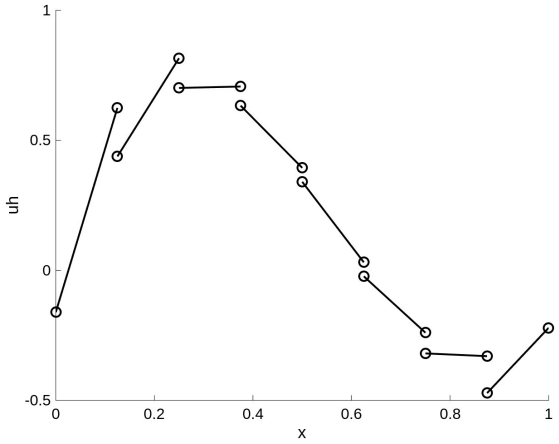

Figure 1.5: non-coercive SIPG-approximation for P^1-elements

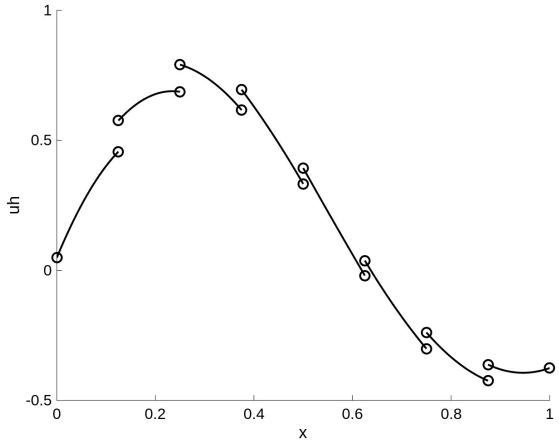

Figure 1.6: non-coercive SIPG-approximation for P^2-elements

set $\sigma = 1.1$, the discontinuities become apparent. The bilinear form loosing it's coercivity means that the system we solve in (1.9) is not invertible anymore and we end up solving a least squares problem. Here we have chosen extreme edge examples of σ on purpose to illustrate the effect of modulating the penalization parameter and to visually show the discontinuity of a DG-solution. Maybe interesting to observe is the effect of enforcing boundary conditions weakly. Clearly here the Dirichlet boundary condition $u(0) = 0$ is not fulfilled by the numerical solutions, which would be the case for strongly enforced boundary conditions.

1.14.3 Influence of the Quadrature Rule on the Convergence Rate

As noted in section 1.7 by using $r + 1$ Gauss-Lobatto nodes (in the context of $\mathcal{P}^r$-elements) as quadrature nodes and basis nodes at the same time introduces an error when assembling the mass matrix of the system. In this subsection we experimentally compare the results of using $r + 1$ Gauss-Lobatto quadrature nodes to approximate the integrals versus using the exact integration values. To be precise we fix a polynomial degree r and our Lagrangian basis with $r + 1$ Gauss-Lobatto nodes as specified in section 1.6 and approximate all integrals first using the Gauss-Lobatto quadrature rule with $r+1$ nodes, then with $r+2$ nodes and compare the two.

First we have to consider a slightly different elliptical problem, which requires a mass matrix.

$$-(c(x)u'(x))' + u(x) = f(x) \qquad \forall x \in \Omega,$$

$$u(0) = g_0, u(1) = g_1,$$

We can apply the exact same tools as in the derivation of the variational formulation in section 1.3 and get the discrete SIPG variational formulation.
Find $u_h \in V_h$ such that:

$$b_h(u_h, v) + (u_h, v)_{L^2(\Omega)} = \ell_h(v), \qquad \forall v \in V_h, \tag{1.46}$$

Now analogously to section 1.5 we write $u_h = \sum_{m=0}^{N} \sum_{j=0}^{r} \alpha_j^m \Phi_j^m \in V_h$ and find that (1.46) is equivalent to

$$\sum_{m=0}^{N} \sum_{j=0}^{r} \alpha_j^m \left(b_h(\Phi_j^m, \Phi_i^n) + (\Phi_j^m, \Phi_i^n)_{L^2(\Omega)} \right) = \ell_h(\Phi_i^n) \qquad \forall i \in \{0, \dots, r\}, n \in \{0, \dots, N\},$$

which in turn is equivalent to the Matrix-Vector system

$$\Big(\mathbf{B} + \mathbf{M}\Big)\mathbf{u} = \mathbf{l}, \tag{1.47}$$

where as before $[\mathbf{B}]_{T(n,i),T(m,j)} = b_h(\Phi_j^m, \Phi_i^n)$, $[\mathbf{u}]_{T(m,j)} = \alpha_j^m$, $[\mathbf{l}]_{T(n,i)} = \ell_h(\Phi_i^n)$ and furthermore $[\mathbf{M}]_{T(n,i),T(m,j)} = (\Phi_j^m, \Phi_i^n)_{L^2(\Omega)}$.

We now consider a similar setting as described in subsection 1.14.1 and test the convergence rates for $\mathcal{P}^1, \mathcal{P}^2$-elements, where we choose the exact solution $u_2(x) = e^{-x}\sin(5x)$ and the strongly oscillating coefficient $c_3(x) = \sin(20x) + 2$. We distinguish the following two methods of calculating numerical solutions.

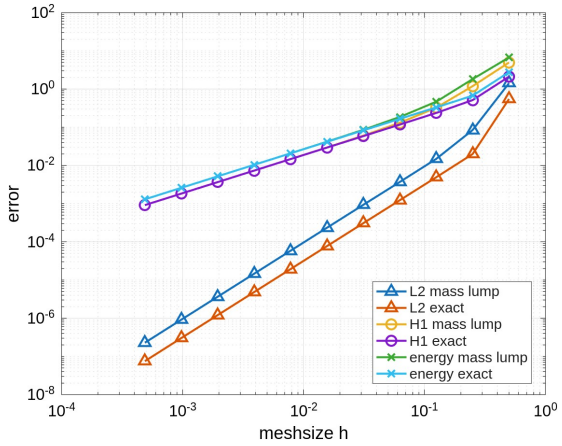

Figure 1.7: Comparison of convergence rates of higher order vs lower order quadrature for $\mathcal{P}^1$-elements

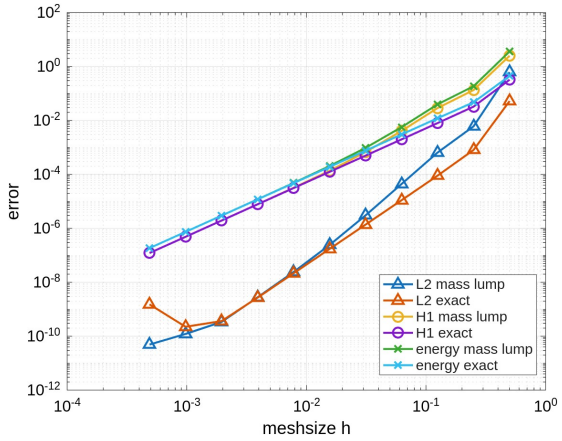

Figure 1.8: Comparison of convergence rates of higher order vs lower order quadrature for $\mathcal{P}^2$-elements

Method A: "Low Order"

We assemble the matrices of the system (1.47) as described in section 1.8 and approximate the integrals appearing in the entries of $\mathbf{A}$, $\mathbf{M}$, $\mathbf{l}$ using the (lower order) $r+1$ node Gauss-Lobatto quadrature rule, where $r \in \{1, 2\}$, meaning the quadrature nodes coincide with the basis nodes and the mass matrix is mass-lumped, hence the name. We calculate the L^2-, $H^1(\mathcal{T}_h)$- and energy error between the numerical solution and the exact solution on all meshes.

Method B: "High Order"

We assemble the matrices of the system (1.47) as described in section 1.8 and approximate the integrals appearing in the entries of $\mathbf{A}$, $\mathbf{M}$, $\mathbf{l}$ using the (higher order) $r+2$ node Gauss-Lobatto quadrature rule, where $r \in \{1, 2\}$, meaning the mass matrix is calculated exactly. We calculate the L^2-, $H^1(\mathcal{T}_h)$- and energy error between the numerical solution and the exact solution on all meshes.

It turns out that there is only a very slight difference in accuracy, plotting the convergence rates of both methods into the same plot does not yield a very informative picture as is visible in Figures 1.7, 1.8. Instead since we expect the error resulting from Method B to be smaller than the error from Method A, we subtract Error B from Error A and plot this difference into a loglog plot, see Figures 1.9, 1.10.

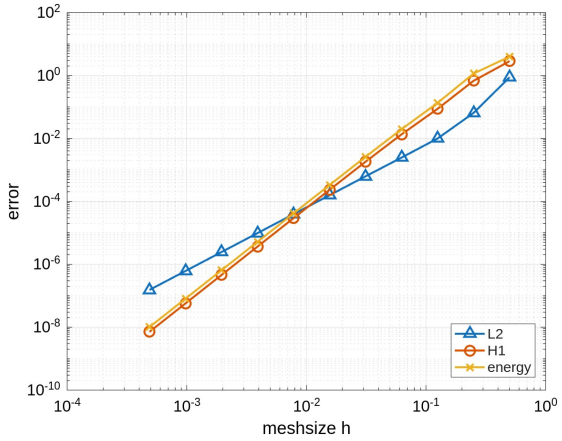

Figure 1.9: Lower order error minus higher order error for $\mathcal{P}^1$-elements

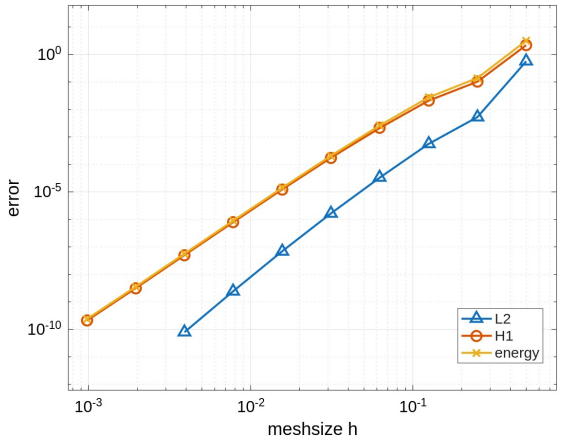

Figure 1.10: Lower order error minus higher order error for $\mathcal{P}^2$-elements

We observe that the convergence rate is not affected by using a mass lumping strategy. Clearly we only gain a slight bit of accuracy by employing a higher order quadrature rule. This speaks for using mass lumping when needed. On the other hand since we already have a block-diagonal mass matrix in DG there is really no benefit to using mass lumping, we simply sacrifice accuracy. In a more realistic application we might be more interested in accuracy than convergence rates and especially if the forcing term f or the coefficient c rapidly change in a certain part of the domain, loosing out on accuracy might be unnecessary. In practice it is the norm to use a higher order quadrature rule if possible.

1.14.4 Modeling an Inhomogeneous Membrane

To round off this chapter we now present a small intuitive example of the pde (1.1a)-(1.1b). This subsection is inspired by the step-5 tutorial program of the `deal.II` [1] website.
Imagine a thin one dimensional membrane of a certain homogeneous material being spanned over the domain $\Omega = (0, 1)$, such that the left end is fixed at the point $x = 0$ to a height of $y = 0$ and the right end is fixed at the point $x = 1$ to a height of $y = 0$. Since the membrane is in a relaxed state and pinned to a fixed height at both ends the picture we get is just a flat membrane. We can describe the vertical displacement of the membrane by a function $u : \Omega \to \mathbb{R}$. This function is harmonic with zero Dirichlet boundary conditions in the relaxed state, so the displacement satisfies $u \equiv 0$.

Now we can introduce a constant force $f \equiv 1$ on Ω pushing the membrane up. The displacement of the membrane is now a solution of the Poisson equation

$$-u'' = f \qquad \text{in } \Omega,$$
$$u(0) = u(1) = 0.$$

[1] www.dealii.org

As a final modification we allow for an inhomogeneous stiffness of the membrane, for example by letting its thickness vary over Ω. The stiffness is described by the coefficient

$$c(x) = \begin{cases} 1, & x \in [0, 0.3) \cup (0.7, 1] \\ 20, & x \in [0.3, 0.7] \end{cases}$$

so that the membrane is twenty times thicker in the center region than closer to the boundary. Now the displacement u is a solution of the problem (1.1a)-(1.1b) with $g_0 = g_1 = 0$ and $f \equiv 1$.

We discretize the domain into a non equidistant partition, where we allow for smaller elements in the region of higher stiffness around $[0.3, 0.7]$. We simulate the displacement u with $\mathcal{P}^2$-elements using the SIPG variational form.

In Figure 1.11 we observe the characteristic shape of a pressurized membrane corresponging to a solution of the Poisson equation. The inhomogeneity in the center, where the membrane displays higher stiffness, barely deforms in response to the pressure introduced by f, this yields a flattened plateau in the interval $[0.3, 0.7]$.

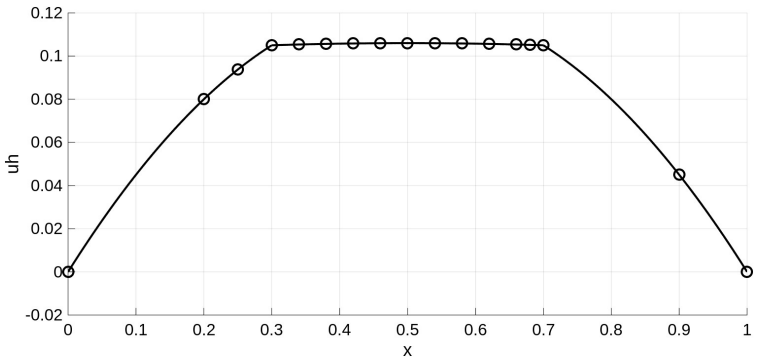

Figure 1.11: Displacement of inhomogeneous membrane with $\mathcal{P}^2$-elements

Chapter 2

DG for the Wave Equation

In the last chapter we have derived the necessary theoretical tools and built code to solve the time-independent elliptic problem. This elliptic problem coincides with a time-independent version of the hyperbolic problem

$$u_{tt} - (cu_x)_x = f.$$

We will use a *method of lines* approach to first discretize in space with discontinuous Galerkin finite elements, using what we have derived in the last chapter, and then discretize in time using leapfrog time-integration to solve the problem numerically. This chapter is structured similarly to the last one. First we pose the problem and recall some analysis on regularity, existence and uniqueness of solutions. Next we discretize both in time and space deriving the fully discrete scheme. We will then again elaborate on the actual implementation of the program as well as recalling convergence theory. Finally we will present the numerical experiments reproducing the expected convergence rates where exact solutions are given and simulate the behavior of waves propagating through a waveguide with material properties varying in space and time.

2.1 Problem

Let $\Omega = (0, L)$ be the domain (waveguide) for some $L > 0$ and let $T > 0$ be the endtime. The 1d wave equation with a time-dependent coefficient is given by

$$u_{tt}(x,t) - (c(x,t)u_x(x,t))_x = f(x,t) \qquad \forall (x,t) \in \Omega \times (0,T], \tag{2.1a}$$

$$u(0,t) = g_0(t), \quad u(L,t) = g_L(t) \qquad \forall t \in [0,T], \tag{2.1b}$$

$$u(x,0) = u_0(x), \quad u_t(x,0) = v_0(x) \qquad \forall x \in \Omega. \tag{2.1c}$$

Similarly to the elliptic case we require the coefficient $c \in C^1(\overline{\Omega} \times [0,T])$ to be bounded by

$$0 < c_{\min} \leq c(x,t) \leq c_{\max} < \infty \qquad \forall (x,t) \in \overline{\Omega} \times [0,T]. \tag{2.2}$$

We assume the forcing term $f \in L^2(0,T;L^2(\Omega))$, the initial displacement $u_0 \in H^1(\Omega)$, and the initial velocity $v_0 \in L^2(\Omega)$. First we assume homogeneous Dirichlet boundary conditions, meaning $g_0 \equiv g_1 \equiv 0$. Multiplying the pde (2.1a) by a test function $v \in C_c^\infty(\Omega)$ and integrating by parts gives us the weak formulation:

Find $u \in L^2(0,T;H_0^1(\Omega))$, with $u_t \in L^2(0,T;L^2(\Omega)), u_{tt} \in L^2(0,T;H^{-1}(\Omega))$, such that

$$\langle u_{tt}(t), v\rangle + a(u(t), v; t) = (f(t), v)_{L^2(\Omega)} \qquad \forall v \in C_c^\infty(\Omega) \quad \text{for a.e. } t \in (0,T), \tag{2.3a}$$

$$u(\cdot, 0) = u_0, \quad u_t(\cdot, 0) = v_0, \tag{2.3b}$$

where

$$a(u(t), v; t) = \int_\Omega c(x,t)u_x(x,t)v_x(x)\mathrm{d}x$$

denotes the standard (time-dependent) elliptic bilinear form and $\langle \cdot, \cdot \rangle$ denotes the duality pairing between $H^{-1}(\Omega)$ and $H_0^1(\Omega)$. Note that both the time derivatives as well as the space derivatives have to be understood in a weak sense. It is well known that the problem (2.3) is well posed (see section 7.2 Theorem 3 in [1]). Furthermore we even have

$$u \in C^1([0,T]; L^2(\Omega)) \cap C([0,T]; H_0^1(\Omega)),$$

which ensures the validity of the initial conditions (2.3b).
We can furthermore deduce existence and uniqueness of the weak problem for inhomogeneous Dirichlet boundary conditions, if they are sufficiently regular. Suppose for example $g_0 = g(0,\cdot), g_L = g(L,\cdot)$ for some $g \in C^2([0,T]; H^2(\Omega))$. We can write $u = w + g$, where $w \in C^1([0,T]; L^2(\Omega)) \cap C([0,T]; H_0^1(\Omega))$ is the unique (weak) solution of

$$\begin{aligned} w_{tt} - (cw_x)_x &= f - g_{tt} + (cg_x)_x && \text{in } \Omega \times (0,T], \\ w(0,\cdot) = w(L,\cdot) &= 0 && \text{in } [0,T], \\ w(\cdot,0) = u_0 - g(\cdot,0), \quad w_t(\cdot,0) &= v_0 - g_t(\cdot,0) && \text{in } \Omega. \end{aligned}$$

With this we can conclude the problem to be well-posed and continue with the discretization procedure.

2.2 Variational Formulation and Fully-Discrete-Scheme

2.2.1 Discretization in Space

We start by discretizing in space. To simplify and clarify calculations we assume the solution to be $u \in C^2([0,T]; H^2(\Omega))$. We fix some time $t \in [0,T]$, the discretization in space is analogous to the elliptical case for each time t (see section 1.3 in chapter 1), so we will only briefly recall the steps.
First we discretize the domain by choosing a partition $\mathcal{T}_h$ of Ω using the notation introduced in the last chapter. Note at this point that the mesh we choose is time-independent, meaning the mesh does not change over time. We multiply (2.1) by a test function $v \in V_h$, where $V_h^r(\mathcal{T}_h)$ is the discontinuous finite element space defined (1.3), multiply over each element $I \in \mathcal{T}_h$ by parts and sum up over all elements. After adding the symmetry and penalty terms necessary for the SIPG variational form we find the semi-discrete SIPG variational formulation:

Find $u_h \in C^2([0,T]; V_h)$ with $u_h(\cdot,0) = \mathcal{I}_h u_0, u_t(\cdot,0) = \mathcal{I}_h v_0$ such that

$$(\partial_t^2 u_h(t), v)_{L^2(\Omega)} + b_h(u_h, v; t) = \ell_h(v; t) \qquad \forall v \in V_h, t \in [0,T], \tag{2.4}$$

where

$$\begin{aligned} b_h(u,v;t) = &\sum_{n=0}^{N} \int_{I_n} c(x,t) u_x(x,t) v_x(x) \,\mathrm{d}x - \sum_{n=0}^{N+1} \{\!\{ c(x_n,t) u_x(x_n,t) \}\!\} [\![v(x_n)]\!] + \{\!\{ c(x_n,t) v_x(x_n) \}\!\} [\![u(x_n,t)]\!] \\ &+ \sum_{n=0}^{N+1} \mathtt{a}_n(t) [\![u(x_n,t)]\!] [\![v(x_n)]\!], \\ \ell_h(v;t) = &(f(t), v)_{L^2(\Omega)} - g_L(t) c(x_{N+1}^-, t) v_x(x_{N+1}^-) + g_0(t) c(x_0^+, t) v_x(x_0^+) \\ &+ \mathtt{a}_{N+1}(t) g_L(t) v(x_{N+1}^-) + \mathtt{a}_0(t) g_0(t) v(x_0^+). \end{aligned}$$

Recall $\mathcal{I}_h : C(\overline{\Omega}) \to V_h$ denotes the interpolation operator (1.34) (one could also use L^2-projection). Clearly we can not expect the semi-discrete solution to satisfy continuous initial conditions, therefore the semi-discrete formulation must introduce projected initial condition.
Next we can rewrite (2.4) into matrix-vector form. To do so let $\Phi_1, \ldots, \Phi_M$ be a basis of V_h and therefore write

$$u_h(x,t) = \sum_{n=1}^{M} \alpha_n(t) \Phi_n(x).$$

We can plug this into the semi-discrete SIPG variational formulation (2.4), using linearity and the fact that testing against elements in a vector space is equivalent to testing against basis functions yields the matrix-vector system of ODEs

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{B}(t)\mathbf{u}(t) = \mathbf{l}(t) \qquad \forall t \in [0,T]. \tag{2.5}$$

Here $[\mathbf{M}]_{i,j} = (\Phi_j, \Phi_i)_{L^2(\Omega)}$ is the *mass matrix*, which in our case is not time-dependent.
$[\mathbf{B}(t)]_{i,j} = b_h(\Phi_j, \Phi_i; t)$ is the *stiffness matrix*, similar to the elliptic case but now with a time-dependent coefficient c and $[\mathbf{l}(t)]_i = \ell_h(\Phi_i; t)$ the *load vector*, also time-dependent. Finally $[\mathbf{u}(t)]_i = \alpha_i(t)$ denotes the solution vector with the time-dependent coefficients as entries uniquely determining the semi-discrete solution. The double dots over $\mathbf{u}$ denote the second order time-derivative, i.e.

$$\ddot{\mathbf{u}} = \frac{\mathrm{d}^2\mathbf{u}}{\mathrm{d}t^2}.$$

2.2.2 Discretization in Time

With this we come to the time-discretization. (2.5) is a second order system of ODEs, meaning we can use a chosen time-integration scheme to discretize the solution in time. One benefit of using discontinuous Galerkin to discretize in space is that the mass matrix $\mathbf{M}$ is block-diagonal, since by the construction of the basis functions, each element is fully decoupled from the rest, this simplifies solving any linear system with the mass matrix as the system matrix. We can capitalize on this property of DG-FEM by choosing an explicit time integration scheme. An explicit scheme, in contrast to an implicit scheme, ensures that we only need to invert the mass matrix and not the stiffness matrix as well.
First we choose a (stable) stepsize $\Delta t > 0$, we write $t_m = m \cdot \Delta t \in [0,T]$ and denote

$$\mathbf{u}_m := \mathbf{u}(t_m), \quad \mathbf{B}_m := \mathbf{B}(t_m), \quad \mathbf{l}_m := \mathbf{l}(t_m).$$

We introduce a second order finite difference quotient to approximate the second order time derivative

$$\ddot{\mathbf{u}}_m \approx \frac{\mathbf{u}_{m+1} - 2\mathbf{u}_m + \mathbf{u}_{m-1}}{\Delta t^2},$$

plugging this into (2.5) and reforming yields the fully discrete, explicit leapfrog scheme

$$\mathbf{M}\mathbf{u}_{m+1} = \Delta t^2 \mathbf{l}_m + (2\mathbf{M} - \Delta t^2 \mathbf{B}_m)\mathbf{u}_m - \mathbf{u}_{m-1}. \tag{2.6}$$

This being a multistep scheme we have to initialize $\mathbf{u}_0, \mathbf{u}_1$. To do so we use the interpolated values of the initial displacement to initialize $\mathbf{u}_0$, meaning we can write $\mathcal{I}_h u_0 = \sum_{n=1}^{M} \alpha_n(0)\Phi_n$ for some coefficients $\alpha_n(0) \in \mathbb{R}$ and get

$$[\mathbf{u}_0]_i = \alpha_i(0).$$

In fact since we have chosen a Lagrangian basis for the space V_h, finding the coefficients $\alpha_i(0)$ corresponds to evaluating the to be interpolated function u_0 at the corresponding node x_i on which the basis function Φ_i is stationed, i.e. $\alpha_i(0) = u_0(x_i)$.
For $\mathbf{u}_1$ we Taylor-expand

$$\mathbf{u}_1 \approx \mathbf{u}(\Delta t) = \mathbf{u}_0 + \Delta t \mathbf{v}_0 + \frac{\Delta t^2}{2}\ddot{\mathbf{u}}(0) + \mathcal{O}(\Delta t^3),$$

using (2.5) we can rewrite $\ddot{\mathbf{u}}(0) = \mathbf{M}^{-1}(\mathbf{l}_0 - \mathbf{B}_0\mathbf{u}_0)$ and define

$$\mathbf{u}_1 := \mathbf{u}_0 + \Delta t \mathbf{v}_0 + \frac{\Delta t^2}{2}\mathbf{M}^{-1}(\mathbf{l}_0 - \mathbf{B}_0\mathbf{u}_0).$$

To guarantee stability of the leapfrog scheme Δt has to be chosen small enough to satisfy the CFL condition, here this means specifically that the matrix $\mathbf{M} - \frac{\Delta t^2}{4}\mathbf{B}(t)$ has to be symmetric positive definite (SPD) for all times. This is equivalent to requiring that

$$\Delta t < 2\frac{\lambda_{\min}(\mathbf{M})}{\lambda_{\max}(\mathbf{B}(t))} \qquad \forall t \in [0,T],$$

where $\lambda_{\min}, \lambda_{\max}$ denote the minimal and maximal (positive) eigenvalues respectively. The expression is well-defined, since $\mathbf{B}(t)$ and $\mathbf{M}$ are SPD for all $t \in [0,T]$. This follows from a slight adaptation of the coercivity proof of the SIPG bilinear form on V_h (Theorem 1.16). For further information on the stability of leapfrog see [3].

2.3 Absorbing Boundary Conditions

When simulating a wave a classical example of a domain on which to do so is a *wave guide*, a rectangular domain through which the wave is propagated. Since we restrict ourselves to the 1d case in this thesis, trivially every domain interval we can choose corresponds to a sort of waveguide. We are interested in simulating a wave propagating through this waveguide. Lets say we want to send a wave through the waveguide from left to right. We can achieve this by imposing a corresponding Dirichlet or Neumann boundary condition at the point of entry. A question arises:

What boundary condition should be imposed at the upper boundary?

The answer may differ depending on the purpose of the simulation. The problem with using a simple Dirichlet or Neumann boundary condition at the upper boundary as well is that the incident wave is directly influenced and modified by the upper boundary condition. Lets say we impose a Dirichlet boundary condition at the lower boundary and a homogeneous Neumann boundary condition at the upper boundary

$$u(0,t) = \sin(x-t), \qquad u_x(L,t) = 0,$$

then the upper boundary will reflect the full wave back. This clearly pollutes the incident wave and if we wanted to observe how the the incident wave changes by passing through an inhomogeneous medium, the reflected wave would overshadow any slight modulation stemming from variations in the material. In reality in this context it would be more interesting to simulate the problem on an unbounded domain. We would like the wave to just propagate through the upper boundary and not return. We could achieve this by choosing a very large domain, large enough such that the incident wave never reaches the upper boundary for $t \in [0,T]$. Clearly this is a suboptimal, since we increase computaional load needlessly, the majority of the simulated wave would be uninteresting. In practice it is more convenient to choose an artifical boundary. To simulate an artificial boundary of this kind we need to impose artificial boundary conditions, these are often called *absorbing-, transparent-,* or *non-reflecting* boundary conditions.
Suppose

$$\operatorname{supp}(f), \operatorname{supp}(c) \subset \Omega \times [0,T],$$

i.e. the artificial boundary has been chosen sufficiently far away from the origin, such that the wave outside of the domain travels at a constant speed $\sqrt{c} = 1$ with no forcing.
Secondly assume that no waves are returning from outside of the domain, meaning all the waves outside the domain travel soley towards infinity. By d'Alembert we find that the solution outside the domain is of the form

$$u(x,t) = \psi(x-t) \qquad \text{for } x > L, t \in [0,T],$$

where $\psi \in C^1((L,\infty))$ only depends on the incident wave. This yields the (exact) first order absorbing boundary condition

$$u_t + u_x = 0 \tag{2.7}$$

at the upper boundary (exit of the waveguide), which corresponds to

$$u_t(L,\cdot) + u_x(L,\cdot) = 0 \qquad \text{in } [0,T].$$

To implement this boundary condition into our time-marching scheme we have to first treat the bilinear form b at the upper boundary as if we were to implement a Neumann boundary condition (see section 1.4), meaning after integrating by parts and summing over all elements we don't introduce the symmetry and penalty terms at the corresponding node x_{N+1}, instead we apply the absorbing boundary condition and get

$$-\{\!\{\underbrace{c(x_{N+1},t)}_{=1}\partial_x u_h(x_{N+1},t)\}\!\}[\![v(x_{N+1})]\!] = \{\!\{\partial_t u_h(x_{N+1},t)\}\!\}[\![v(x_{N+1})]\!].$$

the semi-discrete scheme (2.4) turns into

$$(\partial_t^2 u_h(t), v)_{L^2(\Omega)} + b_h(u_h, v; t) + \{\!\{\partial_t u_h(x_{N+1}, t)\}\!\}[\![v(x_{N+1})]\!] = \ell_h(v; t) \qquad \forall v \in V_h, t \in [0, T],$$

where

$$\begin{aligned} b_h(u, v; t) &= \sum_{n=0}^{N} \int_{I_n} c(x,t) u_x(x,t) v_x(x) \,\mathrm{d}x - \sum_{n=0}^{N} \{\!\{c(x_n,t)u_x(x_n,t)\}\!\}[\![v(x_n)]\!] + \{\!\{c(x_n,t)v_x(x_n)\}\!\}[\![u(x_n,t)]\!] \\ &\quad + \sum_{n=0}^{N} \mathtt{a}_n(t)[\![u(x_n,t)]\!][\![v(x_n)]\!], \\ \ell(v; t) &= (f(t), v)_{L^2(\Omega)} + g_0(t)c(x_0^+, t)v_x(x_0^+) + \mathtt{a}_0(t)g_0(t)v(x_0^+) \end{aligned}$$

are slightly modified.
And the semi-discrete matrix-vector system becomes

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{R}(t)\dot{\mathbf{u}}(t) = \mathbf{l}(t) \qquad \forall t \in [0, T],$$

where $[\mathbf{R}(t)]_{i,j} = \{\!\{\Phi_j(x_{N+1}, t)\}\!\}[\![\Phi_i(x_{N+1})]\!]$ is a matrix where the only non-zero entry is the one at the bottom right corner where $i = j = \dim(V_h)$ and Φ_i the basis centered at x_{N+1} is.

From here on we can use a (second order) centered finite difference to approximate

$$\dot{\mathbf{u}}(t) \approx \frac{\mathbf{u}(t + \Delta t) - \mathbf{u}(t - \Delta t)}{2\Delta t}$$

and similarly as before we find the slightly modified, fully-discrete leapfrog scheme

$$\Big(\mathbf{M} + \frac{\Delta t}{2}\mathbf{R}_m\Big)\mathbf{u}_{m+1} = \Delta t^2 \mathbf{l}_m + (2\mathbf{M} - \Delta t^2 \mathbf{B}_m)\mathbf{u}_m + \Big(\frac{\Delta t}{2}\mathbf{R}_m - \mathbf{I}\Big)\mathbf{u}_{m-1}.$$

We have to also make a slight adjustment to the initialization of the solution at the first time-step $\mathbf{u}_1$. By using a Taylor-expansion and the semi-discrete scheme we have just derived we can define

$$\mathbf{u}_1 := \mathbf{u}_0 + \Delta t \mathbf{v}_0 + \frac{\Delta t^2}{2}\mathbf{M}^{-1}(\mathbf{l}_0 - \mathbf{B}_0\mathbf{u}_0 - \mathbf{R}_0\mathbf{v}_0).$$

2.4 Updating the Stiffness Matrix

By definition the bilinear form b_h and by extension the stiffness matrix $\mathbf{B}(t)$ are time dependent. A priori this means that the matrix $\mathbf{B}_m$ has to be completely reassembled in each timestep $t_m = m\Delta t$. This is very costly, in fact the assembly of the stiffness matrix is by far the most expensive operation and the source of the heaviest computational load. Luckily there is room for optimization. We note that the time-dependency solely comes from the coefficient c in the bilinear form b_h, therefore if the coefficient satisfies certain conditions we can simplify the assembly of $\mathbf{B}_m$.

Time-Independent Coefficient

If c is in fact only a function of space and not of time, meaning $c(x, t) = c(x, 0)$ for all $t \in [0, T]$, then we don't have to update the stiffness matrix at all since

$$\mathbf{B}_m = \mathbf{B}_0 \qquad \forall m.$$

Space-Independent Coefficient

Similarly we can also consider the case of spatially independent coefficients. If $c(x,t) = c(0,t)$ for all $x \in \Omega, t \in [0,T]$ then we can write $c(0,t) = c(t)$ and

$$b_h(\Phi_j, \Phi_i; t) = c(t)\Big(\sum_{n=0}^{N} \int_{I_n} \Phi_i' \Phi_j' \,\mathrm{d}x - \sum_{n=0}^{N+1} \{\!\{\Phi_j'(x_n)\}\!\} [\![\Phi_i(x_n)]\!] + \{\!\{\Phi_i'(x_n)\}\!\} [\![\Phi_j(x_n)]\!] + \sum_{n=0}^{N+1} \frac{\sigma}{\mathtt{h}_n} [\![\Phi_j(x_n)]\!] [\![\Phi_j(x_n)]\!] \Big).$$

We can therefore write

$$\mathbf{B}_m = \frac{c(t)}{c(0)} \mathbf{B}_0,$$

which allows for a fast vectorized update.

Piecewise Constant Coefficient

Following a similar logic as in the space-independent case we can consider a coefficient c which is piecewise constant in space and arbitrary in time. Let $y_1, \dots, y_M \in \Omega$ and $y_0 = 0$ denote the lower boundary and $y_{M+1} = 1$ the upper boundary of Ω. Define the open intervals

$$R_m := (y_i, y_{m+1}) \qquad \text{for } m \in \{0, \dots, M\}$$

and the in space piecewise constant coefficient

$$c(x,t) := \begin{cases} \rho_0(t) & \text{for } x \in R_0 \\ \rho_1(t) & \text{for } x \in R_1 \\ \vdots & \\ \rho_M(t) & \text{for } x \in R_M \end{cases},$$

where $t \in [0,T]$ and

$$c_{\min} \le \rho_m(t) \le c_{\max} \qquad \forall m \in \{0, \dots, M\}, t \in [0,T].$$

The update method could work in this fashion for any type functions ρ_i, but we will only consider $\rho_m \in C^\infty$. Now to properly model the discontinuous coefficient using the DG-FEM we require that the mesh is chosen in such a way that the discontinuities never occur inside an element but rather only at the boundary. If $\mathcal{T}_h = \{I_n\}_{n=0}^N$ denotes the partition of Ω, where $I_n = (x_n, x_{n+1})$ are the elements defined by the face nodes $0 = x_0, \dots, x_{N+1} = L$, then we require

$$\{y_0, \dots, y_{M+1}\} \subset \{x_0, \dots, x_{N+1}\}.$$

In total

1. For each $m \in \{0, \dots, M+1\}$ there exists an index $n_m \in \{0, \dots, N+1\}$ such that $x_{n_m} = y_m$. We denote the index n_m to specify the dependency on m.
2. For each $m \in \{0, \dots, M\}$ there exists an set of indices $\mathcal{J}_m \subset \{0, \dots, N\}$ such that the element $I_n \subset R_m$ for all $n \in \mathcal{J}_m$.

Recall that in the assembly of the stiffness matrix (see 1.8) we had

$$\mathbf{B}(t) = \mathbf{A}(t) - \mathbf{B}_{\text{cons}}(t) + \mathbf{B}_{\text{penal}}(t).$$

We update all three matrices separately. We start by updating $\mathbf{A}$, which proves to be the simplest case. We were able to characterize the assembly of each of the sub-matrices by using local contribution matrices for each element (in the case of $\mathbf{A}$) or for each face node (in the case of $\mathbf{B}_{\text{cons}}, \mathbf{B}_{\text{penal}}$). Fix a subdomain index $m \in \{0, \dots, M\}$ and $n \in \mathcal{J}_m$. The local contribution matrix of $\mathbf{A}$ for the element I_n is given by

$$[\widehat{\mathbf{A}}^{(n)}(t)]_{i,j} = \frac{2}{h_n} \int_{-1}^{1} c(F_n(\xi), t)\, \widehat{\phi}_j'(\xi) \widehat{\phi}_i'(\xi) \,\mathrm{d}\xi = \frac{2\rho_m(t)}{h_n} \int_{-1}^{1} \widehat{\phi}_j'(\xi) \widehat{\phi}_i'(\xi) \,\mathrm{d}\xi.$$

In total the matrix $\mathbf{A}$ is composed of M bigger blocks corresponding to a respective subdomain R_m, each containing a set of smaller block matrices corresponding to the elements contained in R_m:

$$\mathbf{A}(t) = \begin{bmatrix} \frac{\rho_0(t)}{\rho_0(0)}\tilde{\mathbf{A}}_0(0) & & & \\ & \frac{\rho_1(t)}{\rho_1(0)}\tilde{\mathbf{A}}_1(0) & & \\ & & \ddots & \\ & & & \frac{\rho_m(t)}{\rho_m(0)}\tilde{\mathbf{A}}_m(0) \end{bmatrix},$$

or written as a sum

$$\mathbf{A}(t) = \sum_{m=0}^{M} \frac{\rho_m(t)}{\rho_m(0)} \sum_{s\in\mathcal{J}_m} \mathbf{A}^{(s)}(0), \tag{2.8}$$

where the $\mathbf{A}^{(s)}$ are defined in (1.14). An initial matrix can therefore be calculated once and in every time step we get $\mathbf{A}(t)$ by multiplying each block by the corresponding factor $\rho_m(t)$.
The update of the matrices $\mathbf{B}_{\text{cons}}, \mathbf{B}_{\text{penal}}$ is a little more complicated.

Consider an inner face node y_m between two subdomains R_{m-1}, R_m for $m \in \{1,\dots,M\}$, there exists $n \in \{1,\dots,N\}$ with $x_n = y_m$. Recall the local contribution matrices

$$\widehat{\mathbf{B}}^{(n)}_{\text{cons}}(t) = \begin{bmatrix} \mathbf{C}^{(n-1,n-1)}_{\text{cons}}(t) & \mathbf{C}^{(n-1,n)}_{\text{cons}}(t) \\ \mathbf{C}^{(n,n-1)}_{\text{cons}}(t) & \mathbf{C}^{(n,n)}_{\text{cons}}(t) \end{bmatrix}, \qquad \widehat{\mathbf{B}}^{(n)}_{\text{penal}}(t) = \begin{bmatrix} \mathbf{C}^{(n-1,n-1)}_{\text{penal}}(t) & \mathbf{C}^{(n-1,n)}_{\text{penal}}(t) \\ \mathbf{C}^{(n,n-1)}_{\text{penal}}(t) & \mathbf{C}^{(n,n)}_{\text{penal}}(t) \end{bmatrix},$$

where

$$\begin{aligned}
[\mathbf{C}^{(n-1,n-1)}_{\text{cons}}(t)]_{i,j} &= \frac{\rho_{m-1}(t)}{h_{n-1}}\widehat{\phi}_j'(1)\widehat{\phi}_i(1) + \frac{\rho_{m-1}(t)}{h_{n-1}}\widehat{\phi}_i'(1)\widehat{\phi}_j(1),\\
[\mathbf{C}^{(n-1,n)}_{\text{cons}}(t)]_{i,j} &= \frac{\rho_m(t)}{h_n}\widehat{\phi}_j'(-1)\widehat{\phi}_i(1) - \frac{\rho_{m-1}(t)}{h_{n-1}}\widehat{\phi}_i'(1)\widehat{\phi}_j(-1),\\
[\mathbf{C}^{(n,n)}_{\text{cons}}(t)]_{i,j} &= -\frac{\rho_m(t)}{h_n}\widehat{\phi}_j'(-1)\widehat{\phi}_i(-1) - \frac{\rho_m(t)}{h_n}\widehat{\phi}_i'(-1)\widehat{\phi}_j(-1),\\
[\mathbf{C}^{(n-1,n-1)}_{\text{penal}}(t)]_{i,j} &= \mathtt{a}_n(t)\widehat{\phi}_j(1)\widehat{\phi}_i(1),\\
[\mathbf{C}^{(n-1,n)}_{\text{penal}}(t)]_{i,j} &= -\mathtt{a}_n(t)\widehat{\phi}_j(-1)\widehat{\phi}_i(1),\\
[\mathbf{C}^{(n,n)}_{\text{penal}}(t)]_{i,j} &= \mathtt{a}_n(t)\widehat{\phi}_j(-1)\widehat{\phi}_i(-1),
\end{aligned}$$

for

$$\mathtt{a}_n = \frac{\sigma\max(\rho_{m-1}(t),\rho_m(t))}{\min(h_{n-1},h_n)}.$$

The problem is that the consistency terms and penalty terms both contain mixed factors of the coefficient c over face nodes. Therefore if the corresponding face node is a face between two subdomains R_{m-1}, R_m, meaning the face node is a point of discontinuity of c, then we can no longer simply factor out the $\rho_m(t)$. We first have to subtract the local contributions containing such mixed terms, then we can multiply by the time-dependent factor

$$\frac{\rho_m(t)}{\rho_m(0)}$$

per subdomain as done in the case of $\mathbf{A}$ and finally add the missing contribution corresponding to the subdomain face nodes back in. We can rewrite this procedure as the sums

$$\mathbf{B}_{\text{cons}}(t) = \sum_{m=1}^{M} \frac{\rho_m(t)}{\rho_m(0)}\Big(\sum_{s\in\mathcal{J}_m} \mathbf{B}^{(s)}_{\text{cons}}(0) - \mathbf{B}^{(n_m)}_{\text{cons}}(0)\Big) + \sum_{m=1}^{M} \mathbf{B}^{(n_m)}_{\text{cons}}(t) + \sum_{s\in\{0,N+1\}} \frac{\rho_m(t)}{\rho_m(0)}\mathbf{B}^{(s)}_{\text{cons}}(0), \tag{2.9}$$

$$\mathbf{B}_{\text{penal}}(t) = \sum_{m=1}^{M} \frac{\rho_m(t)}{\rho_m(0)}\Big(\sum_{s\in\mathcal{J}_m} \mathbf{B}^{(s)}_{\text{penal}}(0) - \mathbf{B}^{(n_m)}_{\text{penal}}(0)\Big) + \sum_{m=1}^{M} \mathbf{B}^{(n_m)}_{\text{penal}}(t) + \sum_{s\in\{0,N+1\}} \frac{\rho_m(t)}{\rho_m(0)}\mathbf{B}^{(s)}_{\text{penal}}(0). \tag{2.10}$$

Note that the terms corresponding to the boundary nodes x_0, x_{N+1} do not require the subtraction due to the absence of mixed terms.
We see therefore that after having assembled the initial matrices $\mathbf{A}(0), \mathbf{B}_{\text{cons}}(0), \mathbf{B}_{\text{penal}}(0)$ once, we only need to compute the parts $\mathbf{B}_{\text{cons}}^{(n_m)}(t), \mathbf{B}_{\text{penal}}^{(n_m)}(t)$ for $m \in \{1, \dots, M\}$ and the factors $\rho_m(t)$ for $m \in \{0, \dots, M+1\}$. Since M is fixed (in practice $M < 10$), we reduce the computational cost of assembling $\mathbf{B}(t)$ from $\mathcal{O}(N)$ to $\mathcal{O}(M) = \mathcal{O}(1)$.
We can sum up the update procedure as the following algorithm:

Algorithm 1 Calculate $\mathbf{B}(t)$

Assemble initial matrices
$\tilde{\mathbf{A}} := \mathbf{A}(0), \tilde{\mathbf{B}}_{\text{cons}} := \mathbf{B}_{\text{cons}}(0), \tilde{\mathbf{B}}_{\text{penal}} := \mathbf{B}_{\text{penal}}(0)$;

for $m = 1$ to M **do**
  Subtract mixed terms
  $\tilde{\mathbf{B}}_{\text{cons}} := \tilde{\mathbf{B}}_{\text{cons}} - \mathbf{B}_{\text{cons}}^{(n_m)}(0)$;
  $\tilde{\mathbf{B}}_{\text{penal}} := \tilde{\mathbf{B}}_{\text{penal}} - \mathbf{B}_{\text{penal}}^{(n_m)}(0)$;
end for

for $m = 0$ to M **do**
  Define index vector containing indices of all DoFs inside the subdomain R_m
  $\mathbf{j}_m$;
  Divide by initial coefficient
  $\tilde{\mathbf{A}}(\mathbf{j}_m, \mathbf{j}_m) := \tilde{\mathbf{A}}(\mathbf{j}_m, \mathbf{j}_m)/\rho_m(0)$;
  $\tilde{\mathbf{B}}_{\text{cons}}(\mathbf{j}_m, \mathbf{j}_m) := \tilde{\mathbf{B}}_{\text{cons}}(\mathbf{j}_m, \mathbf{j}_m)/\rho_m(0)$;
  $\tilde{\mathbf{B}}_{\text{penal}}(\mathbf{j}_m, \mathbf{j}_m) := \tilde{\mathbf{B}}_{\text{penal}}(\mathbf{j}_m, \mathbf{j}_m)/\rho_m(0)$;
end for
⋮
for $m = 0$ to M **do**
  Define index vector containing indices of all DoFs inside the subdomain R_m
  $\mathbf{j}_m$;
  Multiply by current coefficient
  $\tilde{\mathbf{A}}(\mathbf{j}_m, \mathbf{j}_m) := \tilde{\mathbf{A}}(\mathbf{j}_m, \mathbf{j}_m) \cdot \rho_m(t)$;
  $\tilde{\mathbf{B}}_{\text{cons}}(\mathbf{j}_m, \mathbf{j}_m) := \tilde{\mathbf{B}}_{\text{cons}}(\mathbf{j}_m, \mathbf{j}_m) \cdot \rho_m(t)$;
  $\tilde{\mathbf{B}}_{\text{penal}}(\mathbf{j}_m, \mathbf{j}_m) := \tilde{\mathbf{B}}_{\text{penal}}(\mathbf{j}_m, \mathbf{j}_m) \cdot \rho_m(t)$;
end for

for $m = 1$ to M **do**
  Add current mixed terms
  $\tilde{\mathbf{B}}_{\text{cons}} := \tilde{\mathbf{B}}_{\text{cons}} + \mathbf{B}_{\text{cons}}^{(n_m)}(t)$;
  $\tilde{\mathbf{B}}_{\text{penal}} := \tilde{\mathbf{B}}_{\text{penal}} + \mathbf{B}_{\text{penal}}^{(n_m)}(t)$;
end for
$\mathbf{B}(t) := \tilde{\mathbf{A}} - \tilde{\mathbf{B}}_{\text{cons}} + \tilde{\mathbf{B}}_{\text{penal}}$

The first part of the algorithm should be done before the leapfrog iteration and then the second part can be done repeatedly during the leapfrog iteration using the prepared matrices.

2.5 Numerical Results

2.5.1 Rate of Convergence

To reproduce the expected convergence rates and therein check the validity of our implementation we follow a similar sequence of steps as already presented in section 1.14.1.

Step 1 (*Domain*)**.**
We fix the domain $\Omega = (0, 10)$, an extended version for better visualization, and set the final time to $T = 10$.

Step 2 (*Exact Solution*)**.**
We decide to approximate the smooth exact solutions

$$u_1(x,t) = e^{-(x-t+2)^2}, \qquad u_2(x,t) = \sin(x - t - \pi).$$

Step 3 (*Coefficient*)**.**
We fix the smooth coefficients

$$c_1(x,t) = 1, \quad c_2(x,t) = (\sin(x) + 2)(\cos(t) + 2), \quad c_3(x,t) = \sin(x) + 2, \quad c_4(x,t) = \sin(t) + 2.$$

To ensure our exact solution still satisfies the original pde, we calculate the forcing term f in (1.1a) symbolically, i.e. $f_{i,j} = \partial_t^2 u_i - \partial_x(c_j \partial_x u_i)$.

Step 4 (*Boundary and Initial Conditions*)**.**
As done in the last chapter we focus on reproducing the convergence rates for Dirichlet boundary conditions on both sides, we directly take the values provided by the exact solution u we start with as boundary conditions, meaning

$$g(x,t) = u(x,t) \qquad \forall (x,t) \in \{0, 10\} \times [0, T].$$

We do the same thing for the initial conditions and in total for $i \in \{1, 2\}$ and $j \in \{1, 2, 3, 4\}$ we solve the pde:
Find $u \in C^2([0,T]; H^2(\Omega))$ such that

$$\begin{aligned} u_{tt} - (c_j u_x)_x &= \partial_t^2 u_i - \partial_x(c_j \partial_x u_i) && \text{in } \Omega \times [0,T], \\ u(0,\cdot) = u_i(0,\cdot), \quad u(10,\cdot) &= u_i(10,\cdot) && \text{in } [0,T], \\ u(\cdot,0) = u_i(\cdot,0), \quad u_t(\cdot,0) &= \partial_t u_i(\cdot,0) && \text{in } \Omega. \end{aligned}$$

In the case of the coefficient c_1 we observe that the solution $u \in \{u_1, u_2\}$ satisfies the absorbing boundary condition exactly. When imposing the absorbing boundary condition at the upper boundary and a Dirichlet boundary condition ($g(0,t) = u(0,t)$), or a Neumann boundary condition ($g(0,t) = -u_x(0,t)$) at the lower boundary we observe the exact same expected convergence rates as in the double Dirichlet case.

Step 5 (*Mesh*)**.**
We choose an initial equidistant mesh of meshsize $h = 1$

$$\mathcal{T}_h^{(0)} = \{(0,1), (1,2), (2,3), (3,4), (4,5), (5,6), (6,7), (7,8), (8,9), (9,10)\},$$

but the results do not vary for any kind of initial mesh. We fix a number of refinement cycles and obtain a sequence of nested meshes by refining every element each cycle. We fix a stepsize scaling factor $\gamma > 0$, such that $\Delta t = \gamma h$ on each of the nested meshes. We have found that $\gamma = \frac{1}{50r}$ ensures stability for all chosen coefficients in the case of the equidistant mesh.

Step 6 (*Numerical Approximation*)**.**
Again we only consider $\mathcal{P}^1$- and $\mathcal{P}^2$-elements, so $r \in \{1, 2\}$, and fix a sufficiently large penalization parameter $\sigma = 10(r+1)^2$ to ensure coercivity. We calculate the numerical solution at each timestep using the time-marching scheme (2.6). Due to the time-dependent bilinear and linear forms we now have to reassemble the stiffness matrix $\mathbf{B}(t)$ and the load vector $\mathbf{l}(t)$ at each timestep anew. This is very costly, especially for growing global degrees of freedom (decreasing meshsize). Luckily for c_1, c_3, c_4 we do not in fact have to reassemble everything only for c_2 we are forced to brute force the iteration. In the case of $c \in \{c_1, c_3\}$

the coefficient is not time-dependent at all and therefore neither are the entries of the stiffness matrix, so it holds that

$$\mathbf{B}(t) = \mathbf{B}(0) \qquad \forall t \in [0, T].$$

The majority of the load vector depending on the source term f can also be tested for time-independency. If for example $c = c_1$, the whole system, aside from the boundary conditions, can be reused in every timestep. Since the majority of computation time is required to assemble the system, reusing as much as possible provides a substantial benefit.

Step 7 (*Error Computation*)**.**
We consider the error between the exact and numerical solution at the endtime T, specifically we consider

1. The L^2-error $\|u(T) - u_h(T)\|_{L^2(\Omega)}$;
2. The broken H^1-error $\|u(T) - u_h(T)\|_{H^1(\mathcal{T}_h)}$;
3. The error in the energy norm $\|u(T) - u_h(T)\|_\epsilon$.

We compute all these errors for both exact solutions u_1, u_2 on each mesh in the sequence of nested (equidistant) meshes.

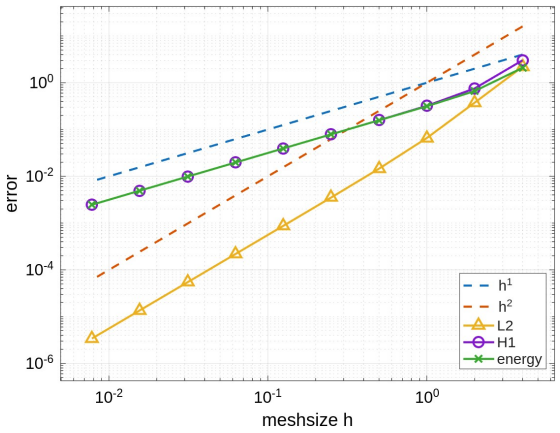

Figure 2.1: Errors of SIPG in space and leapfrog in time for P^1-elements

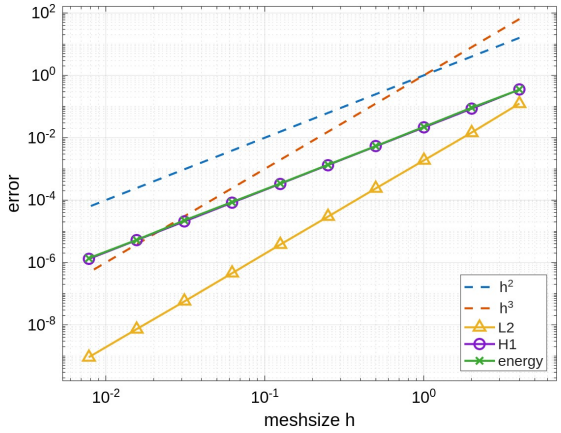

Figure 2.2: Errors of SIPG in space and leapfrog in time for P^2-elements

The results vary only slightly for different permutations of parameters. In Figures 2.1, 2.2 we show the errors computed for the exact solution u_2 and the constant coefficient c_1 for both $\mathcal{P}^1$- and $\mathcal{P}^2$-elements. We observe a considerable gain in accuracy by employing $\mathcal{P}^2$-elements over $\mathcal{P}^1$-elements. It seems that we gain an order of convergence when upgrading to $\mathcal{P}^2$-elements. This is indeed the case in the broken H^1-norm and energy norm, but since leapfrog limits the error to behave like $\mathcal{O}(\Delta t^2)$ and we coupled $\Delta t = \gamma h$ we expect that for $h \to 0$ even the L^2-error curve will flatten and behave like $\mathcal{O}(h^2)$.

2.5.2 Visualization of the SIPG-leapfrog Solution

In Figures 2.3-2.6 we visualize the numerical solutions created in 2.5.1 at the final time $T = 10$ on the once refined mesh $\mathcal{T}_h^{(1)}$. Clearly the choice of the coefficient c does not influence the picture since we artificially modify the pde to output the desired exact solution.

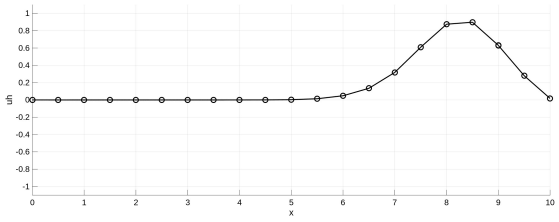

Figure 2.3: u_1 with P^1-elements

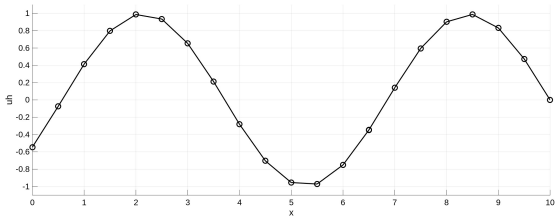

Figure 2.4: u_2 with P^1-elements

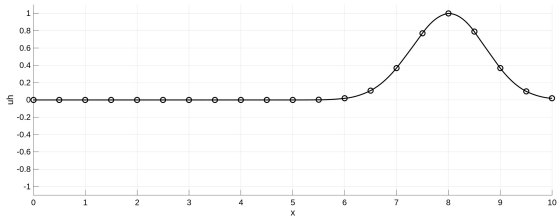

Figure 2.5: u_1 with P^2-elements

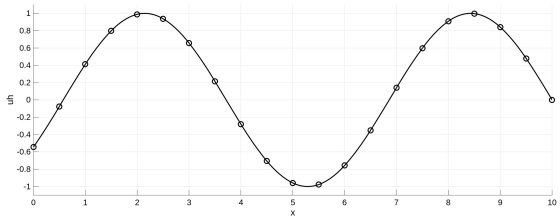

Figure 2.6: u_2 with P^2-elements

Appendix A

Prerequisites

Bibliography

[1] L. C. Evans, *Partial Differential Equations*, vol. 19 of Graduate Studies in Mathematics, American Mathematical Society, 2 ed., 2010.

[2] E. H. Georgoulis, *Discontinuous galerkin methods for linear problems: An introduction*, in Approximation Algorithms for Complex Systems, E. Georgoulis, A. Iske, and J. Levesley, eds., vol. 3 of Springer Proc. Math., Springer, Berlin, Heidelberg, 2011, pp. 91–126.

[3] M. Grote, *Lecture notes numerical methods for wave propagation*, (2024).

[4] M. Grote, A. Schneebeli, and D. Schötznau, *Discontinuous method for the wave equation*, SIAM Journal on Numerical Analysis, 44 (2006), pp. 2408–2431.

[5] D. A. D. Pietro and A. Ern, *Mathematical Aspects of Discontinuous Galerkin Methods*, Springer, Berlin, Heidelberg, 1 ed., 2012.

[6] A. Quarteronia, R. Sacco, and F. Saleri, *Numerical Mathematics*, vol. 2, Springer, Berlin, Heidelberg, 2007.

[7] B. Rivière, *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation*, vol. 35 of Frontiers in Applied Mathematics, SIAM, Philadelphia, 2008.

[8] T. Warburton and J. S. Hesthaven, *On the constants in hp-finite element trace inverse inequalities*, Computer Methods in Applied Mechanics and Engineering, 192 (2003), pp. 2765–2773.