

Comparison of finite element approximation transfer methods between meshes

Subagya Perera

Project submitted in partial fulfillment of the requirements for the degree of
Master of Science Mathematics and Statistics¹

Department of Mathematics and Statistics
Faculty of Science
University of Ottawa

© Subagya Perera, Ottawa, Canada, 2024

¹The M.Sc. program is a joint program with Carleton University, administered by the Ottawa-Carleton Institute of Mathematics and Statistics

Contents

List of Figures	iv
List of Tables	v
1 Basic notations and useful definitions	1
1.1 The finite element method in 1-D case	1
2 Transfer techniques	7
2.1 Transfer techniques in 1-D	7
2.1.1 Lagrange interpolation	8
2.1.2 $L^2(\Omega)$ -projection	8
2.1.3 $L^2(\Omega)$ -projection with mass lumping	10
2.1.4 Alauzet and Mehrenberger formula	11
2.2 Comparison of transfer methods	15
2.2.1 Test case with a regular function	16
2.2.2 Comparing the variants of the AM transfer operator	19
2.2.3 Test case with a discontinuous function	19
3 Applications to PDE	22
3.1 Transport equation	22
3.1.1 SUPG stabilization	23
3.1.2 Numerical example	24
3.2 Heat equation	28
3.2.1 Numerical example	29
Index	32

List of Figures

1.1 Basis function and linear Lagrange interpolant in one dimension	2
1.2 Left and right nodal values of piecewise discontinuous linear functions in element K_i^x and K_{i+1}^x and piecewise discontinuous space nodal basis function on element K_i^x	3
1.3 Basis function and linear approximation in one dimension	6
2.1 Left: Illustration of u_x and its L^2 projection on the space P_y^1 . Right: L^2 projection of a continuous function on the space P_y^1	9
2.2 Intersection region γ_i and integral $\int_{\gamma_i} v_x$ for a mass conservative transfer between meshes	11
2.3 Comparing transfer operators to project the initial function u_1 from the background mesh to the current mesh, using a single transfer step. Top: transferring from a finer mesh τ_x to a coarser mesh τ_y . Bottom: transferring from a coarser mesh τ_y to a finer mesh τ_x	16
2.4 Comparing transfer operators to project the initial function u_1 from the background mesh to the current mesh, when $m = 500$	18
2.5 Comparing the AM transfer operator after replacing step (iii) with equations (2.1.20) and (2.1.21) for a single transfer step. Top: transferring from a finer mesh τ_x to a coarser mesh τ_y . Bottom: transferring from a coarser mesh τ_y to a finer mesh τ_x	20
2.6 Comparing transfer operators to project the function u_2 , using a single transfer step. Figures 2.6a, 2.6c transferring from a finer mesh τ_x to a coarser mesh τ_y . Figures 2.6e, 2.6f transferring from a coarser mesh τ_y to a finer mesh τ_x	21
3.1 Numerical solutions of the transport equation for the manufactured solution u_3 using the fixed mesh τ_x , different CFL numbers and the final time $T = 3$	24

3.2 Numerical solutions of the transport equation for the manufactured solution u_3 using different meshes τ_x and τ_y with the final time $T = 3$, CFL number 0.1 and various transfer operators. Top row: overlapping meshes τ_x and τ_y . Bottom row: nonoverlapping meshes τ_x and τ_y .	25
3.3 Solution comparison of the transport equation using manufactured solution u_4 for different CFL numbers and final time $T = 3$	26
3.4 Comparison of the Galerkin and SUPG methods using the manufactured solution u_4 with CFL number 0.1.	27
3.5 Numerical solutions of the transport equation for the manufactured solution u_4 using different meshes τ_x and τ_y with the CFL number 0.1 and various transfer operators. Top row: overlapping meshes τ_x and τ_y . Bottom row: nonoverlapping meshes τ_x and τ_y .	28
3.6 Numerical solutions of the transport equation for the manufactured solution u_5 using different meshes τ_x and τ_y with the final time $T = 10$, and various transfer operators.	30

List of Tables

2.1 L^2 -error between the function u_1 and the transferred approximations for a single transfer step.	17
2.2 CPU time comparison between the transfer methods for 1000 pro- jections.	19

Chapter 1

Basic notations and useful definitions

1.1 The finite element method in 1-D case

This chapter introduces the finite element method, useful definitions, notations, and concepts needed to understand the following chapters. We first consider the one-dimensional linear case and then move on to higher-dimensional approximation methods for convenience's sake.

Definition 1.1 (Mesh, elements and nodes). Let us assume that $\Omega = (a, b)$ is an interval of \mathbb{R} . A mesh τ_x of the domain $\overline{\Omega}$ is defined as an indexed union/collection of a finite number of non-empty disjoint intervals $K_i^x = [x_{i-1}, x_i]$, for $i = 1, \dots, N_x$, that is;

$$\overline{\Omega} = \bigcup_i K_i^x, \quad \text{with } \text{int}(K_i^x) \cap \text{int}(K_j^x) = \emptyset, \quad \forall i \neq j, \quad (1.1.1)$$

where each K_i^x is called an element of τ_x , $\text{int}(K_i^x)$ is the interior of K_i^x and the list of $N_x + 1$ points $\{x_0, \dots, x_{N_x}\}$ such that $a = x_0 < x_1 < \dots < x_{N_x} = b$ are called the nodes of τ_x . The width of an element is denoted by $|K_i^x| = x_i - x_{i-1}$, for $i = 1, \dots, N_x$. It is common in the literature [3] to set the parameter $h_x = \max_i |K_i^x|$.

Definition 1.2 (Linear polynomial space). Take $K_i^x = [x_{i-1}, x_i]$ to be an interval in $\overline{\Omega}$. Then \mathbb{P}_1 denotes the vector space of linear polynomials on K_i^x , that is

$$\mathbb{P}_1(K_i^x) = \{v : v(x) = \sum_{l=0}^1 c_l x^l, x \in K_i^x, c_l \in \mathbb{R}\}. \quad (1.1.2)$$

Definition 1.3 (Kronecker delta). The Kronecker delta symbol δ_{ij} is defined as

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j; \\ 0, & \text{otherwise.} \end{cases} \quad (1.1.3)$$

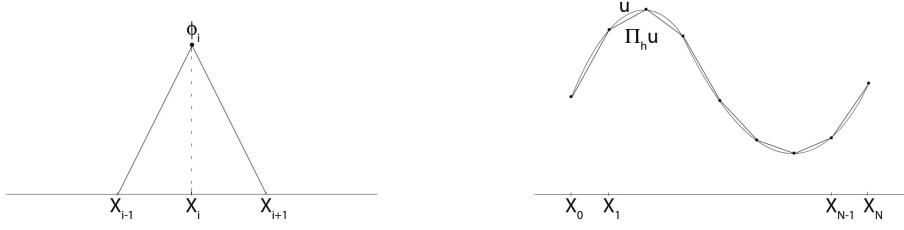
(a) 1-D nodal basis function at i^{th} node (b) Piecewise linear interpolant

Figure 1.1: Basis function and linear Lagrange interpolant in one dimension

Let us now introduce the L^2 function space.

Definition 1.4 ($L^2(\Omega)$ function space). We define the functional space $L^2(\Omega)$ as

$$L^2(\Omega) = \{v : \Omega \rightarrow \mathbb{R}; v \text{ is Lebesgue measurable, } \|v\|_{0,\Omega} < \infty\}, \quad (1.1.4)$$

where $\|v\|_{0,\Omega}^2 = \int_{\Omega} |v(x)|^2 dx$. Furthermore two functions $v = u$ in $L^2(\Omega)$ iff $\|v - u\|_{L^2(\Omega)} = 0$. This implies that $u(x) = v(x)$ for a.e. $x \in \Omega$.

Definition 1.5 (Finite element space and basis functions). Let us consider the vector space of continuous piecewise linear functions on the mesh τ_x as

$$P_x^1 = \{v_x : v_x \in \mathcal{C}^0(\overline{\Omega}); \forall K_i^x \in \tau_x, i \in \{1, \dots, N_x\}, v_x|_{K_i^x} \in \mathbb{P}_1\}. \quad (1.1.5)$$

P_x^1 is the linear finite element space on τ_x . Further introduce the set of functions $\{\phi_0^x, \dots, \phi_{N_x}^x\}$ in P_x^1 , where, for each $i = 0, \dots, N_x$,

$$\phi_i^x(x) = \begin{cases} \frac{1}{|K_i^x|}(x - x_{i-1}), & \text{if } x \in K_i^x; \\ \frac{1}{|K_{i+1}^x|}(x_{i+1} - x), & \text{if } x \in K_{i+1}^x; \\ 0, & \text{otherwise.} \end{cases} \quad (1.1.6)$$

The functions $\phi_0^x, \dots, \phi_{N_x}^x$ are called the nodal basis functions (see figure [1.1a]). We need a modification of the basis functions near the boundary

$$\phi_0^x(x) = \begin{cases} \frac{1}{|K_1^x|}(x_1 - x), & \text{if } x \in K_1^x; \\ 0, & \text{otherwise.} \end{cases}, \quad \phi_{N_x}^x(x) = \begin{cases} \frac{1}{|K_{N_x}^x|}(x - x_{N_x-1}), & \text{if } x \in K_{N_x}^x; \\ 0, & \text{otherwise.} \end{cases}$$

The nodal basis functions satisfy

$$\phi_i^x(x_j) = \delta_{ij}, \quad \forall i, j = 0, \dots, N_x. \quad (1.1.7)$$

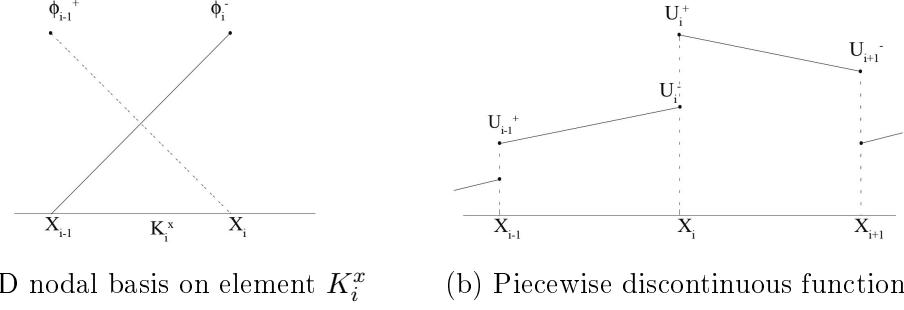
(a) 1-D nodal basis on element K_i^x (b) Piecewise discontinuous function

Figure 1.2: Left and right nodal values of piecewise discontinuous linear functions in element K_i^x and K_{i+1}^x and piecewise discontinuous space nodal basis function on element K_i^x .

In the one-dimensional case, K_i^x is an interval. In the two-dimensional case the elements K_i^x will be taken as triangles. Since $v_x \in P_x^1$,

$$v_x(x) = \sum_{i=0}^{N_x} U_i^x \phi_i^x(x) \quad \forall x \in \Omega \quad (1.1.8)$$

where $\{U_i^x\}_{i=0}^{N_x}$ with $U_i^x = v_x(x_i)$ are the nodal values of v_x on τ_x (see figure 1.1).

Definition 1.6 (Discontinuous finite element space and basis functions). Let us introduce the vector space of discontinuous piecewise linear functions on the mesh τ_x

$$P_x^{1,dc} = \{v_x : v_x \in L^2(\Omega); \forall K_i^x \in \tau_x, i \in \{1, \dots, N_x\}, v_x|_{K_i^x} \in \mathbb{P}_1\}. \quad (1.1.9)$$

Furthermore, we introduce the nodal basis functions ϕ_{i-1}^+ and ϕ_i^- in each element K_i^x for $i = 1, \dots, N_x$ (see figure 1.2a):

$$\phi_{i-1}^+(x) = \begin{cases} \frac{1}{|K_i^x|}(x_i - x), & \text{if } x \in K_i^x; \\ 0, & \text{otherwise.} \end{cases}, \quad \phi_i^-(x) = \begin{cases} \frac{1}{|K_i^x|}(x - x_{i-1}), & \text{if } x \in K_i^x; \\ 0, & \text{otherwise.} \end{cases} \quad (1.1.10)$$

For $v_x \in P_x^{1,dc}$, we have

$$v_x(x) = \sum_{i=1}^{N_x} U_{i-1}^+ \phi_{i-1}^+(x) + U_i^- \phi_i^-(x) \quad \forall x \in \Omega \quad (1.1.11)$$

where U_{i-1}^+ and U_i^- are the nodal values on the element K_i^x (see figure 1.2b).

Definition 1.7 (Transfer operator). Suppose that there are two meshes τ_x and τ_y of the domain Ω , where τ_x represents the background mesh and τ_y represents the current mesh. Assume that P_x^1 and P_y^1 are the linear finite element spaces on τ_x and τ_y , respectively. We can now define $\Pi_h : P_x^1 \rightarrow P_y^1$ as the transfer operator from mesh τ_x to mesh τ_y so that we choose $\Pi_h v_x = \sum_{j=0}^{N_y} U_j^y \phi_j^y \in P_y^1$ "close" to $v_x = \sum_{i=0}^{N_x} U_i^x \phi_i^x \in P_x^1$.

More generally, we define the transfer operator $\Pi_h : B \rightarrow P_x^1$, where B is a Banach space and P_x^1 is the linear finite element space on τ_x . Take, for example, the piecewise linear Lagrange interpolation operator $\Pi_h^{L,1}$. Consider the finite element space P_x^1 and its corresponding nodal basis $\{\phi_i^x\}_{i=0}^{N_x}$. Then the interpolation operator $\Pi_h^{L,1} : C^0(\bar{\Omega}) \rightarrow P_x^1$ is defined as follows:

$$(\Pi_h^{L,1} v)(x) = \sum_{i=0}^{N_x} v(x_i) \phi_i^x(x) \quad (1.1.12)$$

Observe that the linear Lagrange interpolate is such that $\Pi_h^{L,1} v(x_i) = v(x_i)$, $\forall x_i \in \{x_i\}_{i=0}^{N_x}$. The linear Lagrange interpolant is shown on figure 1.1b.

In this project we uses the term supermesh for the intersection of two meshes, as needed when reinterpolating or transferring finite element functions between meshes. In [13, 5], Rueter and Farrell discussed the existence of supermesh and how to construct them for two given meshes over the same polyhedral domain.

Definition 1.8 (Supermesh). Let us take τ_x and τ_y , two meshes of the domain $\Omega \subset \mathbb{R}$ with node sets, $\mathcal{N}_x = \{x_i\}_{i=0}^{N_x}$ and $\mathcal{N}_y = \{y_j\}_{j=0}^{N_y}$, respectively. We define the supermesh τ_z of τ_x and τ_y , as the mesh satisfying

- $\mathcal{N}_z = \mathcal{N}_x \cup \mathcal{N}_y$
- $\text{int}(K^z) \cap \text{int}(K^n) = \emptyset$ or $\text{int}(K^z) \cap \text{int}(K^n) = \text{int}(K^z)$, $\forall K^z \in \tau_z, n \in \{x, y\}$

In words, the first condition express that any node x_i (respectively y_j) in mesh τ_x (respectively τ_y) is in the supermesh τ_z . The second condition says that the intersection of any element K^x (respectively K^y) of τ_x (respectively τ_y) with the supermesh element K^z is either empty or the whole element K^z .

Definition 1.9 (conservation). Consider τ_x and τ_y , two meshes of Ω . Taking P_x^1 and P_y^1 as the finite element spaces on τ_x and τ_y , respectively, and $\Pi_h : P_x^1 \rightarrow P_y^1$ as the transfer operator from τ_x to τ_y . We say that a transfer or interpolation strategy is conservative if it maintains desired properties (such as mass) during the interpolation steps, i.e.

$$\int_{\Omega} v_x dx = \int_{\Omega} \Pi_h v_x dy. \quad (1.1.13)$$

The discrete version of conservation over τ_x and τ_y read as

$$\sum_{K^x \in \tau_x} \int_{K^x} v_x dx = \sum_{K^y \in \tau_y} \int_{K^y} \Pi_h v_x dy \quad (1.1.14)$$

Here Π_h is a generic transfer operator but we will introduce several such operators in chapter 2.

Definition 1.10 (Mesh refinement). Let us take node sets $\mathcal{N}_x, \mathcal{N}_y$ of meshes τ_x, τ_y , respectively, and assume that $\mathcal{N}_x \subset \mathcal{N}_y$. Then we say that τ_y is a refinement of τ_x . Furthermore, $\mathcal{N}_x \subset \mathcal{N}_y \implies P_x^1 \subset P_y^1$.

Besides, τ_x and τ_y are said to be overlapping meshes if τ_y is a refinement of τ_x or τ_x a refinement of τ_y , that is $\mathcal{N}_x \subset \mathcal{N}_y$ or $\mathcal{N}_y \subset \mathcal{N}_x$. If $\mathcal{N}_x \not\subset \mathcal{N}_y$ and $\mathcal{N}_y \not\subset \mathcal{N}_x$, then the meshes τ_x and τ_y are said to be non-overlapping.

Definition 1.11 (P^1 -exactness). [1] Consider a transfer operator $\Pi_h : P_x^1 \rightarrow P_y^1$. Assume that $P_x^1 \subset P_y^1$ and

$$\Pi_h(v_x) = v_x \quad \forall v_x \in P_x^1. \quad (1.1.15)$$

Then we say that the transfer operator Π_h is P^1 -exact. Moreover, if both conservation and P^1 -exactness are satisfied for Π_h , we may say that Π_h is P^1 -conservative.

Definition 1.12 (Local maximum principle for a transfer method). [1] Take any $z \in \bar{\Omega}$ and consider the element $K^z \in \tau_y$ such that $z \in K^z$. Set $S_z = \{\cup K_i^x : K_i^x \cap K^z \neq \emptyset, K_i^x \in \tau_x\}$. Then we say that the transfer operator $\Pi_h : P_x^1 \rightarrow P_y^1$ satisfy a local maximum principle (see figure 1.3b) on $\bar{\Omega}$ iff

$$\min_{x \in S_z} v_x(x) \leq (\Pi_h v_x)(z) \leq \max_{x \in S_z} v_x(x), \forall z \in \bar{\Omega}. \quad (1.1.16)$$

Now we can define barycentric coordinate system to make our computations easy. Barycentric coordinates were first introduced by August Ferdinand Möbius (1790-1868) in his book "Der barycentrische Calcul" [15] published in 1827 and are defined as follows,

Definition 1.13 (Barycentric coordinates). Consider an element $K_i^x = [x_{i-1}, x_i]$ of τ_x . We set the local nodes $\tilde{x}_0 = x_{i-1}$ and $\tilde{x}_1 = x_i$ on the element K_i^x . Then we define the barycentric coordinates $\lambda_i : \mathbb{R} \rightarrow \mathbb{R}$, for $i = 0, 1$, such that

$$\lambda_i(x) = 1 - \frac{(x - \tilde{x}_i)}{|K_i^x|}, \quad \text{with} \quad \sum_{i=0}^1 \lambda_i(x) = 1.$$

Figure 1.3a shows an example of barycentric coordinates of a point x in an element $K = [x_0, x_1]$.

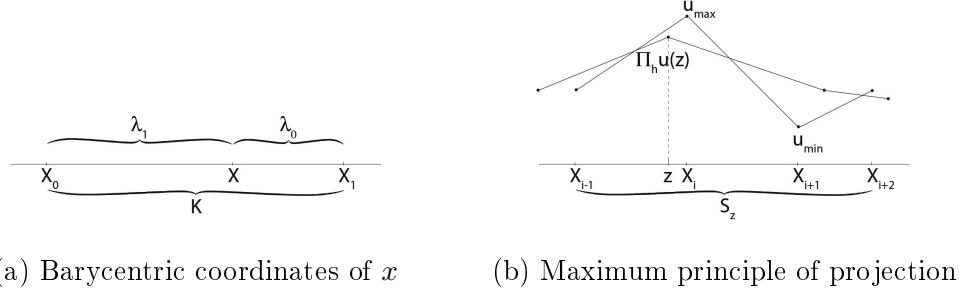


Figure 1.3: Basis function and linear approximation in one dimension

Definition 1.14 (Mass matrix). Consider the mesh τ_x over the domain Ω and the associated linear nodal basis $\{\phi_i^x\}_{i=0}^{N_x}$. Then the mass matrix M_x is defined as

$$M_x = (m_{ij})_{i,j=0,\dots,N_x}, \quad \text{where} \quad m_{ij} = \int_{\Omega} \phi_i^x \phi_j^x dx. \quad (1.1.17)$$

Definition 1.15 (Lumped mass matrix). The lumped mass matrix, noted M_x^L , is obtained by diagonalizing the mass matrix M_x resulting from a Galerkin finite element method by either row or column sum. In this project we only use row sum, i.e.

$$M_x^L = (m_{ij}^l)_{i,j=0,\dots,N_x}, \quad \text{where} \quad m_{ij}^l = \delta_{ij} \sum_{k=1}^{N_x} m_{ik}. \quad (1.1.18)$$

Lumping is useful because the inverse of the lumped matrix is cheap to compute.

Chapter 2

Transfer techniques

In numerical methods, solution transfer between meshes or computational domains is an important strategy. It is used in solving PDEs over moving domains, such as fluid-structure interaction (FSI) solved by Arbitrary Lagrangian-Eulerian (ALE) methods [2] [11]. It is also used to transfer solutions between adapted meshes [1] [5], and coupling of various physical problems or computational codes such as coupling an electrocardiological model of the heart with a structural model of its motion, or coupling a biological activity model with an ocean model [4] [17]. The main goal of transfer techniques is to keep the transmission step as accurate as possible, preserve specific physical quantities such as mass or volume [14], ensure a local maximum principle, and so on. In many cases, having precise solution is critical, as in numerical weather prediction and climate simulation [10] [14] [16]. Failure to maintain some quantities can result in physically impossible solutions such as negative density and instability [6].

2.1 Transfer techniques in 1-D

Our goal is to study and develop several solution transfer strategies. Nonetheless, our goal here is to introduce and demonstrate the accuracy of these transfer operators when used to re-interpolate between two given meshes. We primarily compare transfer techniques with two types of meshes:

- Between overlapping meshes, for example,
 - From one mesh to the same mesh (ideally this gives exact interpolation).
 - From a coarse mesh to a finer mesh or vice versa.
- Between non-overlapping meshes.

With the first type of meshes, one mesh is a refinement of the other, whereas in the second case it is not. We only consider situations where both the background and

current meshes are over the same domain Ω . We will first consider the following transfer operators:

- Lagrange interpolation;
- $L^2(\Omega)$ projection;
- $L^2(\Omega)$ projection with lumped mass matrix;
- Alauzet and Mehrenberger projection method [1].

Below we consider two meshes τ_x, τ_y over $\overline{\Omega}$ and their associated linear finite element spaces P_x^1, P_y^1 , respectively. Assume that $\{U_i^x\}_{i=0}^{N_x}, \{U_j^y\}_{j=0}^{N_y}$ are the set of nodal values on τ_x, τ_y , respectively. We recall that the transfer operator $\Pi_h : P_x^1 \rightarrow P_y^1$ is such that $\Pi_h u_x = \sum_{j=0}^{N_y} U_j^y \phi_j^y \in P_y^1$ is "close" to $u_x = \sum_{i=0}^{N_x} U_i^x \phi_i^x \in P_x^1$, a given function in P_x^1 .

2.1.1 Lagrange interpolation

Lagrange interpolation is one of the most computationally efficient methods for transferring function data between meshes. We simply use the local nodal values on the background mesh to calculate the nodal values on the current mesh. Let $\{\phi_i^x\}_{i=0}^{N_x}, \{\phi_j^y\}_{j=0}^{N_y}$ be the nodal basis functions of the finite element space P_x^1, P_y^1 , respectively. Then we define the Lagrange interpolation operator $\Pi_h^{L,1} : P_x^1 \rightarrow P_y^1$ such that:

$$(\Pi_h^{L,1} u_x) = \sum_{j=0}^{N_y} U_j^y \phi_j^y, \quad \text{where } U_j^y = \sum_{i=0}^{N_x} U_i^x \phi_i^x(y_j), \quad j = 0, \dots, N_y, y_j \in \mathcal{N}_y. \quad (2.1.1)$$

Although this method is faster and cheaper, it does not always preserve some physical properties. See section 2.2.1.

2.1.2 $L^2(\Omega)$ -projection

The $L^2(\Omega)$ -projection is a widely used conservative projection method in which the $L^2(\Omega)$ distance between $u_x \in P_x^1$ and $\Pi_h^0 u_x \in P_y^1$ is minimized, that is, we must solve an $L^2(\Omega)$ minimization problem. Here, Π_h^0 is the $L^2(\Omega)$ -projection operator. Define the function $J : P_y^1 \rightarrow \mathbb{R}$ as follows [7, 9, 11]:

$$J(v_y) = \frac{1}{2} \|u_x - v_y\|_{L^2(\Omega)}^2, \quad \text{for any } v_y \in P_y^1. \quad (2.1.2)$$

The $L^2(\Omega)$ -projection of u_x onto P_y^1 (see figure 2.1a), noted $u_y = \Pi_h^0 u_x$, is the minimizer of J over P_y^1 . From the 1st order optimality condition, $\nabla J(\Pi_h^0 u_x) = 0$. This

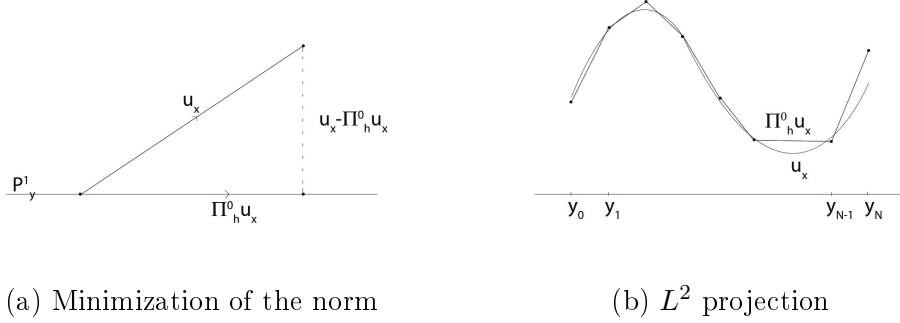


Figure 2.1: Left: Illustration of u_x and its L^2 projection on the space P_y^1 .
Right: L^2 projection of a continuous function on the space P_y^1 .

defines the $L^2(\Omega)$ -projection operator $\Pi_h^0 : P_x^1 \rightarrow P_y^1$, and lead us to the problem:
Find $\Pi_h^0 u_x \in P_y^1$ such that:

$$\int_{\Omega} (\Pi_h^0 u_x) v_y dx = \int_{\Omega} u_x v_y dx, \quad \forall v_y \in P_y^1. \quad (2.1.3)$$

Remark 2.1. $\Pi_h^0 : L^2(\Omega) \rightarrow P_y^1$ is defined in the same way by taking $u_x \in L^2(\Omega)$

The left hand side in equation (2.1.3) is the usual mass matrix in P_y^1 (see definition 1.14), but the right hand side is an integral of a product of functions on two different meshes, typically u_x given on τ_x and v_y on τ_y . In general, this integral is difficult and expensive to compute accurately. Using the basis functions and nodal values, we obtain $u_x = \sum_i U_i^x \phi_i^x$, $u_y = \sum_j U_j^y \phi_j^y$ and write the discrete system of equations to solve:

$$\sum_{j=0}^{N_y} U_j^y \int_{\Omega} \phi_i^y \phi_j^y dx = \int_{\Omega} u_x \phi_i^y dx, \quad \text{for } i = 0, \dots, N_y, \quad (2.1.4)$$

or in matrix form

$$M_y U_y = b, \quad (2.1.5)$$

where

- M_y is the standard mass matrix for the space P_y^1 , such that $M_y = (m_{ij})_{i,j=0,\dots,N_y}$ with $m_{ij} = \int_{\Omega} \phi_i^y \phi_j^y dy$;
- $U_y = [U_0^y, \dots, U_{N_y}^y]^T$ are nodal values on τ_y ;
- b is defined by $b_i = \int_{\Omega} u_x \phi_i^y dx$.

An illustrative example is provided in section 2.2.1.

2.1.3 $L^2(\Omega)$ -projection with mass lumping

For the $L^2(\Omega)$ -projection, a linear system must be solved, which is expensive from a computational perspective. A simple modification prevents the expensive solution of a linear system while ensuring mass conservation. It consists in replacing the mass matrix in the $L^2(\Omega)$ -projection by the lumped mass matrix given in definition 1.15. This defines the $L^2(\Omega)$ -projection with mass lumping transfer operator $\Pi_h^L : P_x^1 \rightarrow P_y^1$ such that

$$M_y^L U_y = b, \quad (2.1.6)$$

where M_y^L is the lumped mass matrix for the space P_y^1 , and U_y, b are defined as for the $L^2(\Omega)$ -projection. As in Farrell [5], we show that mass lumping by row summing has no effect on mass conservation, that is

$$\int_{\Omega} \Pi_h^L v_x dy = \int_{\Omega} v_x dx. \quad (2.1.7)$$

This results from the fact that the lumped mass matrix satisfies

$$\begin{aligned} m_{ii}^l &= \sum_{j=0}^{N_y} m_{ij} = \sum_{j=1}^{N_y} \int_{\Omega} \phi_i^y \phi_j^y dy, \quad (\text{by the definition}); \\ &= \int_{\Omega} \phi_i^y \left(\sum_j \phi_j^y \right) dx = \int_{\Omega} \phi_i^y dx \quad (\text{since } \sum_j \phi_j^y \equiv 1 \text{ on } \bar{\Omega}); \end{aligned} \quad (2.1.8)$$

Each nodal value on the current mesh is then given by

$$\begin{aligned} m_{ii}^l U_i^y &= \int_{\Omega} u_x \phi_i^y dx \Leftrightarrow U_i^y \int_{\Omega} \phi_i^y dx = \int_{\Omega} u_x \phi_i^y dx, \quad \text{for } i = 0, \dots, N_y. \\ \text{Thus, } \int_{\Omega} \Pi_h^L u_x dx &= \sum_{i=1}^{N_y} U_i^y \int_{\Omega} \phi_i^y dx; \\ &= \int_{\Omega} u_x \left(\sum_{i=1}^{N_y} \phi_i^y \right) dx; \\ &= \int_{\Omega} u_x dx \quad (\text{since } \sum_i \phi_i^y \equiv 1 \text{ on } \bar{\Omega}). \end{aligned} \quad (2.1.9)$$

However, one of the significant drawbacks of $L^2(\Omega)$ -projection is the high computational cost. This issue is caused by the need to compute the inner product of the basis functions defined on different supports. According to the available literature, the most accurate method for computing this inner product is to use supermesh construction [4].

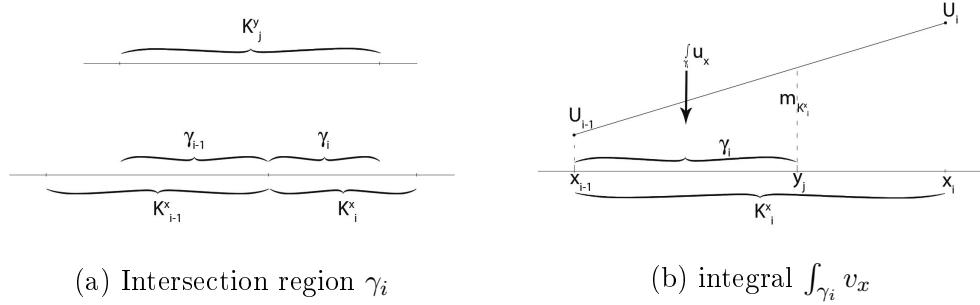


Figure 2.2: Intersection region γ_i and integral $\int_{\gamma_i} v_x$ for a mass conservative transfer between meshes

2.1.4 Alauzet and Mehrenberger formula

Alauzet and Mehrenberger [1] proposed a transfer method that is mass conservative, P^1 exact, and satisfies a local maximum principle, employing classical linear interpolation with a conservative projection operator to project solutions defined on elements. They first reconstruct the average value and gradient of u_x on the elements of the current mesh τ_y , and then recover nodal values on this mesh. The nodal values are obtained by averaging elementwise information rather than evaluating the inner product of the basis functions as we used in $L^2(\Omega)$ -projection. For our convenience, Alauzet and Mehrenberger projection will now be referred to as the **AM** transfer operator.

Let us begin with a three-step explanation of the one-dimension AM transfer operator:

1. Projection of $u_x \in P_x^1 \rightarrow \tilde{u}_y \in P_y^{1,dc}$;
2. Imposing monotonicity to get \bar{u}_y from \tilde{u}_y ;
3. Averaging or projecting from \bar{u}_y on P_y^1 to get $\Pi_h^A u_x$.

Step (i). We first reconstruct a function $\tilde{u}_y \in P_y^{1,dc}$ on the current mesh τ_y using some averaging procedure. Consider K_j^y , with j fixed, an element in τ_y . Borrowing from the definition [1.12], assume that $S_j = \{\cup K_i^x : K_i^x \cap K_j^y \neq \emptyset, K_i^x \in \tau_x\}$. Thus we can introduce the intersection region γ_i for each element $K_i^x \subseteq S_j$ such that $\gamma_i = K_j^y \cap K_i^x$, and compute the integrals $\int_{\gamma_i} u_x$ and $\int_{\gamma_i} \nabla u_x$ for each γ_i . We set the mass and gradient of the function \tilde{u}_y on the element K_j^y of the current mesh τ_y in the following way:

$$m_{K_j^y} = \int_{K_j^y} \tilde{u}_y = \sum_i \int_{\gamma_i} u_x \quad \text{and} \quad \nabla \tilde{u}_y|_{K_j^y} = \frac{\sum_i \int_{\gamma_i} \nabla u_x}{|K_j^y|}. \quad (2.1.10)$$

This yields a conservative and P^1 -exact reconstruction of the function \tilde{u}_y on the current mesh τ_y . Furthermore, the value of \tilde{u}_y at the barycenter g_j^y of K_j^y is set to

$$\tilde{u}_y(g_j^y) = \frac{1}{|K_j^y|} m_{K_j^y}. \quad (2.1.11)$$

For any point y of K_j^y , we set

$$\tilde{u}_y(y) = (\tilde{u}_y)(g_j^y) + (\nabla \tilde{u}_y)|_{K_j^y} \cdot (y - g_j^y). \quad (2.1.12)$$

We may use equation (2.1.12) to find the nodal values $\tilde{U}_{j-1}^+, \tilde{U}_j^-$ on each element K_j^y . Now we define a projection operator $\Pi_h^{A^1} : P_x^1 \rightarrow P_y^{1,dc}$ by setting $\Pi_h^{A^1} u_x = \tilde{u}_y$. This projection operator is P^1 -exact and satisfies the conservation property $\int_{\Omega} u_x = \int_{\Omega} \Pi_h^{A^1} u_x$. It should be noted that the function \tilde{u}_y may not satisfy the local maximum principle and is usually not continuous at the endpoints y_{j-1} and y_j of the element K_j^y .

Step (ii). We now built a function $\bar{u}_y \in P_y^{1,dc}$ that satisfies the local maximum principle. Let us apply corrections to the gradient of the function \tilde{u}_y on each element K_j^y of τ_y if \tilde{u}_y does not satisfy the maximum principle. Consider S_j to be the set of elements K_i^x in τ_x that overlap with K_j^y of τ_y . We want

$$u_{min} = \min_{x \in S_j} u_x(x) \leq (\bar{u}_y)(y) \leq \max_{x \in S_j} u_x(x) = u_{max}, \quad \forall y \in K_j^y. \quad (2.1.13)$$

We may notice that the inequality (2.1.13) is always satisfied by \tilde{u}_y at g_j^y , so we leave the value unchanged, that is $\bar{u}_y(g_j^y) = \tilde{u}_y(g_j^y)$, and update the gradient of \tilde{u}_y in each K_j^y . Suppose that $\tilde{U}_{j-1}^+ \leq \tilde{U}_j^-$ (if $\tilde{U}_j^- < \tilde{U}_{j-1}^+$, we can use the same procedure by permuting j with $j-1$ and $-$ with $+$). Define temporary nodal values \hat{U}_j^- and \hat{U}_{j-1}^+ on τ_y as follows:

$$\begin{aligned} \hat{U}_j^- &= \min\{\tilde{U}_j^-, u_{max}\} \\ \hat{U}_{j-1}^+ &= \tilde{U}_{j-1}^+ + \tilde{U}_j^- - \hat{U}_j^- \end{aligned} \quad (2.1.14)$$

and,

$$\begin{aligned} \bar{U}_{j-1}^+ &= \max\{\hat{U}_{j-1}^+, u_{min}\} \\ \bar{U}_j^- &= \hat{U}_j^- - (\bar{U}_{j-1}^+ - \hat{U}_{j-1}^+) \end{aligned} \quad (2.1.15)$$

These new nodal values \bar{U}_{j-1}^+ and \bar{U}_j^- define a corrected linear function \bar{u}_y on K_j^y . The corrected solution value for any point z in each K_j^y is then given by:

$$\bar{u}_y(z) = \sum_{j=1}^{N_y} U_{j-1}^+ \phi_{j-1}^+(z) + U_j^- \phi_j^-(z), \quad \forall z \in \Omega. \quad (2.1.16)$$

Set $\Pi_h^{A^2} \tilde{u}_y = \bar{u}_y$ to define the transfer operator $\Pi_h^{A^2} : P_y^{1,dc} \rightarrow P_y^{1,dc}$. This operator is P^1 -exact and is mass conservative. We have obtained a piecewise linear function $\bar{u}_y \in P_y^{1,dc}$ that satisfies the local maximum principle.

Step (iii). Now we construct a function $u_y \in P_y^1$ that is continuous on $\bar{\Omega}$, where, $u_y = \sum_{j=0}^{N_y} U_j^y \phi_j^y$. As proposed by Alauzet and Mehrenberger [1], we simply redistribute the solution to each node y_j in τ_y via local averaging:

$$U_j^y = \frac{|K_j^y| \bar{U}_j^- + |K_{j+1}^y| \bar{U}_j^+}{|K_j^y| + |K_{j+1}^y|}, \text{ where, } j = 1, \dots, N_y - 1, U_0^y = \bar{U}_0^+ \text{ and, } U_{N_y}^y = \bar{U}_{N_y}^-.$$
(2.1.17)

By setting $\Pi_h^{A^3} \bar{u}_y = u_y$, we define the transfer operator $\Pi_h^{A^3} : P_y^{1,dc} \rightarrow P_y^1$. This transfer operator is P^1 -exact, and conservative. Here, we mean by P^1 -exact that if $u_x \in P_x^1 \subseteq P_y^{1,dc}$, then $\Pi_h^{A^3} u_x = u_x$. Notice that this averaging process guarantees that the $\Pi_h^{A^3}$ operator satisfies the maximum principle. Finally we produce a continuous function $u_y \in P_y^1$ from $u_x \in P_x^1$ by taking the composition of $\Pi_h^{A^1}$, $\Pi_h^{A^2}$ and $\Pi_h^{A^3}$. We write

$$u_x \xrightarrow{\Pi_h^{A^1}} \tilde{u}_y \xrightarrow{\Pi_h^{A^2}} \bar{u}_y \xrightarrow{\Pi_h^{A^3}} u_y.$$
(2.1.18)

Define the AM operator $\Pi_h^A : P_x^1 \rightarrow P_y^1$, as $\Pi_h^A u_x = \Pi_h^{A^3} \Pi_h^{A^2} \Pi_h^{A^1} u_x = u_y$. Because Πh^{A^1} , Πh^{A^2} and Πh^{A^3} of the AM operator are each P^1 -exact and mass conservative, this AM operator is P^1 -exact and mass conservative. Further AM operator preserves the maximum principle.

We can also replace the local averaging that we used in step (iii) with the following projection methods to obtain a continuous function $u_y \in P_y^1$ on the current mesh τ_y :

- L^2 projection from $P_y^{1,dc}$ to P_y^1 (noted $\Pi_h^{A,0}$);
- L^2 projection with mass lumping (noted $\Pi_h^{A,L}$).

Let \bar{U}_j^- , \bar{U}_j^+ be the nodal values at y_j from step (ii) of the AM formula on elements K_j^y and K_{j+1}^y , respectively (see figure 1.2b), and U_j^y the projected nodal value on P_y^1 at node y_j . Then we rewrite equation (2.1.17):

$$(|K_j^y| + |K_{j+1}^y|)U_j^y = |K_{j+1}^y| \bar{U}_j^+ + |K_j^y| \bar{U}_j^- \quad \forall j = 1, \dots, N_y - 1.$$
(2.1.19)

Here, $U_0^y = \bar{U}_0^+$, and $U_{N_y}^y = \bar{U}_{N_y}^-$ at the boundary. First, by replacing step (iii) of the AM operator with $L^2(\Omega)$ -projection, we get

$$\begin{aligned} \frac{1}{3}(|K_j^y| U_{j-1}^y + 2(|K_j^y| + |K_{j+1}^y|)U_j^y + |K_{j+1}^y| U_{j+1}^y) &= \frac{|K_{j+1}^y|}{3}(2\bar{U}_j^+ + \bar{U}_{j+1}^-) \\ &\quad + \frac{|K_j^y|}{3}(\bar{U}_{j-1}^+ + 2\bar{U}_j^-) \quad \forall j = 1, \dots, N_y - 1. \end{aligned}$$
(2.1.20)

Here, at the boundary $2U_0^y + U_1^y = 2\bar{U}_0^+ + \bar{U}_1^-$ and $U_{N_y-1}^y + 2U_{N_y}^y = \bar{U}_{N_y-1}^+ + 2\bar{U}_{N_y}^-$. As an other choice, for step(iii) of the AM operator, we can use the $L^2(\Omega)$ -projection with mass lumping as follows:

$$\begin{aligned} (|K_j^y| + |K_{j+1}^y|)U_j^y &= \frac{|K_{j+1}^y|}{3}(2\bar{U}_j^+ + \bar{U}_{j+1}^-) \\ &\quad + \frac{|K_j^y|}{3}(\bar{U}_{j-1}^+ + 2\bar{U}_j^-) \quad \forall j = 1, \dots, N_y - 1. \end{aligned} \quad (2.1.21)$$

Here, boundary vertices are giving, $U_0^y = \frac{2\bar{U}_0^+ + \bar{U}_1^-}{3}$ and $U_{N_y}^y = \frac{\bar{U}_{N_y-1}^+ + 2\bar{U}_{N_y}^-}{3}$. Notice that, the left hand sides of both equations (2.1.19) and (2.1.21) lead to a linear system with diagonal matrix. Indeed, we have:

Proposition 2.1. Suppose that $\bar{u}_y \in P_y^{1,dc}$. The functions $u_y \in P_y^1$ solutions of equation (2.1.19) and (2.1.21) are equal iff $\bar{U}_j^+ = \bar{U}_{j+1}^- \forall j = 0, \dots, N_y - 1$.

Proof: We have

$$\begin{aligned} (2.1.19) &\Leftrightarrow (|K_j^y| + |K_{j+1}^y|)U_j^y = |K_{j+1}^y|\bar{U}_j^+ + |K_j^y|\bar{U}_j^-, \quad U_0^y = \bar{U}_0^+ \text{ and, } U_{N_y}^y = \bar{U}_{N_y}^-. \\ (2.1.21) &\Leftrightarrow (|K_j^y| + |K_{j+1}^y|)U_j^y = \frac{|K_{j+1}^y|}{3}(2\bar{U}_j^+ + \bar{U}_{j+1}^-) + \frac{|K_j^y|}{3}(\bar{U}_{j-1}^+ + 2\bar{U}_j^-), \\ U_0^y &= \frac{2\bar{U}_0^+ + \bar{U}_1^-}{3}, \text{ and } U_{N_y}^y = \frac{\bar{U}_{N_y-1}^+ + 2\bar{U}_{N_y}^-}{3}. \end{aligned}$$

U_j^y are the same in equations (2.1.19) and (2.1.21) iff, for $j = 1, \dots, N_y - 1$,

$$\begin{aligned} |K_{j+1}^y|U_j^+ + |K_j^y|U_j^- &= |K_{j+1}^y|[U_j^+ + \frac{(-U_j^+ + U_{j+1}^-)}{3}] + |K_j^y|[\frac{(U_{j-1}^+ - U_j^-)}{3} + U_j^-], \\ \bar{U}_0^+ &= \frac{2\bar{U}_0^+ + \bar{U}_1^-}{3}, \text{ and } \bar{U}_{N_y}^- = \frac{\bar{U}_{N_y-1}^+ + 2\bar{U}_{N_y}^-}{3}; \\ \Leftrightarrow |K_{j+1}^y|(U_{j+1}^- - U_j^+) &= |K_j^y|(U_j^- - U_{j-1}^+), \quad \bar{U}_0^+ = \bar{U}_1^- \text{ and } \bar{U}_{N_y}^- = \bar{U}_{N_y-1}^+; \\ \Leftrightarrow & \bar{U}_j^+ = \bar{U}_{j+1}^- \forall j = 0, \dots, N_y - 1. \end{aligned}$$

■

That is, solutions of equation (2.1.19) and (2.1.21) are equal iff the function \bar{u}_y is constant in each element K_j^y .

Although the right hand side of equations (2.1.20) and (2.1.21) is the same, we cannot find a relationship between the nodal values resulting from both equations because the left hand side of equation (2.1.20) corresponds to a linear system with

tridiagonal matrix, whose solution differs in general from the solution of the diagonal system associated to (2.1.19) and (2.1.21).

We compare the numerical solutions u_y obtained after replacing step (iii) of the AM operator with equations (2.1.20) and (2.1.21) in subsequent sections.

2.2 Comparison of transfer methods

In this section, we analyze the behavior and effectiveness of the previously discussed transfer methods using two closed form functions. We use the 1D domain $\overline{\Omega} = [0, 10]$ for all our test cases. Lagrange interpolation, $L^2(\Omega)$ -projection, $L^2(\Omega)$ -projection with mass lumping and the AM operator are compared, with a focus on mass conservation, P^1 -exactness and monotonicity.

To begin, two overlapping finite element meshes τ_x and τ_y with constant element sizes $h_x = 0.05$ and $h_y = 0.1$, respectively, are considered:

$$\begin{aligned}\tau_x : x_i &= ih_x, \quad i = 0, \dots, 200; \\ \tau_y : y_j &= jh_y, \quad j = 0, \dots, 100.\end{aligned}\tag{2.2.1}$$

In the case of two nonoverlapping finite element meshes, we keep the coarser mesh τ_y and introduce the finer mesh τ_x as follows, with $h = 0.025$:

$$\tau_x : x_0 = 0, \quad x_i = (2i - 1)h \quad \text{for } i = 1, \dots, 200, \quad \text{and } x_{201} = 10. \tag{2.2.2}$$

The same meshes τ_x and τ_y were used for all of the results presented in the following section, with the exact solution drawn in a much finer mesh than τ_x and τ_y . From a background mesh to a current mesh, the following general transfer operators are introduced:

- $u_y = \Pi_h u_x$, where $\Pi_h : P_x^1 \rightarrow P_y^1$;
- $u_x = \tilde{\Pi}_h u_y$, where $\tilde{\Pi}_h : P_y^1 \rightarrow P_x^1$.

We compare the transferred approximations with the exact solution u by computing the error in L^2 -norm, that is

$$\begin{aligned}\|u - u_x\|_{0,\Omega} &= \left(\sum_{i=1}^{N_x} \int_{K_i^x} |u - u_x|^2 dx \right)^{\frac{1}{2}}, \\ \|u - u_y\|_{0,\Omega} &= \left(\sum_{i=1}^{N_y} \int_{K_i^y} |u - u_y|^2 dy \right)^{\frac{1}{2}}.\end{aligned}\tag{2.2.3}$$

2.2.1 Test case with a regular function

The first example is done with a smooth Gaussian function specified by the following closed-form expression:

$$u_1(x) = \exp(-(x - 5)^2), \quad x \in [0, 10]. \quad (2.2.4)$$

The previously discussed transfer methods are considered. More precisely, starting

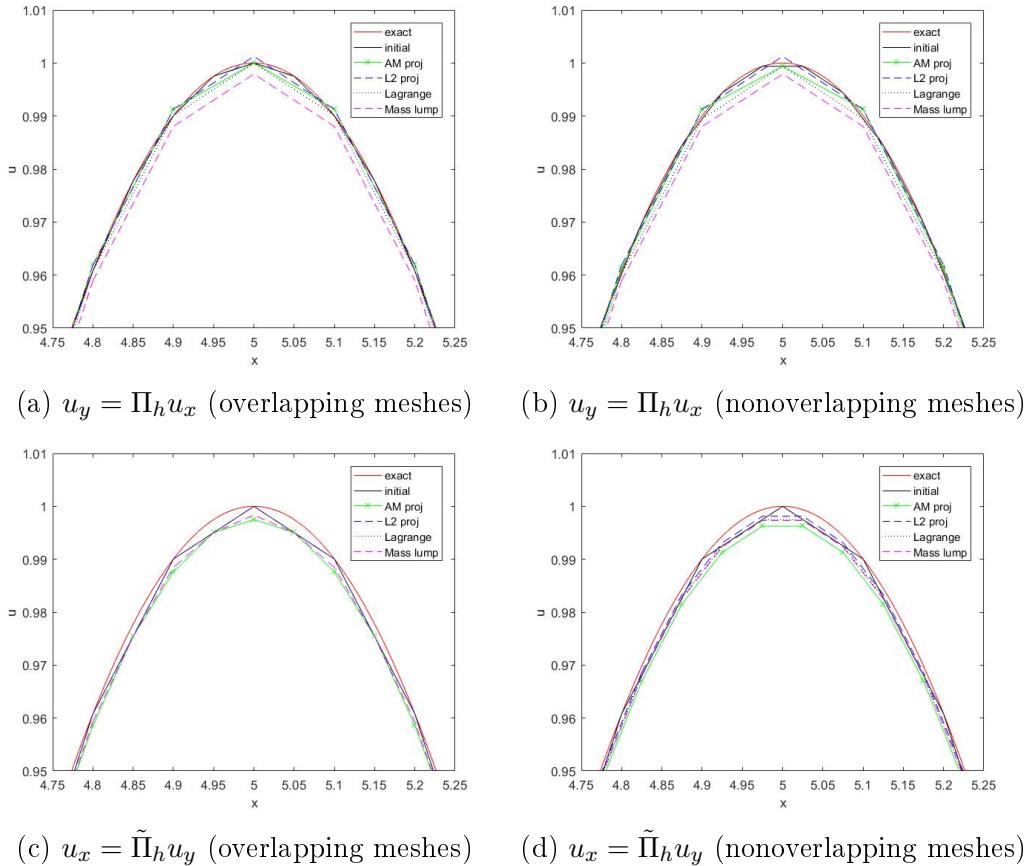


Figure 2.3: Comparing transfer operators to project the initial function u_1 from the background mesh to the current mesh, using a single transfer step. Top: transferring from a finer mesh τ_x to a coarser mesh τ_y . Bottom: transferring from a coarser mesh τ_y to a finer mesh τ_x .

from the function $u_x \in P_x^1$ with nodal values $U_i^x = u_1(x_i)$, $i = 0, \dots, N_x$, we look for $u_y \in P_y^1$ using a single transfer step. We assume here that τ_x is the initial background mesh and τ_y is the current mesh when transfer is done from τ_x to τ_y . Similarly, we can also consider transfer from τ_y to τ_x , in which case τ_y is the background mesh and τ_x is the current mesh. The figure 2.3 shows the zoomed-in region of the graphs with

the largest deviation. Additionally, in the figure [2.3c] the initial function and the Lagrange interpolation are superposed with the $L^2(\Omega)$ -projection. This is possible because $P_x^1 \subseteq P_y^1$. Graphs of the Lagrange interpolation and the $L^2(\Omega)$ -projection with mass lumping show as superposed on figure [2.3d].

Overall, the $L^2(\Omega)$ -projection provides the best approximation for the function given in (2.2.4). In the case of finer to coarser mesh projection, the $L^2(\Omega)$ -projection does not preserve the local maximum principle and the AM operator provides a reasonably accurate approximation. Furthermore, in this case the $L^2(\Omega)$ -projection with mass lumping provides the least accurate approximation (see figures [2.3a], [2.3b]). In the case of coarser to finer mesh projection, the AM projection operator provides the least accurate approximation (see figures [2.3c], [2.3d]). Moreover, all of the graphs in figures [2.3c] and [2.3d] follow the local maximum and minimum principle. We compare the error in L^2 -norm and the results are summarized in the following table.

Table 2.1: L^2 -error between the function u_1 and the transferred approximations for a single transfer step.

Figure	N_{back}	$N_{current}$	L^2 -error			
			AM proj	L^2 proj	Lagrange	Mass lump
[2.3a]	201	101	0.0278	0.0269	0.0535	0.0801
[2.3b]	202	101	0.0425	0.0269	0.0627	0.0801
[2.3c]	101	201	0.0706	0.0535	0.0535	0.0654
[2.3d]	101	202	0.0778	0.0534	0.0626	0.0654

We now investigate a re-projection example that involves applying the projection operator 1000 times between the background and current meshes. For each step, nodal values on the current mesh are calculated using nodal values on the background mesh. More precisely, starting from the function $u_x \in P_x^1$ with nodal values $U_i^x = u_1(x_i)$, $i = 0, \dots, N_x$, we look for $(\tilde{\Pi}_h \Pi_h)^m u_x \in P_x^1$. Similarly, when starting from the function $u_y \in P_y^1$, with nodal values $U_j^y = u_1(y_j)$ for $j = 0, \dots, N_y$, we are looking for $(\Pi_h \tilde{\Pi}_h)^m u_y \in P_y^1$. Figure [2.4] is obtained when $m = 500$. Note that, since $(\tilde{\Pi}_h \Pi_h)^n = \tilde{\Pi}_h (\tilde{\Pi}_h \Pi_h)^{n-1} \Pi_h \approx (\Pi_h \tilde{\Pi}_h)^n$, when n is large, graphs for overlapping meshes are similar (see figures [2.4a], [2.4c]) and graphs for nonoverlapping meshes are similar (see figures [2.4b], [2.4d]). Here, some of the graphs are superposed. In figures [2.4a], [2.4c], the Lagrange interpolation is superposed with the $L^2(\Omega)$ -projection. Furthermore, in all graphs of the figure [2.4], the exact solution and the $L^2(\Omega)$ -projection show as superposed. In figures [2.4b], [2.4d], the AM operator approximation and the Lagrange interpolation are very close to each other.

As illustrated in figure [2.4], the $L^2(\Omega)$ -projection gives the most accurate approximation, while the $L^2(\Omega)$ -projection with mass lumping gives the least accurate

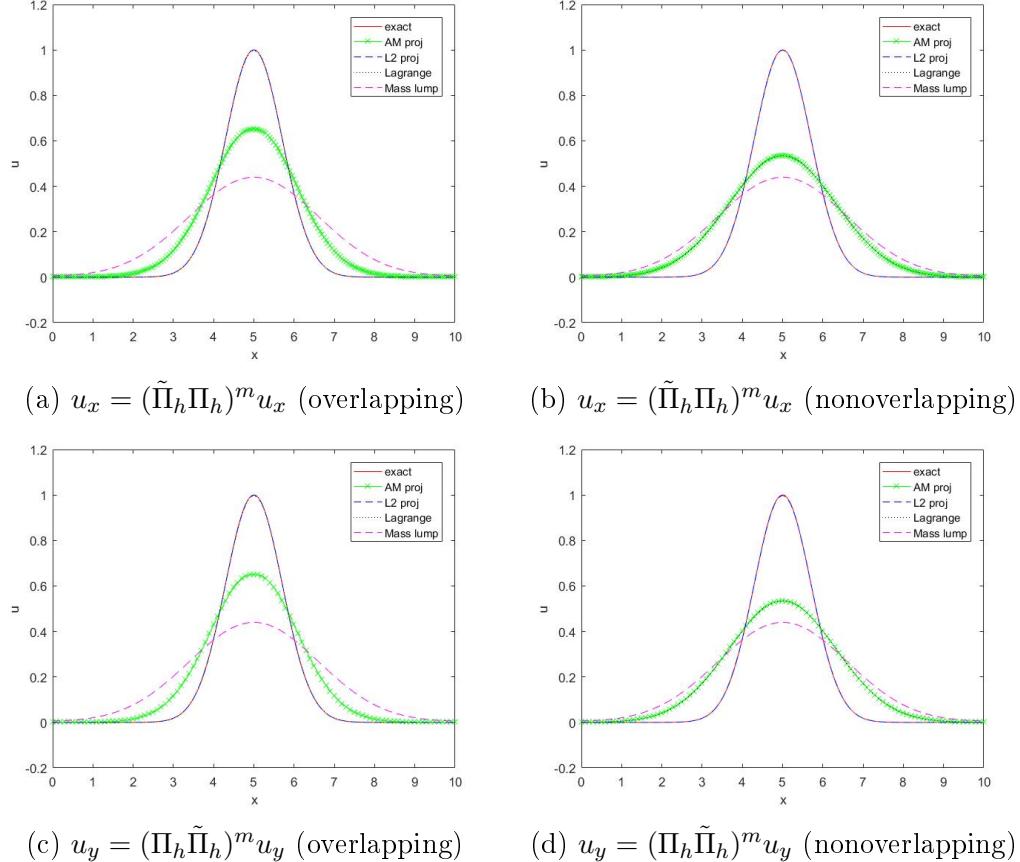


Figure 2.4: Comparing transfer operators to project the initial function u_1 from the background mesh to the current mesh, when $m = 500$.

approximation. In overlapping meshes, the Lagrange interpolation accurately transfers data from the background mesh to the current mesh and is cheaper in terms of computational time. However, when two non-overlapping meshes are used, the Lagrange interpolation method produces significant error, which evolves as the number of iterations increases and causes damping, as shown in the figures 2.4b 2.4d. Due to the damping nature of the Lagrange interpolation operator, the mass of the transferred function corresponding to Lagrange interpolation is reduced after thousand interpolations, which contradicts mass conservation properties.

Concerning CPU time, all transfer methods were compared for four cases on a desktop computer with an Intel Core i5 processor running at 2.50GHz. The results are summarized in table 2.2. The AM transfer operator provides the fastest computational methods, whereas the $L^2(\Omega)$ -projection takes the longest to execute. Unfortunately, there is no explanation for the AM projection being faster than the Lagrange interpolation operator.

Table 2.2: CPU time comparison between the transfer methods for 1000 projections.

Figure	N_{back}	$N_{current}$	CPU time (s)			
			AM proj	L^2 proj	Lagrange	Mass lump
2.4a	201	101	0.135378	0.559129	0.191492	0.462158
2.4b	202	101	0.135186	0.589846	0.279509	0.484269
2.4c	101	201	0.127121	0.550514	0.198814	0.462993
2.4d	101	202	0.135255	0.584624	0.278089	0.466133

2.2.2 Comparing the variants of the AM transfer operator

We now compare the numerical solutions obtained using Π_h^A with the three variants $\Pi_h^{A,3}$, $\Pi_h^{A,0}$ and $\Pi_h^{A,L}$ in step (iii) on the function u_1 for a single transfer step. Figure 2.5 shows a zoomed-in region of the graphs illustrating the greatest deviation. In figures 2.5b, 2.5d, the AM operator and the AM operator with the $L^2(\Omega)$ -projection approximations show as superposed.

The AM operator with $\Pi_h^{A,3}$ provides the best approximation among those with no wiggles, while the AM operator using the $L^2(\Omega)$ -projection with mass lumping, i.e. $\Pi_h^{A,L}$, provides the least accurate approximation. In the case of overlapping meshes, if the transfer is from a finer mesh, the AM transfer operator with the $L^2(\Omega)$ -projection fails to satisfy the local maximum principle, which violates one of the AM formula's main criteria (see figure 2.5a), and when transferring from a coarser mesh, the AM transfer operator with $L^2(\Omega)$ -projection produces wiggles (see figure 2.5c).

2.2.3 Test case with a discontinuous function

Finally, the transfer methods' ability to deal with one-dimensional discontinuous functions is tested. We consider the following square wave

$$u_2(x) = \begin{cases} 1, & \text{if } x \in (4, 7); \\ 0, & \text{otherwise.} \end{cases} \quad (2.2.5)$$

We study the projection of the function u_2 representing two steps in the domain $\bar{\Omega} = [0, 10]$. Results for all transfer methods are presented in figure 2.6. To investigate the deviation of the solution near discontinuities, a zoomed-in version of the graphs is presented on figures 2.6b and 2.6d. Here some of the graphs are superposed. In figures 2.6a, 2.6c, the AM approximation is superposed with the Lagrange interpolant. In the coarser to finer overlapping mesh case (see figure 2.6e), the $L^2(\Omega)$ -projection and Lagrange interpolation approximations are superposed with each other. In fact, all

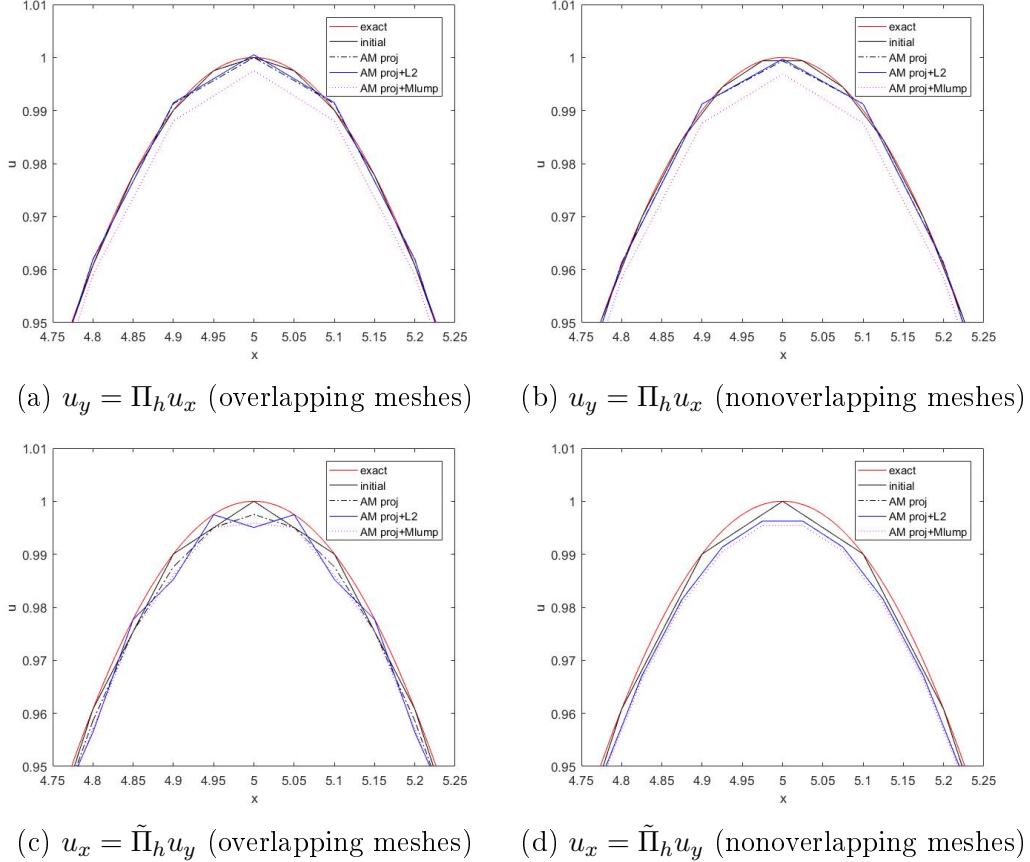


Figure 2.5: Comparing the AM transfer operator after replacing step (iii) with equations (2.1.20) and (2.1.21) for a single transfer step. Top: transferring from a finer mesh τ_x to a coarser mesh τ_y . Bottom: transferring from a coarser mesh τ_y to a finer mesh τ_x .

the graphs show as superposed. In the coarser to finer nonoverlapping case all the graphs are very close to each other (see figure 2.6f).

In this case, the AM transfer operator or the Lagrange interpolation method provides the best approximation. Furthermore, as illustrated in figures 2.6b and 2.6d, the $L^2(\Omega)$ -projection of jump discontinuities results in an approximation with wiggles, a typical example of the Gibbs phenomenon [8], which violates the global maximum principle. In the coarser to finer overlapping case, the AM transfer operator provides the best approximation, whereas in the coarser to finer nonoverlapping case, the Lagrange interpolation provides the best approximation to the function (3.1.6).

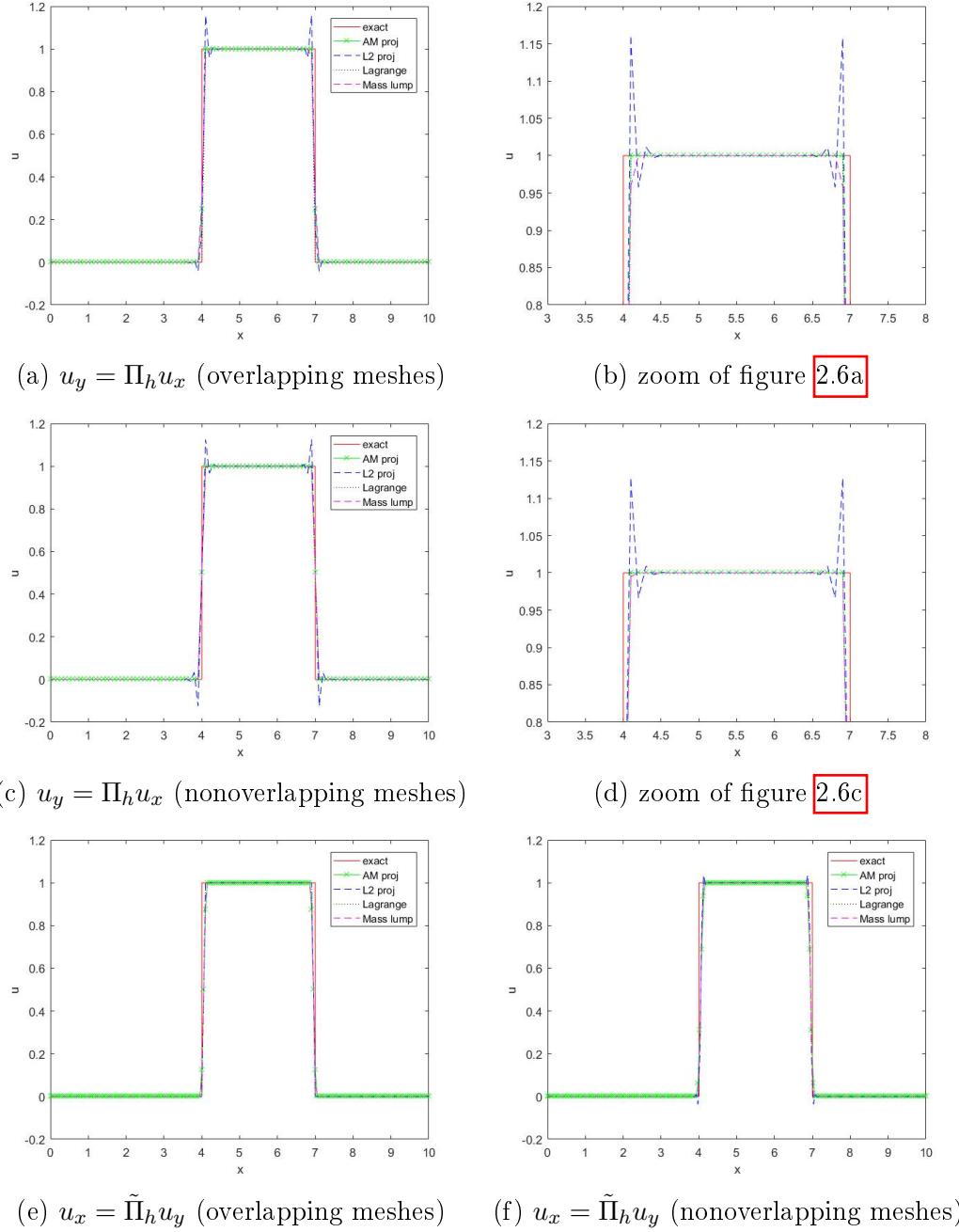


Figure 2.6: Comparing transfer operators to project the function u_2 , using a single transfer step. Figures 2.6a, 2.6c, transferring from a finer mesh τ_x to a coarser mesh τ_y . Figures 2.6e, 2.6f, transferring from a coarser mesh τ_y to a finer mesh τ_x .

Chapter 3

Applications to PDE

The primary goal of this chapter is to solve differential equations numerically using the mesh transfer strategies covered in previous chapters. To carry out this analysis, two well-known partial differential equations (PDE) will be presented and at the end of each section, we also conduct numerical experiments to analyze the performance of transfer methods while solving PDEs.

3.1 Transport equation

To begin, consider the one-dimensional first-order time-dependent transport equation on the domain $\Omega = (a, b) \subset \mathbb{R}$ with the velocity $\alpha(t)$, the initial solution $u_0(x)$, the boundary value $u_a(t)$ and the final time T . Set, $\partial_t u = \frac{\partial u(x,t)}{\partial t}$, $\partial_x u = \frac{\partial u(x,t)}{\partial x}$. We look for a solution $u = u(x, t)$ of the following problem:

$$\begin{aligned} \partial_t u + \alpha(t) \partial_x u &= 0, \quad x \in \Omega, \quad t \in (0, T]; \\ u(x, 0) &= u_0(x), \quad x \in \Omega; \\ u(a, t) &= u_a(t), \quad t \in (0, T]. \end{aligned} \tag{3.1.1}$$

When α is a constant, we know that the closed form solution to equation (3.1.1) is $u(x, t) = u_0(x - \alpha t)$ assuming the proper boundary value is used. Our aim is to compare the numerical approximations for the transport equation using the transfer methods with the closed form solution. For any positive integer N , $\delta t = T/N$ is the time step, $t^n = n\delta t$, $n = 0, \dots, N$, and Π_h (respectively $\tilde{\Pi}_h$) is the generic transfer operators from P_x^1 to P_y^1 (respectively, from P_y^1 to P_x^1). The backward Euler finite difference method in time and finite element discretization in space are considered. Assuming that the initial mesh is τ_x , starting from $u_x^0 = \Pi_h^{L,1} u_0$, for each $n = 1, \dots, N$, given $u_x^{n-1} \in P_x^1$:

- If no mesh adaptation is done during the time step, solve for $u_x^n \in P_x^1$ such that

$$\frac{u_x^n - u_x^{n-1}}{\delta t} + \alpha^n \partial_x u_x^n = 0. \quad (3.1.2)$$

- If mesh adaptation is done to generate τ_y from τ_x , solve for $u_y^n \in P_y^1$ such that

$$\frac{u_y^n - \Pi_h u_x^{n-1}}{\delta t} + \alpha^n \partial_y u_y^n = 0. \quad (3.1.3)$$

where, u_y^n is on the current mesh τ_y , u_x^{n-1} is on the background mesh τ_x and $\alpha^n = \alpha(t^n)$.

For the second case where mesh adaptation is done during the time step, we can also consider a transfer from τ_y to τ_x , in which case u_y^{n-1} is on the background mesh τ_y and u_x^n is on the current mesh τ_x . The variational formulation is obtained by modifying equations (3.1.2) and (3.1.3) accordingly.

3.1.1 SUPG stabilization

We apply the streamline upwind Petrov-Galerkin (SUPG) stabilization to the transport equation with the backward Euler method in time. Taking any element in the current mesh as K , and the SUPG parameter as $\nu_K = \frac{\nu h_x}{\alpha}$ [12], we have:

- If no mesh adaptation is done during the time step, find $u_x^n \in P_x^1$ such that,

$$\sum_{K \in \tau_x} \int_K \left(\frac{u_x^n - u_x^{n-1}}{\delta t} + \alpha^n \partial_x u_x^n \right) (v_x + \nu_K \alpha^n \partial_x v_x) dx = 0, \quad \forall v_x \in P_x^1.$$

- If mesh adaptation is done to generate τ_y from τ_x , find $u_y^n \in P_y^1$ such that,

$$\sum_{K \in \tau_y} \int_K \left(\frac{u_y^n - \Pi_h u_x^{n-1}}{\delta t} + \alpha^n \partial_y u_y^n \right) (v_y + \nu_K \alpha^n \partial_y v_y) dy = 0, \quad \forall v_y \in P_y^1.$$

Note that, for the one dimensional transport equation, the Courant–Friedrichs–Lowy (CFL) number is defined as

$$C = \alpha \frac{\delta t}{h_x}. \quad (3.1.4)$$

There is no stability condition based on the CFL number since our time-stepping method is implicit, but we will use this number to compare solutions in our numerical tests.

3.1.2 Numerical example

Test case 1

The first closed-form example uses a smooth time-dependent Gaussian function given by:

$$u_3(x, t) = \exp(-((x - 2) - t)^2), \quad x \in [0, 10]. \quad (3.1.5)$$

We choose $\alpha(t) = 1$, $u_3(x, 0) = \exp(-(x - 2)^2)$ and $u_3(0, t) = \exp(-(t + 2)^2)$ in problem (3.1.1).

First, we consider the case where no mesh adaptation is involved, i.e., we use over all time steps the same mesh τ_x on domain $\bar{\Omega} = [0, 10]$. The uniform mesh is laid out with element size $h_x = 0.1$ as in equation (2.2.1) for $i = 0, \dots, 100$. Furthermore, we consider the SUPG constant $\nu = 0.1$, the final time $T = 3$, and the constant time steps $\delta t = 0.1, 0.01, 0.001$. We keep h_x unchanged and compute the numerical results with and without SUPG stabilization using each δt . In other words, we compare numerical results where the CFL number is $C = 1, 0.1, 0.01$, respectively.

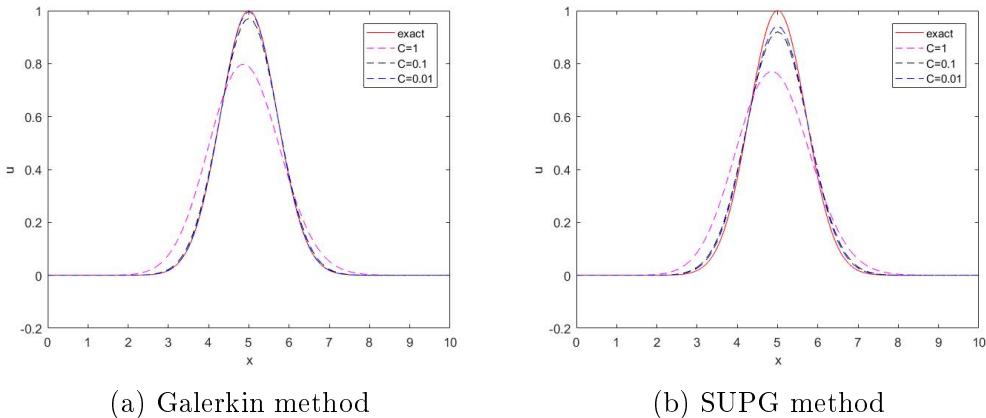


Figure 3.1: Numerical solutions of the transport equation for the manufactured solution u_3 using the fixed mesh τ_x , different CFL numbers and the final time $T = 3$.

Numerical solutions for this test case are plotted on figure 3.1. In figure 3.1a, numerical solution with CFL number $C_3 = 0.01$ shows as superposed with the exact solution. Furthermore, we observe that, as the CFL number decreases, the accuracy of the numerical approximation increases. The best approximation is obtained when the CFL number is 0.01, and the least accurate approximation is obtained when the CFL number is 1 (see figure 3.1). We can see that SUPG stabilization is not necessarily required in this case because the Galerkin solutions are already stable (see figure 3.1a).

Now consider, two overlapping meshes τ_x and τ_y , built on the same domain $\bar{\Omega} = [0, 10]$, with constant element size $h_x = 0.1$ and $h_y = 0.2$, respectively. Layout

of the uniform meshes τ_x and τ_y are as in equation (2.2.1) for $i = 0, \dots, 100$ and $j = 0, \dots, 50$, respectively. In the case of non-overlapping meshes, we keep τ_y unchanged, and introduce the finer mesh τ_x , with $h = 0.05$ as in equation (2.2.2), for $i = 1, \dots, 100$, where, $x_0 = 0$ and $x_{101} = 10$. Further we take the SUPG constant $\nu = 0.1$, final time $T = 3$, and for our convenience the time step $\delta t = 0.01$. Numerical results are reported for each projection method discussed in previous chapters with and without SUPG stabilization. If the initial mesh is τ_x we are starting from the function $u_x^0 = \Pi_h^{L,1} u_3(x, 0)$.

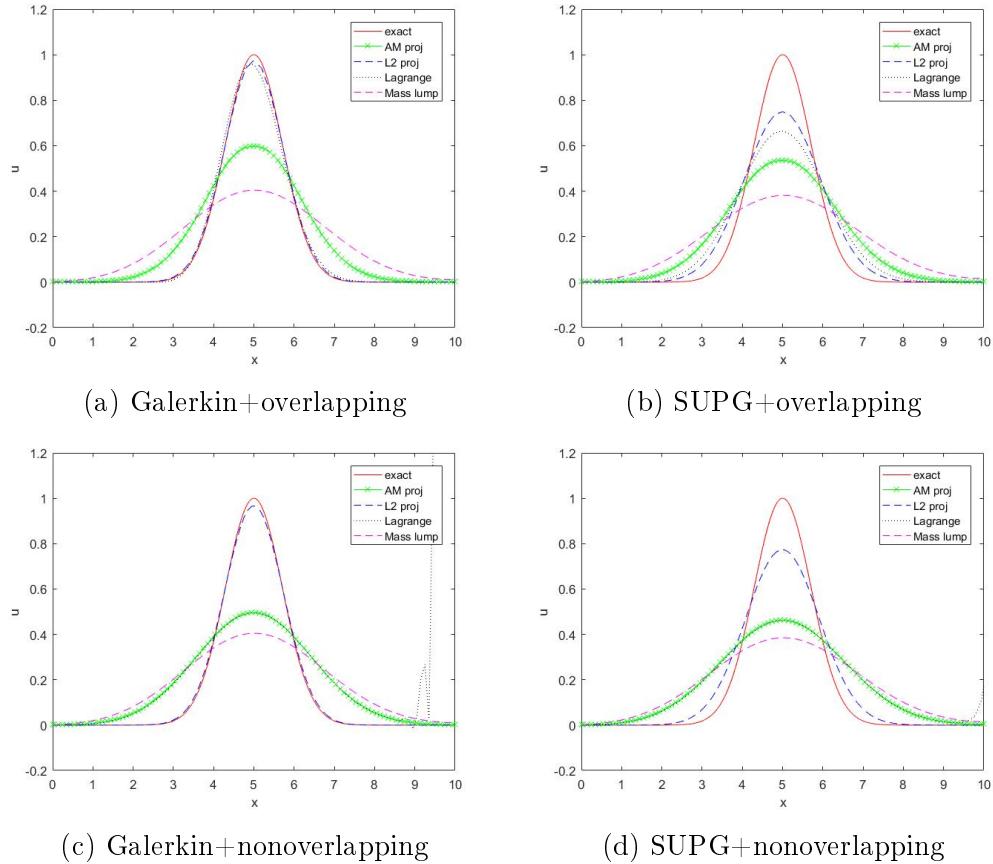


Figure 3.2: Numerical solutions of the transport equation for the manufactured solution u_3 using different meshes τ_x and τ_y with the final time $T = 3$, CFL number 0.1 and various transfer operators. Top row: overlapping meshes τ_x and τ_y . Bottom row: nonoverlapping meshes τ_x and τ_y .

Clearly, in both overlapping and nonoverlapping mesh cases (see figure 3.2), the best results are obtained when using the $L^2(\Omega)$ -projection, worst results corresponding to the $L^2(\Omega)$ -projection with mass lumping. It should be noted that in Galerkin method with overlapping mesh cases, the Lagrange interpolation and the

$L^2(\Omega)$ -projection produce similar results. Also note that, for nonoverlapping meshes, Lagrange interpolation gives an approximation which is similar to the AM projection approximation. In all cases except for the $L^2(\Omega)$ -projection with the Galerkin method, the transfer of the solution between meshes at every time step substantially deteriorates the quality of the solution, and more so with the SUPG method. Again, the SUPG method is not needed here.

Test case 2

Consider the following square wave function given by

$$u_4(x, t) = \begin{cases} 1, & \text{if } (x - t) \in (1, 2); \\ 0, & \text{otherwise.} \end{cases} \quad (3.1.6)$$

We choose $\alpha(t) = 1$, $u_4(0, t) = 0$ and $u_0(x) = u_4(x, 0)$ in the equation (3.1.1).

Consider the case where we use a fixed mesh, τ_x on the domain $\bar{\Omega} \in [0, 10]$, with the element size $h_x = 0.01$. In this case, we take the SUPG constant $\nu = 0.1$, the final time $T = 3$, and the constant time steps $\delta t = 0.01, 0.001$ which correspond to CFL number $C = 1, 0.1$, respectively.

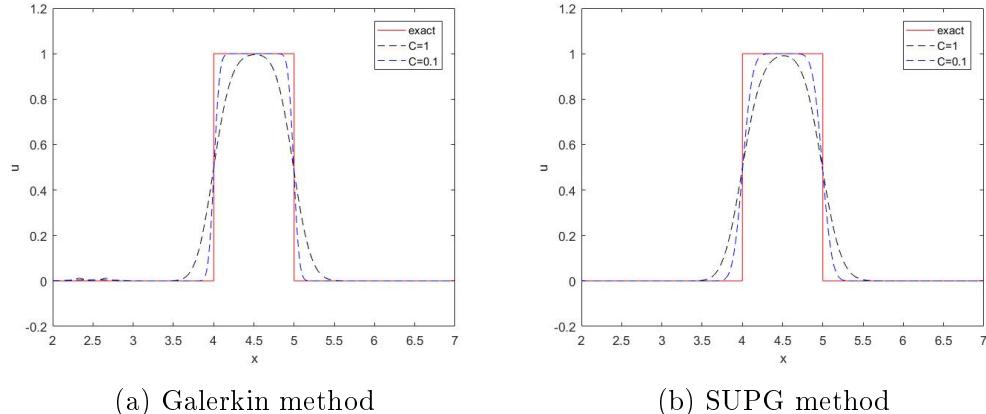


Figure 3.3: Solution comparison of the transport equation using manufactured solution u_4 for different CFL numbers and final time $T = 3$

Numerical solutions for this test case are plotted on figure 3.3. In this figure we observe that, as the CFL number decreases, the accuracy of the numerical approximation increases. Furthermore in the interval $(2, 3)$ of figure 3.3a we observe some wiggles, but with the SUPG stabilization we get a smooth graph with no wiggles. The best approximation is obtained when the CFL number is smaller. The SUPG stabilization is needed in this test case where the exact solution is less regular to remove spurious oscillations from the numerical solution (see figure 3.3b).

In figure 3.4 we plot the graphs for $t = 0.1, 0.2$ with $h_x = 0.01$ and the CFL number 0.1. In figure 3.4b, 3.4d is presented the zoomed-in region of the graphs

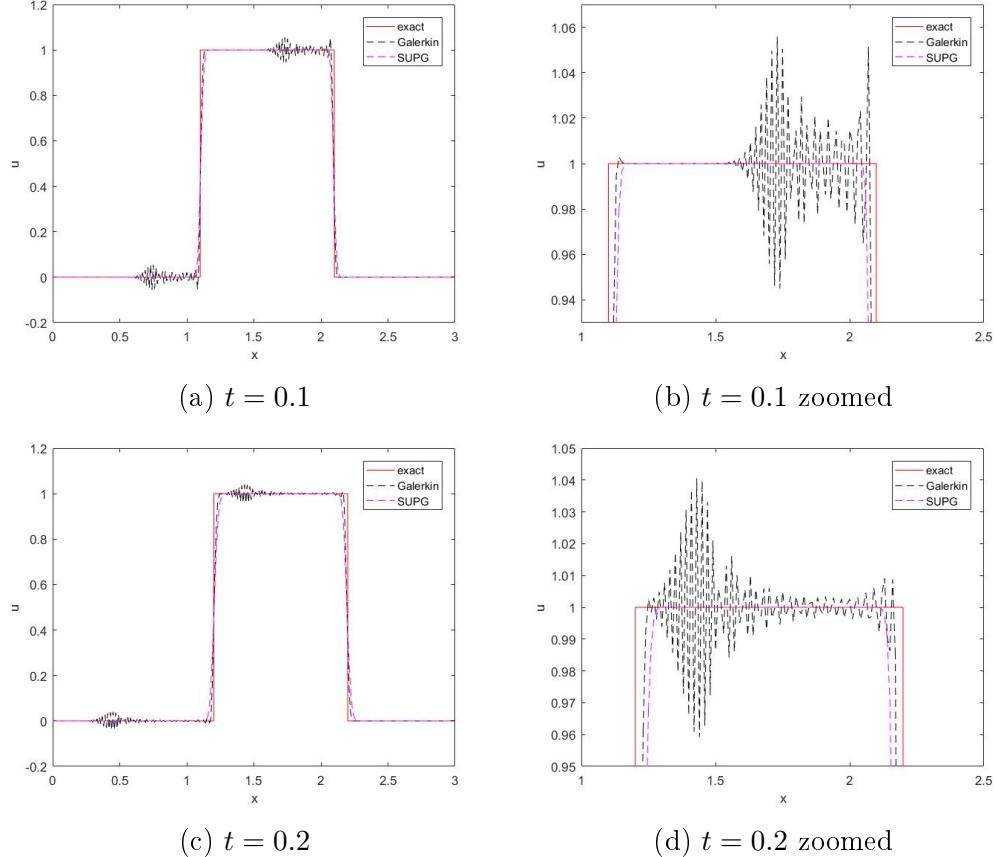


Figure 3.4: Comparison of the Galerkin and SUPG methods using the manufactured solution u_4 with CFL number 0.1.

which shows the largest deviation. Both figures 3.4a and 3.4c show high-frequency wiggles when we use the Galerkin method, but by using SUPG stabilization with $\nu = 0.1$, we remove all these oscillations.

Now we consider, two overlapping meshes on the same domain $\bar{\Omega} = [0, 10]$, with constant element size $h_x = 0.01$ and $h_y = 0.02$ as in equation (2.2.1). In the case of nonoverlapping meshes we keep τ_y unchanged, and introduce τ_x , with $h = 0.005$ as in equation (2.2.2). We take the SUPG constant $\nu = 0.1$.

Clearly, in both overlapping and nonoverlapping mesh cases the best results are obtained when using the $L^2(\Omega)$ -projection, the least accurate approximation corresponding to $L^2(\Omega)$ -projection with mass lumping. In figures 3.5a and 3.5b the $L^2(\Omega)$ projection and Lagrange interpolation produce similar results, and in figure 3.5c and 3.5d the AM operator and Lagrange interpolation show as superposed, illustrating how the solution with Lagrange reinterpolation deteriorates when nonoverlapping meshes are used.

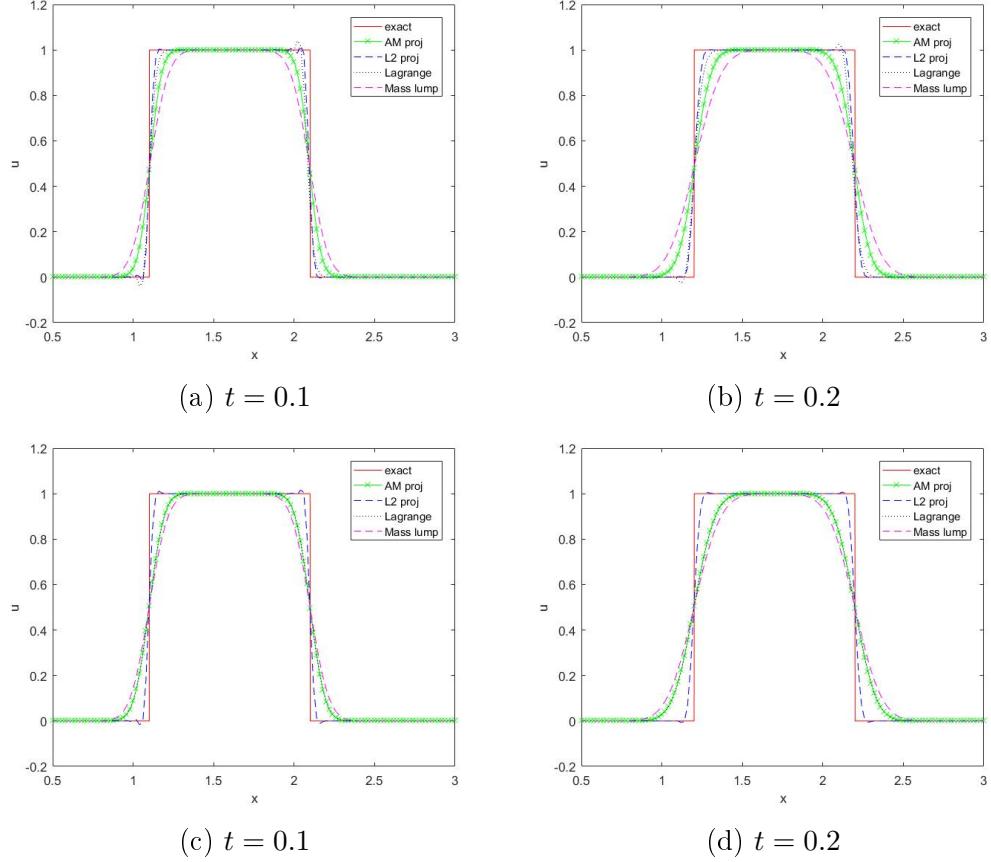


Figure 3.5: Numerical solutions of the transport equation for the manufactured solution u_4 using different meshes τ_x and τ_y with the CFL number 0.1 and various transfer operators. Top row: overlapping meshes τ_x and τ_y . Bottom row: nonoverlapping meshes τ_x and τ_y .

3.2 Heat equation

Consider a problem for the time-dependent heat equation written as,

$$\begin{aligned} \partial_t u(x, t) &= \alpha(t) \partial_{xx} u(x, t) + f(x, t), \quad x \in \Omega = (a, b), \quad t \in (0, T]; \\ u(x, 0) &= u_0(x), \quad x \in \Omega; \\ u(a, t) &= u_a(t), \quad u(b, t) = u_b(t), \quad t \in (0, T]. \end{aligned} \tag{3.2.1}$$

Here, $u(x, t)$ is the unknown function depending on space and time, $f(x, t)$ is the source function, the initial solution $u_0(x)$ is a function of space only, and boundary values $u_a(t), u_b(t)$ are given.

We use the P^1 finite element method for discretization in space and choose the backward Euler scheme in time. For any positive integer N , let $\delta t = T/N$ be the

time step, $t^n = n\delta t$, $n = 0, \dots, N$. We initiate the time-stepping loop from the initial solution $u_x^0 = \Pi_h^{L,1} u_0$. We proceed in the following way for $n = 1, \dots, N$, given $u_x^{n-1} \in P_x^1$:

- If no mesh adaptation is done during the time step, then we are looking for $u_x^n \in P_x^1$ such that

$$\frac{u_x^n - u_x^{n-1}}{\delta t} = \alpha^n \partial_{xx} u_x^n + f^n. \quad (3.2.2)$$

- In the case we use mesh adaptation to generate τ_y from τ_x , then we are looking for $u_y^n \in P_y^1$ such that

$$\frac{u_y^n - \Pi_h u_x^{n-1}}{\delta t} = \alpha^n \partial_{yy} u_y^n + f^n. \quad (3.2.3)$$

In (3.2.2)-(3.2.3), u_y^n is on the current mesh τ_y , u_x^{n-1} is on the background mesh τ_x , $\alpha^n = \alpha(t^n)$, and $f^n = f(x, t^n)$. Similarly, we can also consider a transfer step from τ_y to τ_x , in which case u_y^{n-1} is on the background mesh τ_y and u_x^n is on the current mesh τ_x . The variational formulation is obtained from equations (3.2.2) and (3.2.3) in the usual way.

3.2.1 Numerical example

Consider the following closed-form solution over the domain $\bar{\Omega} = [0, 10]$:

$$u_5(x, t) = 5\sin(\pi x/10)\exp(-t(\pi/10)^2), \quad x \in [0, 10], \quad t \in [0, T]. \quad (3.2.4)$$

We choose $\alpha(t) = 1$, $u_5(x, 0) = 5\sin(\pi x/10)$ and $u_5(0, t) = u_5(10, t) = 0$ in problem (3.2.1).

Consider two meshes τ_x and τ_y with constant mesh size $h_x = 0.25$ and $h_y = 0.5$, respectively. We define two overlapping meshes as introduced in equation (2.2.1) for $i = 0, \dots, 40$ and $j = 0, \dots, 20$. In the case of nonoverlapping meshes, we keep τ_y unchanged and introduce the finer mesh τ_x , with $h = 0.125$ as in equation (2.2.2) for $i = 1, \dots, 40$, where $x_0 = 0$ and $x_{41} = 10$. Further we take final time $T = 10$ and the time step $\delta t = 0.1$. Numerical results are reported for each transfer method discussed in chapter 2. If the initial mesh is τ_x , the initial function can be defined as $u_x^0 = \Pi_h^{L,1} u_5(x, 0)$. For our convenience, the source term f is omitted from equation (3.2.1), vanishes since u_5 is an exact solution of the heat equation.

Figure 3.6 compares each transfer method used to approximate the heat equation with the manufactured solution u_5 . In both overlapping and nonoverlapping mesh cases (see figure 3.6a [3.6b]), the best results are obtained when using the $L^2(\Omega)$ -projection, but the approximated function graph is above the exact solution. The worst results corresponding to the $L^2(\Omega)$ -projection with mass lumping. In both cases, the Lagrange interpolation and the AM transfer operator solutions produce nearly identical results.

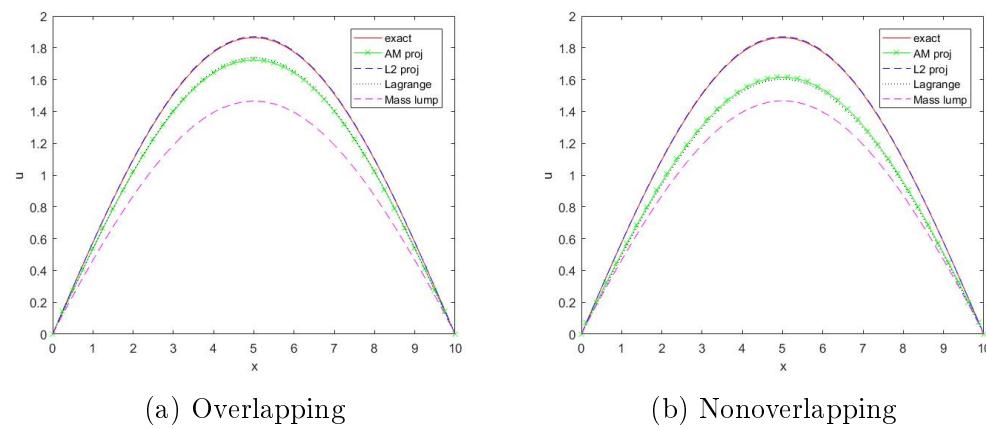


Figure 3.6: Numerical solutions of the heat equation for the manufactured solution u_5 using different meshes τ_x and τ_y with the final time $T = 10$, and various transfer operators.

Bibliography

- [1] F. Alauzet and M. Mehrenberger. P1 - conservative solution interpolation on unstructured triangular meshes. *Int. J. Numer. Methods Eng.*, 84:1552–1588, 2010.
- [2] J. K. Dukowicz and J. W. Kodis. Accurate conservative remapping (rezoning) for arbitrary Lagrangian-Eulerian computations. *SIAM Journal on Scientific and Statistical Computing*, 8:305–321, 1987.
- [3] A. Ern and J. L. Guermond. *Finite elements I*. Springer International Publishing, 2021.
- [4] P. E. Farrell. *Galerkin projection of discrete fields via supermesh construction*. Phdthesis, Imperial College London, 2009.
- [5] P. E. Farrell, M. D. Piggott, C. C. Pain, G. J. Gorman, and C. R. Wilson. Conservative interpolation between unstructured meshes via supermesh construction. *Comput. Methods Appl. Mech. Eng.*, 198:2632–2642, 2009.
- [6] C. W. Gear. Invariants and numerical methods for ODEs. *Phys. D Nonlinear Phenom.*, 60:303–310, 1992.
- [7] C. Geuzaine, B. Meys, F. Henrotte, P. Dular, and W. Legros. A Galerkin projection method for mixed finite elements. *IEEE Transactions on Magnetics*, 35:1438–1441, 1999.
- [8] P. Houston, S. Roggendorf, and K. G. van der Zee. Gibbs phenomena for L^q -best approximation in finite element spaces. *ESAIM: Mathematical Modelling and Numerical Analysis*, 56:177–211, 2022.
- [9] A. El Hraiech. Mechanical field interpolation. In *AIP Conference Proceedings*. AIP, 2005.
- [10] T. N. Krishnamurti. Numerical weather prediction. *Annu. Rev. Fluid Mech.*, 27:195–225, 1995.

- [11] A. Pont, R. Codina, and J. Baiges. Interpolation with restrictions between finite element meshes for flow problems in an ALE setting. *Int. J. Numer. Methods Eng.*, 110:1203–1226, 2016.
- [12] A. Quarteroni. *Numerical Models for Differential Problems*. Springer International Publishing, 2017.
- [13] M. Rüter, S. Korotov, and C. Steenbock. Goal-oriented error estimates based on different FE-spaces for the primal and the dual problem with applications to fracture mechanics. *Comput. Mech.*, 39:787–797, 2006.
- [14] J. Thuburn. Some conservation issues for the dynamical cores of NWP and climate models. *J. Comput. Phys.*, 227:3715–3730, 2008.
- [15] A. A. Ungar. *Barycentric calculus in euclidean and hyperbolic geometry*. World Scientific Publishing Company, 2010.
- [16] T. T. Warner. *Numerical weather and climate prediction*. Cambridge University Press, 2014.
- [17] Q. Zhang and S. Cen. *Multiphysics modeling: Numerical methods and engineering applications*. Academic press Inc., 2015.