Preconditioning the p-FEM Mass Matrix: Theory, Implementation, and Applications

by

Shuai Jiang

M.Sc., Brown University; Providence, RI, 2017

B.A., Cornell University; Ithaca, NY, 2015

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in The Division of Applied Mathematics at Brown University

PROVIDENCE, RHODE ISLAND

May 2020

This dissertation by Shuai Jiang is accepted in its present form
by The Division of Applied Mathematics as satisfying the
dissertation requirement for the degree of Doctor of Philosophy.

Date________________ ________________________________

Mark Ainsworth, Ph.D., Advisor

Recommended to the Graduate Council

Date________________ ________________________________

Chi-Wang Shu, Ph.D., Reader

Date________________ ________________________________

Marcus Sarkis, Ph.D., Reader

Approved by the Graduate Council

Date________________ ________________________________

Andrew G. Campbell, Dean of the Graduate School

Vita

Eduction

Brown University, Providence, RI, USA

M.S., Applied Mathematics, May 2017

Cornell University, Ithaca, NY, USA

B.A., Economics and Mathematics, *magna cum laude*, May 2015

Minor, Computer Science

Honors

National Defense Science and Engineering Graduate Fellow, 2015

Publications

Mark Ainsworth, Shuai Jiang, Manuel Sánchez. *An $\mathcal{O}(p^3)$ hp-version FEM in two dimensions: Preconditioning and post-processing*, Computer Methods in Applied Mechanics and Engineering, 350 (2019), pp. 766-802.

Mark Ainsworth, Shuai Jiang. *Preconditioning the mass matrix for high order finite element approximation on triangles*, SIAM J. Numer. Anal., 57 (2019), pp. 355-377.

Presentations

Uniform Substructuring Preconditioners for High Order FEM on Triangles and The Influence of Nodal Basis Functions, NAHOMCon, July 2019

Preconditioning Matrices Arising from High Order H^1 Conforming FEM on Triangles, SIAM CSE 2019, February 2019

Preconditioning the Mass Matrix for High Order Polynomial Approximation, ICOSAHOM 2018, July 2018

Preconditioning the Mass Matrix for High Order Polynomial Approximation, RPI Applied Math Days, April 2018

Teaching Experiences

Teaching Assistant, Applied Partial Differential Equations, Spring 2017

Teaching Assistant, Numerical Linear Algebra, Fall 2016

Dedication

To my parents, who taught me the importance of grit and perseverance.

Preface and Acknowledgments

I wast to give special thanks to Professor Mark Ainsworth for being a patient mentor and advisor for the past four years. The amount of times our meetings have stretched past the two hour mark really demonstrates how much he cares about both my development and work.

Besides my adviser, I would also want to thank the many faculty and staff in the department who made the whole process smoother. I would also like to express my deepest gratitude towards Professors Chi-Wang Shu and Marcus Sarkis for taking the time to be on my committee.

Finally, to my friends, both old and new, thank you for supporting me.

Contents

List of Tables

List of Figures

Chapter

ONE

Introduction

The finite element method (FEM) is an extremely successful approach to numerically approximating solutions to partial differential equations (PDEs) arising from many real-world scenarios [17]. Accuracy and convergence are generally achieved by refining the mesh (h-version), increasing the polynomial order on the individual elements (p-version), or a combination of the two (hp-version). Raising the polynomial order offers several advantages over pure h-version FEM; chief among them is exponential convergence to the true solution resulting in shorter time-to-solution [22, 67] even in cases involving singularities and boundary layers [32, 54].

The advantages of high-order methods come at a price of poor conditioning of the mass and stiffness matrices. Early endeavors into the construction of high order bases, such as Lagrangian and Peano bases, quickly fell out of favor due partly to the condition numbers of the resulting elemental matrices [73]. The current bases of choice are the hierarchical or Dubiner bases [22, 28, 47] whose mass (and stiffness) matrix has condition numbers which grow at $\mathcal{O}(p^{4(d-1)})$ or faster, where d is the dimension, as we increase the order [5, 52, 56]. Recall that the convergence of all iterative methods, such as conjugate gradient, depends on the condition number, hence an enormous amount of effort have been dedicated to constructing efficient preconditioners, but only for the *stiffness* matrix.

The domain decomposition preconditioner developed by Babuska et al. [12] was shown to reduce the growth of the condition number of the stiffness matrix to $\mathcal{O}(1 + \log^2 p)$ in two dimensions on both quads and triangles. Subsequent works extended these ideas to include preconditioners for the stiffness matrix in higher dimensions, hp-version finite element methods, boundary element methods, along with the use of more efficient approximate solvers on the subspaces [2,7,18,39,60]. Despite the rather extensive work on the analysis and construction of preconditioners for the *stiffness matrix*, virtually no attention has been paid to the question of preconditioning the *mass matrix*, especially on simplices.

The construction of efficient, domain decomposition type preconditioners for the p-version mass matrix is of practical interest, particularly when one turns to applications beyond Poisson-type problems, and this has not escaped the attention of the community completely. Early (unpublished) work of Smith [72] looked at preconditioners for the p-version mass matrix quadrilateral elements in two dimensions using tensor product type arguments. There has also been work generalizing mass lumping to high order elements, but they generally fall short in terms of robustness in p [45, 46].

Chapters 2 and 3 consider the problem of preconditioning the p-version mass matrix on meshes of triangular and tetrahedral elements (respectively). In both cases, a judicious choice of hierarchical basis allows one to construct a preconditioner involving only diagonal solves giving rise to a preconditioned system for which the condition number is bounded independently of the polynomial order p and the mesh size h. The analyses are performed in the framework of an Additive Schwarz Method (ASM) [19, 71, 77] and requires the construction of new polynomial extension theorems, similar to those that were derived in the analysis of the stiffness matrix in [12]. However, in the case of the mass matrix it is necessary to look at traces and extensions from the space L_2 (rather than H^1) and to make sense of the traces of polynomials regarded as functions in L_2.

With the development of the mass matrix preconditioners on both triangles and tetrahedra, the construction of a preconditioner on tensor product elements is a straightforward extension, due to the properties of the L^2 inner-product under a tensor product, which we pursue in Chapter 4. We are able to construct ASM preconditioners for any tensor product element, including quads, hexes and prisms.

In Chapter 5, we first explore the choices of different nodal basis functions in the context of the preconditioner presented in Chapter 2. We next exploit the fact that both the 2D mass matrix preconditioner and stiffness matrix preconditioner of Babuska et al. [12] were built under a similar framework; this allows us to propose an efficient preconditioner for any linear combination of mass and stiffness matrix on triangles.

Finally, we turn from theory to practice in Chapter 6 where we primarily discuss how to implement the 2D mass matrix preconditioner from Chapter 2 in an efficient manner. The proposed solution relies on Bernstein polynomials, a hallmark of computer-aided geometric design (CAGD) and splines among others [30, 31], as the basis for *hp*-FEM. As it turns out, using the Bernstein polynomials allows one to perform essentially *all* computations related to finite element analysis ranging from matrix-free matrix multiply [3], quantity-of-interest computation, visualization and preconditioning in $\mathcal{O}(p^3)$ in 2D.

Chapter

TWO

Triangles

2.1 Introduction

We start by considering the problem of preconditioning the p-version mass matrix on meshes of (possibly curvilinear) triangular elements in two dimensions. Through a judicious choice of hierarchical basis, *it is shown that a preconditioner involving only diagonal solves on the vertices, edges and element interiors gives rise to a preconditioned system for which the condition number is bounded independently of the polynomial order p and the mesh size h.*

The chapter is organized as follows. In section 2, we define the basis functions on a simplex. In section 3, we present the preconditioner, analyze its cost, and state the main theorem. In section 4, we present several illustrative numerical examples. In section 5, we use domain decomposition techniques to prove the key theorems. In section 6, we prove the technical lemmas and estimates required. We finish in section 7 with a conclusion.

2.2 Basis Functions

2.2.1 Basis functions on a triangle

Let T be the reference triangle in $\mathbb{R}^2$ with vertices $v_1 = (-1,-1), v_2 = (1,-1), v_3 = (-1,1)$, and the edges of T be denoted by γ_i for $i = 1,2,3$ such that γ_i is opposite of vertex v_i; see Figure 2.1. Let $p \geq 3$ be a given integer which is fixed throughout, and let $\mathbb{P}_p(T) = \text{span}\{x^\alpha y^\beta : 0 \leq \alpha, \beta, \alpha+\beta \leq p\}$ denote the space of polynomials of total degree p on T. Finally, for $i = 1,2,3$ we let $\lambda_i \in \mathbb{P}_1(T)$ be the barycentric

A version of this chapter has been previously published in [8].

coordinates on T, i.e. the unique polynomial such that $\lambda_i(v_j) = \delta_{ij}$.

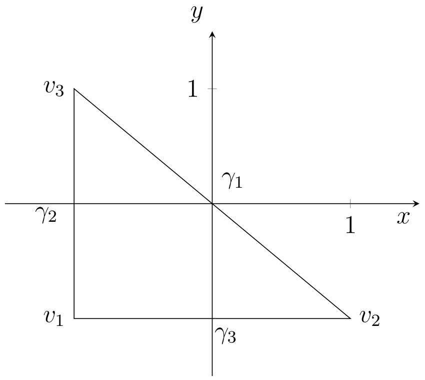


Figure 2.1: Figure of reference triangle T

The classical Jacobi polynomials on $[-1,1]$ are denoted by $P_n^{(\alpha,\beta)}$, where n is the order of the polynomial and $\alpha, \beta > -1$ are weights [1]. These will be used to define the basis functions on triangle T as follows:

Interior Basis Functions

The orthogonalized, interior modified principal functions [47] are given by

$$\psi_{ij}(x,y) = \frac{1-s}{2}\frac{1+s}{2}P_{i-1}^{(2,2)}(s)\left(\frac{1-t}{2}\right)^{i+1}\frac{1+t}{2}P_{j-1}^{(2i+3,2)}(t)$$

for $1 \leq i, j, i+j \leq p-1$, where

$$s = \frac{\lambda_2 - \lambda_1}{1-\lambda_3}, \quad t = 2\lambda_3 - 1$$

and $\lambda_1, \lambda_2, \lambda_3$ are the barycentric coordinates of $(x,y) \in T$. Note that $\{\psi_{ij}\}$ vanishes on the boundary of T and gives a basis for $\mathbb{P}_p(T) \cap H_0^1(T)$.

Edge Basis Functions

On edge γ_1, we define

$$\chi_n^{(1)}(x,y) = 4\lambda_2\lambda_3 P_n^{(2,2)}(\lambda_3 - \lambda_2)$$

for $n = 0, \ldots, p-2$ with $(x,y) \in T$. We note that the factor $\lambda_2\lambda_3$ means that $\chi_n^{(1)}$ vanishes on edges γ_2 and γ_3. The basis functions $\chi_n^{(2)}, \chi_n^{(3)}$ on edges γ_2, γ_3 are defined in an analogous fashion. The key property dictating this particular choice of basis is that $\chi_n^{(i)}|_{\gamma_i} = (1-s^2)P_n^{(2,2)}(s)$ where $s \in [-1,1]$ is a parametrization of γ_i.

Vertex Basis Functions

On vertex v_i for $i = 1,2,3$, we define

$$\varphi_i(x,y) = \frac{(-1)^{\lfloor p/2 \rfloor + 1}}{\lfloor p/2 \rfloor} \lambda_i P_{\lfloor p/2 \rfloor - 1}^{(1,1)}(1 - 2\lambda_i), \qquad (x,y) \in T.$$

Note that $\varphi_i(v_j) = \delta_{ij}$. One could replace $\lfloor p/2 \rfloor$ by p and still obtain a basis for $\mathbb{P}_p(T)$. The reason for choosing $\lfloor p/2 \rfloor$ rather than simply p will become clear later; a partial discussion presented in Lemma 2.6.3 and a full discussion on the choice of nodal basis function is available in chapter 5.

It is not difficult to verify that the functions defined above are linearly independent. Moreover, there are 3 dofs from the vertices, $3p-3$ dofs from the edges and $\frac{1}{2}\left(p^2 - 3p + 2\right)$ from the interior of T which sums to $\frac{1}{2}(p+1)(p+2) = \dim \mathbb{P}_p(T)$.

Hence, we have a basis for $\mathbb{P}_p(T)$ with the following decomposition:

$$\mathbb{P}_p(T) = \operatorname{span}\{\varphi_i\}_{i=1}^3 \oplus \bigoplus_{i=1}^{3} \operatorname{span}\{\chi_n^{(i)}\}_{n=0}^{p-2} \oplus \operatorname{span}\{\psi_{ij}\}_{1\le i,j,i+j\le p-1}. \tag{2.1}$$

This basis bears some similarities to existing [69] bases used for high order FEM and differs slightly in the choice of edge functions, but uses quite non-standard vertex functions. Our choice of vertex function is crucial in what follows. For instance, if the usual hat functions were to be used for the vertices, then the resulting preconditioner would result in a condition number which grows as $\mathcal{O}(p^2)$; see Figure 2.2. For a full discussion on the choice of the nodal basis functions, we refer the reader to Chapter 5.

We enumerate the basis functions in the following order:

1. the vertex functions $\{\varphi_i\}_{i=1}^3$,
2. the edge functions $\{\chi_n^{(1)}\}_{n=0}^{p-2}$, $\{\chi_n^{(2)}\}_{n=0}^{p-2}$, $\{\chi_n^{(3)}\}_{n=0}^{p-2}$
3. the remaining dofs correspond to $\{\psi_{ij}\}_{1\le i,j,i+j\le p-1}$,

then the mass matrix on T will have a block form

$$\hat{\mathbf{M}} = \begin{bmatrix} \hat{\mathbf{M}}_{VV} & \hat{\mathbf{M}}_{VE} & \hat{\mathbf{M}}_{VI} \\ \hat{\mathbf{M}}_{EV} & \hat{\mathbf{M}}_{EE} & \hat{\mathbf{M}}_{EI} \\ \hat{\mathbf{M}}_{IV} & \hat{\mathbf{M}}_{IE} & \hat{\mathbf{M}}_{II} \end{bmatrix}$$

where $\hat{\mathbf{M}}_{VV} = [\int_T \varphi_i\varphi_j\,dx]$ for $i,j = 1,2,3$ and $\hat{\mathbf{M}}_{VE} = [\int_T \varphi_i\chi_n^{(j)}\,dx]$ for $i,j = 1,2,3$ and $n = 0,\ldots,p-2$ etc. Likewise, the element load vector $\vec{f}$ and solution vector $\vec{x}$

take the partitioned forms

$$\vec{f} = \begin{bmatrix} \vec{f}_V \\ \vec{f}_E \\ \vec{f}_I \end{bmatrix}, \text{ and } \vec{x} = \begin{bmatrix} \vec{x}_V \\ \vec{x}_E \\ \vec{x}_I \end{bmatrix}.$$

The condition number of the mass matrix $\hat{\mathbf{M}}$ grows as $\mathcal{O}(p^4)$; Figure 2.2 shows the variation of the condition number versus p. If diagonal scaling is applied as a preconditioner for $\hat{\mathbf{M}}$, then the condition number now grows as $\mathcal{O}(p^2)$; Figure 2.2 also shows the condition number of the diagonally scaled mass matrix (denoted as $\hat{\mathbf{M}}_S$). Our objective is to construct a preconditioner for which the condition number remains bounded.

2.2.2 Basis functions on partitions

Let Ω be a bounded two-dimensional domain, and let $\mathcal{T}$ be a triangulation of Ω. We assume that each element $K \in \mathcal{T}$ is the image of the reference element T under a bijective map $\mathcal{F}_K$ (not necessarily linear) such that the Jacobian $D\mathcal{F}_K$ is bounded uniformly in the sense that there exists non-negative constants θ, Θ such that for all $K \in \mathcal{T}$ there holds

$$\theta|K| \leq |D\mathcal{F}_K| \leq \Theta|K|. \tag{2.2}$$

We remark that this condition places no constraints on the shape regularity of the mesh, and, in particular, allows for “needle” elements.

The basis functions on each element $K \in \mathcal{T}$ are defined in terms of the basis functions on the reference element in the usual way; for example, the first vertex basis functions is defined as

$$\varphi_{1,K}(x) := \varphi_1(\mathcal{F}_K^{-1}(x)).$$

Thanks to the decomposition of the basis into interior contributions and boundary contributions that are only supported on a single entity (i.e. edge or vertex), C^0 global conformity is enforced by matching the corresponding edge and vertex functions.

2.3 Preconditioner and Statement of Main Theorem

2.3.1 Preconditioning on the reference element

We begin by constructing a preconditioner for the mass matrix $\hat{\mathbf{M}}$ on the reference element T. Let $\mathbf{I}_3$ be the 3×3 identity matrix, $\hat{\mathbf{D}}_{VV} = \frac{1}{p^4}\mathbf{I}_3$ and

$$\hat{\mathbf{D}}_{EE} = \text{block diag}(\hat{\mathbf{D}}_{EE}^{(1)}, \hat{\mathbf{D}}_{EE}^{(2)}, \hat{\mathbf{D}}_{EE}^{(3)})$$

where $\hat{\mathbf{D}}_{EE}^{(i)}, i = 1, 2, 3$ is the diagonal matrix $\hat{\mathbf{D}}_{EE}^{(i)} = \text{diag}(q_j)$, with

$$\begin{aligned} q_j &:= \frac{2}{(p+4+j)(p-j+1)} \int_{-1}^{1} (1-x^2)^2 P_j^{(2,2)}(x)^2 \, dx \\ &= \frac{64(j+1)(j+2)}{(p+4+j)(p-j+1)(2j+5)(j+3)(j+4)} \end{aligned} \tag{2.3}$$

for $j = 0, \ldots, p-2$. We define our preconditioner, in the case of the reference element, in terms of its action when applied to a vector $\vec{f}$ in Algorithm 1.

Algorithm 1 Preconditioner on the Reference Element

```
Require: $\hat{\mathbf{M}}, \vec{f}$ as partitioned in Section 2.2
function
    $\vec{x}_I := \hat{\mathbf{M}}_{II}^{-1} \vec{f}_I$                                        ▷ Interior solve
    $\vec{x}_E := \hat{\mathbf{D}}_{EE}^{-1} \left(\vec{f}_E - \hat{\mathbf{M}}_{EI}\vec{x}_I\right)$          ▷ Edges solve
    $\vec{x}_V := \hat{\mathbf{D}}_{VV}^{-1} \left(\vec{f}_V - \hat{\mathbf{M}}_{VI}\vec{x}_I\right)$          ▷ Vertices solve
    $\vec{x}_I := \vec{x}_I - \hat{\mathbf{M}}_{II}^{-1}\hat{\mathbf{M}}_{IV}\vec{x}_V - \hat{\mathbf{M}}_{II}^{-1}\hat{\mathbf{M}}_{IE}\vec{x}_E$   ▷ Interior correction
    return $\vec{x} := \vec{x}_I + \vec{x}_E + \vec{x}_V$
end function
```

Direct manipulation reveals that Algorithm 1 defines a linear mapping $\vec{f} \to \vec{x} := \hat{\mathbf{P}}^{-1}\vec{f}$ where $\hat{\mathbf{P}}^{-1} = \hat{\mathbf{Q}}^{-T}\mathbf{D}^{-1}\hat{\mathbf{Q}}^{-1}$,

$$\hat{\mathbf{Q}} := \begin{bmatrix} \mathbf{I} & 0 & \hat{\mathbf{M}}_{VI}\hat{\mathbf{M}}_{II}^{-1} \\ 0 & \mathbf{I} & \hat{\mathbf{M}}_{EI}\hat{\mathbf{M}}_{II}^{-1} \\ 0 & 0 & \mathbf{I} \end{bmatrix}, \text{ and } \mathbf{D} := \begin{bmatrix} \hat{\mathbf{D}}_{VV} & 0 & 0 \\ 0 & \hat{\mathbf{D}}_{EE} & 0 \\ 0 & 0 & \hat{\mathbf{M}}_{II} \end{bmatrix}.$$

Clearly, $\hat{\mathbf{Q}}$ and $\mathbf{D}$ are invertible, hence

$$\hat{\mathbf{P}} = \hat{\mathbf{Q}}\mathbf{D}\hat{\mathbf{Q}}^T. \tag{2.4}$$

We now state a key result:

Theorem 2.3.1. *There exist positive constants $\hat{c}$ and $\hat{C}$ independent of p such that $\hat{c}\hat{\mathbf{P}} \leq \hat{\mathbf{M}} \leq \hat{C}\hat{\mathbf{P}}$.*[1] *Hence,*

$$\operatorname{cond}(\hat{\mathbf{P}}^{-1}\hat{\mathbf{M}}) \leq \frac{\hat{C}}{\hat{c}}.$$

[1] We use the notation that $\mathbf{A} \leq \mathbf{B}$ implies $\mathbf{B} - \mathbf{A}$ is semi-positive definite.

The proof of Theorem 2.3.1 is postponed to Section 2.5.

2.3.2 Preconditioning on a mesh

The global mass matrix $\mathbf{M}$ on a partition $\mathcal{T}$ is obtained by the standard finite element sub-assembly procedure

$$\mathbf{M} = \sum_{K \in \mathcal{T}} \mathbf{\Lambda}_K \mathbf{M}_K \mathbf{\Lambda}_K^T$$

where $\mathbf{M}_K$ is the element mass matrix, and $\mathbf{\Lambda}_K$ the local assembly matrix. For the global mass matrix, we assume the dofs are numbered in a similar fashion to the one used on a single element, viz.:

1. vertex basis dofs are (first in any order),
2. edge basis dofs grouped by the edge they are supported on, and ordered by the index on the Jacobi polynomial,
3. interior basis dofs grouped by the element on which they are supported.

Thanks to Equation (2.2), it follows that

$$c\frac{|K|}{|T|}\hat{\mathbf{M}} \leq \mathbf{M}_K \leq C\frac{|K|}{|T|}\hat{\mathbf{M}} \qquad \forall K \in \mathcal{T}$$

where the constants c and C depend only on θ and Θ. By the same token, we define a local preconditioner on K in terms of $\hat{\mathbf{P}}$

$$\mathbf{P}_K = \frac{|K|}{|T|}\hat{\mathbf{P}} = \frac{|K|}{|T|}\hat{\mathbf{Q}}\mathbf{D}\hat{\mathbf{Q}}^T \tag{2.5}$$

where the second equality follows from Equation (2.4). The global preconditioner $\mathbf{P}$ is then obtained using sub-assembly to give:

$$\mathbf{P} = \sum_{K\in\mathcal{T}} \boldsymbol{\Lambda}_K \mathbf{P}_K \boldsymbol{\Lambda}_K^T.$$

Let the local assembly matrix $\boldsymbol{\Lambda}_K$ be written in block form

$$\boldsymbol{\Lambda}_K = \begin{bmatrix} \boldsymbol{\Lambda}_{K,V} \\ \boldsymbol{\Lambda}_{K,E} \\ \boldsymbol{\Lambda}_{K,I} \end{bmatrix}$$

where the blocks correspond to the vertex, edge and interior basis functions on element K, and let

$$\mathbf{Q} = \begin{bmatrix} \mathbf{I} & 0 & \mathring{\mathbf{M}}_{VI}(\mathring{\mathbf{M}}_{II})^{-1} \\ 0 & \mathbf{I} & \mathring{\mathbf{M}}_{EI}(\mathring{\mathbf{M}}_{II})^{-1} \\ 0 & 0 & \mathbf{I} \end{bmatrix}$$

where $\mathring{\mathbf{M}}_{EI} = \sum_{K\in\mathcal{T}} \boldsymbol{\Lambda}_{K,E}\hat{\mathbf{M}}_{EI}\boldsymbol{\Lambda}_{K,I}^T$ with $\mathring{\mathbf{M}}_{II}, \mathring{\mathbf{M}}_{VI}$ defined analogously. Observe that if the physical elements K are all affine images of the reference element, then $\mathring{\mathbf{M}}_{II}, \mathring{\mathbf{M}}_{EI}$ will coincide with the global mass matrix blocks $\mathbf{M}_{II}, \mathbf{M}_{EI}$.

The following identity will prove useful in deducing the action of $\mathbf{P}^{-1}$:

Lemma 2.3.2. *For any element $K \in \mathcal{T}$, we have that*

$$\boldsymbol{\Lambda}_K \hat{\mathbf{Q}} = \mathbf{Q}\boldsymbol{\Lambda}_K. \tag{2.6}$$

Proof. It is clear that $\mathbf{\Lambda}_K\hat{\mathbf{Q}}\vec{f} = \mathbf{Q}\mathbf{\Lambda}_K\vec{f}$ if $\vec{f} = [\vec{f}_V; \vec{f}_E; \vec{0}]^2$ since, in that case,

$$\mathbf{\Lambda}_K\hat{\mathbf{Q}}[\vec{f}_V; \vec{f}_E; \vec{0}] = [\mathbf{\Lambda}_{K,V}\vec{f}_V; \mathbf{\Lambda}_{K,E}\vec{f}_E; \vec{0}] = \mathbf{Q}\mathbf{\Lambda}_K[\vec{f}_V; \vec{f}_E; \vec{0}].$$

It remains to show the relation holds for vectors of the form $[\vec{0}; \vec{0}; \vec{f}_I]$. Observe that the interior basis functions are supported on one and only one element. Hence $\mathring{\mathbf{M}}_{II}^{-1} = \sum_{K\in\mathcal{T}} \mathbf{\Lambda}_{K,I}\hat{\mathbf{M}}_{II}^{-1}\mathbf{\Lambda}_{K,I}^T$, and $\mathbf{\Lambda}_{K,I}^T\mathbf{\Lambda}_{K',I} = \delta_{KK'}\mathbf{I}$ for $K, K' \in \mathcal{T}$. Direct computation then shows,

$$\mathbf{Q}\mathbf{\Lambda}_K \begin{bmatrix} 0 \\ 0 \\ \vec{f}_I \end{bmatrix} = \begin{bmatrix} \mathring{\mathbf{M}}_{VI}\mathbf{\Lambda}_{K,I}\hat{\mathbf{M}}_{II}^{-1}\vec{f}_I \\ \mathring{\mathbf{M}}_{EI}\mathbf{\Lambda}_{K,I}\hat{\mathbf{M}}_{II}^{-1}\vec{f}_I \\ \mathbf{\Lambda}_{K,I}\vec{f}_I \end{bmatrix} = \begin{bmatrix} \mathbf{\Lambda}_{K,V}\hat{\mathbf{M}}_{VI}\hat{\mathbf{M}}_{II}^{-1}\vec{f}_I \\ \mathbf{\Lambda}_{K,E}\hat{\mathbf{M}}_{EI}\hat{\mathbf{M}}_{II}^{-1}\vec{f}_I \\ \mathbf{\Lambda}_{K,I}\vec{f}_I \end{bmatrix} = \mathbf{\Lambda}_K\hat{\mathbf{Q}} \begin{bmatrix} 0 \\ 0 \\ \vec{f}_I \end{bmatrix}.$$

□

In view of Lemma 2.3.2 and Equation (2.5), we can rewrite $\mathbf{P}$ in the form

$$\mathbf{P} = \mathbf{Q}\left(\sum_{K\in\mathcal{T}} \mathbf{\Lambda}_K \frac{|K|}{|T|}\mathbf{D}\mathbf{\Lambda}_K^T\right)\mathbf{Q}^T.$$

Moreover, since $\mathbf{D}$ is diagonal, we can rewrite

$$\sum_{K\in\mathcal{T}} \mathbf{\Lambda}_K \frac{|K|}{|T|}\mathbf{D}\mathbf{\Lambda}_K = \text{block diag}(\mathbf{D}_{VV}, \mathbf{D}_{EE}, \mathring{\mathbf{M}}_{II}).$$

where

$$\mathbf{D}_{VV} = \sum_{K\in\mathcal{T}} \frac{|K|}{|T|}\mathbf{\Lambda}_{K,V}\hat{\mathbf{D}}_{VV}\mathbf{\Lambda}_{K,V}^T \text{ and } \mathbf{D}_{EE} = \sum_{K\in\mathcal{T}} \frac{|K|}{|T|}\mathbf{\Lambda}_{K,E}\hat{\mathbf{D}}_{EE}\mathbf{\Lambda}_{K,E}^T.$$

In particular, note that both $\mathbf{D}_{VV}$ and $\mathbf{D}_{EE}$ are diagonal matrices. It follows that $\mathbf{P}$

[2] We use the notation where $[\vec{a}; \vec{b}; \vec{c}]$ denotes the column vector $[\vec{a}^T, \vec{b}^T, \vec{c}^T]^T$.

is invertible, and the action of $\mathbf{P}^{-1}$ on a global right hand side is given by Algorithm 2. A key property of Algorithm 2 is that the global preconditioner requires only diagonal solves over the edges, interior and vertices.

Algorithm 2 Preconditioner for Global Mass Matrix

```
Require: $\mathbf{M}$ global mass matrix, $\vec{f}$ residual vector
function
    $\vec{x}_I := \mathring{\mathbf{M}}_{II}^{-1}\vec{f}_I$
    $\vec{x}_E := \mathbf{D}_{EE}^{-1}\left(\vec{f}_E - \mathring{\mathbf{M}}_{EI}\vec{x}_I\right)$
    $\vec{x}_V := \mathbf{D}_{VV}^{-1}\left(\vec{f}_V - \mathring{\mathbf{M}}_{VI}\vec{x}_I\right)$
    $\vec{x}_I := \vec{x}_I - \mathring{\mathbf{M}}_{II}^{-1}\mathring{\mathbf{M}}_{IV}\vec{x}_V - \mathring{\mathbf{M}}_{II}^{-1}\mathring{\mathbf{M}}_{IE}\vec{x}_E$
    return $\vec{x} := \vec{x}_I + \vec{x}_E + \vec{x}_V$
end function
```

The next result complements Theorem 2.3.1 by showing that $\mathbf{P}$ is a uniform preconditioner for the mass matrix on the entire mesh $\mathcal{T}$:

Corollary 2.3.3. *There exists a constant C independent of h, p such that*

$$\operatorname{cond}(\mathbf{P}^{-1}\mathbf{M}) \leq C.$$

Proof. Bounds Equation (2.2) and a change of variables show that $\theta\hat{\mathbf{M}} \leq \mathbf{M}_K \leq \Theta\hat{\mathbf{M}}$. Then by standard sub-assembly and Theorem 2.3.1

$$\hat{c}\theta\mathbf{P} = \hat{c}\theta \sum_{K\in\mathcal{T}} \mathbf{\Lambda}_K \mathbf{P}_K \mathbf{\Lambda}_K^T \leq \sum_{K\in\mathcal{T}} \mathbf{\Lambda}_K \mathbf{M}_K \mathbf{\Lambda}_K^T = \mathbf{M} \leq \hat{C}\Theta \sum_{K\in\mathcal{T}} \mathbf{\Lambda}_K \mathbf{P}_K \mathbf{\Lambda}_K^T = \hat{C}\Theta\mathbf{P}$$

where $\hat{c}, \hat{C}$ are the constants from Theorem 2.3.1. Hence $\operatorname{cond}(\mathbf{P}^{-1}\mathbf{M}) \leq \frac{\hat{C}\Theta}{\hat{c}\theta}$. □

2.3.3 Cost of Applying the Preconditioner

Line 2 to line 4 of Algorithm 2 all involve inversion of diagonal matrices. Consequently, each interior block can be inverted at a cost of $\frac{1}{2}(p-1)(p-2)$ operations, each edge block at a cost of $p-1$ operations, and the vertex block costs $3|\mathcal{V}|$ operations where $|\mathcal{V}|$ is the number of vertices in mesh $\mathcal{T}$. The dominant cost of the algorithm lies in the matrix-vector multiplication $\mathbf{M}_{EI}^{\text{pre}}\vec{x}_I$, which costs $\mathcal{O}(p^3)$ operations, hence the overall cost of our algorithm is $\mathcal{O}(p^3)$.

2.4 Numerical Examples

In this section, we present results obtained by applying Algorithm 2 to solve linear algebraic systems arising in some representational examples.

2.4.1 Condition number on reference triangle

We start by illustrating the performance of the preconditioner on the reference element (see Theorem 2.3.1). In Figure 2.2, we plot the condition number of $\hat{\mathbf{M}}$, the condition number of the diagonally scaled mass matrix $\hat{\mathbf{M}}_S$ where

$$\hat{\mathbf{M}}_S = \text{diag}(\hat{\mathbf{M}})^{-1/2}\hat{\mathbf{M}}\ \text{diag}(\hat{\mathbf{M}})^{-1/2},$$

and the condition number of the preconditioned mass matrix $\hat{\mathbf{P}}^{-1/2}\hat{\mathbf{M}}\hat{\mathbf{P}}^{-1/2}$.

Figure 2.2 also shows the results obtained if the vertex functions in the choice of basis is replaced by the "full-order" vertex basis functions

$$\ddot{\varphi}_i(x,y) = \frac{(-1)^{p+1}}{p}\lambda_i P_{p-1}^{(1,1)}(1-2\lambda_i), \qquad (x,y) \in T$$

to partially illustrate why the choice $\lfloor p/2 \rfloor$ was made. We will call the preconditioned mass matrix constructed using $\ddot{\varphi}_i$ as $\hat{\mathbf{P}}^{-1/2}\ddot{\mathbf{M}}\hat{\mathbf{P}}^{-1/2}$ It is observed that the condition number no longer remains bounded; see Theorem 5.3.1 for an explanation. Finally, the figure also shows the results obtained if the vertex functions were replaced with the commonly used hat functions

$$\breve{\varphi}_i(x,y) = \lambda_i, \qquad (x,y) \in T.$$

We call the preconditioned mass matrix constructed using $\breve{\varphi}_i$ as $\hat{\mathbf{P}}_L^{-1/2}\breve{\mathbf{M}}\hat{\mathbf{P}}_L^{-1/2}$; the only difference between $\hat{\mathbf{P}}$ and $\hat{\mathbf{P}}_L$ is a more appropriate scaling for $\hat{\mathbf{D}}_{VV}$. Figure 2.2 shows the growth of the condition number is of order $\mathcal{O}(p^2)$.

We note that the mass matrix $\hat{\mathbf{M}}$ and the scaled mass matrix $\hat{\mathbf{M}}_S$ both exhibit algebraic growth with the order p which is typically the case for such basis [5], while, by contrast, the preconditioned system $\hat{\mathbf{P}}^{-1/2}\hat{\mathbf{M}}\hat{\mathbf{P}}^{-1/2}$ remains constant with p as predicted by Theorem 2.3.1 (with an asymptotic value of 24 as $p \to \infty$) .

2.4.2 Condition number on multi-element mesh

We next illustrate Corollary 2.3.3 by considering the mesh shown in Figure 2.3 which consists of 239852 affine elements. We construct the global mass matrix $\mathbf{M}$ explicitly and use ARPACK to approximate the extreme eigenvalues of the preconditioned system to a relative tolerance of 10^{-4}. In Table 2.1, we display the extreme eigenvalues

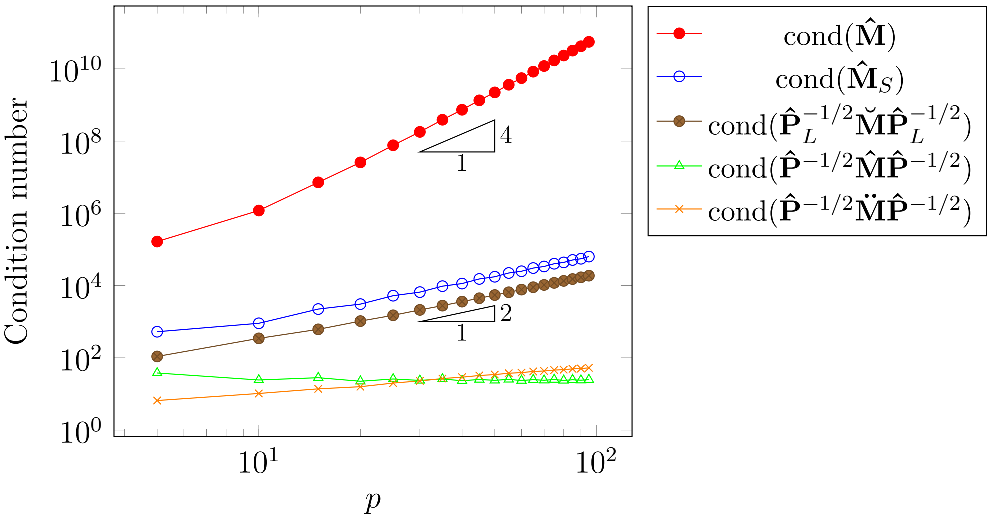


Figure 2.2: The condition numbers of $\hat{\mathbf{M}}, \hat{\mathbf{M}}_S$, cond($\hat{\mathbf{P}}_L^{-1/2}\breve{\mathbf{M}}\hat{\mathbf{P}}_L^{-1/2}$), $\hat{\mathbf{P}}^{-1/2}\hat{\mathbf{M}}\hat{\mathbf{P}}^{-1/2}$ and $\hat{\mathbf{P}}^{-1/2}\ddot{\mathbf{M}}\hat{\mathbf{P}}^{-1/2}$ are plotted on a log-log axis for $p = 5, 10, \ldots, 95$. The algebraic growth of cond($\hat{\mathbf{M}}$) and cond($\hat{\mathbf{M}}_S$) with p are consistent with [5], and the boundedness of cond($\hat{\mathbf{P}}^{-1/2}\hat{\mathbf{M}}\hat{\mathbf{P}}^{-1/2}$) is predicted in Theorem 2.3.1. Finally, we note the importance of our choice of vertex function: the "full-order" vertex basis system cond($\hat{\mathbf{P}}^{-1/2}\ddot{\mathbf{M}}\hat{\mathbf{P}}^{-1/2}$) grows and the hat functions systems cond($\hat{\mathbf{P}}_L^{-1/2}\breve{\mathbf{M}}\hat{\mathbf{P}}_L^{-1/2}$) exhibits $\mathcal{O}(p^2)$ growth.

and condition number of the preconditioned mass matrix on the multi-element mesh, along with the corresponding quantities for the preconditioned mass matrix on the reference element. The condition numbers on the multi-element mesh are bounded by those on the reference element as predicted by Corollary 2.3.3 for affine elements.

Table 2.1: Table to illustrate Corollary 2.3.3 by comparing the extreme eigenvalues of the global mass matrix $\mathbf{M}$ of the mesh as shown in Figure 2.3, to the single element case $\hat{\mathbf{M}}$. The eigenvalues are approximated using ARPACK to a relative tolerance of 10^{-4} for $\mathbf{M}$ and to machine precision for $\hat{\mathbf{M}}$.

		Multi-Element Mesh $\mathbf{M}$			Single Element $\hat{\mathbf{M}}$		
p	#DOF	$\lambda_{\min}$	$\lambda_{\max}$	$\lambda_{\max}/\lambda_{\min}$	$\lambda_{\min}$	$\lambda_{\max}$	$\lambda_{\max}/\lambda_{\min}$
3	1084371	0.0518	2.6077	50.341	0.0518	2.6124	50.386
4	1925541	0.0922	2.3033	24.982	0.0920	2.3064	25.061
5	3006563	0.0793	2.9154	36.764	0.0791	2.9198	36.887

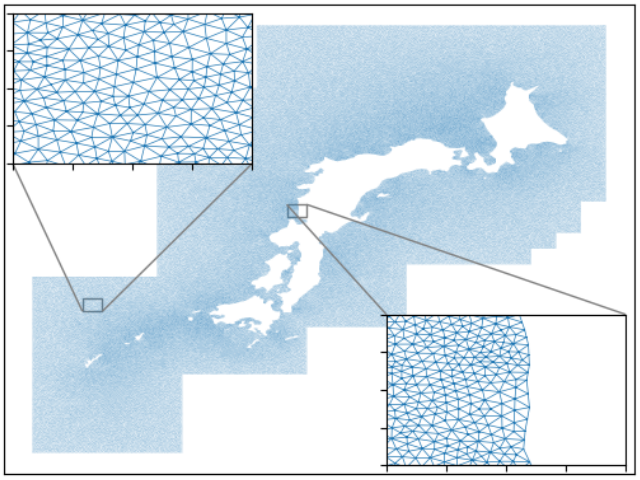


Figure 2.3: Plot of the mesh used to illustrate Corollary 2.3.3; see Table 2.1 for the results.

2.4.3 Explicit time-stepping

We now illustrate the use of the preconditioner in the numerical solution of the wave-equation where the time stepping scheme requires the inversion of the mass matrix at each step. Let $u(x, y, t)$ be defined in $\Omega = [-7, 7] \times [-7, 7]$ be the solution to the wave equation

$$u_{tt} = \Delta u, \qquad (x, y) \in \Omega, t > 0$$

with Neumann boundary condition; the initial condition [16] is

$$u(x, y, 0) = 4\tan^{-1}\exp(x + 1 - 2\,\mathrm{sech}(y + 7) - 2\,\mathrm{sech}(y - 7)), \qquad u_t(x, y, 0) = 0.$$

For the spatial discretization, we use a uniform triangulation of the square. For the time discretization, we use a 4th order Nyström method [40, p. 285], which entails three mass matrix solves per time step; for example, the first substep consists of solving

$$\vec{u}_1^{n+1} := \mathbf{M}^{-1}\left(-\mathbf{S}\vec{u}^n\right)$$

where $\mathbf{S}$ is the stiffness matrix. For each solve, we use the preconditioned conjugate gradient (PCG) with an appropriate initial guess; recall that the error $\vec{e}_k$ at iteration k of preconditioned conjugate gradient satisfies

$$\left\|\vec{e}_k\right\| \leq \left(\frac{\sqrt{\kappa-1}}{\sqrt{\kappa+1}}\right)^k \left\|\vec{e}_0\right\|. \tag{2.7}$$

where κ is is the condition number of the preconditioned matrix and $\vec{e}_0$ is the error of the initial iterate [35, p. 636]. In Table 2.2, we show the minimum, median and max iteration count of PCG over the entire simulation of 10 seconds with $\Delta t = 0.01$.

Corollary 2.3.3 and Equation (2.7) guarantees that the iteration count will not increase with p or with h refinement. In fact, we note that the median iteration count actually *decreases* as we increase p and refine h. This is due to Equation (2.7) being an estimate which only relates the condition number to the error bound, but does not take into account the possible improvements from clustering of eigenvalues. Furthermore, the estimate does not take into account a good initial iterate, which improves as we increase the number of dofs.

2.4.4 Implicit time-stepping

Finally, we illustrate the use of the preconditioner in the solution of the heat equation where the time-stepping scheme requires the inversion of a perturbed mass matrix at each step. Let $u(x, y, t)$ be defined in $\Omega = [-1, 1] \times [-1, 1]$ be the solution to the

Table 2.2: Table illustrates the performance of the preconditioned iterative method of the mass matrix at each time step by displaying the [min, median, max] iteration count of all 3000 PCG solves from using the Nyström method for a period of 10 seconds with a $\Delta t = .01$ on $u_{tt} = \Delta u$ in a uniformly triangulated square. The iteration count does not increase as predicted in Corollary 2.3.3 and Equation (2.7).

Order	16 Elements	64 Elements	256 Elements
4	[21, 27, 34]	[20, 25, 34]	[17, 23, 31]
8	[17, 23, 29]	[16, 21, 30]	[16, 21, 26]
12	[17, 22, 27]	[16, 18, 26]	[16, 17, 25]
16	[16, 18, 25]	[15, 18, 24]	[15, 15, 23]
20	[16, 18, 24]	[15, 15, 23]	

heat equation

$$u_t = \Delta u, \qquad (x, y) \in \Omega, t > 0$$

with Neumann boundary condition; we use a simple initial condition

$$u(x, y, 0) = \exp(-(x^2 + y^2)).$$

The time stepping scheme we use is the Crank-Nicolson method:

$$\left(\mathbf{M} + \frac{\Delta t}{2}\mathbf{S}\right)\vec{u}^{n+1} = \left(\mathbf{M} - \frac{\Delta t}{2}\mathbf{S}\right)\vec{u}^{n}$$

where $\mathbf{S}$ is the stiffness matrix. By Schmidt's inequality [43], there exists a c independent of p, h such that

$$0 \leq \mathbf{S} \leq c\frac{p^4}{h^2}\mathbf{M} \implies \mathbf{M} \leq \mathbf{M} + \frac{\Delta t}{2}\mathbf{S} \leq \left(1 + \frac{1}{2}c\Delta t\frac{p^4}{h^2}\right)\mathbf{M}. \tag{2.8}$$

The preconditioned system will have condition number of

$$\operatorname{cond}\left(\mathbf{P}^{-1}\left(\mathbf{M} + \frac{1}{2}\Delta t\mathbf{S}\right)\right) = \mathcal{O}\left(1 + \Delta t\frac{p^4}{h^2}\right). \tag{2.9}$$

Observe that if we were to use a fully explicit scheme, then the CFL condition is $\Delta t \sim \frac{h^2}{p^4}$ thanks again to Schmidt's inequality being sharp. If we use the choice $\Delta t \sim \frac{h^2}{p^4}$ for the implicit scheme, then Equation (2.9) shows that the iteration count will not increase as we increase p. In practice however, one generally chooses $\Delta t \sim \frac{h^2}{p^2}$ in which case Equation (2.9) shows that the condition number will grow at a rate of at most $\mathcal{O}(p^2)$; hence the iteration count will also increase. These conclusions are illustrated in Table 2.3. In the first two columns, we start with an initial iterate of $\vec{0}$ in each PCG method. In the other two columns, we use the solution from the previous time step as the initial iterate, which results in drastic decreases in iteration counts.

We conjecture Equation (2.9) could be improved to $\mathcal{O}(1 + \log^2 p)$ in Chapter 5 but it would require a significant increase in computational cost.

Table 2.3: Table to illustrate the performance of the preconditioned iterative method to the matrix resulting from Crank-Nicolson scheme by displaying the [min, median, max] iteration count of all PCG solves from using Crank-Nicolson for a period of 1 seconds on 16 elements for $u_t = \Delta u$ in a uniformly triangulated square. For the latter two columns, the initial guess is the previous time-step. The behaviors as we increase p is predicted by Equation (2.9).

	Initial Iterate: $\vec{0}$		Initial Iterate: $\vec{u}^n$	
p	$\Delta t \sim \frac{h^2}{p^4}$	$\Delta t \sim \frac{h^2}{p^2}$	$\Delta t \sim \frac{h^2}{p^4}$	$\Delta t \sim \frac{h^2}{p^2}$
4	[35, 36, 37]	[35, 36, 37]	[34, 34, 36]	[34, 34, 36]
8	[38, 39, 39]	[66, 67, 73]	[9, 17, 35]	[49, 51, 73]
12	[34, 35, 35]	[87, 91, 103]	[4, 8, 29]	[51, 55, 101]
16	[32, 33, 33]	[108, 114, 127]	[2, 7, 24]	[48, 55, 124]
20	[16, 19, 19]	[129, 130, 151]	[1, 1, 9]	[47, 55, 149]

2.5 Additive Schwarz Theory

Thanks to Corollary 2.3.3, the analysis of the preconditioner reduces to bounding the condition number on the reference element as in Theorem 2.3.1. Consequently, for the remainder of this article we confine our attention to the reference triangle.

Let $X := \mathbb{P}_p(T)$ be equipped with the standard L^2 inner-product denoted by $(\cdot,\cdot)$ with the respective norm denoted by $\|\cdot\|$, and let $X_I := H_0^1(T) \cap \mathbb{P}_p(T)$ be the interior space equipped with the $L^2(T)$ inner-product. The orthogonal complement of the (closed) subspace X_I in X is denoted by $\widetilde{X}_B$, i.e.

$$X = X_I \oplus \widetilde{X}_B, \qquad X_I \perp \widetilde{X}_B. \tag{2.10}$$

We begin by exploring the structure of the space $\widetilde{X}_B$. Let $\mathbb{P}_p(\partial T)$ denote the space of traces of $\mathbb{P}_p(T)$ on the boundary ∂T of the reference triangle:

$$\mathbb{P}_p(\partial T) = \{u : u = v|_{\partial T} \text{ for some } v \in \mathbb{P}_p(T)\}. \tag{2.11}$$

The next result shows that there is a one-to-one correspondence between $\widetilde{X}_B$ and $\mathbb{P}_p(\partial T)$.

Lemma 2.5.1. *For every $u \in \mathbb{P}_p(\partial T)$, there exists a unique $\widetilde{u} \in \widetilde{X}_B$ which satisfies $\widetilde{u} = u$ on ∂T, and $(\widetilde{u}, v) = 0$ for all $v \in X_I$. Furthermore, $\widetilde{u}$ is a minimal L^2 extension of u in the sense that for all $w \in \mathbb{P}_p(T)$ with $w|_{\partial T} = u$ we have $\|\widetilde{u}\| \leq \|w\|$.*

Proof. Let $u \in \mathbb{P}_p(\partial T)$ be given. According to Equation (2.11), u is equal to the trace of a polynomial in $\mathbb{P}_p(T)$, which we again denote by u. We can construct a $\widetilde{u} \in \widetilde{X}_B$ with the claimed properties as follows.

Let

$$u_I \in X_I : (u_I, v_I) = -(u, v_I) \qquad \forall v_I \in X_I.$$

Set $\widetilde{u} = u + u_I$; clearly $\widetilde{u}|_{\partial T} = u$ and $(\widetilde{u}, v_I) = 0$ for all $v_I \in X_I$; this gives existence. For uniqueness, let $\tilde{w} \in \mathbb{P}_p(T) : \tilde{w}|_{\partial T} = u, (\widetilde{w}, v_I) = 0$ for all $v_I \in X_I$, then

$$(\tilde{u} - \tilde{w}, v_I) = 0 \qquad \forall v_I \in X_I.$$

Hence $\tilde{u} - \tilde{w} = 0$ as $\tilde{u} - \tilde{w} \in X_I$. The minimal L^2 extension property follows from the Pythagorean identity. □

We say that $\tilde{u}$ is the "minimal L^2 extension" or "minimal extension" of $u \in \mathbb{P}_p(\partial T)$. Lemma 2.5.1 shows that $\tilde{u}$ is uniquely determined by the boundary values of u and the degree of the space.

We decompose the space $\widetilde{X}_B$ further. Let $\widetilde{\varphi}_i$ and $\widetilde{\chi}_n^{(i)}$ be the minimal extension, constructed as described in Lemma 2.5.1, of the vertex basis function and edge basis function defined in Section 2.2 respectively. Let

$$\widetilde{X}_V = \operatorname{span}\{\widetilde{\varphi}_i : i = 1, 2, 3\}$$

and

$$\widetilde{X}_{E_i} = \operatorname{span}\{\widetilde{\chi}_n^{(i)} : n = 0, \ldots, p - 2\}, \qquad i = 1, 2, 3.$$

By the construction of the basis functions on the boundary and, thanks to Equation (2.1) and Equation (2.10), we have

$$X = X_I \oplus \widetilde{X}_V \oplus \bigoplus_{i=1}^{3} \widetilde{X}_{E_i}. \tag{2.12}$$

Let $\vec{\varphi} = [\varphi_1; \varphi_2; \varphi_3]$ where φ_i are the vertex basis functions with $\vec{\psi}$ defined similarly for the interior basis functions, and, using the notation of Section 2.2, define

$$\vec{\tilde{\varphi}} = \vec{\varphi} - \hat{\mathbf{M}}_{VI}\hat{\mathbf{M}}_{II}^{-1}\vec{\psi}. \tag{2.13}$$

Then for $\vec{u} \in \mathbb{R}^3$, we have for all $X_I \ni w = \vec{w}^T\vec{\psi}$,

$$\begin{aligned}(\vec{u}^T\vec{\tilde{\varphi}}, w) = \left(\vec{u}^T\vec{\tilde{\varphi}}, \vec{w}^T\vec{\psi}\right) &= \left(\vec{u}^T(\vec{\varphi} - \hat{\mathbf{M}}_{VI}\hat{\mathbf{M}}_{II}^{-1}\vec{\psi}), \vec{w}^T\vec{\psi}\right) \\ &= \vec{u}^T\hat{\mathbf{M}}_{VI}\vec{w} - \vec{u}^T\hat{\mathbf{M}}_{VI}\hat{\mathbf{M}}_{II}^{-1}\hat{\mathbf{M}}_{II}\vec{w} = 0.\end{aligned}$$

Hence $\{\tilde{\varphi}_1, \tilde{\varphi}_2, \tilde{\varphi}_3\} \in \widetilde{X}_B$, and as a consequence forms a basis for $\widetilde{X}_V$ (since $\tilde{\varphi}_i|_{\partial T} = \varphi_i|_{\partial T}$). A basis for $\widetilde{X}_{E_i}$ with $i = 1, 2, 3$ can be constructed in the same fashion.

Next, we define the bilinear forms on each subspace in the decomposition (2.12):

- Interior space X_I:

$$a_I(u, w) := (u, w), \qquad u, w \in X_I.$$

- Vertex space $\widetilde{X}_V$:

$$a_V(u, w) := \frac{1}{p^4}\sum_{i=1}^{3} u(v_i)w(v_i), \qquad u, w \in \widetilde{X}_V$$

  where v_1, v_2, v_3 are the vertices of T.

- Edge spaces $\tilde{X}_{E_i}$ $(i = 1, 2, 3)$:

$$a_{E_i}(u, w) := \sum_{n=0}^{p-2} q_n\mu_n(u)\mu_n(w), \qquad u, w \in \widetilde{X}_{E_i}$$

with q_n defined as in Equation (2.3), and μ_n is the weighted moment given by

$$\mu_n(u) := \frac{(2n+5)(n+3)(n+4)}{32(n+1)(n+2)} \int_{-1}^{1} \chi_n^{(i)}(x) u(x)\, dx$$

where we use a linear parametrization such that $\gamma_i = [-1, 1]$.

The spaces and inner-products defined above give rise to an Additive Schwarz Method (ASM) preconditioner [19, 71, 77] whose action on a given residual $f \in X$ is defined as:

(i) $u_I \in X_I : a_I(u_I, v_I) = (f, v_I) \quad \forall v_I \in X_I$.

(ii) $u_V \in \widetilde{X}_V : a_V(u_V, v_V) = (f, v_V) \quad \forall v_V \in \widetilde{X}_V$.

(iii) For $i = 1, 2, 3$, $u_{E_i} \in \widetilde{X}_{E_i} : a_{E_i}(u_{E_i}, v_{E_i}) = (f, v_{E_i}) \quad \forall v_{E_i} \in \widetilde{X}_{E_i}$.

(iv) $u := u_I + u_V + \sum_{i=1}^{3} u_{E_i}$ is our solution.

2.5.1 Matrix Formulation of the ASM

In practice, it is convenient to reformulate steps (i)-(iv) in terms of matrix operations.

1) Recall that $X_I = \operatorname{span}\{\psi_{ij}\}$ and let $u_I = \vec{u}_I^T \vec{\psi}$ where $\vec{\psi}$ is the column vector of all the interior basis functions. The matrix form of (i) is

$$\hat{\mathbf{M}}_{II} \vec{u}_I = a_I(u_I, \vec{\psi}) = (f, \vec{\psi}) = \vec{f}_I.$$

2) Let $u_V = \vec{u}_V^T \vec{\widetilde{\varphi}}$ where $\vec{\widetilde{\varphi}}$ is the basis for $\widetilde{X}_V$ in column form. As $\widetilde{\varphi}_i(v_j) = \delta_{ij}$, we

have

$$\frac{1}{p^4}\mathbf{I}_{VV}\vec{u}_V = a_V(u_V, \vec{\tilde{\varphi}}) = (f, \vec{\tilde{\varphi}}).$$

Inserting identity Equation (2.13) in the right hand side gives

$$\begin{aligned}(f, \vec{\tilde{\varphi}}) &= (f, \vec{\varphi}) - \mathbf{M}_{VI}\mathbf{M}_{II}^{-1}(f, \vec{\psi}) \\ &= \vec{f}_V - \hat{\mathbf{M}}_{VI}\hat{\mathbf{M}}_{II}^{-1}\vec{f}_I.\end{aligned}$$

3) Let $u_{E_1} = \vec{u}_{E_1}^T \vec{\tilde{\chi}}$ where $\vec{\tilde{\chi}}$ is the basis for $\widetilde{X}_{E_1}$ in column form. By the orthogonality properties of $P_i^{(2,2)}(x)$ in Equation (2.3), the weighted moments in $a_V(\cdot,\cdot)$ of (iii) simplifies to $\mu_n(\widetilde{\chi}_i)\mu_n(\widetilde{\chi}_j) = \delta_{ij}$, and hence we have

$$\hat{\mathbf{D}}_{EE}^{(1)}\vec{u}_{E_1} = a_{E_1}(u_{E_1}, \vec{\tilde{\chi}}) = (f, \vec{\tilde{\chi}}).$$

The same reasoning holds for edges γ_2, γ_3. The right-hand side modification follows from 2).

4) The vector solution $\vec{x}_V$ to step (ii) corresponds to the function $\tilde{u}_V := \vec{x}_V^T \vec{\tilde{\varphi}}$. Applying identity Equation (2.13) again, we have

$$\tilde{u}_V = \vec{x}_V^T \left(\vec{\varphi} - \hat{\mathbf{M}}_{VI}\hat{\mathbf{M}}_{II}^{-1}\vec{\psi}\right).$$

Therefore, our minimal energy solution contains interior functions of the form $-\hat{\mathbf{M}}_{II}^{-1}\hat{\mathbf{M}}_{IV}\vec{x}_V$ which we have to add back to $\vec{x}_I$. A similar correction term is needed for the three edge terms.

Theorem 2.5.2. *The abstract Additive Schwarz Method defined above corresponds to Algorithm 1.*

Proof. Steps 1), 2), 3), 4) above corresponds to line 2, line 4, line 3 and line 5 respectively from Algorithm 1. □

2.5.2 Proof of Theorem 3.1

We apply the standard theory [19,71,77] for the analysis of additive Schwarz methods to the scenario as described above. In particular, we will follow the framework as laid out in [77, §2].

Lemma 2.5.3 (Local Stability). *For a constant C independent of p, each of our local bilinear forms are coercive in the sense that*

$$\begin{aligned}
(u,u) &= a_I(u,u) && \forall u \in X_I,\\
(u,u) &= a_{E_i}(u,u) && \forall u \in \widetilde{X}_{E_i}, i = 1,2,3,\\
(u,u) &\le 3Ca_V(u,u) && \forall u \in \widetilde{X}_V.
\end{aligned}$$

Proof. The first equality holds as X_I is a subspace of X and inherits the inner-product. For $\widetilde{X}_{E_i}$, identity Equation (2.16) of Lemma 2.6.4 gives us the equality

$$a_{E_i}(u,u) = \sum_{n=0}^{p-2} q_n \mu_n(u)^2 = \|u\|^2 .$$

Finally, for $u \in \widetilde{X}_V$, we rewrite $u = \sum_{i=1}^3 u(v_i)\widetilde{\varphi}_i$. Using the triangle inequality and the estimate $\|\widetilde{\varphi}_i\|^2 \le Cp^{-4}$ of Lemma 2.6.3, we have

$$\|u\|^2 \le 3\sum_{i=1}^{3} \|u(v_i)\widetilde{\varphi}_i\|^2 \le \frac{3C}{p^4}\sum_{i=1}^{3} |u(v_i)|^2 = 3Ca_V(u,u).$$

□

The next result gives an estimate for the largest eigenvalue, and is an immediate consequence of the triangle inequality and Lemma 2.5.3:

Lemma 2.5.4. *There exists a constant C independent of p such that for all $u \in X$, the unique decomposition*

$$u = u_I + u_V + \sum_{i=1}^{3} u_{E_i},$$

with $u_I \in X_I, u_V \in \widetilde{X}_V, u_{E_i} \in \widetilde{X}_{E_i}$, satisfies

$$\|u\|^2 \leq C \left(a_I(u_I, u_I) + a_V(u_V, u_V) + \sum_{i=1}^{3} a_{E_i}(u_{E_i}, u_{E_i}) \right).$$

The final ingredient is the following bound for the smallest eigenvalue of the additive Schwarz operator, whose proof is the subject of Section 2.6:

Theorem 2.5.5 (Stable Decomposition)**.** *For all $u \in X$, with the decomposition as in Lemma 2.5.4, there exists a constant C independent of p such that*

$$a_I(u_I, u_I) + a_V(u_V, u_V) + \sum_{i=1}^{3} a_{E_i}(u_{E_i}, u_{E_i}) \leq C\|u\|^2.$$

The proof of Theorem 2.3.1 is now an immediate consequence of Lemmas 2.5.3 and 2.5.4 and theorem 2.5.5 thanks to Theorem 2.7 of [77].

2.6 Technical Lemmas

In this section, we present the technical lemmas that were used in the proof of Theorem 2.3.1. For notational purposes, we let $\|\cdot\|_\omega$ define the L^2-norm over a domain ω, and we shall omit the subscript in the case $\omega = T$ the reference element.

We begin with a bound relating the vertex values of a polynomial to its L^2 norm over the triangle. The constant appearing in Lemma 2.6.1 is the best one possible; a related result was proved in [79].

Lemma 2.6.1. *For $u \in \mathbb{P}_p(T)$, we have that*

$$\max_{i\in\{1,2,3\}} |u(v_i)| \leq \frac{1}{2\sqrt{2}}(p+1)(p+2)\|u\| \,.$$

Proof. For $0 \leq i, j, i+j \leq p$ define

$$\Psi_{ij}(x,y) = \sqrt{\frac{(2i+1)(i+j+1)}{2}} P_i^{(0,0)}(\xi) \left(\frac{1-\eta}{2}\right)^i P_j^{(2i+1,0)}(\eta), \tag{2.14}$$

where $\xi = \frac{2(1+x)}{1-y} - 1$ and $\eta = y$ [47, §3]. These functions form an orthonormal basis for $\mathbb{P}_p(T)$. Hence, $u \in \mathbb{P}_p(T)$ can be written in the form $u = \sum_{i+j\leq p} u_{ij}\Psi_{ij}$ and $\|u\|^2 = \sum_{i+j\leq p} u_{ij}^2$. It suffices to prove the inequality in the case of vertex $(-1,-1)$. Using Cauchy-Schwarz gives

$$\begin{aligned} |u(-1,-1)|^2 &= \left(\sum_{i+j\leq p} (-1)^{i+j} u_{ij} \sqrt{\frac{(2i+1)(i+j+1)}{2}} \right)^2 \\ &\leq \sum_{i+j\leq p} u_{ij}^2 \sum_{i+j\leq p} \frac{(2i+1)(i+j+1)}{2} = \frac{1}{8}(p+1)^2(p+2)^2\|u\|^2 \,. \end{aligned}$$

□

Next, we prove an equality needed to bound the minimal extension of the vertex functions.

Lemma 2.6.2. *Define*

$$\xi_p(x) = \frac{(-1)^{p+1}}{p(p+1)} P_p'(x)(1-x) = \frac{(-1)^{p+1}}{p} \frac{1-x}{2} P_{p-1}^{(1,1)}(x), \quad x \in [-1,1]$$

where P_p is the Legendre polynomial. Then

$$\left\| \xi_p \right\|_{[-1,1]}^2 = \frac{4}{(p+1)(2p+1)}. \tag{2.15}$$

Proof. We note that $\xi_p(-1) = 1, \xi_p(1) = 0$, and $\xi_p(x_i) = 0$ where $x_i, i = 2, \ldots, p$ are the roots of $P_p'(x)$. Hence, using the $(p+1)$ point Gauss-Lobatto quadrature gives

$$\int_{-1}^{1} \xi_p^2(x)\, dx = w_1 + \sum_{i=2}^{p} w_i \xi_p^2(x_i) + E$$

where E is the error term

$$E = -\frac{(p+1)p^3 2^{2p+1}[(p-1)!]^4}{(2p+1)[(2p)!]^3} \frac{\mathrm{d}^{2p}}{\mathrm{d}x^{2p}} \xi_p^2(x) \Big|_{x=\eta}, \qquad \eta \in [-1,1].$$

for some $\eta \in [-1,1]$. Direct calculation shows that $E = -\frac{2}{(2p+1)(p+1)p}$ which, along with the fact that $w_1 = \frac{2}{p(p+1)}$, gives the result claimed. □

Using the function defined in Lemma 2.6.2, we can bound the minimal extensions of the vertex functions.

Lemma 2.6.3. *The minimal extension of the vertex basis function of degree p satisfies the bound*

$$\frac{c}{p^4} \leq \|\widetilde{\varphi}_i\|^2 \leq \frac{C}{p^4}$$

where c and C are positive constants independent of p.

Remark. Surprisingly, if $\lfloor p/2 \rfloor$ is replaced with the full order p, then the above estimate no longer holds. Instead, we show in Lemma 5.7.3 that $\frac{c}{p^4} \leq \|\widetilde{\varphi}_i\|^2 \leq \frac{C \log p}{p^4}$ implying that the condition number of the preconditioned system will grow as $\mathcal{O}(\log p)$.

Proof. Without loss of generality, assume that $i = 1$ which corresponds to $v_1 = (-1, -1)$ of the reference triangle T. Using the minimal L^2 property of $\widetilde{\varphi}_1$, and $\mathbb{Q}_{\lfloor p/2 \rfloor} \subset \mathbb{P}_p$ where $\mathbb{Q}_r = \{x^\alpha y^\beta : 0 \leq \alpha, \beta \leq r\}$, gives:

$$\|\widetilde{\varphi}_1\|^2 = \min_{\substack{u=\varphi_1 \text{ on } \partial T \\ u \in \mathbb{P}_p}} \|u\|^2 \leq \min_{\substack{u=\varphi_1 \text{ on } \partial T \\ u \in \mathbb{Q}_{\lfloor p/2 \rfloor}}} \|u\|^2 .$$

Consider the polynomial $\zeta_r \in \mathbb{Q}_{2r}$ defined by

$$\zeta_r(x, y) = \xi_r(x)\xi_r(y) - \xi_r(-x)\xi_r(-y)$$

where $\xi_r(x)$ is defined in Lemma 2.6.2. By construction, $\zeta_{\lfloor p/2 \rfloor} = \varphi_1$ on ∂T, and using Equation (2.15) gives

$$\left\|\zeta_{\lfloor p/2 \rfloor}\right\|^2 = \frac{4(2\lfloor p/2 \rfloor - 1)}{\lfloor p/2 \rfloor^2(\lfloor p/2 \rfloor + 1)^2(2\lfloor p/2 \rfloor + 1)} \leq \frac{C}{p^4}$$

which proves the upper bound.

The lower bound is an immediate consequence of Lemma 2.6.1 (choosing $u = \widetilde{\varphi}_i$). □

Remark. The $\lfloor p/2 \rfloor$ order on the vertex functions is crucial here to guarantee that $\mathbb{Q}_{\lfloor p/2 \rfloor}$ is a smaller space than $\mathbb{P}_p$. Using p as the order on the Legendre polynomial will result in a growing condition number; see Figure 2.2 of Chapter 5 for more information.

The next result gives an explicit expression for the norm of a minimal extension of an edge function:

Lemma 2.6.4. *Let $u \in \mathbb{P}_p(\gamma)$ be a polynomial on edge $\gamma \subset \partial T$, which vanishes at the endpoints, be written in the form*

$$u(x) = (1-x^2)\sum_{i=0}^{p-2} w_i P_i^{(2,2)}(x),$$

where $x \in [-1,1]$ is a parametrization of γ. Then the norm of the the minimal energy extension $\widetilde{u} \in \mathbb{P}_p(T)$, satisfying $\widetilde{u} = 0$ on $\partial T \setminus \gamma$ and $u = \widetilde{u}$ on γ, is given by

$$\|\widetilde{u}\|^2 = \sum_{i=0}^{p-2} \frac{2\mu_i w_i^2}{(p+i+4)(p-i-1)} \tag{2.16}$$

where $\mu_i = \int_{-1}^{1}(1-x^2)^2 P_i^{(2,2)}(x)^2\,dx = \frac{32}{2i+5}\frac{(i+1)(i+2)}{(i+3)(i+4)}$.

Proof. Without loss of generality, take the edge to be $\gamma = \{(x,y) : y = -1, -1 \leq x \leq 1\}$ of the reference triangle. We construct a basis for the space of polynomials which vanish on $\partial T \setminus \gamma_i$ and express $\widetilde{u}$ in the form

$$\widetilde{u}(x,y) = (1-\xi^2)\left(\frac{1-\eta}{2}\right)^2 \sum_{i+j\leq p-2} \widetilde{u}_{ij} P_i^{(2,2)}(\xi)\left(\frac{1-\eta}{2}\right)^i P_j^{(2i+5,0)}(\eta)$$

for suitable coefficients $\{\widetilde{u}_{ij} \in \mathbb{R} : i+j \leq p-2\}$ where $\xi = \frac{2(1+x)}{1-y} - 1$ and $\eta = y$. The L^2 norm to minimize can be expressed in terms of $\{\widetilde{u}_{ij}\}$

$$\|\widetilde{u}\|^2 = \int_{-1}^{1}\int_{-1}^{1} \widetilde{u}^2(x,y)\left(\frac{1-\eta}{2}\right) d\eta d\xi = \sum_{i+j\leq p-2} \widetilde{u}_{ij}^2 \mu_i \nu_{ij}$$

where $\nu_{ij} = \int_{-1}^{1}\left(\frac{1-\eta}{2}\right)^{2i+5} P_j^{(2i+5,0)}(\eta)^2\,d\eta = \frac{1}{i+j+3}$ and μ_i as defined in the lemma statement. The requirement for $\widetilde{u} = u$ on γ means that

$$\widetilde{u}(x,-1) = (1-x^2) \sum_{i+j\leq p-2} (-1)^j \widetilde{u}_{ij} P_i^{(2,2)}(x) \implies w_i = \sum_{j=0}^{p-2-i} (-1)^j \widetilde{u}_{ij}.$$

The Cauchy-Schwarz inequality gives

$$w_i^2 \leq \left(\sum_{j=0}^{p-2-i} \nu_{ij}^{-1} \right) \left(\sum_{j=0}^{p-2-i} \widetilde{u}_{ij}^2 \nu_{ij} \right) = \frac{1}{2}(p-i-1)(p+i+4) \sum_{j=0}^{p-2-i} \widetilde{u}_{ij}^2 \nu_{ij} \tag{2.17}$$

with equality if there exists a constant λ, such that for all $j \in [0, p-2-i]$ and fixed i, such that $(-1)^j \widetilde{u}_{ij} \nu_{ij}^{1/2} = \lambda \nu_{ij}^{-1/2}$, or equally well, $\widetilde{u}_{ij} = (-1)^j \lambda (i+j+3)$. The choice $\lambda = \frac{w_i}{\sum_{j=0}^{p-2-i} i+j+3}$ gives $w_i = \sum_{j=0}^{p-2-i} (-1)^j \widetilde{u}_{ij}$. Hence, the case of strict equality in Equation (2.17) is achieved.

Direct computation reveals that

$$\|\widetilde{u}\|^2 = \sum_{i=0}^{p-2} \mu_i \sum_{j=0}^{p-2-i} \widetilde{u}_{ij}^2 \nu_{ij} = \sum_{i=0}^{p-2} \frac{\mu_i w_i^2}{\frac{1}{2}(p-i-1)(p+i+4)}$$

and the result follows. □

The next result gives a bound on the norm of the minimal extension of a polynomial supported on a single edge of a triangle:

Lemma 2.6.5. *Let $u \in \mathbb{P}_p(T)$, such that $u(v_i) = 0$ for v_i the vertices of T. Let γ be any edge of T, and let $U \in \mathbb{P}_p(\partial T)$ such that $U|_\gamma = u|_\gamma$ and $U = 0$ on the remaining two edges. Let $\widetilde{U}$ denote the minimal L^2 extension of U, then there exists a constant C independent of p such that*

$$\left\| \widetilde{U} \right\| \leq C \|u\|.$$

Proof. Without loss of generality, we assume $\gamma = \{(x, y) : y = -1, -1 \leq x \leq 1\}$ and let Ψ_{ij} be given by Equation (2.14). Since $\{\Psi_{ij}\}_{0\leq i,j,i+j\leq p}$ forms a basis, we may write $u = \sum_{i+j\leq p} u_{ij}\Psi_{ij}$, and denote

$$f = u|_\gamma = \sum_{i+j\leq p} (-1)^j u_{ij} \sqrt{\frac{(2i+1)(i+j+1)}{2}} P_i^{(0,0)}(x).$$

Our technique is to express f as a sum of $(1-x^2)P_i^{(2,2)}$, $i = 0, \dots, p-2$, and to then use Lemma 2.6.4 to calculate $\left\|\widetilde{U}\right\|$. Define

$$v_i = \sum_{j=0}^{p-i} (-1)^j u_{ij} \sqrt{\frac{(2i+1)(i+j+1)}{2}}, \tag{2.18}$$

then in order to use Lemma 2.6.4, we seek coefficients w_i such that

$$f = \sum_{i=0}^{p} v_i P_i^{(0,0)}(x) = (1-x^2) \sum_{i=0}^{p-2} w_i P_i^{(2,2)}(x).$$

Observe that since u vanishes at the vertices of T, we have $u(\pm 1, -1) = 0$, which in turn implies $\sum_{i=0}^{p} v_i = 0$ and $\sum_{i=0}^{p} (-1)^i v_i = 0$, or equally well

$$\sum_{i=0,\text{even}}^{p} v_i = 0, \qquad \sum_{i=1,\text{odd}}^{p} v_i = 0. \tag{2.19}$$

Consequently, we can rewrite f as

$$f = \sum_{i=2,\text{even}}^{p} (P_i^{(0,0)} - P_{i-2}^{(0,0)}) S_i + \sum_{i=3,\text{odd}}^{p} (P_i^{(0,0)} - P_{i-2}^{(0,0)}) S_i$$

where

$$S_i = v_i + v_{i+2} + \cdots + \begin{cases} v_p \\ v_{p-1} \end{cases} = \begin{cases} v_0 + \cdots + v_{i-2} \text{ if } i \text{ even} \\ v_1 + \cdots + v_{i-2} \text{ else} \end{cases}$$

depending on the parity.

Using the identity

$$-\frac{1-x^2}{2(n-1)} \left(\frac{(n+1)(n+2)}{2n} P_{n-2}^{(2,2)} - \frac{n-1}{2} P_{n-4}^{(2,2)} \right) = P_n^{(0,0)} - P_{n-2}^{(0,0)}, \qquad n \geq 2$$

which follows from identities (22.7.15) to (22.7.19) from [1] where P_{n-4} is understood to be 0 for $n < 4$, we have

$$\sum_{i=2}^{p} \left(-\frac{(i+1)(i+2)}{4i(i-1)} P_{i-2}^{(2,2)} + \frac{1}{4} P_{i-4}^{(2,2)} \right) S_i = \sum_{i=0}^{p-2} w_i P_i^{(2,2)}$$

and we deduce that $w_i = \frac{S_{i+4}}{4} - \frac{(i+3)(i+4)}{4(i+1)(i+2)} S_{i+2}$. Writing $S_{i+4} = S_{i+2} - v_{i+2}$, we have

$$w_i = -\frac{v_{i+2}}{4} - \frac{5+2i}{2(i+1)(i+2)} S_{i+2}. \tag{2.20}$$

The Cauchy-Schwarz inequality applied to Equation (2.18) gives

$$v_i^2 \leq \sum_{j=0}^{p-i} u_{ij}^2 \sum_{j=0}^{p-i} \frac{(2i+1)(i+j+1)}{2} = \frac{(2i+1)(i+p+2)(p-i+1)}{4} \sum_{j=0}^{p-i} u_{ij}^2.$$

which in turn gives

$$\sum_{i=0}^{p} \frac{4v_i^2}{(2i+1)(i+p+2)(p-i+1)} \leq \sum_{i=0}^{p} \sum_{j=0}^{p-i} u_{ij}^2 = \|u\|^2. \tag{2.21}$$

Using Lemma 2.6.4 and the inequality $w_i^2 \leq \frac{v_{i+2}^2}{8} + \frac{1}{2}k_i^2 S_{i+2}^2$ where $k_i = \frac{5+2i}{2(i+1)(i+2)}$ deduced from Equation (2.20), we have

$$\begin{aligned}\left\|\widetilde{U}\right\|^2 &= \sum_{i=0}^{p-2} \frac{2\mu_i w_i^2}{(p+i+4)(p-i-1)} \\ &\leq C\left(\sum_{i=0}^{p-2} \frac{v_{i+2}^2}{(p+i+4)(p-i-1)(2i+5)} + \sum_{i=0}^{p-2} \frac{k_i^2 S_{i+2}^2}{(p+i+4)(p-i-1)(2i+5)}\right).\end{aligned}$$

Turning to the first term, thanks to Equation (2.21), we have

$$\sum_{i=0}^{p-2} \frac{v_{i+2}^2}{(p+i+4)(p-i-1)(2i+5)} \leq \frac{1}{4}\sum_{i=0}^{p} \frac{4v_i^2}{(2i+1)(i+p+2)(p-i+1)} \leq C\|u\|^2 .$$

For the second term, we first denote

$$\widetilde{S}_i = \begin{cases} |v_0| + \cdots + |v_{i-2}| \text{ if } i \text{ even} \\ |v_1| + \cdots + |v_{i-2}| \text{ else} \end{cases}$$

so that $S_i^2 \leq \widetilde{S}_i^2$. We first note that $k_i \leq \frac{2}{i+1}$ and change the index of the summation, then using Lemma 3.4.10 in the case of $j = 1$ and Equation (2.21), we obtain

$$\begin{aligned}&\sum_{i=2}^{p} \frac{S_i^2}{(i-1)^2(2i+1)(p+i+2)(p-i+1)} \\ &\leq \sum_{i=2}^{p} \frac{\widetilde{S}_i^2}{(i-1)^2(2i+1)(p+i+2)(p-i+1)} \\ &\leq C\sum_{i=0}^{p} \frac{v_i^2}{(2i+1)(i+p+2)(p-i+1)} \leq C\|u\|^2\end{aligned}$$

and the result follows as claimed. □

Finally, we are in a position to give the proof of Theorem 2.5.5:

Proof. The first step is to construct a suitable decomposition for $u \in X$. Let

$$u_V = \sum_{i=1}^{3} u(v_i)\widetilde{\varphi}_i \in X_V$$

be the interpolant to u at the vertices using the minimal L^2 vertex functions.

Consequently $(u - u_V)|_{\partial T} \in \mathbb{P}_p(\partial T)$ vanishes at the element vertices, and can therefore be written in the form

$$u - u_V|_{\partial T} = U_1 + U_2 + U_3$$

where $U_i \in \mathbb{P}_p(\partial T)$ is supported on edge γ_i. We then let

$$u_{E_i} \in X_{E_i}$$

be the minimal L^2 extension of U_i into the triangle. It follows that

$$u - u_V - \sum_{i=1}^{3} u_{E_i} = u_I \in X_I$$

Thus $u = u_V + \sum_{i=1}^{3} u_{E_i} + u_I$ is a decomposition of u. It remains to show the decomposition is uniformly bounded.

Firstly, by Lemma 2.6.1:

$$a_V(u_V, u_V) = \frac{1}{p^4}\sum_{i=1}^{3} u(v_i)^2 \leq \frac{3}{p^4}\max_{i\in\{1,2,3\}} u^2(v_i) \leq 3C\|u\|^2 . \tag{2.22}$$

For the edge contributions, we use Lemma 2.6.5 to bound

$$a_{E_i}(u_{E_i}, u_{E_i}) = \left\|u_{E_i}\right\|^2 \leq C\|u - u_V\|^2 \leq 2C\left(\|u\|^2 + \|u_V\|^2\right),$$

and then use the estimate $\|u_V\|^2 \leq C a_V(u_V, u_V)$ from Lemma 2.5.3 and Equation (2.22), to deduce $\|u_V\|^2 \leq \|u\|^2$ and hence $a_{E_i}(u_{E_i}, u_{E_i}) \leq C\|u\|^2$.

Finally, as $u_V + \sum_{i=1}^{3} u_{E_i} \in \widetilde{X}_B$, Lemma 2.5.1 gives us $\left(u_I, u_V + \sum_{i=1}^{3} u_{E_i}\right) = 0$, hence

$$a_I(u_I, u_I) = \|u_I\|^2 \leq \|u_I\|^2 + \left\|u_V + \sum_{i=1}^{3} u_{E_i}\right\|^2 = \|u\|^2,$$

and our result follows. □

2.7 Conclusions

The current chapter has developed an Additive Schwarz method which results in a uniform condition number in both mesh size h and polynomial order p on the triangle. The key idea is the construction of a new basis which is used to define the subspace decomposition for the Additive Schwarz method. It is not our intention to suggest that this basis be adopted thoroughly; e.g. for Poisson-type problems for which the mass matrix is absent and only the stiffness matrix appears. The key point is that, although the spaces used in the description of the ASM are constructed using the specific basis described in Section 2.2, the resulting abstract form of the ASM means that the preconditioner can be applied to whatever basis the reader may care to use through applying a change of basis. For instance, Chapter 6 shows how the algorithm can be applied to the Bernstein basis at a cost of $\mathcal{O}(p^3)$ operations.

Chapter THREE

Tetrahedra

3.1 Introduction

We now turn preconditioning the p-version mass matrix on tetrahedra. Much as in the previous chapter, we develop a new high-order basis with the usual property that the individual functions can be associated with distinct geometric entities of the tetrahedron; the distinguishing new property is that the resulting *mass matrix is spectrally equivalent to its own diagonal* with constants independent of h, p. The basis can then used as the foundation for an Additive Schwarz method (ASM), implying that one only needs to implement an appropriate change-of-basis in existing codes to utilize the preconditioner.

With the result (p uniformity) and theoretical framework (ASM) both mimicking the 2D case, this begs the question of the triviality of the present chapter; the 3D is far from a simple consequence of the 2D case. For one, the techniques used to prove the stable decomposition on the face spaces has no correspondence in the 2D case and is quite technical. In addition, the question of how to extend edge and nodal basis functions onto faces is highly nontrivial. In fact, if one were to use the tetrahedron basis as a template for the triangle case, one can obtain a preconditioner whose condition number is more than half of those presented in the previous chapter.

The remainder of the chapter is organized as follows. In section 2, we define the basis functions and state the main result. In section 3, we present illustrative numerical examples such as singularly perturbed problem and time-stepping. Finally in section 4, we prove the inequalities and polynomial extension lemmas needed for the main result.

3.2 Basis Definition and Main Result

Let T be the reference tetrahedron in $\mathbb{R}^3$ with vertices $v_1 = (-1,-1,-1), v_2 = (1,-1,-1), v_3 = (-1,1,-1)$, $v_4 = (-1,-1,1)$, and let F_1 and E_1 be the face and edge given by

$$\begin{aligned} F_1 &:= T \cap \{z=-1\}, \\ E_1 &:= T \cap \{z=-1\} \cap \{y=-1\}. \end{aligned}$$

Let $p \geq 4$ be a given integer, and let $\mathbb{P}_p(D)$ be the space of polynomials of total degree p on a domain D. Let $X := \mathbb{P}_p(T)$, and $\lambda_i \in \mathbb{P}_1(T)$ for $i = 1,2,3,4$ be the barycentric coordinates of T associated with vertex v_i; i.e. $\lambda_i(v_j) = \delta_{ij}$.

We begin by introducing a particular basis for $\mathbb{P}_p(T)$ which, as usual, consists of functions associated with vertices, edges, faces and the interior of the tetrahedron. However, the actual choice of functions differs from those typically used in the literature.

3.2.1 Basis functions

The classical Jacobi polynomials [1] on $[-1,1]$ are denoted by $P_n^{(\alpha,\beta)}$, where n is the order of the polynomial and $\alpha, \beta > -1$ are weights, and satisfy

$$\int_{-1}^{1} \left(\frac{1-x}{2}\right)^{\alpha} \left(\frac{1+x}{2}\right)^{\beta} P_n^{(\alpha,\beta)}(x)^2 \, dx = \frac{2(\alpha+n)!(\beta+n)!}{n!(\alpha+\beta+2n+1)(\alpha+\beta+n)!}.$$

For non-negative integers m, q, let $\Phi_q^{(m)}(x) \in \mathbb{P}_q([-1,1])$ be defined by

$$\Phi_q^{(m)}(x) := \frac{(-1)^q}{q+1} P_q^{(m,1)}(x), \tag{3.1}$$

and $\Xi_q \in \mathbb{P}_q([0,1]^2)$ be given by

$$\Xi_q(l_1, l_2) := P_q^{(2,2)}\left(\frac{2l_2}{l_1+l_2} - 1\right)(l_1+l_2)^q. \tag{3.2}$$

Interior Basis Functions

Let

$$\omega_{ijk} := \lambda_1\lambda_2\lambda_3\lambda_4\Xi_i(\lambda_1,\lambda_2)P_j^{(2i+5,2)}\left(\frac{2\lambda_3}{1-\lambda_4} - 1\right)(1-\lambda_4)^j\, P_k^{(2i+2j+8,2)}(2\lambda_4 - 1)$$

for $0 \leq i, j, k, i+j+k \leq p-4$. Note that ω_{ijk} vanishes on the boundary of T due to the factor $\lambda_1\lambda_2\lambda_3\lambda_4$. The set $\{\omega_{ijk}\}$ is an orthogonal basis for $X_I := X \cap H_0^1(T)$ with respect to the $L^2(T)$ inner product (see Lemma 3.4.1).

Face Basis Functions

The basis functions associated with the face F_1 are defined by

$$\psi_{ij}^{(1)} := \lambda_1\lambda_2\lambda_3\Xi_i(\lambda_1,\lambda_2)P_j^{(2i+5,2)}\left(\frac{2\lambda_3}{1-\lambda_4} - 1\right)(1-\lambda_4)^j\, \Phi_{p-3-i-j}^{(2i+2j+8)}(2\lambda_4 - 1)$$

for $0 \leq i, j, i+j \leq p-3$. In particular, the presence of the factor $\lambda_1\lambda_2\lambda_3$ means that these functions vanish on the remaining three faces. The basis functions on the remaining faces are defined in an analogous fashion to give the face spaces $X_{F_k} :=$

span$\{\psi_{ij}^{(k)}\}$. The functions provide an orthogonal basis for X_{F_k} (e.g. $(\psi_{ij}^{(k)}, \psi_{mn}^{(k)}) \propto \delta_{ij,mn}$ where $(\cdot, \cdot)$ is the L^2 inner-product over T); see Lemma 3.4.1.

Edge Basis Functions

The basis functions associated with the edge E_1 are chosen as follows:

$$\chi_i^{(1)} := \lambda_1 \lambda_2 \Xi_i(\lambda_1, \lambda_2) \left(\frac{q_i(\lambda_3, \lambda_4) + q_i(\lambda_4, \lambda_3)}{2} \right), \qquad 0 \leq i \leq p-2,$$

where the function q_i is given by

$$q_i(l_1, l_2) := \Phi_j^{(2i+5)} \left(\frac{2l_1}{1-l_2} - 1 \right) (1-l_2)^j \, \Phi_{p-2-i-j}^{(2i+2j+6)}(2l_2 - 1) \tag{3.3}$$

with $j = \lfloor (p-i-2)/2 \rfloor$. The basis functions on the remaining edges are defined analogously to give the edge spaces $X_{E_k} :=$ span$\{\chi_i^{(k)}\}$.

The edge basis functions have the following properties:

1. locally supported: vanish on the two faces which do not contain edge E_1 (owing to the factor $\lambda_1 \lambda_2$);
2. symmetry: the values on the two non-zero faces satisfy the condition that $\chi(r, s, t, 0) = \chi(r, s, 0, t)$ for all r, s, t;
3. orthogonality: $(\chi_i^{(k)}, \chi_j^{(k)}) \propto \delta_{ij}$ (see Lemma 3.4.1).

Vertex Basis Functions

The function associated with the vertex v_1 is given by

$$\varphi_1 := \frac{1}{3}\lambda_1\left(q(\lambda_2,\lambda_3,\lambda_4)+q(\lambda_3,\lambda_4,\lambda_2)+q(\lambda_4,\lambda_2,\lambda_3)\right)$$

where

$$\begin{aligned} q(l_1,l_2,l_3) := \Phi_i^{(2)}\left(\frac{2l_1}{1-l_2-l_3}-1\right)(1-l_2-l_3)^i\,\Phi_j^{(2i+3)}\left(\frac{2l_2}{1-l_3}-1\right) \\ \times (1-l_3)^j\,\Phi_{p-1-i-j}^{(2i+2j+4)}(2l_3-1), \end{aligned} \tag{3.4}$$

with $i = \lfloor \frac{p}{2} \rfloor$ and $j = \lfloor \frac{i}{2} \rfloor$. The basis functions on the remaining vertices are defined in an analogous manner to give the vertex spaces $X_{V_k} := \text{span}\{\varphi_k\}$.

The vertex basis functions have the following properties:

1. local support: $\varphi_1(v_1) = 1$ and vanishes at the remaining vertices;
2. symmetry: $\varphi_1(r,s,0,0) = \varphi_1(r,0,s,0) = \varphi_1(r,0,0,s)$ for all r,s.

It is not difficult to see that the basis functions are linearly independent and a simple counting argument shows that the union of the sets give a basis for X.

Basis Functions on a Mesh

Let Ω be a bounded three-dimensional domain, and let $\mathcal{P}$ be a partitioning of Ω into the union of disjoint tetrahedra such that the intersection of any two distinct elements is either a single common vertex, edge or face. Each element $K \in \mathcal{P}$ is the image of the reference element T under a (possibly non-affine) map $\mathcal{F}_K$ such that there exists positive constants θ, Θ such that the Jacobian $D\mathcal{F}_K$ satisfies

$$\theta|K| \leq |D\mathcal{F}_K(x)| \leq \Theta|K| \qquad \forall x \in K. \tag{3.5}$$

It is worth noting that this condition does not place constraints on the shape regularity of the mesh, and, in particular, allows for “needle” or “slab” elements.

The basis functions on an element $K \in \mathcal{P}$ are defined to be pull-backs using the map $\mathcal{F}_K$ in the usual manner, e.g.

$$\varphi_{1,K}(x) := \varphi_1(\mathcal{F}_K^{-1}(x)), \qquad x \in K.$$

The fact that the basis functions are associated with vertices, edges and faces, together with the symmetry properties means that enforcing global conformity follows the same procedure for hierarchic bases. In particular, one needs to number the degrees of freedom in a systematic manner to ensure that the edge and face basis functions will be oriented correctly. The standard finite element sub-assembly gives the global mass matrix

$$\mathbf{M} = \sum_{K \in \mathcal{P}} \mathbf{\Lambda}_K \mathbf{M}_K \mathbf{\Lambda}_K^T$$

where $\mathbf{\Lambda}_K$ is the local assembly matrix and $\mathbf{M}_K$ is the element mass matrix expressed using the above basis. For more details about the assembly process, see [6].

3.2.2 Main result

The main result states that the diagonal of the mass matrix is spectrally equivalent to the full matrix:

Theorem 3.2.1. *There exists constants c, C independent of h, p such that*

$$c\,\mathrm{diag}(\mathbf{M}) \leq \mathbf{M} \leq C\,\mathrm{diag}(\mathbf{M}).$$

Proof. Let $\hat{\mathbf{M}}$ be the mass matrix on the reference element, then Equation (3.5) implies that

$$\theta|K|\hat{\mathbf{M}} \leq \mathbf{M}_K \leq \Theta|K|\hat{\mathbf{M}}. \tag{3.6}$$

We shall show below that the following condition holds with constants c, C independent of p:

$$c\,\mathrm{diag}(\hat{\mathbf{M}}) \leq \hat{\mathbf{M}} \leq C\,\mathrm{diag}(\hat{\mathbf{M}}). \tag{3.7}$$

Thus, standard sub-assembly together with Equation (3.6) and Equation (3.7) shows that

$$\begin{aligned} c\,\mathrm{diag}(\mathbf{M}) &= c\sum_{K\in\mathcal{P}} \mathbf{\Lambda}_K \mathrm{diag}\left(\mathbf{M}_K\right)\mathbf{\Lambda}_K^T \leq c\sum_{K\in\mathcal{P}} |K|\mathbf{\Lambda}_K \mathrm{diag}\left(\hat{\mathbf{M}}\right)\mathbf{\Lambda}_K^T \\ &\leq \sum_{K\in\mathcal{P}} |K|\mathbf{\Lambda}_K \hat{\mathbf{M}}\mathbf{\Lambda}_K^T \\ &\leq C\sum_{K\in\mathcal{P}} |K|\mathbf{\Lambda}_K \mathrm{diag}\left(\hat{\mathbf{M}}\right)\mathbf{\Lambda}_K^T \leq C\sum_{K\in\mathcal{P}} \mathbf{\Lambda}_K \mathrm{diag}\left(\mathbf{M}_K\right)\mathbf{\Lambda}_K^T = C\,\mathrm{diag}(\mathbf{M}). \end{aligned}$$

It remains to show that condition Equation (3.7) holds: that is, there exists constants c, C independent of p such that

$$c\vec{u}^T \text{diag}(\hat{\mathbf{M}})\vec{u} \leq \vec{u}^T \hat{\mathbf{M}} \vec{u} \leq C\vec{u}^T \text{diag}(\hat{\mathbf{M}})\vec{u}, \qquad \forall \vec{u}.$$

Let $u \in X$ be the function corresponding to $\vec{u}$ so that $\vec{u}^T \hat{\mathbf{M}} \vec{u} = \|u\|^2$ where $\|\cdot\|$ is the standard L^2 inner-product over T. The vector $\vec{u}$ can be decomposed as follows:

$$\vec{u} = [\vec{u}_I, \vec{u}_{F_1}, \ldots, \vec{u}_{F_4}, \vec{u}_{E_1}, \ldots, \vec{u}_{E_6}, \vec{u}_{V_1}, \ldots, \vec{u}_{V_4}]$$

where $\vec{u}_I$ corresponds to the coefficients of the interior basis functions ω_{ijk} or, equally well, a function $u_I \in X_I$ etc. This partitioning induces a partitioning of the mass matrix into subblocks. Moreover, the orthogonality of the basis functions *within* each block (not between different blocks) means that

$$\text{diag}(\hat{\mathbf{M}}) = \begin{bmatrix} \hat{\mathbf{M}}_I & & & \\ & \hat{\mathbf{M}}_{F_1} & & \\ & & \ddots & \\ & & & \hat{\mathbf{M}}_{V_4} \end{bmatrix}.$$

Thus,

$$\vec{u}^T \text{diag}(\hat{\mathbf{M}})\vec{u} = \|u_I\|^2 + \sum_{i=1}^{4} \left\|u_{F_i}\right\|^2 + \sum_{i=1}^{6} \left\|u_{E_i}\right\|^2 + \sum_{i=1}^{4} \left\|u_{V_i}\right\|^2$$

where $u_I \in X_I$, $u_{F_i} \in X_{F_i}$, $u_{E_i} \in X_{E_i}$ and $u_{V_i} \in X_{V_i}$.

Condition Equation (3.7) hence reduces to showing that for all $u \in X$, there exist positive constants c, C independent of p such that

$$\begin{aligned} c\left(\|u_I\|^2 + \sum_{i=1}^{4}\left\|u_{F_i}\right\|^2 + \sum_{i=1}^{6}\left\|u_{E_i}\right\|^2 + \sum_{i=1}^{4}\left\|u_{V_i}\right\|^2\right) \leq \|u\|^2 \leq \\ C\left(\|u_I\|^2 + \sum_{i=1}^{4}\left\|u_{F_i}\right\|^2 + \sum_{i=1}^{6}\left\|u_{E_i}\right\|^2 + \sum_{i=1}^{4}\left\|u_{V_i}\right\|^2\right). \end{aligned} \tag{3.8}$$

The upper-bound follows at once thanks to the triangle inequality. The proof of the lower bounds is less straight forward and relies on a number of technical estimates whose proof is postponed to Section 3.4.

Lemma 3.4.4 and the fact that $\|u\|_\infty \leq Cp^3\|u\|$ [79] gives a bound on the vertex portions:

$$\left\|u_{V_i}\right\| = \|u(v_i)\varphi_i\| \leq \|\varphi_i\|\|u\|_\infty \leq C\|u\|, \qquad i = 1, \ldots, 4.$$

Now, by Lemma 3.4.5, we obtain

$$\left\|u_{E_i}\right\| \leq C\left\|u - \sum_{i=1}^{4} u_{V_i}\right\| \leq C\|u\|, \qquad i = 1, \ldots 6.$$

We next apply Corollary 3.4.7 to each individual face

$$\left\|u_{F_i}\right\| \leq C\left\|u - \sum_{i=1}^{4} u_{V_i} - \sum_{i=1}^{6} u_{E_i}\right\| \leq C\|u\|, \qquad i = 1, 2, 3, 4.$$

Finally, a bound for u_I follows bbfrom triangle inequality

$$\|u_I\| \leq C\left\|u - \sum_{i=1}^{4} u_{V_i} - \sum_{i=1}^{6} u_{E_i} - \sum_{i=1}^{4} u_{F_i}\right\| \leq C\|u\|.$$

Collecting these estimates establishes the lower bound in Equation (3.8).

□

3.3 Numerical Examples

3.3.1 Preconditioned mass matrix

We first illustrate Theorem 2.3.1 for a single (reference) element, and for a mesh obtained by partitioning the cube into 24 elements. Let

$$\hat{\mathbf{M}}_s := \hat{\mathbf{P}}^{-1/2}\hat{\mathbf{M}}\hat{\mathbf{P}}^{-1/2}, \qquad \mathbf{M}_s := \mathbf{P}^{-1/2}\mathbf{M}\mathbf{P}^{-1/2}$$

where $\hat{\mathbf{P}} = \operatorname{diag}(\hat{\mathbf{M}})$, $\mathbf{M}$ is the global mass matrix corresponding to partitioning the cube into 24 elements, and $\mathbf{P} = \operatorname{diag}(\mathbf{M})$. In Section 3.3.1, we show the condition numbers of both $\hat{\mathbf{M}}_s$ and $\mathbf{M}_s$. As predicted by Theorem 2.3.1, the condition numbers of both remain bounded for all p.

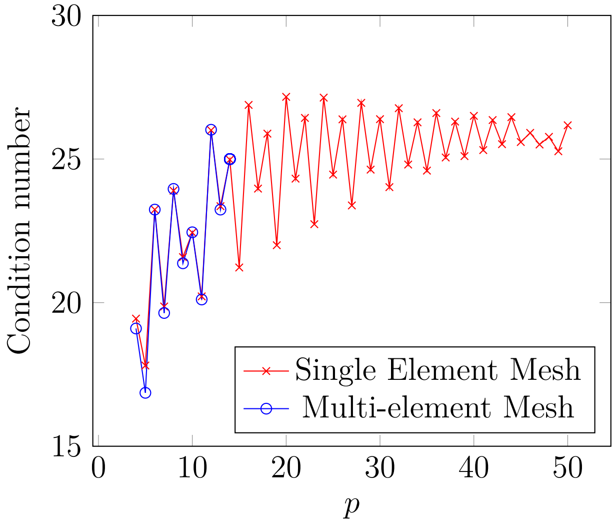


Figure 3.1: Figure illustrates the condition number of $\hat{\mathbf{M}}_s$ and $\mathbf{M}_s$. The bounded condition number of the preconditioned system is predicted in Theorem 2.3.1.

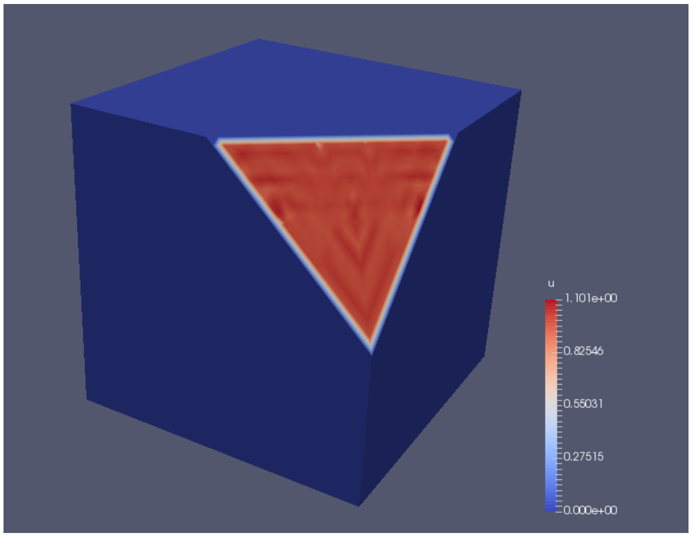


Figure 3.2: Cross-section of the solution to Equation (3.9) for $\varepsilon^2 = 10^{-4}$ and $p = 10$ on a corner of the cube. Observe the presence of a boundary layer.

3.3.2 Singularly Perturbed Problem

The utility of the preconditioner is not confined to the pure mass matrix. Consider the following problem

$$\begin{aligned} u - \varepsilon^2 \Delta u &= f, \qquad x \in \Omega, \\ u &= 0, \qquad x \in \partial\Omega, \end{aligned} \tag{3.9}$$

where $0 < \varepsilon \ll 1$ and $f \in L^2(\Omega)$ which is prototypical of several problems arising in mechanics [11, 38]. The p-version Galerkin discretization of Equation (3.9) leads to an algebraic problem of the form

$$(\mathbf{M} + \varepsilon^2 \mathbf{S})\vec{u} = \vec{f} \tag{3.10}$$

where $\mathbf{S}$ is the stiffness matrix and $\vec{f}$ is the load vector corresponding to f. Solutions to the above problem generally exhibit boundary layers which become steeper as $\varepsilon \to 0$; see Figure 3.2 for a plot of the solution for $f = 1$.

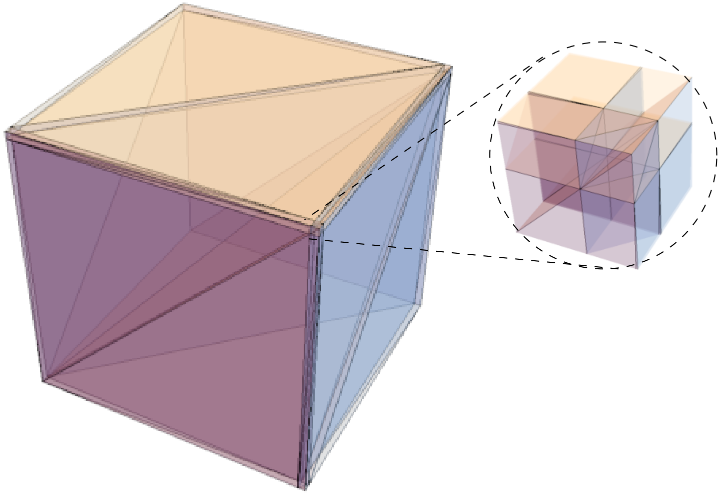


Figure 3.3: Figure illustrating the mesh for the singularly perturbed problem on an octant of the cube. The inset shows the submesh of elements in the corner. Note the needle and slab elements of width $\mathcal{O}(p\varepsilon)$ encompassing the boundary of the cube.

In order to resolve the boundary layers present in the solution, anisotropic elements are needed to obtain robust convergence. It suffices [68] to use a *single* layer of anisotropic elements of width $\mathcal{O}(p\varepsilon)$ around the boundary to obtain robust exponential convergence in p independent of ε. An undesirable side-effect of the anisotropic elements is that the condition number of Equation (3.10) will grow meteorically as $\varepsilon \to 0$ due to the increasing aspect ratio of the anisotropic elements. This difficulty has not gone unnoticed by other researchers: Toselli and Vasseur [75,76] developed a domain decomposition preconditioner for tensor product elements with a condition number independent of ε and only growing as $1 + \log^2 p$. However the types of mesh considered here differ from those of [75,76] which rely strongly on the tensor product structure and only hold on a geometrically graded mesh rather than the (optimal), single layer mesh advocated in [68].

An alternative approach [9], was applied in two dimensions on meshes with a single layer of (needle) anisotropic elements, based on the following norm equivalence

Table 3.1: Condition number of the singularly perturbed matrices obtained using the preconditioner for the pure mass matrix. Observe the condition number exhibits moderate growth in p but remains independent of ε.

ε^2	$p = 4$	$p = 5$	$p = 6$	$p = 7$	$p = 8$	$p = 9$
1e-3	22.61	21.17	30.65	30.02	39.20	39.04
1e-5	23.24	22.09	32.75	31.41	42.32	40.15
1e-7	23.31	22.25	33.08	31.67	42.78	40.38
1e-9	23.31	22.27	33.11	31.70	42.83	40.41

obtained from applying a scaling argument to Schmidt's inequality [23] in conjunction with Theorem 2.3.1:

$$c\,\mathrm{diag}(\mathbf{M}) \leq \mathbf{M} + \varepsilon^2\mathbf{S} \leq \left(1 + C\varepsilon^2\frac{p^4}{(p\varepsilon)^2}\right)\mathbf{M} \leq Cp^2\mathrm{diag}(\mathbf{M}).$$

The same estimate remains valid in three dimensions on meshes containing both "needle" and "slab" elements with aspect ratio ε. If the mass matrix preconditioner is used to precondition the system Equation (3.10), then the condition number of the preconditioned system grows as $\mathcal{O}(p^2)$ but, crucially, remains *independent* of ε, even on an unstructured mesh.

To illustrate the effectiveness of this strategy, we consider problem Equation (3.9) with $f = 1$ and $\Omega = (-100, 100)^3$. Due to symmetry of the problem, it suffices to only consider the octant of the cube given by $(0, 100)^3$. Figure 3.3 illustrates the discretization of the domain with a single layer of anisotropic elements bordering the Dirichlet boundary condition. The condition number of the preconditioned matrices

$$\mathrm{diag}(\mathbf{M})^{-1/2}\left(\mathbf{M} + \varepsilon^2\mathbf{S}\right)\mathrm{diag}(\mathbf{M})^{-1/2}$$

is reported in Table 3.1 where it is seen that the condition number is independent of ε. Interestingly, the asymptotic $\mathcal{O}(p^2)$ growth is not seen for orders for $p < 10$. Results for the 2D case [9] shows that $\mathcal{O}(p^2)$ growth is obtained for $p > 15$.

3.3.3 Time-Stepping

Finally, we discuss applying our preconditioner to time-stepping applications. Let

$$\mathbf{A}(\mu,\nu) := \mu\mathbf{M} + \nu\Delta t\mathbf{S}.$$

For a fully explicit scheme $\nu = 0$, and Theorem 2.3.1 implies that the preconditioner will be uniform in the polynomial order p. For a implicit scheme $\nu > 0$, we once again take advantage of Schmidt's inequality to deduce that

$$\mu\mathbf{M} \leq \mathbf{A}(\mu,\nu) \leq (\mu + C_S p^4 \nu\Delta t)\mathbf{M} \leq 2\max\left(\mu, C_S p^4 \nu\Delta t\right)\mathbf{M}$$

where C_S is the constant arising from Schmidt's inequality. In other words, preconditioning using the diagonal of the mass matrix gives

$$\operatorname{cond}(\tilde{\mathbf{A}}(\mu,\nu)) \leq 2\max\left(1, \frac{C_S\nu\Delta t}{\mu}p^4\right) \tag{3.11}$$

where $\tilde{\mathbf{A}}(\mu,\nu) = \operatorname{diag}(\mathbf{M})^{-1/2}\mathbf{A}(\mu,\nu)\operatorname{diag}(\mathbf{M})^{-1/2}$; in practice one does not see the $\mathcal{O}(p^4)$ growth owing to the small multiplicative factor $C_S\nu\Delta t/\mu$.

For a concrete example, consider a system of nonlinear reaction-diffusion equations [36] on the hemisphere which exhibits pattern formation [61]:

$$\begin{aligned} \frac{\partial u}{\partial t} &= -uv^2 + \alpha(1-u) + d_u\Delta u \\ \frac{\partial v}{\partial t} &= uv^2 - (\alpha+\beta)v + d_v\Delta v \end{aligned} \qquad (x,y) \in \Omega, t > 0, \tag{3.12}$$

where $\alpha = .05, \beta = .02, d_u = 2\times 10^{-5}, d_v = 10^{-5}$ and Ω a hemisphere with radius 1. Section 3.3.3 illustrates the solution u at $t = 1500$.

Using a standard Galerkin approximation in the spatial dimensions and an IMEX scheme [66] for the temporal dimension, one arrives at the follow linear systems:

$$\begin{aligned}\frac{\mathbf{M}\vec{u}^{n+1} - \mathbf{M}\vec{u}^{n}}{\Delta t} &= -\vec{g}^{n} + \alpha\vec{1} - \alpha\mathbf{M}\vec{u}^{n+1} - \frac{d_u}{2}\left(\mathbf{S}u^{n+1} + \mathbf{S}u^{n}\right)\\ \frac{\mathbf{M}\vec{v}^{n+1} - \mathbf{M}\vec{v}^{n}}{\Delta t} &= \vec{g}^{n} - (\alpha+\beta)\mathbf{M}\vec{v}^{n+1} - \frac{d_v}{2}\left(\mathbf{S}v^{n+1} + \mathbf{S}v^{n}\right)\end{aligned} \tag{3.13}$$

where $\vec{u}^n, \vec{v}^n$ is the finite element approximation at time step n and $\vec{g}^n$ is the nonlinear moment associated with uv^2 at time step n.

The first equation of Equation (3.13) involves inverting the matrix $\mathbf{A}\left(1 + \alpha\Delta t, d_u/2\right)$ at each time step. In this example $\mu \gg \nu$ whilst numerical evidence suggests that the constant $C_S < \frac{1}{5}$ [58], hence the constant in front of the $\mathcal{O}(p^4)$ growth in Equation (3.11) is quite small. In Section 3.3.3 we show the condition number of the preconditioned system $\tilde{\mathbf{A}}\left(1 + \alpha\Delta t, d_u/2\right)$ with different Δt and order p. In practice, one generally chooses Δt depending on p, but for illustrative purposes here, we vary Δt and p independently. Note that the condition number for $p \leq 10$ does not yet attain the asymptotic $\mathcal{O}(p^4)$ growth even for incredibly large values of Δt. Results presented for the case $\Delta t = 5$ also exhibits a transition from constant condition number to a slight growth with p as predicted by Equation (3.11).

3.3.4 Applicability to Other Types of Basis

The discussion thus far might leave the reader with the (false) impression that our preconditioner is only applicable provided one chooses the basis presented in Section 3.2.1. This is not the case and the preconditioner is applicable to any choice of basis. Indeed, our preconditioner can be regarded as an Additive Schwarz method (ASM) [19, 77].

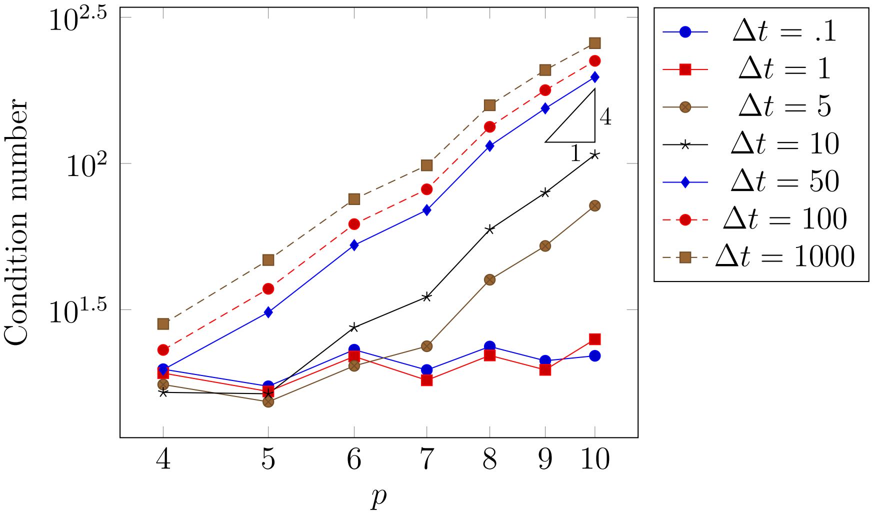


Figure 3.4: Figure illustrating the condition number of the preconditioned system arising from the discretization of the reaction-diffusion system on the hemisphere consisting of 60 elements. Note that we do not yet observe the $\mathcal{O}(p^4)$ growth for $p \leq 10$ even for very large Δt.

The ASM is defined by the following subspace decomposition:

$$X = X_I \oplus \bigoplus_{k=1}^{4} X_{F_k} \oplus \bigoplus_{k=1}^{6} X_{E_k} \oplus \bigoplus_{k=1}^{4} X_{V_k},$$

where, in the parlance of ASM methods, an exact solver is used on each subspace. Specifically, given a residual $f \in X$, the action of the ASM is defined as follows:

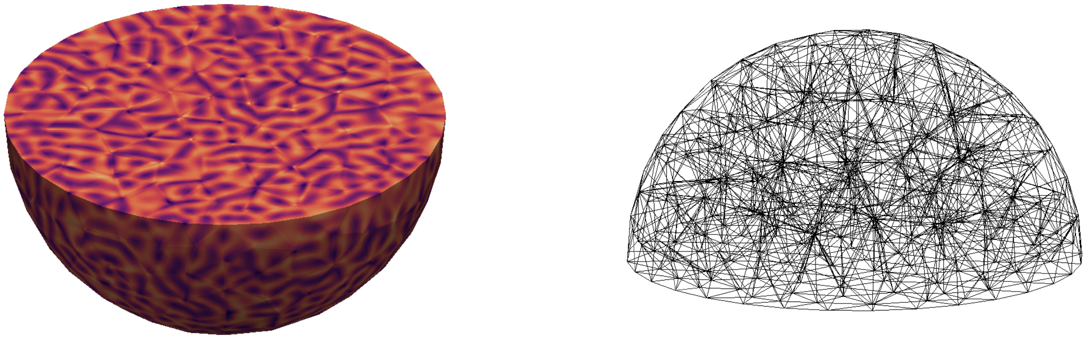


Figure 3.5: Plot of u from above in the Gray-Scott equations Equation (3.12) with $p = 6$ on a mesh of the hemisphere with 1159 elements at $t = 1500$ with $\Delta t = 1$.

- $u_I \in X_I : (u_I, v_I) = (f, v_I) \quad \forall v_I \in X_I,$
- $u_{F_k} \in X_{F_k} : (u_{F_k}, v_{F_k}) = (f, v_{F_k}) \quad \forall v_{F_k} \in X_{F_k},$
- $u_{E_k} \in X_{E_k} : (u_{E_k}, v_{E_k}) = (f, v_{E_k}) \quad \forall v_{E_k} \in X_{E_k},$
- $u_{V_k} \in X_{V_k} : (u_{V_k}, v_{V_k}) = (f, v_{V_k}) \quad \forall v_{V_k} \in X_{V_k},$

and returns $u := u_I + \sum_{k=1}^{4} u_{F_k} + \sum_{k=1}^{6} u_{E_k} + \sum_{k=1}^{4} u_{V_k}$. The proof that the ASM gives rise to an uniform bound on the condition number follows from the fact that the constants c, C in Equation (3.8) are independent of p [77, Theorem 2.7].

The action of the preconditioner for a general choice of basis begins by statically condensing out the interior degrees of freedom. Lemma 3.4.3 states that X_I is L^2 orthogonal to the remaining subspaces:

$$X_I \perp \bigoplus_{k=1}^{4} X_{F_k} \oplus \bigoplus_{k=1}^{6} X_{E_k} \oplus \bigoplus_{k=1}^{4} X_{V_k}$$

which means that one can first reduce the system to the Schur complement matrix. Once the Schur complement is in hand, a change of basis can be applied on the interface to map to the spaces X_{F_k}, X_{E_k} and X_{V_k} corresponds to the preconditioner presented here. Further details in the 2D setting can be found in [9].

3.4 Technical Lemmas

In this section, we turn to the proof of the technical results which were used in proving Theorem 2.3.1.

3.4.1 Orthogonality

The Duffy transformation [47, §3.2] given by

$$\xi := \frac{2\lambda_2}{1-\lambda_3-\lambda_4} - 1, \qquad \eta := \frac{2\lambda_3}{1-\lambda_4} - 1, \qquad \theta := 2\lambda_4 - 1$$

maps the reference tetrahedron T onto the cube $\{(\xi,\eta,\theta) : -1 \le \xi,\eta,\theta \le 1\}$. For reference, the edge $E_1 = \{(\xi,\eta,\theta) : -1 \le \xi \le 1, \eta = -1, \theta = -1\}$ and the face $F_1 = \{(\xi,\eta,\theta) : -1 \le \xi,\eta \le 1, \theta = -1\}$.

We begin by establishing the orthogonality properties of the basis functions:

Lemma 3.4.1. *The functions $\{\omega_{ijk}\}, \{\psi_{ij}^{(k)}\}, \{\chi_i^{(k)}\}$ provide an L^2-orthogonal basis for X_I, X_{F_k}, X_{E_k} respectively.*

Proof. It suffices to show that

$$(\omega_{i_1j_1k_1}, \omega_{i_2j_2k_2}) \propto \delta_{i_1j_1k_1,i_2j_2k_2}, \quad (\psi_{i_1j_1}^{(1)}, \psi_{i_2j_2}^{(1)}) \propto \delta_{i_1j_1,i_2j_2}, \quad (\chi_{i_1}^{(1)}, \chi_{i_2}^{(1)}) \propto \delta_{i_1,i_2}.$$

Transforming the basis functions using the Duffy transformation gives

$$\begin{aligned}
\omega_{ijk} &= \frac{1-\xi}{2}\frac{1+\xi}{2}P_i^{(2,2)}(\xi)\left(\frac{1-\eta}{2}\right)^{i+2}\frac{1+\eta}{2}P_j^{(2i+5,2)}(\eta) \\
&\qquad \times \left(\frac{1-\theta}{2}\right)^{i+j+3}\frac{1+\theta}{2}P_k^{(2i+2j+8,2)}(\theta), \\
\psi_{ij}^{(1)} &= \frac{1-\xi}{2}\frac{1+\xi}{2}P_i^{(2,2)}(\xi)\left(\frac{1-\eta}{2}\right)^{i+2}\frac{1+\eta}{2}P_j^{(2i+5,2)}(\eta) \\
&\qquad \times \left(\frac{1-\theta}{2}\right)^{i+j+3}\Phi_{p-3-i-j}^{(2i+2j+8)}(\theta), \\
\chi_i^{(1)} &= \frac{1-\xi}{2}\frac{1+\xi}{2}P_i^{(2,2)}(\xi)\left(\frac{1-\eta}{2}\right)^{i+2}\left(\frac{1-\theta}{2}\right)^{i+2}F(\eta,\theta)
\end{aligned}$$

where $F(\eta, \theta)$ is a polynomial in η and θ.

The Jacobian of the Duffy transformation is given by

$$J = \frac{1-\eta}{2}\left(\frac{1-\theta}{2}\right)^2,$$

and, as a consequence, we find

$$\begin{aligned}\int_T \omega_{i_1 j_1 k_1}\omega_{i_2 j_2 k_2}\,dx &= \int_{-1}^{1}\left(\frac{1-\xi}{2}\right)^2\left(\frac{1+\xi}{2}\right)^2 P_{i_1}^{(2,2)}P_{i_2}^{(2,2)}\,d\xi \\ &\times \int_{-1}^{1}\left(\frac{1-\eta}{2}\right)^{i_1+i_2+5}\left(\frac{1+\eta}{2}\right)^2 P_{j_1}^{(2i_1+5,2)}P_{j_2}^{(2i_2+5,2)}\,d\eta \\ &\times \int_{-1}^{1}\left(\frac{1-\theta}{2}\right)^{i_1+i_2+j_1+j_2+8}\left(\frac{1+\theta}{2}\right)^2 P_{k_1}^{(2i_1+2j_1+8,2)}P_{k_2}^{(2i_2+2j_2+8,2)}\,d\theta \\ &= C\delta_{i_1,i_2}\delta_{j_1,j_2}\delta_{k_1,k_2}.\end{aligned}$$

The result for the edge $\psi_{ij}^{(1)}$ and face $\chi_i^{(1)}$ functions follows the same lines. □

The next lemma enumerates the pertinent properties of the function $\Phi_p^{(m)}$ which was used in several places in defining the basis functions:

Lemma 3.4.2. *For non-negative integers m, q, the polynomial defined in Equation* (3.1) *has the following properties:*

1. $\Phi_q^{(m)}(-1) = 1$,
2. *Weighted norm*

$$I_{m,q} := \int_{-1}^{1}\left(\frac{1-x}{2}\right)^m\left(\Phi_q^{(m)}(x)\right)^2\,dx = \frac{2}{(q+1)(m+q+1)}, \tag{3.14}$$

3. Orthogonality property

$$\int_{-1}^{1}\left(\frac{1-x}{2}\right)^{m}\frac{1+x}{2}\Phi_q^{(m)}(x)w(x)\,dx = 0$$

for all $w \in \mathbb{P}_r([-1,1])$ with $r < q$.

Proof. The first property comes from the fact that $P_q^{(m,1)}(-1) = (-1)^q\binom{q+1}{q}$ [1, §22.2.1], and the third property follows straight from the orthogonality property of $P_q^{(m,1)}$. For the second result, relation (22.7.19) in [1] gives us

$$\frac{2q+m+1}{q+m+1}P_q^{(m,0)} - \frac{q+m}{q+m+1}P_{q-1}^{(m,1)} = P_q^{(m,1)}.$$

Equation Equation (3.14) in the case of $q = 0$ trivially holds. Suppose that Equation (3.14) holds in the case of $q-1$, then

$$\begin{aligned}
I_{m,q} &= \frac{1}{(q+1)^2}\int_{-1}^{1}\left(\frac{1-x}{2}\right)^{m} P_q^{(m,1)}(x)P_q^{(m,1)}(x)\,dx \\
&= \frac{1}{(q+1)^2}\int_{-1}^{1}\left(\frac{1-x}{2}\right)^{m}\left(\frac{(2q+m+1)^2}{(q+m+1)^2}P_q^{(m,0)}(x)P_q^{(m,0)}(x)\right)\,dx \\
&\quad + \frac{1}{(q+1)^2}\frac{(q+m)^2}{(q+m+1)^2}q^2 I_{m,q-1} \\
&= \frac{1}{(q+1)^2}\frac{(2q+m+1)^2}{(q+m+1)^2}\frac{2}{2q+m+1} + \frac{1}{(q+1)^2}\frac{(q+m)^2}{(q+m+1)^2}q^2\frac{2}{q(m+q)} \\
&= \frac{2}{(q+1)(q+m+1)}
\end{aligned}$$

and the result Equation (3.14) holds by induction.

□

The above result implies that the interior basis functions are orthogonal to the face/edge/vertex functions:

Lemma 3.4.3. *Let $X_B = \bigoplus_{k=1}^4 X_{F_k} \oplus \bigoplus_{k=1}^6 X_{E_k} \oplus \bigoplus_{k=1}^4 X_{V_k}$, then the space X can be decomposed as $X = X_I \oplus X_B$ such that $X_I \perp X_B$.*

Proof. Recall Ξ_i, q_i and q from Equations (3.2) to (3.4) respectively, and define $\bar{\chi}_i^{(1)}, \bar{\varphi}_1$ as

$$\begin{aligned}
\bar{\chi}_i^{(1)} &:= \lambda_1\lambda_2\Xi_i(\lambda_1,\lambda_2)q_i(\lambda_3,\lambda_4)\\
&= \frac{1-\xi}{2}\frac{1+\xi}{2}P_i^{(2,2)}(\xi)\left(\frac{1-\eta}{2}\right)^{i+2}\Phi_j^{(2i+5)}(\eta)\left(\frac{1-\theta}{2}\right)^{i+j+2}\Phi_{p-2-i-j}^{(2i+2j+6)}(\theta),\\
\bar{\varphi}_1 &:= \lambda_1 q(\lambda_2,\lambda_3,\lambda_4)\\
&= \frac{1-\xi}{2}\Phi_i^{(2)}(\xi)\left(\frac{1-\eta}{2}\right)^{i+1}\Phi_j^{(2i+3)}(\eta)\left(\frac{1-\theta}{2}\right)^{i+j+1}\Phi_{p-1-i-j}^{(2i+2j+4)}(\theta).
\end{aligned} \tag{3.15}$$

By permutation of the barycentric coordinates, it suffices to show that for any interior basis function ω_{lmn} with $0 \le l, m, n, l+m+n \le p-4$, the inner product vanishes

$$\begin{aligned}
(\bar{\varphi}_1, \omega_{lmn}) &= 0,\\
(\bar{\chi}_i^{(1)}, \omega_{lmn}) &= 0, \quad i = 0, \ldots, p-2,\\
(\psi_{ij}^{(1)}, \omega_{lmn}) &= 0, \quad 0 \le i, j, i+j \le p-3.
\end{aligned}$$

Calculating the inner-product for the face functions first:

$$\begin{aligned}(\psi_{ij}^{(1)},\omega_{lmn}) &= \int_{-1}^{1}\left(\frac{1-\xi}{2}\right)^{2}\left(\frac{1+\xi}{2}\right)^{2} P_i^{(2,2)}(\xi)P_l^{(2,2)}(\xi)\,d\xi \\ &\quad\times\int_{-1}^{1}\left(\frac{1-\eta}{2}\right)^{i+l+5}\left(\frac{1+\eta}{2}\right)^{2} P_j^{(2i+5,2)}(\eta)P_m^{(2l+5,2)}(\eta)\,d\eta \\ &\quad\times\int_{-1}^{1}\left(\frac{1-\theta}{2}\right)^{i+l+j+m+8}\left(\frac{1+\theta}{2}\right)\Phi_{p-3-i-j}^{(2i+2j+8)}(\theta)P_n^{(2l+2m+8,2)}(\theta)\,d\theta \\ &\propto \delta_{il}\delta_{jm}\int_{-1}^{1}\left(\frac{1-\theta}{2}\right)^{2i+2j+8}\left(\frac{1+\theta}{2}\right)\Phi_{p-3-i-j}^{(2i+2j+8)}(\theta)P_n^{(2l+2m+8,2)}(\theta)\,d\theta.\end{aligned}$$

The inner-product vanishes if $i \neq l, j \neq m$. Assuming otherwise, we have that $p-3-i-j > n$ as $l+m+n \leq p-4$, hence the inner-product is 0 by Lemma 3.4.2.

For the edges, we have

$$\begin{aligned}(\bar{\chi}_i^{(1)},\omega_{lmn}) &\propto \delta_{il}\int_{-1}^{1}\left(\frac{1-\eta}{2}\right)^{i+l+5}\frac{1+\eta}{2}P_j^{(2i+5,1)}(\eta)P_m^{(2l+5,2)}(\eta)\,d\eta \\ &\quad\times\int_{-1}^{1}\left(\frac{1-\theta}{2}\right)^{i+j+l+m+7}\frac{1+\theta}{2}P_{p-2-i-j}^{(2i+2j+6,1)}(\theta)P_n^{(2l+2m+8,2)}(\theta)\,d\theta.\end{aligned}$$

The inner product is trivially zero if $i \neq l$ or $m < j$. Assuming otherwise, we have for the θ variable

$$\int_{-1}^{1}\left(\frac{1-\theta}{2}\right)^{2i+2j+6}\frac{1+\theta}{2}\left[\left(\frac{1-\theta}{2}\right)^{1+m-j}P_n^{(2l+2m+8,2)}(\theta)\right]P_{p-2-i-j}^{(2i+2j+6,1)}(\theta)\,d\theta.$$

The above vanishes if

$$1+m-j+n < p-2-i-j$$

which follows from the fact that $l+m+n \leq p-4$.

Finally, we have

$$(\bar{\varphi}_1, \omega_{lmn}) \propto \int_{-1}^{1} \left(\frac{1-\xi}{2}\right)^2 \frac{1+\xi}{2} P_i^{(2,1)}(\xi) P_l^{(2,2)}(\xi)\, d\xi$$
$$\int_{-1}^{1} \left(\frac{1-\eta}{2}\right)^{i+l+4} \frac{1+\eta}{2} P_j^{(2i+3,1)}(\eta) P_m^{(2l+5,2)}(\eta)\, d\eta$$
$$\int_{-1}^{1} \left(\frac{1-\theta}{2}\right)^{i+j+l+m+6} \frac{1+\theta}{2} P_k^{(2i+2j+4,1)}(\theta) P_l^{(2l+2m+8,2)}(\theta)\, d\theta.$$

If $i > l$, then there is nothing to prove, otherwise the η integral can be written as

$$\int_{-1}^{1} \left(\frac{1-\eta}{2}\right)^{2i+3} \frac{1+\eta}{2} \left[\left(\frac{1-\eta}{2}\right)^{1+l-i} P_m^{(2l+5,2)}(\eta)\right] P_j^{(2i+3,1)}(\eta)\, d\eta$$

which vanishes if $j > 1+l-i+m$. Finally, assuming otherwise, the θ integral can be written as

$$\int_{-1}^{1} \left(\frac{1-\theta}{2}\right)^{2i+2j+4} \frac{1+\theta}{2} \left[\left(\frac{1-\theta}{2}\right)^{l+m-i-j+2} P_n^{(2l+2m+8,2)}(\theta)\right] P_{p-1-i-j}^{(2i+2j+4,1)}(\theta)\, d\theta.$$

The above quantity vanishes if

$$l+m-i-j+2+n < p-1-i-j$$

which follows from the fact that $l+m+n \leq p-4$. □

Now we show the stability of the subspace decomposition.

3.4.2 Vertex Contributions

The following lemma corresponds to Lemma 5.4 and 6.1 of [8] and allows us to bound the vertex contribution:

Lemma 3.4.4. *The vertex basis functions of degree p satisfy the bound*

$$cp^{-3} \leq \|\varphi\| \leq Cp^{-3}$$

for constants c, C independent of p.

Proof. Note that

$$\begin{aligned}\|\varphi_1\| &= \|\bar{\varphi}_1/3 + \lambda_1 q(\lambda_3, \lambda_4, \lambda_2)/3 + \lambda_1 q(\lambda_4, \lambda_2, \lambda_3)/3\| \\ &\leq \|\bar{\varphi}_1/3\| + \|\lambda_1 q(\lambda_3, \lambda_4, \lambda_2)/3\| + \|\lambda_1 q(\lambda_4, \lambda_2, \lambda_3)/3\| = \|\bar{\varphi}_1\|\end{aligned}$$

where $\bar{\varphi}_1$ is defined in Equation (3.15).

Using Lemma 3.4.2,

$$\begin{aligned}\|\bar{\varphi}\|^2 &= \int_{-1}^{1} \frac{(1-\xi)^2}{4} \Phi_i^{(2)}\, d\xi \int_{-1}^{1} \left(\frac{1-\eta}{2}\right)^{2i+3} \Phi_j^{(2i+3)}\, d\eta \\ &\times \int_{-1}^{1} \left(\frac{1-\theta}{2}\right)^{2i+2j+4} \Phi_{p-1-i-j}^{(2i+2j+4)}\, d\theta \\ &= \frac{8}{(i+1)(i+3)(j+1)(2i+j+4)(p-i-j)(i+j+p+4)} \leq Cp^{-6}.\end{aligned}$$

For the lower bound, let $0 \leq i, j, k, i+j+k \leq p$ and define

$$\Psi_{ijk} := c_{ijk} P_i^{(0,0)}(\xi) \left(\frac{1-\eta}{2}\right)^{i} P_j^{(2i+1,0)}(\eta) \left(\frac{1-\theta}{2}\right)^{i+j} P_k^{(2i+2j+2,0)}(\theta), \tag{3.16}$$

where $c_{ijk} = \frac{1}{2}\sqrt{(2i+1)(i+j+1)(2i+2j+2k+3)}$. These functions form an orthonormal basis for X hence φ can be written in the form $\varphi = \sum_{i+j+k\leq p} u_{ijk}\Psi_{ijk}$ where u_{ijk} are the appropriate coefficients and $\|\varphi\|^2 = \sum_{i+j+k\leq p} u_{ijk}^2$. It suffices to prove the inequality in the case of φ_1. Cauchy-Schwarz gives

$$
\begin{aligned}
1 = |\varphi(-1,-1,-1)|^2 &= \left(\sum_{i+j+k \leq p} (-1)^{i+j+k} c_{ijk} u_{ijk} \right)^2 \\
&\leq \sum_{i+j+k \leq p} u_{ij}^2 \sum_{i+j+k \leq p} c_{ijk}^2 = \frac{(p+1)^2(p+2)^2(p+3)^2}{48} \|\varphi\|^2 .
\end{aligned}
$$

□

We now proceed to the edge contributions.

3.4.3 Edge contributions

The following lemma bounds the contribution on an edge:

Lemma 3.4.5. *Let $u \in X$ be such that u vanishes at the vertices of T. Let γ be an arbitrary edge of T and let $U \in X_{E_\gamma}$ such that $U|_\gamma = u|_\gamma$. Then there exists a constant C independent of p such that*

$$
\|U\| \leq C\|u\| . \tag{3.17}
$$

Proof. Without loss of generality, we assume that $\gamma := E_1$. Let $U = \sum_{i=0}^{p-2} w_i \chi_i^{(1)}$ where the coefficients w_i are chosen such that $U|_\gamma = u|_\gamma$. It is more convenient to work with the function $\bar{\chi}_i^{(1)}$ defined in Equation (3.15). Observe that $\bar{\chi}_i^{(1)}|_{E_1} = \chi_i^{(1)}|_{E_1}$, and $(\bar{\chi}_i^{(1)}, \bar{\chi}_j^{(1)}) \propto \delta_{ij}$. Let $\bar{U} = \sum_{i=0}^{p-2} w_i \bar{\chi}_i^{(1)}$, then $\bar{U} = U$ on edge γ and $\|U\| \leq \left\|\bar{U}\right\|$ as

$$
\begin{aligned}
\left\|\chi_i^{(1)}\right\| &= \left\|\bar{\chi}_i^{(1)}/2 + \lambda_1 \lambda_2 p_i(\lambda_2 - \lambda_1) q_j(\lambda_4, \lambda_3)/2\right\| \\
&\leq \left\|\bar{\chi}_i^{(1)}/2\right\| + \left\|\lambda_1 \lambda_2 p_i(\lambda_2 - \lambda_1) q_j(\lambda_4, \lambda_3)/2\right\| = \left\|\bar{\chi}_i^{(1)}\right\| ,
\end{aligned}
$$

thus it suffices to show that $\left\|\bar{U}\right\| \leq C\|u\|$.

To this end, recall the orthonormal basis Ψ_{ijk} defined in Equation (3.16), then we may write $u = \sum_{i+j+k \leq p} u_{ijk}\Psi_{ijk}$. Let

$$f := u|_\gamma = \sum_{i=0}^{p} v_i P_i^{(0,0)}(x)$$

where

$$v_i := \sum_{j=0}^{p-i} \sum_{k=0}^{p-i-j} \frac{(-1)^{j+k}}{2} u_{ijk} \sqrt{(2i+1)(i+j+1)(2i+2j+2k+3)}, \tag{3.18}$$

Furthermore, since u vanishes at the vertices of T, then $f(\pm 1) = 0$ thus

$$\sum_{i=0,\text{even}}^{p} v_i = 0, \qquad \sum_{i=1,\text{odd}}^{p} v_i = 0. \tag{3.19}$$

Consequently, we can rewrite $f = \sum_{i=2}^{p}(P_i^{(0,0)} - P_{i-2}^{(0,0)})S_i$ where

$$S_i = v_i + v_{i+2} + \cdots + \begin{cases} v_p \\ v_{p-1} \end{cases} = \begin{cases} -v_0 - \cdots - v_{i-2} \text{ if } i \text{ even} \\ -v_1 - \cdots - v_{i-2} \text{ else} \end{cases}$$

depending on the parity.

Turning to the coefficients w_i, we must have on edge γ

$$\bar{U}|_\gamma = \frac{1-\xi}{2}\frac{1+\xi}{2}\sum_{i=0}^{p-2} w_i P_i^{(2,2)}(\xi) = \sum_{i=2}^{p}(P_i^{(0,0)} - P_{i-2}^{(0,0)})S_i$$

Recall the following identity from Lemma 6.6 of [8]

$$-\frac{1-x^2}{2(n-1)}\left(\frac{(n+1)(n+2)}{2n}P_{n-2}^{(2,2)} - \frac{n-1}{2}P_{n-4}^{(2,2)}\right) = P_n^{(0,0)} - P_{n-2}^{(0,0)}, \qquad n \geq 2$$

where P_{n-4} is understood to be 0 for $n<4$, then we have

$$\sum_{i=0}^{p-2} w_i P_i^{(2,2)} = \sum_{i=2}^{p} \left(-\frac{(i+1)(i+2)}{i(i-1)} P_{i-2}^{(2,2)} + P_{i-4}^{(2,2)} \right) S_i$$

and we deduce by matching coefficients that

$$\begin{aligned} w_i &= S_{i+4} - \frac{(i+3)(i+4)}{(i+1)(i+2)} S_{i+2} \\ &= -v_{i+2} - \frac{2(5+2i)}{(i+1)(i+2)} S_{i+2}. \end{aligned} \tag{3.20}$$

With Equation (3.20) in hand, we can now analyze $\left\|\bar{U}\right\|$ and $\|u\|$. The Cauchy-Schwarz inequality applied to Equation (3.18) gives

$$\begin{aligned} v_i^2 &\leq \sum_{j=0}^{p-i} \sum_{k=0}^{p-i-j} u_{ijk}^2 \sum_{j=0}^{p-i} \sum_{k=0}^{p-i-j} \frac{(2i+1)(i+j+1)(2i+2j+2k+3)}{4} \\ &= \frac{1}{16}(2i+1)(i-p-2)(i-p-1)(i+p+2)(i+p+3) \sum_{j=0}^{p-i} \sum_{k=0}^{p-i-j} u_{ijk}^2, \end{aligned}$$

hence, rearranging and summing over the index i, we have a lower bound for $\|u\|$

$$\begin{aligned} &\sum_{i=0}^{p} \frac{16 v_i^2}{(2i+1)(i-p-2)(i-p-1)(i+p+2)(i+p+3)} \\ &\qquad \approx \sum_{i=0}^{p} \frac{v_i^2}{(i+1)(i-p-1)^2(i+p+1)^2} \leq \|u\|^2 . \end{aligned} \tag{3.21}$$

Using Lemma 3.4.2, the fact that $j = \lfloor \frac{p-i-2}{2} \rfloor$, and Cauchy-Schwarz on Equation (3.20) gives

$$
\begin{aligned}
\left\|\bar{U}\right\|^2 &= \sum_{i=0}^{p-2} \frac{2(i+1)(i+2)w_i^2}{(i+3)(i+4)(2i+5)} \frac{2}{(j+1)(2i+j+6)} \frac{2}{(p-i-j-1)(i+j+p+5)} \\
&\approx \sum_{i=0}^{p-2} \frac{w_i^2}{(i+1)} \frac{1}{(p-i+1)(p+i+1)} \frac{1}{(p-i+1)(i+p+1)} \\
&\leq C\left(\sum_{i=0}^{p-2} \frac{v_{i+2}^2}{(i+1)(p-i+1)^2(p+i+1)^2} + \frac{S_{i+2}^2}{(i+1)^3(p-i+1)^2(p+i+1)^2}\right).
\end{aligned}
$$

The first term is bounded easily by using Equation (3.21)

$$
\sum_{i=0}^{p-2} \frac{v_{i+2}^2}{(i+1)(p-i+1)^2(p+i+1)^2} \leq C \sum_{i=0}^{p} \frac{v_i^2}{(i+1)(i-p-1)^2(i+p+1)^2} \leq C\|u\|^2 .
$$

Hence, the theorem follows if there exists a constant C independent of p such that

$$
\sum_{i=0}^{p-2} \frac{S_{i+2}^2}{(i+1)^3(p-i+1)^2(p+i+1)^2} \leq C \sum_{i=0}^{p} \frac{v_i^2}{(i+1)(i-p-1)^2(i+p+1)^2},
$$

but this follows by applying Lemma 3.4.10 with $j = 2$. □

3.4.4 Face contributions

Finally, it remains to show that the face contributions are bounded. Let F be an arbitrary face of T, and let S be a subset of the remaining faces of T. We remark that $S \cup F$ need not necessarily coincide with the set of *all* faces of T. Let $Y_F = \{u \in X : u = 0 \text{ on all the edges of } F\}$, and define the operator $\mathcal{E}_{S,F} : Y_F \mapsto Y_F$ by

$$
\mathcal{E}_{S,F} u = \underset{\substack{v|_F = u|_F \\ v|_S = 0 \\ v \in Y_F}}{\operatorname{argmin}} \|v\|^2 . \tag{3.22}
$$

Existence to the minimization problem is trivial, while uniqueness comes from the strict convexity of the squared L^2 norm. Clearly,

$$\left\|\mathcal{E}_{S\setminus F',F}u\right\| \leq \left\|\mathcal{E}_{S,F}u\right\|, \quad \forall F' \subset S$$

since $\mathcal{E}_{S,F}u = u$ on F and also vanishes on $S \setminus F'$. The proof that the converse inequality is independent of p is less obvious:

Lemma 3.4.6. *Let F be an arbitrary face of T, and let S be a subset of the remaining faces of T. There exists a constant C independent of p such that*

$$\left\|\mathcal{E}_{S,F}u\right\| \leq C\left\|\mathcal{E}_{S\setminus F',F}u\right\|, \qquad \forall u \in Y_F,$$

for all $F' \subset S$.

Before giving the proof, we note the following consequence of Lemma 3.4.6 which was used in the proof of Theorem 2.3.1:

Corollary 3.4.7. *Let F_i be any face of T and $u \in Y_{F_i}$, then there exists a polynomial $U \in X_{F_i}$ such that $U|_{F_i} = u|_{F_i}$ and*

$$\|U\| \leq C\|u\|$$

where C is independent of p.

Proof. Choosing $S = \partial T \setminus F_i$, $F' = S$, and let $U = \mathcal{E}_{S,F_i}u$. Clearly, $U \in X_{F_i}$ as U vanishes on S the three remaining faces. Furthermore, Lemma 3.4.6 gives the bound

$$\|U\| = \left\|\mathcal{E}_{S,F_i}u\right\| \leq C\left\|\mathcal{E}_{S\setminus F',F_i}u\right\| \leq C\|u\|.$$

□

All that remains is to prove Lemma 3.4.6; to this end, for $l, m, n \in \{0,1\}$ define the polynomials

$$\begin{aligned}\zeta_{ij}^{(l,m,n)} = &\left(\frac{1-\xi}{2}\right)^m \left(\frac{1+\xi}{2}\right)^n P_i^{(2m,2n)}(\xi) \left(\frac{1-\eta}{2}\right)^{i+m+n} \left(\frac{1+\eta}{2}\right)^l \\ &\times P_j^{(2i+2m+2n+1,2l)}(\eta) \left(\frac{1-\theta}{2}\right)^{j+i+m+n+l} \Phi_{p-i-j-m-n-l}^{(2(j+i+m+n+l)+2)}(\theta)\end{aligned} \tag{3.23}$$

for $0 \le i, j, i+j \le p-l-m-n$.

Lemma 3.4.8. *The following properties hold:*

1. $\zeta_{ij}^{(l,m,n)} \in X$,
2. $\zeta_{ij}^{(1,1,1)}$ *vanishes on* $\{\xi = \pm 1, \eta = 1\}$, $\zeta_{ij}^{(0,1,1)}$ *vanishes on* $\{\xi = \pm 1\}$ *etc.,*
3. $\zeta_{ij}^{(1,1,1)} = \psi_{ij}^{(1)}$, *our face basis functions,*
4. $\{\zeta_{ij}^{(l,m,n)}\}$ *are orthogonal on* T *for a fixed* l, m, n,
5. $\{\zeta_{ij}^{(l,m,n)}|_{F_1}\}$ *spans* $\mathbb{P}_p(F_1) \cap H_0^1(F_1)$.

Proof. The first three statements can be deduced by inspection. For the orthogonality property, we note that

$$\begin{aligned}(\zeta_{i_1j_1}^{(l,m,n)}, \zeta_{i_2j_2}^{(l,m,n)}) \propto F(\theta) \int_{-1}^{1} \left(\frac{1-\xi}{2}\right)^{2m} \left(\frac{1+\xi}{2}\right)^{2n} P_{i_1}^{(2m,2n)} P_{i_2}^{(2m,2n)}\, d\xi \\ \times \int_{-1}^{1} \left(\frac{1-\eta}{2}\right)^{i_1+i_2+2m+2n+1} \left(\frac{1+\eta}{2}\right)^{2l} P_{j_1}^{(2i_1+2m+2n+1,2l)} P_{j_2}^{(2i_2+2m+2n+1,2l)}\, d\eta.\end{aligned}$$

The quantity vanishes if $i_1 \ne i_2$ or $j_1 \ne j_2$.

The last statement follows from linear independence, and recognizing that the restriction of the 3D Duffy transformation onto F_1 reduces to the 2D Duffy transformation. □

The following lemma gives an explicit expression for the operator $\mathcal{E}_{S,F}$ defined in Equation (3.22):

Lemma 3.4.9. *Let $u \in Y_{F_1}$ then*

$$\mathcal{E}_{S,F_1} u = \sum_{i+j \leq p-l-m-n} u_{ij}^{(l,m,n)} \zeta_{ij}^{(l,m,n)} \tag{3.24}$$

where $u_{ij}^{(l,m,n)}$ are determined by the condition

$$\sum_{i+j \leq p-l-m-n} u_{ij}^{(l,m,n)} \zeta_{ij}^{(l,m,n)}(\xi, \eta, -1) = u(\xi, \eta, -1) \tag{3.25}$$

and the coefficients l, m, n are given by one of the following conditions depending on S:

1. $S = \{\xi = -1\} \cup \{\xi = 1\} \cup \{\eta = -1\}$, $m = n = l = 1$.
2. $S = \{\xi = -1\} \cup \{\xi = 1\}$, $m = n = 1, l = 0$.
3. $S = \{\xi = -1\} \cup \{\eta = -1\}$, $m = 1, n = 0, l = 1$.
4. $S = \{\xi = 1\} \cup \{\eta = -1\}$, $m = 0, n = l = 1$.
5. $S = \{\xi = -1\}$, $m = 1, n = l = 0$.
6. $S = \{\eta = -1\}$, $m = n = 0, l = 1$.
7. $S = \{\xi = 1\}$, $m = 0, n = 1, l = 0$.
8. $S = \emptyset$, $m = n = l = 0$.

Proof. Clearly, the coefficients $u_{ij}^{(l,m,n)}$ are uniquely defined by Equation (3.25) thanks to properties 4 and 5 of Lemma 3.4.8. For the sake of notation, we will drop the (l, m, n) notation in the remainder of the proof. It suffices to show that the right hand side of Equation (3.24) solves the minimization problem Equation (3.22).

By statement 4 of Lemma 3.4.8, and statement 2 of Lemma 3.4.2, we can calculate

$$\begin{aligned}\left\|\sum_{i+j\leq p-l-m-n} u_{ij}\zeta_{ij}\right\|^2 &= \sum_{i+j\leq p-l-m-n} u_{ij}^2\left\|\zeta_{ij}\right\|^2 \\ = \sum_{i+j\leq p-l-m-n} u_{ij}^2\mu_i\nu_j & \frac{2}{(p-i-j-m-n-l+1)(p+i+j+m+n+l+3)}\end{aligned} \tag{3.26}$$

where

$$\begin{aligned}\mu_i &= \int\left(\frac{1-x}{2}\right)^{2m}\left(\frac{1+x}{2}\right)^{2n}(P_i^{(2m,2n)})^2\,dx \\ \nu_j &= \int\left(\frac{1-x}{2}\right)^{2i+2m+2n+1}\left(\frac{1+x}{2}\right)^{2l}(P_j^{(2i+2m+2n+1,2l)})^2\,dx.\end{aligned}$$

We will show below that $\left\|\mathcal{E}_{S,F_1}u\right\|^2$ equals the above quantity Equation (3.26).

For $i+j+k\leq p-l-m-n$, let

$$\begin{aligned}\Psi_{ijk} := &\left(\frac{1-\xi}{2}\right)^m\left(\frac{1+\xi}{2}\right)^n P_i^{(2m,2n)}(\xi)\left(\frac{1-\eta}{2}\right)^{i+m+n}\left(\frac{1+\eta}{2}\right)^l \\ &\times P_j^{(2i+2m+2n+1,2l)}(\eta)\left(\frac{1-\theta}{2}\right)^{i+m+n+l+j} P_k^{(2(i+m+n+l+j)+2,0)}(\theta).\end{aligned}$$

By construction, Ψ_{ijk} vanish on S and are orthogonal to each other, hence there exists coefficients $\widetilde{u}_{ijk}$ such that $\mathcal{E}_{S,F_1}u = \sum_{i+j+k\leq p-m-n-l}\widetilde{u}_{ijk}\Psi_{ijk}$ with

$$\left\|\mathcal{E}_{S,F_1}u\right\|^2 = \sum_{i+j+k\leq p-m-n-l}\widetilde{u}_{ijk}^2\mu_i\nu_j\rho_k$$

where

$$\rho_k = \int \left(\frac{1-x}{2}\right)^{2(i+m+n+l+j)+2} (P_k^{(2(i+m+n+l+j)+2,0)})^2 \, dx.$$

We now turn to the relationship between u_{ij} and $\widetilde{u}_{ijk}$. First, note that $\zeta_{ij}|_{F_1} = \Psi_{ijk}|_{F_1}$ hence in order to satisfy the constraint that $\sum u_{ij}\zeta_{ij}|_{F_1} = \sum \widetilde{u}_{ijk}\Psi_{ijk}|_{F_1}$, we have

$$u_{ij} = \sum_{k=0}^{p-i-j-m-n-l} \widetilde{u}_{ijk} P_k^{(2(i+m+n+l+j)+2,0)}(-1) = \sum_{k=0}^{p-i-j-m-n-l} (-1)^k \widetilde{u}_{ijk}. \tag{3.27}$$

By Cauchy-Schwarz inequality, we have that

$$u_{ij}^2 \leq \sum_{k=0}^{p-i-j-m-n-l} \widetilde{u}_{ijk}^2 \rho_k \sum_{k=0}^{p-i-j-m-n-l} \rho_k^{-1} \tag{3.28}$$

which implies a lower bound for the norm of the extension in terms of u_{ij}

$$\begin{aligned} \left\|\mathcal{E}_{S,F_1} u\right\|^2 &= \sum_{i+j+k \leq p-m-n-l} \widetilde{u}_{ijk}^2 \mu_i \nu_j \rho_k \\ &\geq \sum_{i=0}^{p-m-n-l} \mu_i \sum_{j=0}^{p-m-n-l-i} \nu_j \frac{u_{ij}^2}{\sum_{k=0}^{p-i-j-m-n-l} \rho_k^{-1}}. \end{aligned} \tag{3.29}$$

In fact, equality can be achieved in Equation (3.28) if we let

$$\widetilde{u}_{ijk} = (-1)^k \rho_k^{-1} \left(\frac{u_{ij}}{\sum_{k=0}^{p-i-j-m-n-l} \rho_k^{-1}} \right).$$

One can verify that with this choice of coefficients that Equation (3.27) is still satisfied. As $\rho_k = \frac{2}{2(i+j+l+m+n)+2k+3}$, thus

$$\sum_{k=0}^{p-i-j-m-n-l} \rho_k^{-1} = \frac{1}{2}(p-i-j-l-m-n+1)(i+j+l+m+n+p+3).$$

Comparing Equation (3.29) with Equation (3.26), we see that they are indeed equal.

□

Finally we are in a position to give the proof of Lemma 3.4.6:

Proof. We first prove the case where F' consists of a single face. Without loss of generality, we can assume that $F = F_1 = \{\theta = -1\}$ the reference face, and $F' = \{\eta = -1\}$. There are three cases corresponding to $S \setminus F'$ consisting of the empty set, a single face or two faces:

Case 1. If $S = F'$, we choose $m = n = 0$.

Case 2. If $S \setminus F'$ is a single face, we choose $m = 0, n = 1$ or $m = 1, n = 0$.

Case 3. If $S \setminus F'$ consists of the two remaining faces, we choose $m = n = 1$.

Let $\alpha, \beta \in X$ of form

$$\alpha = \sum_{i+j \leq p-1-m-n} \alpha_{ij} \zeta_{ij}^{(1,m,n)}, \qquad \beta = \sum_{i+j \leq p-m-n} \beta_{ij} \zeta_{ij}^{(0,m,n)}$$

with coefficients α_{ij}, β_{ij} such that α and β coincides with u on face F_1 (i.e. $u|_{F_1} = \alpha(\xi, \eta, -1) = \beta(\xi, \eta, -1)$). Lemma 3.4.9 implies that

$$\alpha = \mathcal{E}_{S,F_1} u, \qquad \beta = \mathcal{E}_{S \setminus F', F_1} u,$$

and it suffices to show that there exists a C independent of p such that $\|\alpha\| \leq C\|\beta\|$.

Using orthogonality of the basis functions and Lemma 3.4.2 gives

$$\begin{aligned}
\|\alpha\|^2 = &\sum_{i+j\leq p-1-m-n} \frac{2(i+2m)!(i+2n)!\alpha_{ij}^2}{i!(2i+2m+2n+1)(i+2(m+n))!} \\
&\times \frac{(j+1)(j+2)}{(i+j+m+n+2)(2i+j+2m+2n+3)(2i+j+2(m+n+1))} \\
&\times \frac{2}{(p-i-j-m-n)(i+j+m+n+p+4)} \\
\approx &\sum_{i+j\leq p-1-m-n} \frac{2(i+2m)!(i+2n)!\alpha_{ij}^2}{i!(2i+2m+2n+1)(i+2(m+n))!} \\
&\times \frac{(j+1)^2}{(i+j+1)^3} \frac{1}{(p-i-j)(i+j+p)}
\end{aligned} \tag{3.30}$$

and

$$\begin{aligned}
\|\beta\|^2 = &\sum_{i+j\leq p-m-n} \frac{2(i+2m)!(i+2n)!\beta_{ij}^2}{i!(2i+2m+2n+1)(i+2(m+n))!} \\
&\times \frac{1}{i+j+m+n+1} \frac{2}{(p-i-j-m-n+1)(i+j+m+n+p+3)} \\
\approx &\sum_{i+j\leq p-m-n} \frac{2(i+2m)!(i+2n)!\beta_{ij}^2}{i!(2i+2m+2n+1)(i+2(m+n))!} \\
&\times \frac{1}{i+j+1} \frac{1}{(p-i-j+1)(i+j+p)}.
\end{aligned} \tag{3.31}$$

We thus have to show for all $0 \leq i \leq p-m-n-1$ that

$$\begin{aligned}
&\sum_{j=0}^{p-1-m-n-i} \frac{(j+1)^2\alpha_{ij}^2}{(i+j+1)^3} \frac{1}{(p-i-j)(i+j+p)} \\
&\qquad \leq C \sum_{j=0}^{p-m-n-i} \frac{\beta_{ij}^2}{i+j+1} \frac{1}{(p-i-j+1)(i+j+p)}.
\end{aligned} \tag{3.32}$$

Now, we turn to the relationship between the coefficients α_{ij} and β_{ij}. First, note that since $u \in Y_{F_1}$, it vanishes on the edges of F_1; in particular $u|_{F_1\cap\{\eta=-1\}} = 0$. We have $\alpha|_{F_1\cap\{\eta=-1\}} = 0$ as $\zeta_{ij}^{(1,m,n)}$ vanishes on $\eta = -1$, but the basis functions of β

does not vanishes trivially on $\eta = -1$. We see that

$$\begin{aligned} \beta|_{F_1 \cap \{\eta = -1\}} &= \sum_{i+j \le p-m-n} \left(\frac{1-\xi}{2}\right)^m \left(\frac{1+\xi}{2}\right)^n P_i^{(2m,2n)}(\xi)(-1)^j \beta_{ij} \\ &= \sum_{i=0}^{p-m-n} \left(\frac{1-\xi}{2}\right)^m \left(\frac{1+\xi}{2}\right)^n P_i^{(2m,2n)}(\xi) \sum_{j=0}^{p-m-n-i} (-1)^j \beta_{ij}, \end{aligned}$$

hence by linear independence,

$$\sum_{j=0}^{p-m-n-i} (-1)^j \beta_{ij} = 0 \tag{3.33}$$

in order for $\beta|_{F_1 \cap \{\eta = -1\}}$ to vanish.

Now returning to the face F_1, let $\gamma = 2i + 2m + 2n + 1$, then

$$\begin{aligned} \alpha|_{F_1} = &\sum_{i=0}^{p-1-m-n} \left(\frac{1-\xi}{2}\right)^m \left(\frac{1+\xi}{2}\right)^n P_i^{(2m,2n)}(\xi) \left(\frac{1-\eta}{2}\right)^{i+m+n} \\ &\times \sum_{j=0}^{p-1-m-n-i} \left(\frac{1+\eta}{2}\right) P_j^{(\gamma,2)}(\eta) \alpha_{ij} \end{aligned}$$

By Equation (3.33), $\beta_{p-m-n,0} = 0$ hence

$$\begin{aligned} \beta|_{F_1} &= \sum_{i=0}^{p-m-n} \left(\frac{1-\xi}{2}\right)^m \left(\frac{1+\xi}{2}\right)^n P_i^{(2m,2n)}(\xi) \left(\frac{1-\eta}{2}\right)^{i+m+n} \sum_{j=0}^{p-m-n-i} P_j^{(\gamma,0)}(\eta) \beta_{ij} \\ &= \sum_{i=0}^{p-m-n-1} \left(\frac{1-\xi}{2}\right)^m \left(\frac{1+\xi}{2}\right)^n P_i^{(2m,2n)}(\xi) \left(\frac{1-\eta}{2}\right)^{i+m+n} \sum_{j=0}^{p-m-n-i} P_j^{(\gamma,0)}(\eta) \beta_{ij}. \end{aligned}$$

As $\alpha|_{F_1} = \beta|_{F_1}$, then we must have that for a fixed $0 \le i \le p - 1 - m - n$

$$\sum_{j=0}^{p-m-n-i-1} \alpha_{ij} \left(\frac{1+\eta}{2}\right) P_j^{(\gamma,2)}(\eta) = \sum_{j=0}^{p-m-n-i} \beta_{ij} P_j^{(\gamma,0)}(\eta).$$

By telescoping the sum, we have

$$\sum_{j=0}^{p-m-n-i} \beta_{ij} P_j^{(\gamma,0)}(\eta) = \sum_{j=0}^{p-m-n-i} S_{ij}(P_{j+1}^{(\gamma,0)}(\eta) + P_j^{(\gamma,0)}(\eta)) \tag{3.34}$$

where $S_{ij} = \sum_{k=0}^{j} (-1)^{k+j} \beta_{ik}$ with $S_{i,p-m-n-i} = 0$ due to Equation (3.33).

Combining (22.7.16) and (22.7.19) of [1] gives the following relation

$$P_{j+1}^{(\gamma,0)}(x) + P_j^{(\gamma,0)}(x) = \frac{x+1}{2} \left(\frac{(\gamma+j)}{j+1} P_{j-1}^{(\gamma,2)}(x) + \frac{\gamma+j+2}{j+1} P_j^{(\gamma,2)}(x) \right) \tag{3.35}$$

for non-negative j where we assume that $P_{-1}^{(\gamma,2)} = 0$. Hence, substituting Equation (3.35) into Equation (3.34), we have

$$\sum_{j=0}^{p-m-n-i} \beta_{ij} P_j^{(\gamma,0)}(\eta) = \sum_{j=0}^{p-m-n-i} S_{ij} \frac{\eta+1}{2} \left(\frac{(\gamma+j)}{j+1} P_{j-1}^{(\gamma,2)}(\eta) + \frac{\gamma+j+2}{j+1} P_j^{(\gamma,2)}(\eta) \right).$$

Matching coefficients, we have that

$$\alpha_{ij} = \frac{\gamma+j+2}{j+1} S_{ij} + \frac{\gamma+j+1}{j+2} S_{i,j+1} = \frac{\gamma+j+1}{j+2} \beta_{i,j+1} + \frac{\gamma+2j+3}{(j+1)(j+2)} S_{ij}.$$

Using the inequality $(a+b)^2 \leq 2a^2 + 2b^2$, we have that

$$\alpha_{ij}^2 \leq 2 \left(\frac{\gamma+j+1}{j+2} \right)^2 \beta_{i,j+1}^2 + 2 \left(\frac{\gamma+2j+3}{(j+1)(j+2)} \right)^2 S_{ij}^2.$$

Inserting the above into Equation (3.32), it suffices to show that there exists a constant C independent of p and i such that

$$\sum_{j=0}^{p-1-m-n-i} \frac{(j+1)^2\left(\frac{\gamma+j+1}{j+2}\right)^2}{(i+j+1)^3} \frac{\beta_{i,j+1}^2}{(p-i-j)(i+j+p)} \\ \leq C \sum_{j=0}^{p-m-n-i} \frac{\beta_{ij}^2}{i+j+1} \frac{1}{(p-i-j+1)(i+j+p)}.$$

and

$$\sum_{j=0}^{p-1-m-n-i} \frac{(j+1)^2\left(\frac{\gamma+2j+3}{(j+1)(j+2)}\right)^2}{(i+j+1)^3} \frac{S_{ij}^2}{(p-i-j)(i+j+p)} \\ \leq C \sum_{j=0}^{p-m-n-i} \frac{\beta_{ij}^2}{i+j+1} \frac{1}{(p-i-j+1)(i+j+p)}.$$

For the first expression, we note that $\gamma + j + 1 \approx i + j + 1$ hence the inequality follows trivially. As for the second expression, we note that

$$\left(\frac{\gamma+2j+3}{(j+1)(j+2)}\right)^2 \approx \frac{(i+j+1)^2}{(j+1)^4}$$

Hence, we wish to show that

$$\sum_{j=0}^{p-1-m-n-i} \frac{S_{ij}^2}{(j+1)^2(i+j+1)} \frac{1}{(p-i-j)(i+j+p)} \\ \leq C \sum_{j=0}^{p-m-n-i} \frac{\beta_{ij}^2}{i+j+1} \frac{1}{(p-i-j+1)(i+j+p)}.$$

By Corollary 3.4.11, there exists a C independent of p and i, and we are done with the case of F' consisting of a single face.

In the case where F' consists of two or three faces, we can simply bootstrap the argument. For example, if $F' = F_1' \cup F_2'$ where F_1', F_2' are two distinct faces, then

$$\left\|\mathcal{E}_{S,F} u\right\| \leq C \left\|\mathcal{E}_{S \setminus F_1', F} u\right\| \leq C \left\|\mathcal{E}_{S \setminus (F_1' \cup F_2'), F} u\right\| = C \left\|\mathcal{E}_{S \setminus F', F} u\right\| .$$

□

3.4.5 Hardy Inequalities

It remains to prove the Hardy inequalities used.

Lemma 3.4.10. *Let $\{v_i\}_{i=0}^p \in \mathbb{R}$ satisfy*

$$\sum_{i=0}^{p} v_i = 0, \tag{3.36}$$

then for j a positive integer, there exists a constant $C(j)$ independent of p such that

$$\sum_{i=0}^{p} \frac{S_i^2}{(i+1)^3 (i+p+1)^j (p-i+1)^j} \leq C \sum_{i=0}^{p} \frac{v_i^2}{(i+1)(i+p+1)^j (p-i+1)^j}$$

where $S_i = \sum_{k=0}^{i} v_k$.

Proof. By Equation (3.36), we have that $S_i = -\sum_{k=i+1}^{p} v_k$, our inequality follows if

$$\sum_{i=0}^{p/2} \frac{\left(\sum_{k=0}^{i} v_k\right)^2}{(i+1)^3 (i+p+1)^j (p-i+1)^j} \leq C \sum_{i=0}^{p/2} \frac{v_i^2}{(i+1)(i+p+1)^j (p-i+1)^j} \tag{3.37}$$

and

$$\sum_{i=p/2+1}^{p} \frac{\left(-\sum_{k=i+1}^{p} v_k\right)^2}{(i+1)^3 (i+p+1)^j (p-i+1)^j} \leq C \sum_{i=p/2+1}^{p} \frac{v_i^2}{(i+1)(i+p+1)^j (p-i+1)^j} \tag{3.38}$$

both hold with the constant C independent of p.

Hardy's inequality for weighted sums states that for non-negative a_k, b_i, c_i,

$$\sum_{i=0}^{\infty}\left(\sum_{k=0}^{i} a_k\right)^2 b_i \leq C\sum_{i=0}^{\infty} a_i^2 c_i \tag{3.39}$$

with $C \leq 2\sqrt{2}A$ where $A := \sup_{n\in\mathbb{N}}\left(\sum_{i=n}^{\infty} b_i\right)^{1/2}\left(\sum_{i=0}^{n} c_i^{-1}\right)^{1/2} < \infty$ [50, p. 57]. For Equation (3.37) our result follows if we set $a_i = |v_i|$, $b_i^{-1} = (i+1)^3(i+p+1)^j(p-i+1)^j$ and $c_i^{-1} = (i+1)(i+p+1)^j(p-i+1)^j$ for $i = 0, \ldots, p/2$, and let $a_i = 0, b_i = 0, c_i = 1$ for $i > p/2$. It remains to show that A does not grow with p.

We note that

$$\sum_{i=0}^{n} c_i^{-1} \leq p^{2j}\sum_{i=0}^{n}(i+1) \approx n^2 p^{2j}.$$

Furthermore, the supremum can be reduced to over the interval $n \in [0, p/2]$ due to the padding of zeros, hence

$$\begin{aligned} A^2 &\approx \sup_{n\in[0,p/2]} n^2 p^{2j}\sum_{i=n}^{p/2}\frac{1}{(i+1)^3(i+p+1)^j(p-i+1)^j} \\ &\leq \sup_{n\in[0,p/2]} n^2 p^{2j}\int_n^{p/2}\frac{1}{(x+1)^3(p-p/2+1)^j p^j}\,dx \\ &\approx \sup_{n\in[0,p/2]} n^2\left(\frac{1}{2(n+1)^2}-\frac{2}{(p+2)^2}\right) < \infty. \end{aligned}$$

For Equation (3.38), we first transform the sum such that the index starts at 0 by mapping the indices $i \to p-i, k \to p-k$

$$\sum_{i=0}^{p/2-1}\frac{\left(-\sum_{k=0}^{i-1} v_{p-k}\right)^2}{(p-i+1)^3(2p-i+1)^j(i+1)^j} \leq C\sum_{i=0}^{p/2-1}\frac{v_{p-i}^2}{(p-i+1)(2p-i+1)^j(i+1)^j}.$$

Our result follows if we set $a_i = |v_{p-i}|, b_i^{-1} = (p-i+1)^3(2p-i+1)^j(i+1)^j, c_i^{-1} = (p-i+1)(2p-i+1)^j(i+1)^j$ for $i = 0, \dots, p/2-1$, and let $a_i = 0, b_i = 0, c_i = 1$ for $i \geq p/2$. It remains to show that A does with not grow with p.

Proceeding similarly as before, note that $\sum_{i=0}^n c_i^{-1} \leq (2p)^{j+1} \sum_{i=0}^n (i+1)^j \approx p^{j+1}n^{j+1}$. The supremum can be reduced to over the interval $n \in [0, p/2-1]$ as before. Calculating, we have

$$\begin{aligned}
A^2 &\approx \sup_{n \in [0,p/2-1]} n^{j+1}p^{j+1} \sum_{i=n}^{p/2-1} \frac{1}{(p-i+1)^3(2p-i+1)^j(i+1)^j} \\
&\leq \sup_{n \in [0,p/2-1]} n^{j+1}p \int_n^{p/2} \frac{1}{(p-p/2+1)^3(x+1)^j}\, dx \\
&\approx \sup_{n \in [0,p/2-1]} \frac{n^{j+1}}{p^2} \begin{cases} \frac{2(n+1)(p+2)^j - 2^j(p+2)(n+1)^j}{2(j-1)(n+1)^j(p+2)^j} & j > 1 \\ \log\left(\frac{p}{2n}\right) & j = 1 \end{cases} \\
&< \infty.
\end{aligned}$$

□

Lemma 3.4.10 deals with the general case $j \in \mathbb{N}$ and in addition proves explicitly that $C(j)$ is independent of p.

The following Hardy inequality is required for the face extension inequalities:

Corollary 3.4.11. *Let $\{v_i\}_{i=0}^{p-k} \in \mathbb{R}$ where k is an integer $1 \leq k \leq p$, and $S_i = \sum_{j=0}^{i}(-1)^j v_j$, then there exists a constant C independent of p, k such that*

$$\sum_{i=0}^{p-k} \frac{S_i^2}{(i+1)^2(i+k)(p-k-i+1)(p+k+i)} \leq C \sum_{i=0}^{p-k} \frac{v_i^2}{(i+k)(p-k-i+1)(p+k+i)}$$

Proof. Since the proof technique is the same as Lemma 3.4.10, we will only tersely discuss the details below.

As before, split the inequality into two, similar to Equations (3.37) and (3.38). For the first sum, we set $a_i = |v_i|$, $b_i^{-1} = (i+1)^2(i+k)(p-k-i+1)(p+k+i)$ and $c_i^{-1} = (i+k)(p-k-i+1)(p+k+i)$ for $i = 0, \ldots, \frac{p-k}{2}$. Then, $\sum_{i=0}^n c_i^{-1} \leq (p+k)(p-k)\sum_{i=0}^n (i+k) \approx (p+k)(p-k)(n^2+kn)$ and the following calculation gives that A is bounded:

$$\begin{aligned} A^2 &\approx \sup_{n\in[0,\frac{p-k}{2}]} (p+k)(p-k)(n^2+kn) \sum_{i=n}^{\frac{p-k}{2}} \frac{1}{(i+1)^2(i+k)(p-k-i+1)(p+k+i)} \\ &\leq \sup_{n\in[0,\frac{p-k}{2}]} (n^2+kn) \int_n^{(p-k)/2} \frac{1}{(x+1)^2(x+k)}\,dx \\ &\leq \sup_{n\in[0,\frac{p-k}{2}]} n^2 \int_n^{\frac{p-k}{2}} \frac{1}{(x+1)^3}\,dx + kn \int_n^{\frac{p-k}{2}} \frac{1}{(x+1)^2(x+k)}\,dx < \infty. \end{aligned}$$

For the second sum, first transform the sum to start the index 0 again. Next, set $a_i = |v_{p-k-i}|$, $b_i^{-1} = (p-k-i+1)^2(p-i)(2p-i)(i+1)$, $c_i^{-1} = (p-i)(2p-i)(i+1)$ for $i = 0, \ldots, \frac{p-k}{2} - 1$. Calculating, we have $\sum_{i=0}^n c_i^{-1} \leq p^2 \sum_{i=0}^n (i+1) \approx p^2n^2$ and thus

$$\begin{aligned} A^2 &\approx \sup_{n\in[0,\frac{p-k}{2}-1]} p^2n^2 \sum_{i=n}^{\frac{p-k}{2}-1} \frac{1}{(p-k-i+1)^2(p-i)(2p-i)(i+1)} \\ &\leq \sup_{n\in[0,\frac{p-k}{2}-1]} pn^2 \int_n^{\frac{p-k}{2}} \frac{1}{(p-k-(p-k)/2+1)^2(p-(p-k)/2)(x+1)}\,dx \\ &\approx \sup_{n\in[0,\frac{p-k}{2}-1]} \frac{pn^2}{(p-k)^2(p+k)} \log\left(\frac{p-k}{2n}\right) < \infty. \end{aligned}$$

□

3.5 Appendix: Triangle Basis Functions

In Chapter 2, we defined the triangle vertex and edge basis functions, and also the notion of "minimal L^2 extension" (or simply "minimal extension"). The minimal extensions were implicitly constructed in the preconditioner by using the Schur complement, and is needed in the theory at multiple instances in Chapter 2 to prove the existence of a stable decomposition.

Through the experience of developing the 3D preconditioner, we are able to now define new vertex and edge basis functions which *themselves are explicitly the minimal L^2 extension.* This was mentioned in the introduction of the chapter, with the claim that the condition number using this basis is better than those presented in Chapter 2, which we will list for completeness here.

Recall the reference triangle T (c.f. Figure 2.1) with vertices $v_1 = (-1,-1), v_2 = (1,-1), v_3 = (-1,1)$ and barycentric coordinates λ_i. The vertex basis function associated with v_1 of T is defined as

$$\varphi_1 = \frac{1}{2}\lambda_1(q(\lambda_2,\lambda_3) + q(\lambda_3,\lambda_2)) \tag{3.40}$$

where

$$q(l_1.l_2) = \Phi_i^{(2)}\left(\frac{2l_1}{1-l_2} - 1\right)(1-l_2)^i\,\Phi_{p-i-1}^{(2i+3)}(2l_2-1)$$

with $i = \lfloor p/2 \rfloor$. The basis functions associated with the edge $E_1 := \{y = -1\} \cap T$ are chosen as follows:

$$\chi_i^{(1)} := \lambda_1\lambda_2 P_i^{(2,2)}\left(\frac{2\lambda_2}{1-\lambda_3} - 1\right)(1-\lambda_3)^i\Phi_{p-i-2}^{(2i+5)}(2\lambda_3 - 1), \qquad 0 \le i \le p-2,$$

The basis functions on the other two edges are defined by a permutation of the barycentric coordinates. Similar to Lemma 3.4.3, it is not difficult to show that the above basis is orthogonal to all interior basis functions. Furthermore, one can show that $\|\varphi_1\|_T^2 \approx Cp^{-4}$ by direct calculation, thus by Theorem 5.3.1, the preconditioner using the above basis is uniform in p.

Chapter

FOUR

Tensor Product Elements

4.1 Stable Decomposition on Tensor Products Elements

An Additive Schwarz method (ASM) is defined by a subspace decomposition, and a choice of inner product on each subspace [19, 77]. In the case of the mass matrix, Chapters 2 and 3 developed ASM preconditioners for triangles and tetrahedra with condition numbers independent of the polynomial order p. A reasonable question now is how do we develop preconditioners for the mass matrix on tensor product elements such as quads, hexes and prisms with a condition number that is also bounded in p.

It is useful to first recall the structure and the estimates one has to prove in the case of the simplicial elements before proceeding. Let K be a single reference triangle or tetrahedron and $X := \mathbb{P}_p(K)$. In each case Chapters 2 and 3, we defined a number of non-overlapping subspaces $\{X_i\}_{i=1}^k$ each associated with a geometric entity of K with the property that $\oplus_{i=1}^k X_i = X$; an exact solver is used on each of the subspaces. The key estimates one needed to prove in the simplicial cases is the following: for all $u \in X$, there exists a decomposition $\sum_{i=1}^k u_i = u$ with $u_i \in X_i$ such that there exists constants c, C independent of p such that

$$c\sum_{i=1}^{k}\|u_i\|_K^2 \leq \|u\|_K^2 \leq C\sum_{i=1}^{k}\|u_i\|_K^2 \tag{4.1}$$

where $\|\cdot\|_K$ is the L^2 norm over K. The condition number of the ASM preconditioner is then C/c by standard ASM theory (see [77, Theorem 2.7]). The generalization to a mesh of elements readily follows by a subassembly argument.

We will also follow the above simple, yet effective, framework in the case of tensor product elements: a non-overlapping subspace decomposition of the polynomial space (e.g. $\mathbb{Q}_p$) each associated with an exact solver. The effectiveness of a tensor product mass matrix preconditioner then reduces to the ratio of C/c in Equation (4.1).

The upper bound C is obtained easily due to an application of the triangle inequality. The lower bound, sometimes known as a *stable decomposition* in the parlance of ASM [77, Assumption 2.2], is the difficult aspect of the proofs in Chapters 2 and 3. This difficulty can be actually be avoided for a tensor product element as the following lemma shows:

Lemma 4.1.1. *Let K be a geometric entity (e.g. interval, triangle etc.) and let X be polynomial space defined on K. Assume that there exists a subspace decomposition $\{X_i\}_{i=0}^{k}$ of X with $\oplus_{i=0}^{k} X_i = X$ such that $\forall u \in X$, we have*

$$\sum_{i=0}^{k} \|u_i\|_K^2 \leq C_K \|u\|_K^2 \tag{4.2}$$

with $u_i \in X_i$, and $\sum_{i=0}^{k} u_i = u$. Furthermore, let L be another geometric entity and let Y be the polynomial space on L. Assume that there exists a subspace decomposition $\{Y_i\}_{i=0}^{l}$ of X with $\oplus_{i=0}^{l} Y_i = Y$ such that $\forall v \in Y$, we have

$$\sum_{i=0}^{l} \|v_i\|_L^2 \leq C_L \|v\|_L^2 \tag{4.3}$$

with $v_i \in Y_i$ and $\sum_{i=0}^{l} v_i = v$. Then for all $w \in Z := X \otimes Y$ defined on the element $P = K \otimes L$, there exists a decomposition satisfying

$$\sum_{i=0}^{k} \sum_{j=0}^{l} \left\| w_{ij} \right\|_P^2 \leq C_K C_L \|w\|_P^2$$

with $w_{ij} \in Z_{ij} := X_i \otimes Y_j$, and $\sum_{i=0}^{k} \sum_{j=0}^{l} w_{ij} = w$.

Proof. Let $\mathbf{M}_K, \mathbf{M}_L$ be the mass matrices on entities K, L over spaces X, Y respectively. The subspaces $\{X_i\}_{i=0}^k, \{Y_i\}_{i=0}^l$ induce a natural partitioning of the mass matrices

$$\mathbf{M}_K := \begin{bmatrix} \mathbf{M}_{K,11} & \cdots & \mathbf{M}_{K,1k} \\ \vdots & \ddots & \vdots \\ \mathbf{M}_{K,k1} & \cdots & \mathbf{M}_{K,kk} \end{bmatrix} \qquad \mathbf{M}_L := \begin{bmatrix} \mathbf{M}_{L,11} & \cdots & \mathbf{M}_{L,1l} \\ \vdots & \ddots & \vdots \\ \mathbf{M}_{L,l1} & \cdots & \mathbf{M}_{L,ll} \end{bmatrix}$$

with the $\mathbf{M}_{K,ij}$ block corresponding to the interaction between subspace X_i, X_j etc. Furthermore, we may pick a basis such that the blocks lying on the diagonal (e.g. $\{\mathbf{M}_{K,ii}\}_{i=0}^k, \{\mathbf{M}_{L,ii}\}_{i=0}^l$) are *themselves diagonal.*

Let $\vec{u}$ be the vector corresponding to $u \in X$ such that $\|u\|_K^2 = \vec{u}^T \mathbf{M}_K \vec{u}$ and analogously for $\vec{v}$ to a function in $v \in Y$. The vectors $\vec{u}, \vec{v}$ can be decomposed similar to the mass matrices

$$\vec{u} = [\vec{u}_1, \ldots, \vec{u}_k] \qquad \vec{v} = [\vec{v}_1, \ldots, \vec{v}_l]$$

with $\vec{u}_1$ corresponding to a function in X_1 etc, hence $\vec{u}_1^T \mathbf{M}_{K,11} \vec{u}_1 = \|u_1\|_K^2$ etc.

Define the diagonal matrices

$$\mathbf{D}_K := \operatorname{diag}(\mathbf{M}_K), \qquad \mathbf{D}_L := \operatorname{diag}(\mathbf{M}_L),$$

then the statements Equations (4.2) and (4.3) can be rewritten as

$$\vec{u}^T \mathbf{D}_K \vec{u} \leq C_K \vec{u}^T \mathbf{M}_K \vec{u}, \qquad \vec{v}^T \mathbf{D}_L \vec{v} \leq C_L \vec{v}^T \mathbf{M}_L \vec{v}$$

for all $\vec{u}, \vec{v}$; hence $\lambda_{\min}(\mathbf{D}_K^{-1} \mathbf{M}_K) \geq \frac{1}{C_K}$ and $\lambda_{\min}(\mathbf{D}_L^{-1} \mathbf{M}_L) \geq \frac{1}{C_L}$.

Turning to P, the corresponding mass matrix $\mathbf{M}_P := \mathbf{M}_K \otimes \mathbf{M}_L$ is simply the Kronecker product owing to the tensor product structure of P. For any $w \in Z$, let $\vec{w}$ be the corresponding vector such that $\vec{w}^T \mathbf{M}^P \vec{w} = \|w\|_P^2$. Similar to before, we can decompose $\vec{w}$ as

$$\vec{w} = [\vec{w}_{11}, \ldots, \vec{w}_{1l}, \vec{w}_{21}, \ldots, \vec{w}_{2l}, \vec{w}_{31}, \ldots, \vec{w}_{kl}]$$

where $\vec{w}_{ij}$ corresponds to a function in Z_{ij}. By the properties of Kronecker product [35], we have that

$$\begin{aligned}\lambda_{\min}((\mathbf{D}_K \otimes \mathbf{D}_L)^{-1} (\mathbf{M}_K \otimes \mathbf{M}_L)) &= \lambda_{\min}\left(\left(\mathbf{D}_K^{-1} \otimes \mathbf{D}_L^{-1}\right)(\mathbf{M}_K \otimes \mathbf{M}_L)\right) \\ &= \lambda_{\min}\left(\left(\mathbf{D}_K^{-1}\mathbf{M}_K \otimes \mathbf{D}_L^{-1}\mathbf{M}_L\right)\right) \geq \frac{1}{C_K C_L}\end{aligned}$$

hence $\mathbf{D}_K \otimes \mathbf{D}_L \leq C_K C_L \mathbf{M}_P$.

Finally, note that since Z_{ij} is the tensor product of X_i and Y_j then the mass matrix corresponding to Z_{ij} is $\mathbf{M}_{K,ii} \otimes \mathbf{M}_{L,jj}$, which is also a diagonal matrix. Hence,

$$\vec{w}^T (\mathbf{D}_K \otimes \mathbf{D}_L) \vec{w} = \sum_{i=0}^{k} \sum_{j=0}^{l} \left\| w_{ij} \right\|_P^2$$

and we are done. □

4.2 Applications of Lemma 4.1.1

With the above lemma, it suffices to define the subspaces on lower dimensional objects, such as the interval, and prove Equations (4.2) and (4.3) on it.

4.2.1 Stable Decomposition on an Interval

It is useful to first define decompositions on the reference interval $\hat{I} = [-1, 1]$ which one can apply Lemma 4.1.1 to. Let $\varphi_{V_1} \in \mathbb{P}_p(\hat{I})$ be a nodal basis function such that

1. $\varphi_{V_1}(-1) = 1, \varphi_{V_1}(1) = 0,$
2. there exists a constant C independent of p such that $\|\varphi_{V_1}\|_{\hat{I}} \leq Cp^{-1}$.

Let $\varphi_{V_2}(x) = \varphi_{V_1}(-x)$, then define the vertex spaces as $X_{V_1} = \text{span}\{\varphi_{V_1}\}$ and $X_{V_2} = \text{span}\{\varphi_{V_2}\}$. Finally, let $X_I = \text{span}\{(1-x^2)P_i^{(2,2)}(x)\}_{i=0}^{p-2}$ be the interior space. It is trivial to see that $X_{V_1} \oplus X_{V_2} \oplus X_I = \mathbb{P}_p(\hat{I})$.

Lemma 4.2.1. *For all $u \in \mathbb{P}_p(\hat{I})$, then there exists a constant C independent of p such that*

$$\|u_{V_1}\|_{\hat{I}}^2 + \|u_{V_2}\|_{\hat{I}}^2 + \|u_I\|_{\hat{I}}^2 \leq C\|u\|_{\hat{I}}^2$$

where $u_{V_1} = u(-1)\varphi_{V_1} \in X_{V_1}, u_{V_2} = u(1)\varphi_{V_2} \in X_{V_2}, u_I = u - u_{V_1} - u_{V_2} \in X_I$.

Proof. For the two nodal functions, recall that $\|u\|_\infty \leq p\|u\|_{\hat{I}}$ [79] and $\|\varphi_{V_1}\|_{\hat{I}} \leq C/p$, hence

$$\|u_1\|_{\hat{I}} \leq \|\varphi_{V_1}\|_{\hat{I}}\|u\|_\infty \leq C\|u\|_{\hat{I}}\,.$$

The same holds for u_2. By triangle inequality $\|u_I\|_{\hat{I}} \leq \|u - u_{V_1} - u_{V_2}\|_{\hat{I}} \leq C\|u\|_{\hat{I}}$. □

Lemma 4.2.1 implies that a large class of stable decompositions exists on the interval which are characterized by the choice of nodal function. A key question is which nodal basis function should one choose if one were to use a hybrid mesh consisting of *both* quads and triangles (e.g. Figure 4.2).

To this end, let us recall the nodal basis functions for the preconditioner on the triangle. Let T be the a triangle with vertices v_1, v_2, v_3, and let λ_i be the barycentric coordinates of T. For positive integer k, let

$$\xi_k(x) := \frac{(-1)^{k+1}}{k}\frac{1-x}{2}P_{k-1}^{(1,1)}(x), \tag{4.4}$$

then the nodal basis function defined in Chapter 2 is $\xi_{\lfloor p/2\rfloor}(1-2\lambda_i)$. By Lemma 2.6.3, we have $\left\|\xi_{\lfloor p/2\rfloor}\right\|_{\hat{I}} \leq Cp^{-1}$, hence the following decomposition is stable on the interval:

$$\begin{aligned} X_{V_1} &= \operatorname{span}\{\xi_{\lfloor p/2\rfloor}(x)\} \\ X_{V_2} &= \operatorname{span}\{\xi_{\lfloor p/2\rfloor}(-x)\} \\ X_I &= \operatorname{span}\{(1-x^2)P_i^{(2,2)}(x)\}_{i=0}^{p-2}. \end{aligned} \tag{4.5}$$

4.2.2 Quad and Hex Elements

Suppose $K = L = [-1,1], X = Y = \mathbb{P}_p(I)$ with the stable decomposition on $\mathbb{P}_p(I)$ given by 4.5. Let $\hat{Q} = K \otimes L$ the reference quadrilateral, and let

$$\begin{aligned} &Z_{V_1} = X_{V_1} \otimes X_{V_1}, Z_{V_2} = X_{V_1} \otimes X_{V_2}, Z_{V_3} = X_{V_2} \otimes X_{V_1}, Z_{V_4} = X_{V_2} \otimes X_{V_2} \\ &Z_{E_1} = X_{V_1} \otimes X_I, Z_{E_2} = X_{V_2} \otimes X_I, Z_{E_3} = X_I \otimes X_{V_1}, Z_{E_4} = X_I \otimes X_{V_2} \\ &Z_I = X_I \otimes X_I \end{aligned}$$

be the four vertex spaces, four edge spaces, and interior spaces of $\mathbb{Q}_p(\hat{Q})$ respectively. The above decomposition gives rise to an ASM if we associate each subspace with an exact local solve as follows: given a residual $f \in \mathbb{Q}_p(\hat{Q})$, the action of the ASM is find

1. For $i = 1, 2, 3, 4$, $u_{V_i} \in Z_{V_i} : (u_{V_i}, v_{V_i}) = (f, v_{V_i}), \forall v_{V_i} \in Z_{V_i}$
2. For $i = 1, 2, 3, 4$, $u_{E_i} \in Z_{E_i} : (u_{E_i}, v_{E_i}) = (f, v_{E_i}), \forall v_{E_i} \in Z_{E_i}$
3. $u_I \in Z_I : (u_I, v_I) = (f, v_I), \forall v_I \in Z_I$

where $(\cdot, \cdot)$ is the L^2 inner-product over $\hat{Q}$, and return $u = u_I + \sum_{k=1}^{4} u_{V_i} + u_{E_i}$.

The proof that the above ASM preconditioner is uniform in p is a simple application of Lemma 4.1.1:

Corollary 4.2.2. *Let $u \in \mathbb{Q}_p(\hat{Q})$, then there exists constants c, C independent of p such that*

$$c\left(\sum_{i=1}^{4}\left\|u_{V_i}\right\|_{\hat{Q}}^2 + \left\|u_{E_i}\right\|_{\hat{Q}}^2 + \|u_I\|_{\hat{Q}}^2\right) \leq \|u\|_{\hat{Q}}^2 \leq C\left(\sum_{i=1}^{4}\left\|u_{V_i}\right\|_{\hat{Q}}^2 + \left\|u_{E_i}\right\|_{\hat{Q}}^2 + \|u_I\|_{\hat{Q}}^2\right)$$

with $u_{V_i} \in Z_{V_i}, u_{E_i} \in Z_{E_i}, u_I \in Z_I$ and $u = \sum_{i=1}^{4} u_{V_i} + u_{E_i} + u_I$.

Proof. The upper bound follows immediately from an application of the triangle inequality while the lower bound follows from an application of Lemma 4.1.1 to the stable decomposition on the interval Equation (4.5). □

The use of a non-traditional nodal basis $\xi_{\lfloor p/2 \rfloor}(x)$ might seem odd, but whose choice is more apparent on a mesh consisting of quads and triangles. For example, consider the simplified mesh Section 4.2.2 which consists of two elements, one quad element and a triangle element, sharing the inter-elemental edge γ and vertex $\boldsymbol{v}$.

Suppose one uses the nodal function $\varphi^T := \xi_{\lfloor p/2 \rfloor}(1-2\lambda)$ on $\boldsymbol{v}$ as advocated by [8] in order to precondition the mass matrix on the triangle; how would this choice influence the nodal basis on the quad? The nodal basis function for $\boldsymbol{v}$ on the quad is simply $\varphi^Q := \xi_{\lfloor p/2 \rfloor}(x)\xi_{\lfloor p/2 \rfloor}(y) \in Z_{V_i}$. It is clear that conformity is now enforced as $\varphi^T|_\gamma = \varphi^Q|_\gamma$.

On the other hand, we also presented the nodal basis φ_i in Equation (3.40) at the end of Chapter 3; what if we had chosen that nodal basis to be our preconditioner on the triangle? In this case, the use of $\xi_{\lfloor p/2 \rfloor}(x)$ is inappropriate, and a new nodal basis for the interval needs to be defined. Recall the definition of $\Phi_q^{(m)}(x) \in \mathbb{P}_q([-1,1])$

$$\Phi_q^{(m)}(x) := \frac{(-1)^q}{q+1} P_q^{(m,1)}(x) \tag{4.6}$$

where q, m or non-negative integers. Let

$$f(x) := \frac{1}{2}\left(\frac{1-x}{2}\Phi_i^{(2)}(x) + \left(\frac{1-x}{2}\right)^{i+1} \Phi_{p-1-i}^{(2i+3)}(x)\right).$$

Note that $\varphi|_\gamma = f$, then if we choose the subspaces of the interval as

$$\begin{aligned} X_{V_1} &= \operatorname{span}\{f(x)\} \\ X_{V_2} &= \operatorname{span}\{f(-x)\} \\ X_I &= \operatorname{span}\{(1-x^2)P_i^{(2,2)}(x)\}_{i=0}^{p-2}. \end{aligned} \tag{4.7}$$

and proceed as before, we arrive at another ASM preconditioner for the quad. By Lemma 4.2.1, it suffices to prove that $\|f\|_{\hat{I}} \leq Cp^{-1}$, but this follows from the inverse inequality from an edge [79] and the fact that $\|\varphi_i\|_{\hat{T}} \leq Cp^{-2}$:

$$\|f\|_{\hat{I}} \leq p\|\varphi_i\|_{\hat{T}} \leq Cp^{-1}.$$

Section 4.2.2 illustrates the performance of a preconditioner utilizing both the tensor product preconditioner utilizing $\xi_{\lfloor p/2 \rfloor}(x)$ above and the triangle preconditioner Chapter 2; note that the condition number stays bounded as predicted. The generalization to hexahedral elements is a analogous to the above exposition.

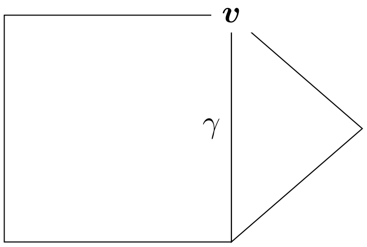


Figure 4.1: Figure of a simple hybrid mesh to illustrate the issue of choice of nodal basis functions.

4.2.3 Prism Elements

Now suppose $K = [-1, 1], X = \mathbb{P}_p(\hat{I})$ paired with the same subspace decomposition as Equation (4.5), and consider $L = \hat{T}$ the reference triangle and $Y = \mathbb{P}_p(\hat{T})$. Let $\{Y_{V_i}\}_{i=1}^{3}, \{Y_{E_i}\}_{i=1}^{3}, \{Y_I\}$ be the three nodal spaces, three edge spaces and the interior space defined in Chapter 2. Let $\hat{P} := \hat{I} \otimes \hat{T}$ be the reference prism and $Z = \mathbb{P}_P(\hat{I}) \otimes \mathbb{P}_p(\hat{T})$ the corresponding polynomial space. Define the subspaces of Z as follows:

1. Six nodal spaces Z_{V_i} due to the permutations of $X_{V_i} \otimes Y_{V_j}$,
2. Nine edge spaces Z_{E_i} due to the six permutations of $X_{V_i} \otimes Y_{E_j}$ and the three permutations of $X_I \otimes Y_{V_i}$,
3. Five face spaces Z_{F_i} due to the two permutations of $X_{V_i} \otimes Y_I$ and three permutations of $X_I \otimes Y_{E_i}$
4. One interior space Z_I arising from $X_I \otimes Y_I$.

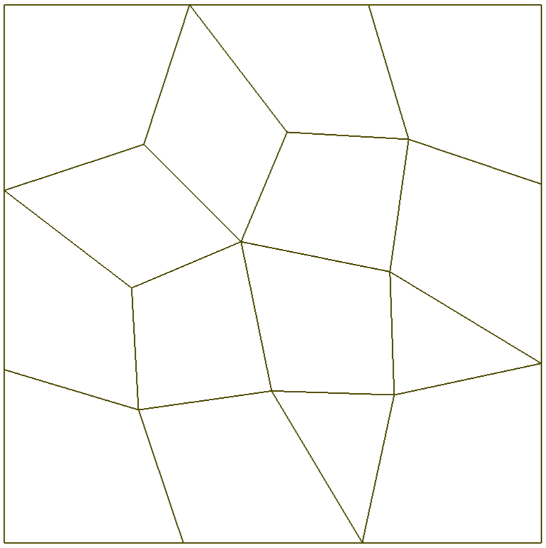


Figure 4.2: Figure of mesh consisting of both quads and triangular elements.

The ASM is thus defined by the subspace decomposition above, with an exact local solve on each individual subspace. For completeness, we again give the action of the preconditioner on a residual $f \in Z$,

1. For $i = 1, \cdots, 6$, $u_{V_i} \in Z_{V_i} : (u_{V_i}, v_{V_i}) = (f, v_{V_i}), \forall v_{V_i} \in Z_{V_i}$
2. For $i = 1, \cdots, 9$, $u_{E_i} \in Z_{E_i} : (u_{E_i}, v_{E_i}) = (f, v_{E_i}), \forall v_{E_i} \in Z_{E_i}$
3. For $i = 1, \cdots, 5$, $u_{F_i} \in Z_{F_i} : (u_{F_i}, v_{F_i}) = (f, v_{F_i}), \forall v_{F_i} \in Z_{F_i}$
4. $u_I \in Z_I : (u_I, v_I) = (f, v_I), \forall v_I \in Z_I$

where this time, $(\cdot, \cdot)$ is the L^2 inner-product over $\hat{P}$, and return $\sum_{i=1}^{6} u_{V_i} + \sum_{i=1}^{9} u_{E_i} + \sum_{i=1}^{5} u_{F_i} + u_I$.

The ASM is uniform in p due to the following lemma whose proof follows from triangle inequality and Lemma 4.1.1:

Corollary 4.2.3. *Let $u \in \mathbb{P}_P(\hat{I}) \otimes \mathbb{P}_p(\hat{T})$, then there exists constants c, C independent*

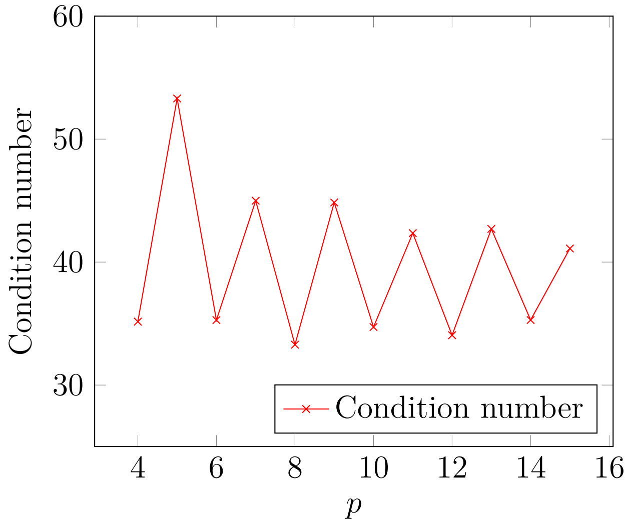


Figure 4.3: Figure illustrates the condition number of the preconditioned mass matrix on Figure 4.2; note that the condition number stays bounded for all p. The oscillatory behavior is also observed in Chapter 2, and is due to the floor function in the nodal basis function.

of p such that

$$c\left(\sum_{i=1}^{6}\left\|u_{V_i}\right\|^2_{\hat{P}}+\sum_{i=1}^{9}\left\|u_{E_i}\right\|^2_{\hat{P}}+\sum_{i=1}^{5}\left\|u_{F_i}\right\|^2_{\hat{P}}+\|u_I\|^2_{\hat{P}}\right)\leq\|u\|^2_{\hat{P}}\leq$$

$$C\left(\sum_{i=1}^{6}\left\|u_{V_i}\right\|^2_{\hat{P}}+\sum_{i=1}^{9}\left\|u_{E_i}\right\|^2_{\hat{P}}+\sum_{i=1}^{5}\left\|u_{F_i}\right\|^2_{\hat{P}}+\|u_I\|^2_{\hat{P}}\right)$$

with $u_{V_i}\in Z_{V_i}, u_{E_i}\in Z_{E_i}, u_{F_i}\in Z_{F_i}, u_I\in Z_I$ and $u=\sum_{i=1}^{6}u_{V_i}+\sum_{i=1}^{9}u_{E_i}+\sum_{i=1}^{5}u_{F_i}+u_I$.

CHAPTER

FIVE

Uniform Substructuring Preconditioner and the Influence of Nodal Basis Functions

5.1 Introduction

A key step in the substructuring preconditioner Chapter 2 is that *vertex* basis functions are chosen to be

$$\phi^\star = \frac{(-1)^{p+1}}{p}\lambda P_{r-1}^{(1,1)}(1-2\lambda). \tag{5.1}$$

where λ is the usual barycentric coordinate on the triangle, r is chosen to be the integer part of $p/2$ and $P_r^{(1,1)}$ denotes the Jacobi polynomials of degree r. In contrast, the choice of the edge and interior functions is (as we shall later show) not crucial and one is free to exercise one's own preference.

One might well ask what would happen if a more standard choice of vertex function was used rather than Equation (5.1)? A finite element practitioner would probably prefer to use the barycentric coordinate λ as the vertex function whilst an aficionado of spectral elements might well prefer to use the function Equation (5.1) in conjunction with the choice $r = p$. The first part of the chapter shows that, with the above two choices, the condition number of the preconditioned mass matrix will now grow as $\mathcal{O}(p^2)$ and $\mathcal{O}(1 + \log p)$ respectively. These results are special cases of a more general result which is actually shown here: if the vertex function is chosen to be ϕ, then the condition number will grow as $\mathcal{O}(p^4\Upsilon_p(\phi))$ where

$$\Upsilon_p(\phi) = \min_{\substack{u=\phi \text{ on } \partial T\\ u\in\mathbb{P}_p(T)}} \|u\|_{L^2}^2 . \tag{5.2}$$

The main part of the current chapter is then devoted to developing a robust substructuring type preconditioner for high order approximation of the problems for which the element matrix takes the form

$$\mathbf{A}_\kappa := (1-\kappa)\mathbf{L} + \kappa\mathbf{M} \tag{5.3}$$

where $\kappa \in (0,1)$ and $\mathbf{M}, \mathbf{L}$ are the mass and stiffness matrices respectively. Leaving aside the case of the pure mass matrix ($\kappa = 1$), one can simply use the substructuring preconditioner for the stiffness matrix developed in [12] as a preconditioner for cases where $\kappa \in [0,1)$, resulting in the same $\mathcal{O}(1+\log^2 p)$ bound on the condition number mentioned earlier. However, whilst undoubtedly correct, this conclusion fails to recognize that the hidden constant in this bound is *dependent on* κ and may degenerate badly for κ values close to 1; e.g. corresponding to the stepsize tends to zero in an implicit timestepping scheme or the singular perturbation parameter tends to zero in a singularly perturbed problem. The concern is that the scheme may fail to be robust in the limit $\kappa \to 1$. We show that such concerns are well-founded in the sense the *best uniform bound in* κ that one can hope for is actually $\mathcal{O}(p^2)$. More precisely, we prove that the upper envelope of the bound $C_\kappa(1+\log^2 p)$ is Cp^2 for all κ.

This serves as motivation for the final part of this work where we turn to the question of what can be done to obtain a preconditioner that is robust for all $\kappa \in [0,1]$. The solution turns out to be a relatively minor modification of the basic substructuring algorithm [12]: one simply augments the preconditioner with a Jacobi smoothener over the coarse grid degrees of freedom expressed using the basis function $\phi^\star$. This is numerically shown to result in a condition number bounded by $\mathcal{O}(1+\log^2 p)$ where the constant is independent of $\kappa \in [0,1]$.

Finally, for good measure, we provide a generalization of this result for the reader who would prefer to use a more standard choice of vertex function ϕ for the vertex smoothener rather than $\phi^\star$. The condition number is then conjectured to be bounded by $\mathcal{O}(\max\{1+\log^2 p, (1+\log p)p^4\Upsilon_p(\phi)\})$ where, as before, the constant is independent of κ and $\Upsilon_p(\phi)$ is the same quantity Equation (5.2) which arose in the analysis of the pure mass matrix.

The remainder of this chapter is organized as follows. In section 2, we give a brief introduction to substructuring preconditioners in the context of p-FEM. In section 3, we discuss the effects of the choice of nodal basis function in the cases of the pure mass matrix. In section 4, we generalize the previous section to the case of $\mathbf{A}_\kappa$. In section 5, we present two numerical examples. We finish with section 6 and 7 which contains the proofs and technical lemmas, and a conclusion in section 8.

5.2 Model problem and substructuring preconditioners

5.2.1 Model problem

In view of the foregoing discussion, we consider the issue of preconditioning the operator $\mathbf{A}_\kappa$. The case where $\kappa \to 0$ corresponds to the pure stiffness matrix, while $\kappa \to 1$ corresponds to the pure mass matrix. We seek to construct a preconditioner for $\mathbf{A}_\kappa$ which is, as far as possible, robust in κ, h and p. We first restrict our analysis to a single reference element, and generalize to multiple element meshes in Section 5.4.1.

Let $\hat{T}$ be the reference triangle in $\mathbb{R}^2$ with vertices $v_1 = (-1,1), v_2 = (1,-1), v_3 = (-1,-1)$ and edges $\gamma_1, \gamma_2, \gamma_3$ where γ_i is opposite v_i; see Figure 2.1. For a fixed integer $p \geq 3$, let $X := \mathbb{P}_p(\hat{T}) = \operatorname{span}\{x^\alpha y^\beta : 0 \leq \alpha, \beta, \alpha+\beta \leq p\}$ be the space of

A version of this chapter has been submitted for publication.

polynomials of total degree p on $\hat{T}$.

5.2.2 Preconditioners for $\mathbf{A}_\kappa$

We shall construct an additive Schwarz method (ASM) preconditioner [19, 71, 77] for Equation (5.3) based on the decomposition

$$X = \underbrace{X_I}_{\text{Interior space}} \oplus \underbrace{X_{E_1} \oplus X_{E_2} \oplus X_{E_3}}_{\text{Edge spaces}} \oplus \underbrace{X_V}_{\text{Vertex space}} \tag{5.4}$$

where X_I denotes the interior space $X \cap H_0^1$ and $X_{E_i} = \{u \in X : u = 0 \text{ on } \partial\hat{T} \backslash \gamma_i\}, i \in \{1, 2, 3\}$ denote the edge spaces. The vertex space is defined as $X_V = \text{span}\{\varphi_i\}_{i=1}^3$ where $\varphi_i \in X$ such that $\varphi_i(v_j) = \delta_{ij}$ for $i, j \in \{1, 2, 3\}$. Possible choices for φ_i range from the affine hat functions popular with the finite element community [73] through to the high order polynomials which vanish at quadrature points commonly adopted for spectral element methods [59]. The specific choice of φ_i plays a crucial role in the performance of the ASM preconditioner which will be studied in detail for the above (and other) cases.

The decomposition Equation (5.4) naturally leads to a partitioning of $\mathbf{A}_\kappa$ into blocks as follows:

$$\mathbf{A}_\kappa = \begin{bmatrix} \mathbf{A}_{VV} & \mathbf{A}_{VE} & \mathbf{A}_{VI} \\ \mathbf{A}_{EV} & \mathbf{A}_{EE} & \mathbf{A}_{EI} \\ \mathbf{A}_{IV} & \mathbf{A}_{IE} & \mathbf{A}_{II} \end{bmatrix}.$$

The blocks involving the edge space $X_E := \oplus_{i=1}^3 X_{E_i}$ (e.g. $\mathbf{A}_{EE}, \mathbf{A}_{VE}$) can be further decomposed into subblocks corresponding to the three individual edge spaces of the triangle, viz. $\mathbf{A}_{VE} = [\mathbf{A}_{VE_1}, \mathbf{A}_{VE_2}, \mathbf{A}_{VE_3}]$.

The interior block $\mathbf{A}_{II}$ corresponds to basis functions which are supported locally on $\hat{T}$ and hence can be eliminated locally (even in the case of a mesh of elements). The *static condensation* of interior dofs leads to a reduced system in which the vertex and edge degrees of freedom are coupled by the Schur complement matrix:

$$\mathbf{S}_\kappa := \begin{bmatrix} \mathbf{A}_{VV} & \mathbf{A}_{VE} \\ \mathbf{A}_{EV} & \mathbf{A}_{EE} \end{bmatrix} - \begin{bmatrix} \mathbf{A}_{VI} \\ \mathbf{A}_{EI} \end{bmatrix} \mathbf{A}_{II}^{-1} \begin{bmatrix} \mathbf{A}_{IV} & \mathbf{A}_{IE} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{VV} & \mathbf{S}_{VE} \\ \mathbf{S}_{EV} & \mathbf{S}_{EE} \end{bmatrix}.$$

This procedure of eliminating the interior degrees of freedom, also known as *substructuring*, has been utilized since the early days of finite element analysis [25, 80] initially simply as a means of reducing the number of global unknowns in the linear system. Subsequently, it was realized that [15, 27] the Schur complement matrix arising from substructuring methods can be effectively preconditioned using block-Jacobi preconditioners of the form

$$\mathbf{P}_\kappa := \begin{bmatrix} \mathbf{S}_{VV} & 0 \\ 0 & \operatorname{blockdiag}(\mathbf{S}_{EE}) \end{bmatrix} \tag{5.5}$$

where blockdiag($\mathbf{S}_{EE}$) is the block diagonal matrix with entries $\mathbf{S}_{E_iE_i}, i \in \{1, 2, 3\}$.

Remark. The choice of blockdiag($\mathbf{S}_{EE}$) corresponds to simply discarding the off-diagonal blocks $\mathbf{S}_{E_iE_j}$ with $i \neq j$, resulting in a decoupling of interactions *between* distinct edges whilst maintaining full interaction of the basis functions *within* each of the edge spaces X_{E_i}. In other words, the preconditioner depends only on the space X_{E_i} and *not* on the particular choice basis for X_{E_i}.

Using $\mathbf{P}_\kappa$ as a preconditioner gives rise to a condition number which can be bounded analytically using the standard ASM framework as described, for instance, in [19, 71, 77].

The earliest example of such a preconditioner for the p-version FEM arises in [12] which leads to a condition number which grows as $\mathcal{O}(1+\log^2 p)$ in the case of where the stiffness matrix dominates ($\kappa \to 0$). Conversely in the mass matrix dominated case ($\kappa \to 1$), an ASM preconditioner was constructed in Chapter 2 and shown to have a condition number which is uniformly bounded in p. The key idea in Chapter 2 was the use of a non-standard choice for the vertex functions φ_i used to define the space X_V. In the next section, we investigate how the preconditioner would perform for one of the more standard choices of φ_i mentioned earlier.

5.3 Influence of the choice of nodal basis on the condition number

The performance of the preconditioner Equation (5.5) depends on the choice of basis for the nodal space X_V but not on the choice of basis for the remaining spaces used in the decomposition Equation (5.4).

In particular, it is easy to see that the condition number is independent of the choice of basis for X_I. The elimination of the interior basis functions results in the partial orthogonalization between X_I, and the vertex and edge spaces; specifically, the linear system after elimination is

$$\begin{bmatrix} \mathbf{A}_{VV} & \mathbf{A}_{VE} & \mathbf{A}_{VI} \\ \mathbf{A}_{EV} & \mathbf{A}_{EE} & \mathbf{A}_{EI} \\ \mathbf{A}_{IV} & \mathbf{A}_{IE} & \mathbf{A}_{II} \end{bmatrix} \overset{\text{Elimination}}{\longrightarrow} \begin{bmatrix} \mathbf{S}_{VV} & \mathbf{S}_{VE} & \mathbf{0} \\ \mathbf{S}_{EV} & \mathbf{S}_{EE} & \mathbf{0} \\ \mathbf{A}_{IV} & \mathbf{A}_{IE} & \mathbf{A}_{II} \end{bmatrix},$$

where the block zero matrices in the first two rows imply that the vertex and edge basis functions are now orthogonal to X_I. The condition number is independent of the choice of bases for X_{E_i} thanks to the discussion in Section 5.2.2. Hence, only the choice of a basis for X_V can affect the value of $\mathrm{cond}(\mathbf{P}_\kappa^{-1}\mathbf{S}_\kappa)$.

5.3.1 The pure mass matrix case

We begin by numerically illustrating the impact of different choices of basis for X_V in the case of the pure mass matrix; as we are strictly examining the case of $\kappa = 1$, we will drop the κ notation from $\mathbf{P}$ and $\mathbf{S}$.

The standard nodal basis used in the finite element community is

$$\phi_i^L = \lambda_i, \qquad i \in \{1, 2, 3\} \tag{5.6}$$

where λ_i is the barycentric coordinate on $\hat{T}$ associated with the ith vertex. Denote the resulting Schur complement and preconditioner by $\mathbf{S}_L$ and $\mathbf{P}_L$ respectively. Figure 5.1 shows a quadratic growth $\mathcal{O}(p^2)$ of the condition number $\mathrm{cond}(\mathbf{P}_L^{-1}\mathbf{S}_L)$ for this choice of nodal function.

A spectral element code might well use [63] a nodal basis function whose value at a node is unity and is zero on the Gauss-Lobatto quadrature points on the edges

$$\phi_i^{GL} := \frac{(-1)^{p+1}}{p}\lambda_i P_{p-1}^{(1,1)}(1 - 2\lambda_i), \qquad i \in \{1, 2, 3\}. \tag{5.7}$$

Denote the resulting Schur complement and preconditioner by $\mathbf{S}_{GL}$ and $\mathbf{P}_{GL}$ respectively. The results in Figure 5.1 suggest a logarithmic growth of the condition number $\mathrm{cond}(\mathbf{P}_{GL}^{-1}\mathbf{S}_{GL})$.

While the preconditioner associated with ϕ^{GL} is a drastic improvement over the hat functions λ, it is certainly not uniform with respect to the polynomial order p. In order to achieve a uniform preconditioner, Chapter 2 considers a basis given by

$$\phi_i^\star := \frac{(-1)^{\lfloor p/2 \rfloor + 1}}{\lfloor p/2 \rfloor} \lambda_i P_{\lfloor p/2 \rfloor - 1}^{(1,1)}(1 - 2\lambda_i), \qquad i \in \{1,2,3\}. \tag{5.8}$$

Note that this is the same formula as the one used to define ϕ^{GL} apart from the (essential) difference that p is replaced by $\lfloor p/2 \rfloor$. We denote the resulting Schur complement and preconditioner by $\mathbf{S}_\star$ and $\mathbf{P}_\star$ respectively. It was shown in Chapter 2 that this combination is in fact optimal and does not exhibit *any* growth (i.e. $\operatorname{cond}(\mathbf{P}_\star^{-1}\mathbf{S}_\star) = \mathcal{O}(1)$); see Figure 5.1 for a plot of the condition number. Furthermore, Chapter 6 showed that the preconditioner can be implemented efficiently at a cost of $\mathcal{O}(p^3)$ in a matrix-free manner.

In the next section, a general result relating the growth of the condition number to the choice of nodal basis functions will be given in Theorem 5.3.1. Corollary 5.3.2 provides a theoretical confirmation of the numerical results observed in this section.

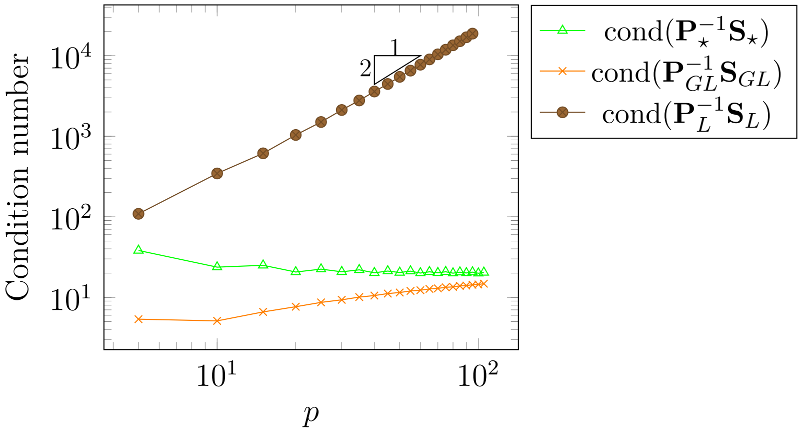


Figure 5.1: Figure illustrating the growth of the condition numbers of the various preconditioned Schur complement system in the case $\kappa = 1$. It is clear that $\operatorname{cond}(\mathbf{P}_\star^{-1}\mathbf{S}_\star)$ remains bounded for all p, while $\operatorname{cond}(\mathbf{P}_{GL}^{-1}\mathbf{S}_{GL})$ exhibits logarithmic growth and $\operatorname{cond}(\mathbf{P}_L^{-1}\mathbf{S}_L)$ exhibits $\mathcal{O}(p^2)$ growth.

5.3.2 Theoretical explanation for the mass matrix case

We begin by stating the main result (whose proof is delayed until Section 5.6) which gives the explicit dependence of the condition number of the ASM preconditioner on the choice of the basis function for X_V for the mass matrix:

Theorem 5.3.1. *Let $\varphi \in X_V$ be any nodal basis function, and let*

$$\Upsilon_p(\varphi) = \min_{\substack{u=\varphi \text{ on } \partial T \\ u \in X}} \|u\|^2 \tag{5.9}$$

where $\|\cdot\|$ denotes the L^2 norm on the triangle $\hat{T}$. Let $\mathbf{S}_\varphi$ and $\mathbf{P}_\varphi$ be the Schur complement and preconditioner constructed using φ as the nodal basis function. Then there exists a constant C independent of p such that the condition number of the preconditioned system satisfies

$$\operatorname{cond}(\mathbf{P}_\varphi^{-1}\mathbf{S}_\varphi) \leq C(1+p^4)\Upsilon_p(\varphi)$$

Equally well, one could use the equivalent definition $\Upsilon_p(\varphi) = \|\varphi - \Pi\varphi\|^2$ where Π is the L^2 orthogonal projection of X onto X_I.

Theorem 5.3.1 shows that it suffices to estimate $\Upsilon_p(\varphi)$ in order to gauge the effect of using φ as the nodal basis function in the substructuring preconditioner. Lemmas 5.7.3 and 5.7.4, and Lemma 6.3 of Chapter 2 show that

$$\Upsilon_p(\phi^{GL}) \sim p^{-4}\log p, \quad \Upsilon_p(\phi^\lambda) \sim p^{-2}, \quad \Upsilon_p(\phi^\star) \sim p^{-4}$$

respectively. An immediate application of Theorem 5.3.1 results in the following:

Corollary 5.3.2. *There exists a constant C independent of p such that*

$$\operatorname{cond}(\mathbf{P}_{GL}^{-1}\mathbf{S}_{GL}) \leq C(1+\log p), \quad \operatorname{cond}(\mathbf{P}_{L}^{-1}\mathbf{S}_{L}) \leq Cp^2, \quad \operatorname{cond}(\mathbf{P}_{\star}^{-1}\mathbf{S}_{\star}) \leq C.$$

The results proven in Corollary 5.3.2 agree with the numerical observations in the previous section.

Theorem 5.3.1 can also be used to predict the performance of other choices of nodal basis. Recently, there has been some interest in the use of Bernstein polynomials for high order finite element analysis [3, 49]. The Bernstein vertex functions are given by

$$\phi_i^B = \lambda_i^p, \qquad i \in \{1,2,3\}. \tag{5.10}$$

How will the Bernstein basis fair in the context of the substructuring preconditioner?

Denote the resulting Schur complement and preconditioner as $\mathbf{S}_B$ and $\mathbf{P}_B$ respectively. We show in Lemma 5.7.7 that $\Upsilon_p(\phi^B) \leq Cp^{-3}$, hence by Theorem 5.3.1, the condition number of the preconditioner grows as $\operatorname{cond}(\mathbf{P}_B^{-1}\mathbf{S}_B) = \mathcal{O}(p)$. Figure 5.2 shows the predicted linear growth of the condition number $\operatorname{cond}(\mathbf{P}_B^{-1}\mathbf{S}_B)$.

5.4 Preconditioner for $0 < \kappa < 1$

We now turn to the problem of developing a preconditioner which is robust in $\kappa \in (0,1)$ for the matrix $\mathbf{A}_\kappa$. Figure 5.3 shows that the preconditioner for the pure mass matrix described in the previous section fails to be robust in the limit $\kappa \to 0$. Equally

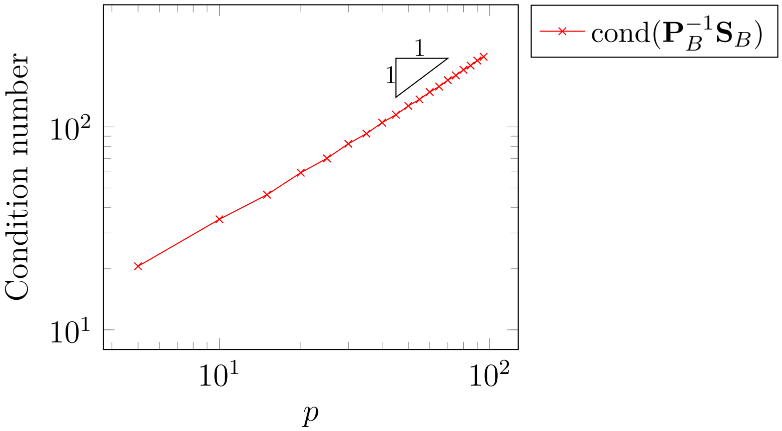


Figure 5.2: Figure illustrating the linear growth of the condition number of the preconditioned Schur complement system constructed using the Bernstein nodal basis for the mass matrix.

well, Figure 5.4 shows that the BCMP preconditioner [12] for the pure stiffness matrix fails to be robust in the limit $\kappa \to 1$.

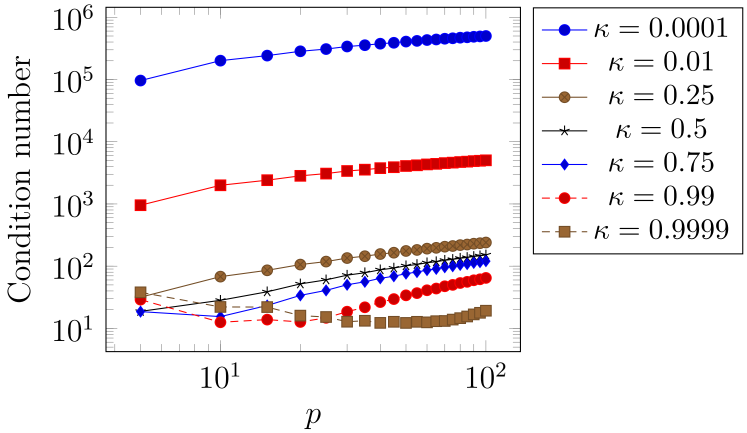


Figure 5.3: Figure illustrating the growth of the condition numbers of the preconditioned Schur complement system constructed using $\phi^\star$ with respect to κ. Note that while the condition number is quite good for the mass-dominant cases of $\kappa \geq .5$ based on Chapter 2, the condition number scales poorly as $\kappa \to 0$.

More precisely, the BCMP preconditioner performs well *asymptotically* with respect to p for a *fixed* κ. It seems that for a fixed κ, BCMP has condition number which has an asymptotic growth of $\mathcal{O}(1 + \log^2 p)$. However, as the value of κ approaches unity $\kappa \to 1$, the condition number exhibits a growth at a rate $\mathcal{O}(p^2)$ in the pre-asymptotic regime before tapering off to $\mathcal{O}(1 + \log^2 p)$ growth as $p \to \infty$ as shown in the following result:

Theorem 5.4.1. *Let* $\mathbf{S}_{\kappa,L}$ *be the Schur complement constructed using the hat functions, and let* $\mathbf{P}_{\kappa,L}$ *be the associated substructuring preconditioner as in Equation* (5.5). *For* $\kappa \in [0,1]$, *there exists constants* C_1, C_2, C_3 *independent of* κ *and* p *such that*

$$\operatorname{cond}(\mathbf{P}_{\kappa,L}^{-1}\mathbf{S}_{\kappa,L}) \leq \begin{cases} \min\{C_1\frac{\kappa}{1-\kappa}(1+\log^2 p), C_2 p^2\} & \kappa \geq 0.5 \\ C_3(1+\log^2 p) & \kappa \leq 0.5 \end{cases}.$$

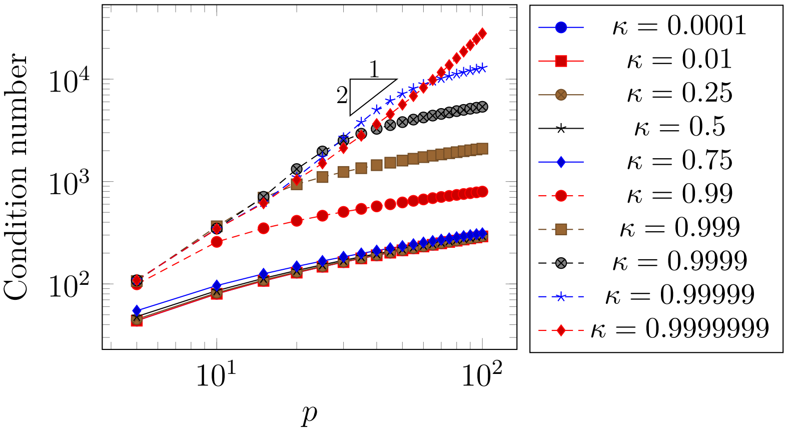


Figure 5.4: Figure illustrating the growth of the condition numbers of the preconditioned Schur complement system constructed using λ vertex basis for different κ (i.e. BCMP). The condition number is quite good for the more stiffness-dominant case as expected. In the mass-dominant case, the condition number grows as $\mathcal{O}(p^2)$ before tapering off to $\mathcal{O}(\log^2 p)$ growth (Theorem 5.4.1).

In search of a uniform in κ preconditioner, first let

$$\mathbf{S}_{\kappa,L} = \begin{bmatrix} \mathbf{S}_{VV,L} & \mathbf{S}_{VE,L} \\ \mathbf{S}_{EV,L} & \mathbf{S}_{EE} \end{bmatrix} \text{ and } \mathbf{S}_{\kappa,\star} = \begin{bmatrix} \mathbf{S}_{VV,\star} & \mathbf{S}_{VE,\star} \\ \mathbf{S}_{EV,\star} & \mathbf{S}_{EE} \end{bmatrix}$$

denote the Schur complement constructed using the hat functions and $\phi^\star$ basis functions respectively. Then, there exists a transformation matrix $\mathbf{\Gamma} \in \mathbb{R}^{3,3(p-1)}$ such that

$$\mathbf{S}_{\kappa,\star} = \begin{bmatrix} \mathbf{I} & \mathbf{\Gamma} \\ 0 & \mathbf{I} \end{bmatrix} \mathbf{S}_{\kappa,L} \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{\Gamma}^T & \mathbf{I} \end{bmatrix}.$$

Details regarding the construction of $\mathbf{\Gamma}$ can be found in Chapter 6. We describe the action (i.e. the inverse) of the preconditioner on the Schur complement $\mathbf{S}_{\kappa,L}$. The new preconditioner for $\mathbf{S}_{\kappa,L}$ is defined as

$$\bar{\mathbf{P}}_{\kappa,\star}^{-1} := \begin{bmatrix} \mathbf{S}_{VV,L}^{-1} & 0 \\ 0 & \text{blockdiag}(\mathbf{S}_{EE}^{-1}) \end{bmatrix} + \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{\Gamma}^T & \mathbf{I} \end{bmatrix} \begin{bmatrix} \text{diag}(\mathbf{S}_{VV,\star})^{-1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{\Gamma} \\ 0 & \mathbf{I} \end{bmatrix}. \tag{5.11}$$

The first term in Equation (5.11) is simply the BCMP substructuring preconditioner whilst the second term constitutes an additional smoothing step on the vertex components (after an appropriate change of basis to $\phi^\star$). Figure 5.5 shows the condition number $\text{cond}(\bar{\mathbf{P}}_{\kappa,\star}^{-1}\mathbf{S}_{\kappa,L})$ for various κ versus p for the preconditioner Equation (5.11). Observe that the simple expedient augmentation with a nodal smoothening step results in an improvement of the condition number by a factor up to two orders of magnitude. We conjecture that the condition number grows as $C(1 + \log^2 p)$ where the constant C is independent of κ and p.

Of course, one is free to use a different choice of nodal basis function such as those discussed in Section 5.3.1 (e.g. ϕ^{GL}, ϕ^{B}), and obtain a different preconditioner by changing $\mathbf{\Gamma}$ and $\mathbf{S}_{VV,\star}$ appropriately. We conjecture that the resulting condition number is bounded below where, by analogy to Theorem 5.3.1, the condition number is governed by $\Upsilon_p(\varphi)$.

Conjecture 5.4.2. *Let $\varphi \in X_V$ be any nodal basis function. For any $\kappa \in [0,1]$, there exists a constant C independent of p and κ such that the condition number*

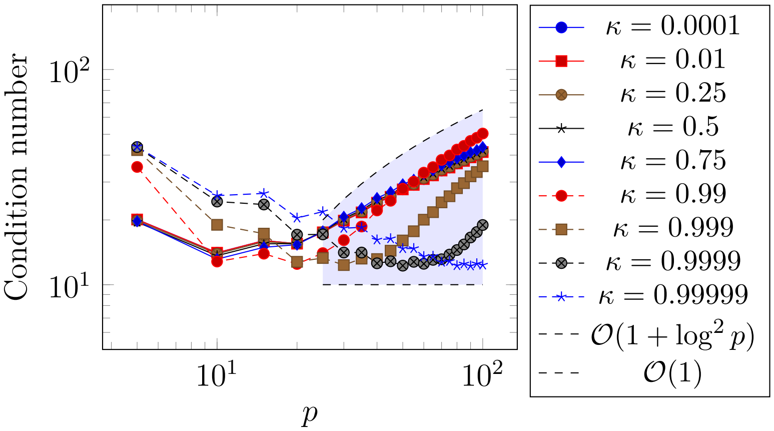


Figure 5.5: Figure illustrating the growth of the condition numbers of the preconditioned Schur complement system constructed using Equation (5.11). We observe $\mathcal{O}(1 + \log^2 p)$ growth in the condition number with no κ degeneration.

using preconditioner Equation (5.11) *with* φ *satisfies*

$$\operatorname{cond}(\bar{\mathbf{P}}_{\kappa,\varphi}^{-1}\mathbf{S}_{\kappa,L}) \leq C \max\{1 + \log^2 p, (1 + \log p)p^4 \Upsilon_p(\varphi)\}.$$

If one assumes that Conjecture 5.4.2 is true, then using the bounds for Υ_p obtained before:

Corollary 5.4.3. *There exists a constant* C *independent of* κ *and* p *such that*

$$\operatorname{cond}(\bar{\mathbf{P}}_{\kappa,GL}^{-1}\mathbf{S}_{\kappa,L}) \leq C \log^2 p, \quad \operatorname{cond}(\bar{\mathbf{P}}_{\kappa,B}^{-1}\mathbf{S}_{\kappa,L}) \leq C p \log p.$$

where $\bar{\mathbf{P}}_{\kappa,GL}^{-1}$ *is the preconditioner constructed with* ϕ^{GL} *and* $\bar{\mathbf{P}}_{\kappa,B}^{-1}$ *is the preconditioner constructed with* ϕ^{B}.

Figure 5.6 and Figure 5.7 shows the actual condition numbers obtained using nodal smootheners based on the Gauss-Lobatto and Bernstein vertex functions respectively.

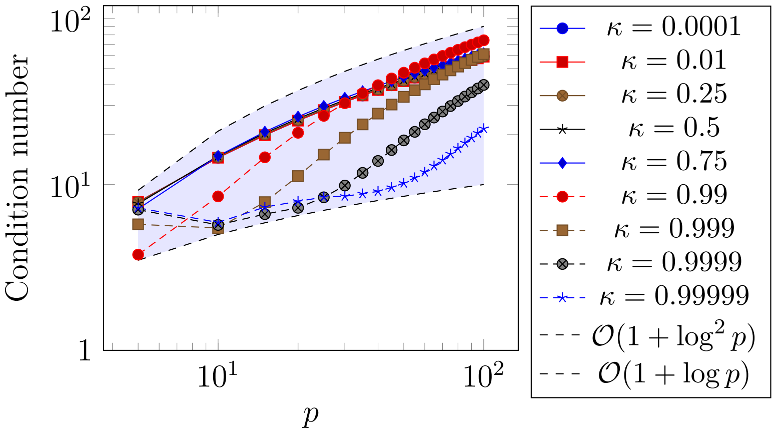


Figure 5.6: Figure illustrating the growth of the condition numbers of the preconditioned Schur complement system using both linear and ϕ^{GL} vertex solves. We observe that the condition number is bounded by $C(1 + \log^2 p)$ from above as predicted by Conjecture 5.4.2 and $c(1 + \log p)$ from below.

5.4.1 Extension to Meshes

The foregoing results were stated in the context of a single element. We can readily generalize the results above to a mesh with multiple elements. Let $\mathcal{T}$ be a partition of a domain Ω into the union of non-overlapping triangular elements such that the non-empty intersection of any two distinct elements is either a common vertex or a single common edge. We further assume the mesh is: shape-regular, there exists constant $\tau > 0$ such that $\rho_K \geq h_K/\tau$ for all element $K \in \mathcal{T}$ where ρ_K, h_K is the inradius and diameter respectively; and quasi-uniform, there exists a constant $1 > c$ such that $h_K \geq c \max_{K \in \mathcal{T}} h_K$. Let $\mathcal{F}_K : \hat{T} \to K$ be a mapping from $\hat{T}$ to an element $K \in \mathcal{T}$ for which there exists constants θ, Θ such that, for all K,

$$\theta h_K^2 \leq |D\mathcal{F}_K| \leq \Theta h_K^2, \qquad \frac{\theta}{h_K^2}\mathbf{I} \leq D\mathcal{F}_K^{-1} D\mathcal{F}_K^{-T} \leq \frac{\Theta}{h_K^2}\mathbf{I} \tag{5.12}$$

where $D\mathcal{F}_K$ is the Jacobian of $\mathcal{F}_K$ and $|D\mathcal{F}_K|$ is its determinant.

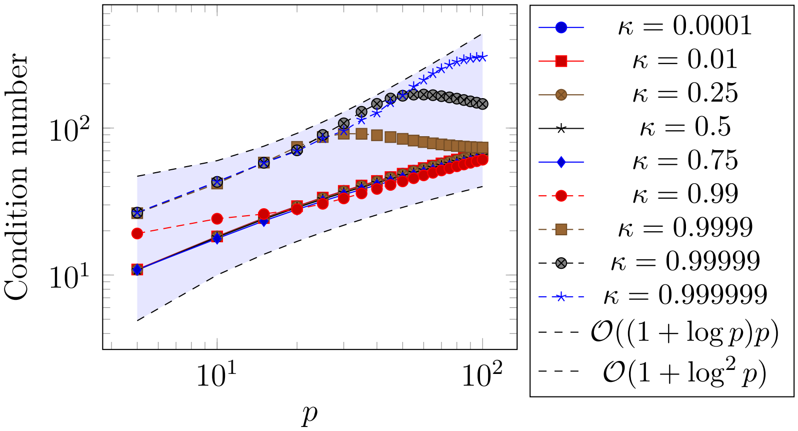


Figure 5.7: Figure illustrating the growth of the condition number of the preconditioned Schur complement system using both linear and ϕ^B vertex solves. We observe that the condition number is bounded by $C(1 + \log p)p$ from above as predicted by Conjecture 5.4.2, and $c(1 + \log^2 p)$ from below.

Let $u \in \mathbb{P}_p(K)$ then by a straightforward change of variables there holds

$$\begin{aligned} \theta h_K^2 \|\hat{u}\|_{\hat{T}}^2 \le \|u\|_K^2 = |D\mathcal{F}_K| \|\hat{u}\|_{\hat{T}}^2 \le \Theta h_K^2 \|\hat{u}\|_{\hat{T}}^2 \\ \theta^2 \|\nabla \hat{u}\|_{\hat{T}}^2 \le \|\nabla u\|_K^2 = |D\mathcal{F}_K| \left\| D\mathcal{F}^{-T} \nabla \hat{u} \right\|_{\hat{T}}^2 \le \Theta^2 \|\nabla \hat{u}\|_{\hat{T}}^2 . \end{aligned} \tag{5.13}$$

where $x = \mathcal{F}_K \hat{x}$, $\hat{u}(\hat{x}) = u(\mathcal{F}_K \hat{x})$, and $\|\cdot\|_\omega^2$ is the L^2-norm on domain ω.

Now let $u \in V = \{u \in H^1(\Omega) : u|_K \in \mathbb{P}_p(K), \forall K \in \mathcal{T}\}$. Applying Equation (5.13) to each element gives

$$\sum_{K \in \mathcal{T}} c_K \|\hat{u}\|_{\kappa_L, \hat{T}}^2 \le \|u\|_{\kappa, \Omega}^2 \le \sum_{K \in \mathcal{T}} C_K \|\hat{u}\|_{\kappa_U, \hat{T}}^2$$

where

$$C_K = \Theta^2(1 - \kappa) + \Theta h_K^2 \kappa, \; \kappa_U = \frac{\Theta h_k^2 \kappa}{C_K}, \; c_K = \theta^2(1 - \kappa) + \theta h_K^2 \kappa, \; \kappa_L = \frac{\theta h_k^2 \kappa}{c_K}$$

and $\|\cdot\|_{\kappa,\omega}^2 = (1 - \kappa)\|\nabla \cdot\|_\omega^2 + \kappa \|\cdot\|_\omega^2$. Note that as $\theta < \Theta$ implies that $\kappa_U < \kappa_L$, we have $\|\hat{u}\|^2_{\kappa_U,\hat{T}} \leq (1-\kappa_U)/(1-\kappa_L)\|\hat{u}\|^2_{\kappa_L,\hat{T}}$ giving

$$\sum_{K\in\mathcal{T}} c_K \|u\|^2_{\kappa_L,\hat{T}} \leq \|u\|^2_{\kappa,\Omega} \leq \sum_{K\in\mathcal{T}} C_K \frac{1-\kappa_U}{1-\kappa_L}\|u\|^2_{\kappa_L,\hat{T}}\,. \tag{5.14}$$

The preconditioner on the mesh is then defined by

$$\bar{\mathbf{P}} = \sum_{K\in\mathcal{T}} \mathbf{\Lambda}_K \bar{\mathbf{P}}_{\kappa_L,\varphi} \mathbf{\Lambda}_K^T$$

where $\mathbf{\Lambda}_K$ is the usual subassembly operator mapping global degrees of freedom to local degrees of freedom on element K.

Assuming the conjecture holds on a single element, then the following result shows that $\bar{\mathbf{P}}$ is robust in κ on a mesh:

Corollary 5.4.4. *There exists a constant C independent of h, p, κ*

$$\mathrm{cond}(\bar{\mathbf{P}}^{-1}\mathbf{S}) \leq C\max\{1+\log^2 p, p^4\Upsilon_p(\varphi)(1+\log p)\}$$

where $\mathbf{S}$ is the Schur complement on the mesh $\mathcal{T}$.

Proof. By quasi-uniformity of the mesh, $\theta^2(1-\kappa) + c\theta h^2\kappa \leq c_K$, and direct manipulation results in

$$\frac{C_K\frac{1-\kappa_U}{1-\kappa_L}}{c_K} \leq \frac{\Theta((c-1)h^2\kappa + (\kappa-1)\Theta)}{\theta((1-c)h^2\kappa + (\kappa-1)\theta)} \leq \frac{\Theta^2}{\theta^2}.$$

Let $\mathbf{S}_{\kappa,K}$ be the local Schur complement matrix, then Equation (5.14) with the above implies

$$\sum_{K\in\mathcal{T}} \mathbf{\Lambda}_K \mathbf{S}_{\kappa,K} \mathbf{\Lambda}_K^T \approx \sum_{K\in\mathcal{T}} \mathbf{\Lambda}_K \mathbf{S}_{\kappa_L,\hat{T}} \mathbf{\Lambda}_K^T$$

with constant independent of h. Finally, Conjecture 5.4.2 implies that

$$\bar{\mathbf{P}}_{\kappa_L,\varphi} \leq \mathbf{S}_{\kappa_L,\hat{T}} \leq C \max\{1+\log^2 p, (1+\log p)p^4 \Upsilon_p(\varphi)\} \bar{\mathbf{P}}_{\kappa_L,\varphi},$$

hence the result follows by taking the summation over the elements. □

5.5 Applications

In this section, we illustrate the performance of the preconditioner Equation (5.11) for two representative applications.

5.5.1 Implicit Time Stepping

First, consider the Gray-Scott system [37, 61], a model of autocatalytic chemical reactions which consists of finding $u(t), v(t)$ such that

$$\begin{aligned} \frac{\partial u}{\partial t} &= -uv^2 + \alpha(1-u) + d_u \Delta u \\ \frac{\partial v}{\partial t} &= uv^2 - (\alpha+\beta)v + d_v \Delta v \end{aligned} \qquad (x,y) \in \Omega, t > 0, \tag{5.15}$$

where α, β, d_u, d_v are constants and Ω is the discretization of the surface of a torus with major radius of 1 and minor radius of $\frac{1}{2}$ (see Figure 5.8 for the meshes). Note that since we are solving the system on a closed surface of a torus, there is no

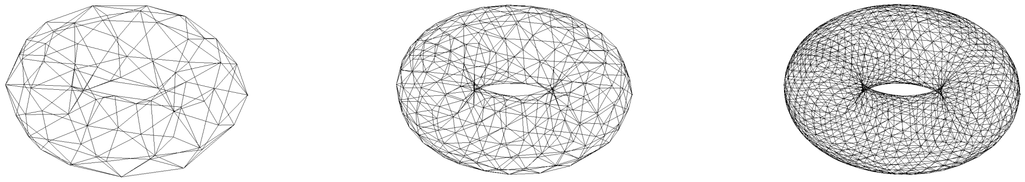


Figure 5.8: Figure to illustrate the meshes used to compute the Gray-Scott example. From left to right, the mesh is of 148, 592 and 2368 triangular elements.

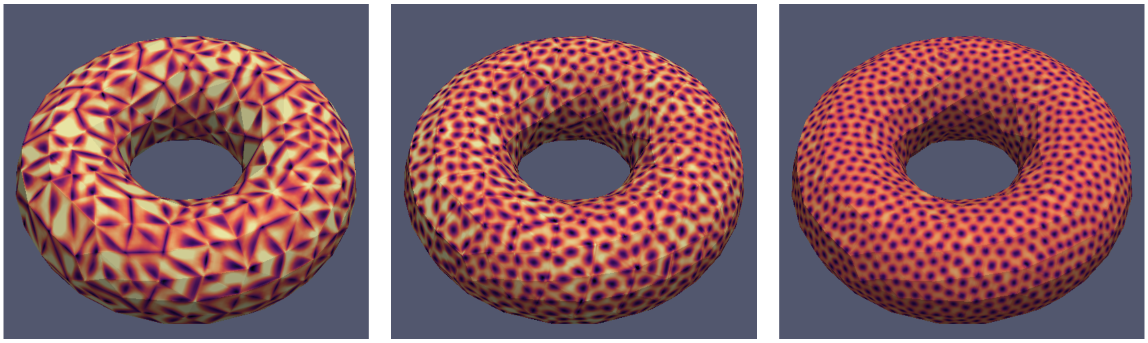


Figure 5.9: Plot of the variable u of the Gray-Scott equations for the constants $\alpha = 0.1, \beta = .05$ on the mesh with 592 elements of order 4, 8, 16 respectively at $t = 10000$ with $\Delta t = 1$.

boundary on which conditions need to be imposed. We take $d_u = 2 \times 10^{-5}, d_v = 10^{-5}$ and initial conditions as specified in [61]; see Figure 5.9 for a plot of the solutions.

An IMEX scheme [66] is used to evolve the solution in time:

$$\begin{aligned} \frac{\mathbf{M}\vec{u}^{n+1} - \mathbf{M}\vec{u}^{n}}{\Delta t} &= -\vec{g}^{n} + \alpha\vec{1} - \alpha\mathbf{M}\vec{u}^{n+1} - \frac{d_u}{2}\left(\mathbf{L}u^{n+1} + \mathbf{L}u^{n}\right) \\ \frac{\mathbf{M}\vec{v}^{n+1} - \mathbf{M}\vec{v}^{n}}{\Delta t} &= \vec{g}^{n} - (\alpha+\beta)\mathbf{M}\vec{v}^{n+1} - \frac{d_v}{2}\left(\mathbf{L}v^{n+1} + \mathbf{L}v^{n}\right) \end{aligned} \tag{5.16}$$

where $\vec{u}^n, \vec{v}^n$ is the finite element approximation at time step n and $\vec{g}^n$ is the nonlinear moment associated with uv^2 at time step n. Observe that Equation (5.16) entails solving systems at each time step involving the matrices

$$\mathbf{M} + \Delta t\left(\alpha\mathbf{M} + \frac{d_u}{2}\mathbf{L}\right) \text{ and } \mathbf{M} + \Delta t\left((\alpha+\beta)\mathbf{M} + \frac{d_v}{2}\mathbf{L}\right), \tag{5.17}$$

each of which is mass-dominated for $\Delta t \to 0$.

In Tables 5.1 to 5.3, we display the condition number of the preconditioned system of the first matrix in Equation (5.17) for $\Delta t = 1, 100, 10000$ respectively for varying mesh size and polynomial order. The condition number does not degenerate in the number of elements as stated in Corollary 5.4.4. Finally, comparing the condition numbers between the three tables, little change is observed in the condition number as the preconditioner is robust in κ.

Table 5.1: Table illustrates the condition number of the preconditioner $\bar{\mathbf{P}}_{\kappa,\star}^{-1}$ applied to the Gray-Scott problem on the torus for $\Delta t = 1$.

Order	148 Elements	592 Elements	2368 Elements
4	27.00	26.61	25.11
8	23.09	21.36	18.42
12	20.57	17.66	14.57
16	18.46	15.09	13.30
20	16.72	13.69	13.10

Table 5.2: Table illustrates the condition number of the preconditioner $\bar{\mathbf{P}}_{\kappa,\star}^{-1}$ applied to the Gray-Scott problem on the torus for $\Delta t = 100$.

Order	148 Elements	592 Elements	2368 Elements
4	24.01	21.10	17.45
8	17.76	13.37	12.41
12	14.31	11.99	13.16
16	13.36	12.33	13.94
20	13.15	12.99	14.95

Table 5.3: Table illustrates the condition number of the preconditioner $\bar{\mathbf{P}}_{\kappa,\star}^{-1}$ applied to the Gray-Scott problem on the torus for $\Delta t = 10000$.

Order	148 Elements	592 Elements	2368 Elements
4	23.58	20.56	16.74
8	17.15	12.83	12.46
12	13.93	11.97	13.23
16	13.20	12.41	14.03
20	13.10	13.16	15.09

5.5.2 Hierarchical Modeling

We now illustrate how the preconditioner can be applied to the hierarchical modeling of thin plates [67]. Let $\Omega_t = \Omega \times (-t,t) \subset \mathbb{R}^3$ where $\Omega \subset \mathbb{R}^2$, the diameter of Ω is of order 1 and $t \ll 1$, and suppose we are solving

$$\begin{aligned} -\Delta u &= 1 && \text{in } \Omega_t, \\ \text{subject to } u &= 0 && \text{on } \partial\Omega \times (-t,t), \\ \text{and } \left.\frac{\partial u}{\partial z}\right|_{z=\pm t} &= f^{\pm} && \text{on } \Omega. \end{aligned} \tag{5.18}$$

where $f^{\pm} \in L^2(\Omega)$. The bilinear form associated with Equation (5.18) is

$$B(u,v) = \int_{-t}^{t}\int_{\Omega} \text{grad } u \cdot \text{grad } v \, dxdydz \tag{5.19}$$

where $u, v \in \mathcal{H}(\Omega_t) = \{w | w \in H^1(\Omega_t), w = 0 \text{ on } \partial\Omega \times (-t,t)\}$.

We perform a standard *hp*-FEM discretization of the space $H_0^1(\Omega)$ paired with a modal expansion of degree n in the transverse coordinate z. That is to say, we seek an approximation from the space

$$V_p^n = \{v : v = \sum_{i=0}^{n} \alpha_i(x,y)\psi_i(z/t) : \alpha_i \in V \cap H_0^1(\Omega), \psi_i \in \mathbb{P}_i([-1,1])\}.$$

Inserting test functions from V_p^n into the bilinear form, we can simplify Equation (5.19)

$$(\psi_i,\psi_j)\left(\sum_{i=0}^{n}\nabla\alpha_i(x,y), \sum_{i=0}^{n}\nabla\beta_i(x,y)\right) + \frac{1}{t^2}(\psi_i',\psi_j')\left(\sum_{i=0}^{n}\alpha_i(x,y), \sum_{i=0}^{n}\beta_i(x,y)\right). \tag{5.20}$$

Let $\mathbf{M}_\psi = (\psi_i,\psi_j), \mathbf{L}_\psi = (\psi_i',\psi_j')$ be the 1D mass and stiffness matrix associated with

$\{\psi_i\}_{i=0}^n$, then the matrix-vector formulation of Equation (5.20) is a sum of Kronecker products

$$\mathbf{M}_\psi \otimes \mathbf{L} + \frac{1}{t^2}\mathbf{L}_\psi \otimes \mathbf{M}.$$

Rather than working with the matrix above directly, we perform the following simplification: we use the polynomials $\vec{\chi} = \mathbf{Q}\vec{\psi}$ where $\mathbf{Q}$ is the matrix of the eigenvectors of the following generalized eigenvalue problem

$$\mathbf{L}_\psi q = \lambda_i \mathbf{M}_\psi q \tag{5.21}$$

with the normalization such that $q^T\mathbf{M}_\psi q = 1$. Using the transformed basis $\{\chi_i\}_{i=0}^n$ will result in the following matrix

$$\mathbf{I}_\psi \otimes \mathbf{L} + \frac{1}{t^2}\mathbf{\Lambda}_\psi \otimes \mathbf{M} \tag{5.22}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{\Lambda}$ is the diagonal matrix of the eigenvalues.

We first note that Equation (5.22) is a block diagonal matrix with entries $\mathbf{L}+\frac{\lambda_i}{t^2}\mathbf{M}$ on the diagonal. The eigenvalues λ_i of Equation (5.21) include 0 and grow as $\mathcal{O}(n^4)$, hence for large model orders, the diagonal blocks can easily range from the pure stiffness matrix to heavily mass-dominated operators. An effective preconditioner for Equation (5.22) is a block-diagonal preconditioner with $\bar{\mathbf{P}}_{\kappa,\star}$ on its diagonal where we set κ appropriately.

In Table 5.4, we display the condition number of the preconditioned system using four elements on the domain $\Omega = (-1,1)^2$ with $t = 1/100$. The condition number does not degenerate as we increase the model order as our preconditioner $\bar{\mathbf{P}}_{\kappa,\star}$ is

Table 5.4: Condition number of the preconditioned system arising from the hierarchical modeling example with $t = 1/100$. The condition number does not degenerate as we increase model order n.

p	$n = 5$	$n = 10$	$n = 15$
4	23.24	23.24	23.24
8	21.95	21.97	21.97
12	21.24	21.33	21.33
16	20.65	20.89	20.90
20	20.07	20.58	20.62

robust for all eigenvalues of Equation (5.21) as shown in Conjecture 5.4.2.

5.6 Proofs of the main results

Proof of Theorem 5.3.1. We first define the appropriate subspaces needed for the ASM framework. Following Equation (2.10), we define $\widetilde{X}$ where $X = X_I \oplus \widetilde{X}$ and $X_I \perp \widetilde{X}$ with respect to the L^2 inner-product. We further decompose $\widetilde{X}$ as

$$\widetilde{X} = \widetilde{X}_V \oplus \widetilde{X}_{E_1} \oplus \widetilde{X}_{E_2} \oplus \widetilde{X}_{E_3} \tag{5.23}$$

where $\widetilde{X}_{E_i} := \{(I - \Pi)u_{E_i} : u_{E_i} \in X_{E_i}\}, \widetilde{X}_V := \{(I - \Pi)u_V : u_V \in X_V\}$ and Π is the L^2 orthogonal projection into X_I. Since we condensed out the interiors and are solving the Schur complement system, we seek the solution in the space $\widetilde{X}$.

Let $(\cdot, \cdot)$ denote the standard L^2 inner-product on $\hat{T}$. The preconditioner Equation (5.5) can then be defined as the following ASM preconditioner

(i) $u_V \in \widetilde{X}_V : (u_V, v_V) = (f, v_V) \quad \forall v_V \in \widetilde{X}_V$.

(ii) For $i \in \{1, 2, 3\}$, $u_{E_i} \in \widetilde{X}_{E_i} : (u_{E_i}, v_{E_i}) = (f, v_{E_i}) \quad \forall v_{E_i} \in \widetilde{X}_{E_i}$.

(iii) $u := u_V + \sum_{i=1}^3 u_{E_i}$ is our solution.

We first give a specific decomposition for all $u \in \widetilde{X}$ in the manner of Equation (5.23). Let $\widetilde{X}_V \ni u_V = \sum_{i=1}^3 u(v_i)\widetilde{\varphi}_i$ where $\widetilde{\varphi}_i = (I - \Pi)\varphi_i$, hence $u - u_V$ vanishes on the vertices and can be written as $u_{E_1} + u_{E_2} + u_{E_3} = u - u_V$ where $u_{E_i} \in \widetilde{X}_{E_i}$.

Combining the above decomposition for $u \in \widetilde{X}$ and triangle inequality, we obtain

$$\|u\|^2 \leq C \left(\|u_V\|^2 + \sum_{i=1}^3 \left\|u_{E_i}\right\|^2 \right).$$

The dependence on p of the condition number of our ASM preconditioner is then simply $\Lambda(p)$ [77], where $\Lambda(p)$ is the function such that for all $u \in \widetilde{X}$

$$\|u_V\|^2 + \sum_{i=1}^3 \left\|u_{E_i}\right\|^2 \leq \Lambda(p)\|u\|^2.$$

By Lemma 2.6.1,

$$\|u_V\|^2 \leq C\|\widetilde{\varphi}\|^2 \max_{i\in\{1,2,3\}} |u_V(v_i)|^2 \leq Cp^4\Upsilon_p(\varphi)\|u\|^2.$$

For the edge terms, we use Lemma 2.6.5 which states that $\left\|u_{E_i}\right\| \leq C\|u - u_V\|$,

$$\left\|u_{E_i}\right\|^2 \leq C\|u - u_V\|^2 \leq C(\|u\|^2 + \|u_V\|^2) \leq C(1 + p^4\Upsilon_p(\varphi))\|u\|^2,$$

hence $\Lambda(p) = C(1 + p^4\Upsilon_p(\varphi))$. □

Proof of Theorem 5.4.1. We first prove the result in the case of $\kappa \geq 0.5$. Schmidt's inequality gives us

$$\kappa\|u\|^2 \leq \|u\|_\kappa^2 := (1-\kappa)\|\nabla u\|^2 + \kappa\|u\|^2 \leq (C(1-\kappa)p^4 + \kappa)\|u\|^2$$

hence $\|u\|_\kappa^2$ is equivalent to the L^2 norm with a constant depending on $Cp^4\frac{1-\kappa}{\kappa} + 1$. Let $p^\star = \left(\frac{\kappa}{C(1-\kappa)}\right)^{1/4}$. For $p \leq p^\star$, we have that $\|u\|_\kappa^2 \approx \|u\|^2$, thus we use the results from Section 5.3.2 regarding the barycentric nodal basis: the condition number of the ASM method grows as $\mathcal{O}(p^2)$.

For the case where $p > p^\star$, we first note that the proof for Lemma 3.3 of [12] easily holds for the H^1 norm (BCMP is actually a preconditioner for the full H^1 norm corresponding to $\mathbf{M} + \mathbf{L}$). Next, we trivially note that

$$(1-\kappa)\|u\|_{H^1}^2 \leq \|u\|_\kappa^2 \leq \kappa\|u\|_{H^1}^2 .$$

Hence, $\|u\|_\kappa^2$ is equivalent to the H^1 norm with the constant $\frac{\kappa}{1-\kappa}$. Thus the growth of applying BCMP will be bounded by $\frac{\kappa}{1-\kappa}C_1(1+\log^2 p)$ where C_1 is the constant from Lemma 3.3 of BCMP.

As for the $\kappa \leq 0.5$ case, we first define the subspaces and inner-products for the ASM. We now associate X with the norm $\|\cdot\|_\kappa = (1-\kappa)\|\nabla\cdot\|^2 + \kappa\|\cdot\|^2$ and denote $(\cdot,\cdot)_\kappa$ as the associated inner-product. Similar to the proof of Theorem 5.3.1, we define $\widetilde{X}$ such that $X = \widetilde{X} + X_I$ and $\widetilde{X} \perp X_I$ with respect to the $(\cdot,\cdot)_\kappa$ inner product. We further decompose $\widetilde{X}$ as

$$\widetilde{X} = \widetilde{X}_L + \sum_{i=1}^{3} \widetilde{X}_{E_i} \tag{5.24}$$

where $\widetilde{X}_{E_i} = \{(I - \Pi_\kappa)u_{E_i} : u_{E_i} \in X_{E_i}\}$, and $\widetilde{X}_L = \{(I - \Pi_\kappa)v : v \in \mathbb{P}_1\}$ where $\Pi_\kappa : X \to X_I$ is the orthogonal projection onto X_I with respect to the $(\cdot,\cdot)_\kappa$ inner-product. A critical property of $\widetilde{X}$ and its subspaces is that for all $u \in \widetilde{X}$

$$\|u\|_\kappa^2 = \min_{v=u \text{ on } \partial\hat{T}, v\in X} \|v\|_\kappa^2 . \tag{5.25}$$

As we performed static condensation, we again seek the solution in the space $\widetilde{X}$.

The action of BCMP for $f \in \widetilde{X}$ is given by u defined as follows

(i) $u_L \in \widetilde{X}_L : (u_L, v_L) = (f, v_L) \quad \forall v_L \in \widetilde{X}_L$.

(ii) for $i = 1, 2, 3$: $u_{E_i} \in \widetilde{X}_{E_i} : (u_{E_i}, v_{E_i}) = (f, v_{E_i}) \quad \forall v_{E_i} \in \widetilde{X}_{E_i}$.

(iii) $u := u_L + \sum_{i=1}^3 u_{E_i}$.

Similar to the proof of Theorem 5.3.1, we wish to find a function $\Lambda(p)$ such that for all $u \in \widetilde{X}$

$$\|u_L\|_\kappa^2 + \sum_{i=1}^3 \left\|u_{E_i}\right\|_\kappa^2 \leq \Lambda(p)\|u\|_\kappa^2 .$$

By [12], there exists functions $u_L^*, u_{E_i}^* \in X$ (not necessarily in $\widetilde{X}_L, \widetilde{X}_{E_i}$ due to the difference in norms) such that for all $u \in \widetilde{X}$

$$\|\nabla u_L^*\|^2 + \sum_{i=1}^3 \left\|\nabla u_{E_i}^*\right\|^2 \leq C(1 + \log^2 p)\|\nabla u\|^2 . \tag{5.26}$$

Define $u_L \in \widetilde{X}_L$ and $u_{E_i} \in \widetilde{X}_{E_i}$ such that $u_L|_{\partial\hat{T}} = u_L^*|_{\partial\hat{T}}$ and $u_{E_i}|_{\partial\hat{T}} = u_{E_i}^*|_{\partial\hat{T}}$. By Equation (5.25), Equation (5.26) and an application of Poincare's ienquality, we have for $i = 1, 2, 3$

$$\begin{aligned}\left\|u_{E_i}\right\|_\kappa^2 &\leq \left\|u_{E_i}^*\right\|_\kappa^2 \\ &= (1-\kappa)\left\|\nabla u_{E_i}^*\right\|^2 + \kappa\left\|u_{E_i}^*\right\|^2 \\ &\leq (1-\kappa)\left\|\nabla u_{E_i}^*\right\|^2 + C\kappa\left\|\nabla u_{E_i}^*\right\|^2 \\ &\leq C(1+\log^2 p)\|\nabla u\| \leq C(1+\log^2 p)\|u\|_\kappa^2 .\end{aligned}$$

Finally, by an application of triangle's inequality, we have that

$$\|u_L\|_\kappa^2 = \left\|u - \sum_{i=1}^{3} u_{E_i}\right\|_\kappa^2 \leq C(1+\log^2 p)\|u\|_\kappa^2$$

and we are done.

□

Sketch of of Conjecture 5.4.2. We again define the appropriate spaces for the ASM framework. Let $\widetilde{X}$ be such that $X = \widetilde{X} + X_I$ and $\widetilde{X} \perp X_I$ with respect to the $(\cdot,\cdot)_\kappa$ inner product. We further decompose $\widetilde{X}$ as

$$\widetilde{X} = \widetilde{X}_L + \sum_{i=1}^{3} \widetilde{X}_{V_i} + \sum_{i=1}^{3} \widetilde{X}_{E_i} \tag{5.27}$$

where $\widetilde{X}_{E_i} = \{(I - \Pi_\kappa)u_{E_i} : u_{E_i} \in X_{E_i}\}$, $\widetilde{X}_L = \{(I - \Pi_\kappa)v : v \in \mathbb{P}_1\}$ and $\widetilde{X}_{V_i} = \text{range}\{(I - \Pi_\kappa)\varphi_i\}$ where $\Pi_\kappa : X \to X_I$ is the orthogonal projection onto X_I with respect to the $(\cdot,\cdot)_\kappa$ inner-product.

The action of the preconditioner Equation (5.11) for $f \in \widetilde{X}$ is given by u defined as follows

(i) $u_0 \in \widetilde{X}_L : (u_0, v_0) = (f, v_0) \quad \forall v_0 \in \widetilde{X}_L$.

(ii) for $i = 1, 2, 3$: $u_{V_i} \in \widetilde{X}_{V_i} : (u_{V_i}, v_{V_i}) = (f, v_{V_i}) \quad \forall v_{V_i} \in \widetilde{X}_{V_i}$.

(iii) for $i = 1, 2, 3$: $u_{E_i} \in \widetilde{X}_{E_i} : (u_{E_i}, v_{E_i}) = (f, v_{E_i}) \quad \forall v_{E_i} \in \widetilde{X}_{E_i}$.

(iv) $u := u_0 + \sum_{i=1}^{3} u_{V_i} + \sum_{i=1}^{3} u_{E_i}$.

For the case of $\kappa \geq .5$, which is the mass dominated case, we proceed much as the proof of Theorem 5.3.1. We need to first define the decomposition

$$u = u_0 + \sum_{i=1}^{3} \Big(u_{V_i} + u_{E_i} \Big)$$

for every $u \in \widetilde{X}$ where $u_0 \in \widetilde{X}_L, u_{V_i} \in \widetilde{X}_{V_i}, u_{E_i} \in \widetilde{X}_{E_i}$.

Let $u_0 = 0$ and $u_{V_i} = u(v_i)\widetilde{\varphi}_i \in \widetilde{X}_{V_i}$ where $\widetilde{\varphi}_i = (I - \Pi_\kappa)\varphi_i$. Furthermore, let $\widetilde{\varphi}_{i,L^2} = (I - \Pi)\varphi_i$ where Π is the L^2 projection on X_I as before; note that $\left\|\widetilde{\varphi}_{i,L^2}\right\|^2 = \Upsilon_p(\varphi)$ by definition, and that $\widetilde{\varphi}_i = \widetilde{\varphi}_{i,L^2}$ on $\partial\hat{T}$. We have for $i = 1, 2, 3$

$$\begin{aligned}
\left\|u_{V_i}\right\|_\kappa^2 &= \min_{v = u_{V_i} \text{ on } \partial\hat{T}, v \in X} (1-\kappa)\|\nabla v\|^2 + \kappa\|v\|^2 \\
&\leq (1-\kappa)\left\|u(v_i)\nabla\widetilde{\varphi}_{i,L^2}\right\|^2 + \kappa\left\|u(v_i)\widetilde{\varphi}_{i,L^2}\right\|^2 \\
&\leq (1-\kappa)\|u\|_{L^\infty}^2\left\|\nabla\widetilde{\varphi}_{i,L^2}\right\|^2 + \kappa\|u\|_{L^\infty}^2\left\|\widetilde{\varphi}_{i,L^2}\right\|^2 \\
&\leq (1-\kappa)\|u\|_{L^\infty}^2\, p^4\Upsilon_p(\varphi) + \kappa\|u\|_{L^\infty}^2\, \Upsilon_p(\varphi) \\
&\leq C(1-\kappa)(1+\log p)\|u\|_{H^1}^2\, p^4\Upsilon_p(\varphi) + \kappa p^4\|u\|^2\, \Upsilon_p(\varphi) \\
&\leq C(1+\log p)p^4\Upsilon_p(\varphi)\|u\|_\kappa^2 .
\end{aligned}$$

where we used Schmidt's inequality to bound $\left\|\nabla\widetilde{\varphi}_{i,L^2}\right\|^2$ with $p^4\Upsilon_p(\varphi)$, Theorem 6.2 from [12], Lemma 2.6.1.

We see that $u_1 := u - \sum_{i=1}^{3} u_{V_i}$ vanishes at the vertices, hence by triangle inequality $\|u_1\|_\kappa^2 \leq C(1+\log p)p^4 \Upsilon_p(\varphi)\|u\|_\kappa^2$. Unfortunately, we do not have an extension theorem for the edges similar to Theorem 7.4 of [12] or Lemma 2.6.5 of Chapter 2. Without such an estimate, we are only able to bound the edge contributions in a suboptimal way: let $u_{E_i^*}$ be the 2D Munoz-Sola extension [42, 55] such that $u_{E_i}^*|_{\gamma_i} = u_1|_{\gamma_i}$. Theorem 3.2 of [42] states that $\left\|u_{E_i}^*\right\|^2 \leq \|u_1\|_{\gamma_i}^2$ where $\|\cdot\|_{\gamma_i}^2$ is the L^2 norm over the edge γ_i, and Theorem 6.6 and 7.4 of [12] states that $\left\|u_{E_i}^*\right\|_{H^1}^2 \leq (1+\log^2 p)\|u\|_{H^1}^2$. Let $u_{E_i}|_{\partial\hat{T}} = u_{E_i}^*|_{\partial\hat{T}}$, then by Equation (5.25), inverse inequality on the edge $\|u_1\|_{\gamma_i} \leq p\|u_1\|$ [79], and the inequalities above, we have

$$\begin{aligned}
\left\|u_{E_i}\right\|_\kappa^2 \leq \left\|u_{E_i}^*\right\|_\kappa^2 &= (1-\kappa)\left\|\nabla u_{E_i}^*\right\|^2 + \kappa\left\|u_{E_i}^*\right\|^2 \\
&\leq (1-\kappa)\left\|u_{E_i}^*\right\|_{H^1}^2 + \kappa\|u_1\|_{\gamma_i}^2 \\
&\leq (1-\kappa)(1+\log^2 p)\|u\|_{H^1}^2 + \kappa p^2\|u_1\|_{\gamma_i}^2 \\
&\leq (1-\kappa)(1+\log^2 p)\|u\|_{H^1}^2 + p^2\|u_1\|_\kappa^2 \\
&\leq (1-\kappa)(1+\log^2 p)\|u\|_{H^1}^2 + (1+\log p)p^6\Upsilon_p(\varphi)\|u\|_\kappa^2 \\
&\leq C\max\{1+\log^2 p, (1+\log p)p^6\Upsilon_p(\varphi)\}\|u\|_\kappa^2
\end{aligned}$$

for each $i = 1, 2, 3$. Note that this does not reflect the numerical results, and the poor bound is due to the lack of an extension result which is independent of κ.

For the case of $\kappa \leq .5$, which is the stiffness dominated case, we only use the subspaces $\widetilde{X}_0, \widetilde{X}_{E_i}$. Since the preconditioner is equivalent to the result in Theorem 5.4.1, the result follows trivially and the condition number is strictly bounded by $\mathcal{O}(1+\log^2 p)$, independent of the choice of the augmented nodal basis.

□

5.7 Technical Lemmas for the calculation of Υ_p

In this section, we estimate $\Upsilon_p(\varphi)$ for ϕ^{GL}, λ, and ϕ^B.

5.7.1 Gauss-Lobatto Vertex Function

We first prove two auxiliary lemmas needed to calculate $\Upsilon_p(\phi^{GL})$.

Lemma 5.7.1. *For $i \geq 1, j > 0$, we have the following equality if $j \geq i-1$*

$$I_{i-1,j} := \int_{-1}^{1} (1-x)^2(1+x) P_{i-1}^{(2,1)} P_j^{(2,2)}\, dx = (-1)^{j-i+1} \frac{16i}{(j+3)(j+4)}$$

else $I_{i-1,j} = 0$.

Proof. For $j < i-1$, $I_{i-1,j} = 0$ by orthogonality. For $j \geq i-1$, first let $c_{nm} := \int_{-1}^{1} (1-x)^2(1+x) P_n^{(2,1)} P_m^{(2,1)}\, dx = \delta_{nm} \frac{16}{2n+4} \frac{n+1}{n+3}$. From [24], we have that

$$P_n^{(2,2)} = a_n P_n^{(2,1)} - b_n P_{n-1}^{(2,2)}$$

where $a_n = \frac{2n+4}{n+4}$ and $b_n = \frac{n+2}{n+4}$. Substituting the identity into the integral

$$\begin{aligned} \int_{-1}^{1} (1-x)^2(1+x) P_{i-1}^{(2,1)} P_j^{(2,2)}\, dx &= a_j c_{i-1,j} - b_j \int_{-1}^{1} (1-x)^2(1+x) P_{i-1}^{(2,1)} P_{j-1}^{(2,2)}\, dx \\ &= -b_j I_{i-1,j-1} \\ &\ \ \vdots \\ &= (-1)^{j-i+1} a_{i-1} c_{i-1,i-1} \Pi_{k=i}^{j} b_k \end{aligned}$$

where we iterate this process until $I_{i-1,i-2} = 0$. Direct simplification yields the result.

□

Lemma 5.7.2. *For $i \geq 1, j > 0$, we have the following equality*

$$J_{i-1,j} := \int_{-1}^{1} (1-x)^2(1+x)P_{i-1}^{(1,1)}P_j^{(2,2)}\,dx = \begin{cases} (-1)^{j-i+1}\frac{16i}{(j+3)(j+4)} & j \geq i-1 \\ -\frac{16(i-1)i}{(2i+1)(i+1)(i+2)} & j = i-2 \\ 0 & j < i-2 \end{cases}$$

else $J_{i-1,j} = 0$.

Proof. From [24], we have that

$$P_{i-1}^{(1,1)} = \frac{i+2}{2i+1}P_{i-1}^{(2,1)} - \frac{i}{2i+1}P_{i-2}^{(2,1)}.$$

Hence

$$\begin{aligned} J_{i-1,j} &= \int_{-1}^{1} (1-x)^2(1+x)\left(\frac{i+2}{2i+1}P_{i-1}^{(2,1)} - \frac{i}{2i+1}P_{i-2}^{(2,1)}\right)P_j^{(2,2)}\,dx \\ &= \frac{i+2}{2i+1}I_{i-1,j} - \frac{i}{2i+1}I_{i-2,j}. \end{aligned}$$

From here, we distinguish between three cases depending on $I_{i-1,j}$. If $j \geq i-1$, then $I_{i-1,j}$ is non-zero, and $J_{i-1,j} = \frac{i+2}{2i+1}I_{i-1,j} - \frac{i}{2i+1}I_{i-2,j} = (-1)^{j-i+1}\frac{16i}{(j+3)(j+4)}$. If $j = i-2$, then $I_{i-1,j} = 0$, and we have $J_{i-1,j} = -\frac{i}{2i+1}I_{i-2,j} = -\frac{16(i-1)i}{(2i+1)(i+1)(i+2)}$. Otherwise for $j < i-2$, we have that integral is 0. □

With the above integrals, we can bound the norm of the minimal extension of the Gauss-Lobatto basis functions.

Lemma 5.7.3. *The Lobatto basis function of degree p satisfies the bound*

$$\Upsilon_p(\phi^{GL}) \sim p^{-4}\log p$$

Proof. First, let p be even and let $q = p/2$. Let w_j be coefficients such that

$$\phi^{GL} - \phi^{\star}|_{\gamma} = (1-x)\left(c_p P_{p-1}^{(1,1)} - c_q P_{q-1}^{(1,1)}\right) = \sum_{j=0}^{p-2} w_j (1-x^2) P_j^{(2,2)}(x)$$

where $\gamma = \{(x,-1) : -1 \leq x \leq 1\}$ and $c_p = \frac{(-1)^{p+1}}{2p}$. Due to orthogonality, we have that

$$\begin{aligned} w_j &= \frac{\int_{-1}^{1}(1-x)\left(c_p P_{p-1}^{(1,1)} - c_q P_{q-1}^{(1,1)}\right)(1-x^2)P_j^{(2,2)}\,dx}{\left\|(1-x^2)P_j^{(2,2)}\right\|_{[-1,1]}^2} \\ &= \frac{(2j+5)(j+3)(j+4)}{32(j+1)(j+2)}(c_p J_{p-1,j} - c_q J_{q-1,j}). \end{aligned}$$

Using Lemma 5.7.2, we have that

1. $j = 0, \ldots, q-3$: $w_j = 0$.
2. $j = q-2$: $w_j = \frac{(2q+1)(q+1)(q+2)(-c_q J_{q-1,q-2})}{32q(q-1)} = \frac{(-1)^{q+1}}{4q}$
3. $j = q-1, \ldots, p-3$: $w_j = -c_q J_{q-1,j}\frac{(2j+5)(j+3)(j+4)}{32(j+1)(j+2)} = \frac{(-1)^{j+1}(2j+5)}{4(j+1)(j+2)}$.
4. $j = p-2$: $w_{p-2} = \frac{(2j+5)(j+3)(j+4)}{32(j+1)(j+2)}(c_p J_{p-1,p-2} - c_q J_{q-1,p-2}) = \frac{(-1)^{p+1}(p+2)}{4(p-1)p}$.

Using Lemma 2.6.3, we can calculate

$$\begin{aligned} \Upsilon_p(\phi^{GL} - \phi^{\star}|_{\gamma}) &\sim p^{-4} + \sum_{j=q-1}^{p-3} \frac{1}{j^3 p(p-j-1)} \\ &\sim p^{-4} + \frac{1}{p}\int_{p/2-1}^{p-3} \frac{1}{x^3(p-x-1)}\,dx \sim \frac{\log(p)}{p^4}. \end{aligned}$$

The asymptotics follows by recalling that $\Upsilon_p(\phi^\star) \sim p^{-4}$, Lemma 2.6.5 and using the triangle inequality. For odd values of p, the result follows in an analogous manner with $q = \lfloor p/2 \rfloor$. □

5.7.2 Barycentric Vertex Functions

We can now easily bound $\Upsilon_p(\phi^\lambda)$ using $\Upsilon_p(\phi^{GL})$.

Lemma 5.7.4. *The barycentric coordinates satisfies the bound*

$$\Upsilon_p(\phi^\lambda) \sim p^{-2}$$

Proof. We proceed similarly to Lemma 5.7.3 and seek w_j such that

$$\phi^\lambda - \phi^{GL}|_\gamma = (1-x)\left(\frac{1}{2} - c_p P_{p-1}^{(1,1)}\right) = \sum_{j=0}^{p-2} w_j (1-x^2) P_j^{(2,2)}(x)$$

where $\gamma = \{(x,-1) : -1 \leq x \leq 1\}$ and $c_p = \frac{(-1)^{p+1}}{2p}$. Due to orthogonality, we have for $j = 0, \ldots, p-2$

$$w_j = \frac{\int_{-1}^{1} (1-x)\left(\frac{1}{2} - c_p P_{p-1}^{(1,1)}\right)(1-x^2) P_j^{(2,2)}\, dx}{\left\| (1-x^2) P_j^{(2,2)} \right\|_{[-1,1]}^2} = \frac{\frac{8(-1)^j}{(j+3)(j+4)} - c_p J_{p-1,j}}{\frac{32(j+1)(j+2)}{(2j+5)(j+3)(j+4)}}.$$

Hence by Lemma 5.7.2, $w_j = \frac{(-1)^j(2j+5)}{4(j+1)(j+2)}$ for $j = 0, \ldots, p-3$ and $w_{p-2} = \frac{(-1)^p(p+2)}{4(p-1)p}$.

Using Lemma 2.6.3

$$\begin{aligned}\Upsilon_p(\phi^\lambda - \phi^{GL}|_\gamma) &\sim p^{-4} + \sum_{j=0}^{p-3} \frac{1}{(j+1)^3 p(p-j-1)} \\ &\sim p^{-4} + \frac{1}{p}\int_0^{p-3} \frac{1}{(x+1)^3(p-x-1)}\,dx \sim \frac{1}{p^2}.\end{aligned}$$

The results follows as $\Upsilon_p(\phi^{GL}) \sim p^{-4}\log p$. □

5.7.3 Bernstein Vertex Functions

We will need the following two lemmas to bound the Bernstein polynomial nodal functions

Lemma 5.7.5. *For all* $p \geq 2$*,*

$$\left(\frac{1-x}{2}\right)^p - \left(\frac{1-x}{2}\right) = (1-x^2)\sum_{j=0}^{p-2} c_j^p P_j^{(2,2)}$$

where

$$c_j^p = -\frac{(-1)^j\binom{2p+1}{p-2-j}}{4(2p+1)\binom{2p}{p-1}}\frac{(2j+5)(j^2+5j+p+6)}{(j+1)(j+2)}.$$

Proof. We proceed with an inductive proof on the polynomial order p. For $p = 2$, it is trivial to verify. Assume that we have proven the identity for p, we consider the $P_j^{(2,2)}$ expansion of

$$\begin{aligned}\left[\left(\frac{1-x}{2}\right)^{p+1} - \left(\frac{1-x}{2}\right)\right] - \left[\left(\frac{1-x}{2}\right)^p - \left(\frac{1-x}{2}\right)\right] &= -2^{-p-1}(x+1)(1-x)^p \\ &= (1-x^2)\sum_{j=0}^{p-1} d_j^p P_j^{(2,2)}.\end{aligned}$$

By orthogonality, we have that

$$d_j^p = -\frac{\int_{-1}^{1}(1-x)^p(1+x)(1-x^2)P_j^{(2,2)}\,dx}{2^{p+1}\left\|(1-x^2)P_j^{(2,2)}\right\|^2}.$$

The numerator can be evaluated using identity 18.17.36 of [24]. Finally, it is straightforward manipulation to verify that $c_j^p + d_j^p = c_j^{p+1}$. □

Lemma 5.7.6. *For $j = 0, \dots, p-2$, let*

$$Q_j := 1 - \frac{(j^2+5j+p+6)}{(2p+1)\binom{2p}{p-1}}\binom{2p+1}{p-2-j}.$$

Then $Q_j \leq 1$ for all $j = 0, \dots, p-2$. Furthermore, suppose that $(j+1)(j+2) \leq \frac{1}{2}(p+1)$, then $Q_j \leq \frac{(j+1)(j+2)}{p}$.

Proof. We note that $Q_0 \leq 1$. Furthermore, it is not hard to show that $(j^2+5j+p+6)\binom{2p+1}{p-2-j}$ is a decreasing sequence in j, hence $Q_j \leq 1$.

For the second inequality, first bring Q_j to a common denominator of $p\binom{2p+1}{p}$, then the numerator is

$$\underbrace{p\binom{2p+1}{p} - p\binom{2p+1}{p-2-j}}_{A} - \underbrace{(j+2)(j+3)\binom{2p+1}{p-2-j}}_{B}.$$

Examining A at a greater detail, we have

$$A = p\sum_{r=0}^{j+1}\binom{2p+1}{p-r} - \binom{2p+1}{p-r-1} = \frac{p}{p+1}\sum_{r=0}^{j+1}(r+1)\binom{2p+2}{p-r}.$$

Now using the identity $\sum_{j=1}^{n+1} jc_j = (n+1)^2c_{n+1} + \sum_{j=1}^{n} j(jc_j - (j+1)c_{j+1})$, then A

is equal to

$$\frac{p(j+2)^2}{p+1}\binom{2p+2}{p-j-1}+\frac{p}{p+1}\sum_{r=0}^{j}(r+1)\left((r+1)\binom{2p+2}{p-r}-(r+2)\binom{2p+2}{p-r-1}\right)$$
$$=\frac{p(j+2)^2}{p+1}\binom{2p+2}{p-j-1}+\frac{p}{(p+1)(2p+3)}\sum_{r=0}^{j}(r+1)(2r^2+6r+3-p)\binom{2p+3}{p-r}.$$

We note that $(j+1)(j+2)\le\frac{1}{2}(p+1) \implies 2j^2+6j+3-p\le 0$, hence

$$A\le\frac{p(j+2)^2}{p+1}\binom{2p+2}{p-j-1}.$$

Finally, note that by simplification,

$$A+B\le\frac{p(j+2)^2}{p+1}\binom{2p+2}{p-j-1}+B=(j+1)(j+2)\binom{2p+1}{p-j-1}.$$

Hence,

$$Q_j\le\frac{(j+1)(j+2)\binom{2p+1}{p-j-1}}{p\binom{2p+1}{p}}\le\frac{(j+1)(j+2)}{p}.$$

□

Now, we can bound the Bernstein nodal basis.

Lemma 5.7.7. *The minimal extension of the degree p nodal Bernstein polynomial satisfies the bound*

$$\Upsilon_p(\phi^B)\le Cp^{-3}.$$

Proof. Proceeding as before, let w_j such that

$$\begin{aligned}\lambda^p - \phi^{GL}|_\gamma &= \left(\left(\frac{1-x}{2}\right)^p - \left(\frac{1-x}{2}\right)\right) - \left(\left(\frac{1-x}{2}\right) - \frac{(-1)^{p+1}}{2p}(1-x)P_{p-1}^{(1,1)}\right) \\ &= (1-x^2)\sum_{j=0}^{p-2} w_j P_j^{(2,2)}.\end{aligned}$$

Combining Lemma 5.7.5 and the expansion in Lemma 5.7.4, we have that for $j = 0, \dots, p-3$

$$\begin{aligned} w_j &= \frac{(-1)^j(2j+5)}{4(j+1)(j+2)} - \frac{(-1)^j}{4(2p+1)\binom{2p}{p-1}}\binom{2p+1}{p-2-j}\frac{(2j+5)(j^2+5j+p+6)}{(j+1)(j+2)} \\ &= \frac{(-1)^j(2j+5)}{4(j+1)(j+2)}\left(1 - \frac{(j^2+5j+p+6)}{(2p+1)\binom{2p}{p-1}}\binom{2p+1}{p-2-j}\right) = \frac{(-1)^j(2j+5)}{4(j+1)(j+2)}Q_j\end{aligned}$$

and $j = p-2$ we have

$$w_{p-2} = -\frac{(-1)^p(p+2)(p-\binom{2p}{p-1})}{4(p-1)p\binom{2p}{p-1}} \sim 1/p.$$

Now, using the fact that $\sqrt{p} \le \frac{1}{2}(p+1)$, $Q_j \le 1$ and Lemma 5.7.6, we have

$$\begin{aligned}\Upsilon_p(\phi^B - \phi^{GL}) &\le \mathcal{O}(p^{-4}) + \sum_{j=0}^{\sqrt{p}} \frac{Q_j^2}{(j+1)^3(p+j+1)(p-j-1)} + \sum_{j=\sqrt{p}}^{p-3}\frac{1}{j^3(p+j)(p-j-1)} \\ &\le \mathcal{O}(p^{-4}) + \frac{1}{p^4}\sum_{j=0}^{\sqrt{p}} j + \frac{1}{p}\int_{\sqrt{p}}^{p-3}\frac{1}{x^3(p-x-1)}\,dx \le \mathcal{O}(p^{-3})\end{aligned}$$

The results follows from recalling $\Upsilon_p(\phi^{GL}) \sim p^{-4}\log p$. □

5.8 Conclusions

The current work first analyzed the impact of the nodal space in the construction of substructuring preconditioners for the mass matrix, and developed an ASM preconditioner Equation (5.11) which is robust in κ for $\mathbf{A}_\kappa$. We concluded that $\Upsilon_p(\phi)$ is the key quantity which affects the quality of the mass matrix preconditioner. As for the preconditioner for $\mathbf{A}_\kappa$, we showed that simply complementing the existing BCMP preconditioner [12] with an appropriate Jacobi smoothening step allowed one to obtain robustness in κ. Surprisingly, $\Upsilon_p(\phi)$ also plays an impactful role in the condition number of Equation (5.11).

We also wish to mention the computational drawbacks of the preconditioner Equation (5.11). Unlike in Chapter 6, there does not yet exist efficient methods to invert the interior dofs for $\mathbf{A}_\kappa$ or edges in the Schur complement. Thus, the preconditioner is more costly than the pure mass matrix preconditioner and, in our experience, we recommend the mass matrix preconditioner as implemented in Chapter 6 for the transient problems.

Chapter

SIX

Efficient Implementation of p-FEM in 2D

6.1 Introduction

Finally, we turn the implementation aspects of *hp*-FEM. The root cause of many issues of *hp*-FEM can often be traced to the selection of an appropriate basis for the implementation. Early endeavors into the construction of high order bases such as Lagrangian and Peano bases quickly fell out of favor due to the condition numbers of the resulting mass and stiffness matrices [73]. Although the current bases of choice are the hierarchical or Dubiner bases [22,28,47], recently attention has been drawn to favorable properties of the Bernstein polynomials [3,49]. The Bernstein polynomials [30] are widely used in the spline literature [62], computer aided geometric design (CAGD) [31], and computer graphics (e.g. PS/TT fonts) [41] but have hitherto not been widely adopted as a basis for high order finite element approximations.

One immediate benefit of using the Bernstein basis is the ease with which one can visualize and post-process finite element solutions owing to the ubiquitous usage of the Bernstein basis in CAGD and the computer graphics community. Generally, visualization and post-processing, including computing iso-surfaces and gradients, of a high order approximation is considerably more complicated than for a low order approximation [64]. For example, visualizing a high order approximation expressed using hierarchical bases typically requires the explicit evaluation of Jacobi polynomials at large number of points which can become prohibitively expensive [47]. Nevertheless, if the approximation is expressed in Bernstein-Bézier form, then techniques developed in the CAGD community enable one to visualize a degree p approximation in $\mathcal{O}(p^3)$ operations as described in Section 6.3.

The suitability of the Bernstein basis for finite element approximations is less clear-cut. Here, among other things, one needs to compute moments of the data with the basis functions (e.g. when constructing the load vector) which, along with the assembly of the element matrices can dominate the computational costs. In the case of tensor product elements, one can use the sum factorization approach, pioneered by Orszag [57], to efficiently compute matrix-vector products and, although less well-known, to evaluate moments of the data. Standard hierarchical bases on a triangle do not naturally have a tensorial structure and are therefore not amenable to sum factorization approaches. Tensorial hierarchical bases [28,47] circumvent this difficulty, but lack rotational symmetry and are sub-optimal when it comes to the evaluation of the element matrices. Perhaps surprisingly, the Bernstein basis was shown in [3] to *naturally* have the tensorial property, which is needed for the sum factorization approach, despite having been known for decades prior to the realization of the importance of the tensorial property. We briefly discuss how the AAD algorithms [3] can be used to construct moments efficiently, and enable the evaluation of the element matrices in optimal complexity in Section 6.3.5. In particular, we show that these algorithms can be used to calculate quantities of interest of the solution in $\mathcal{O}(p^3)$ operations per element.

The aforementioned computational properties of the Bernstein basis come at a price: the ill-conditioning of the resulting matrices. For example, the mass matrix for the Bernstein basis has a condition number which grows as $\mathcal{O}(2^{2p}p^{-1/2})$ [51], whereas the mass (and stiffness) matrix for hierarchical bases has condition numbers which grow at $\mathcal{O}(p^4)$ or faster as we increase the order [5,52,56]. The preconditioner in Chapter 2 is basis independent, and therefore applies to both hierarchical and Bernstein bases. In Section 6.4, we present algorithms which implement the ASM preconditioner in $\mathcal{O}(p^3)$ operations in the case of the Bernstein basis. We exploit a number of properties of the Bernstein basis to reduce computational costs. A key component of the algorithm is the static condensation of the interior degrees of freedom on each element: we present an algorithm which allows one to achieve this in $\mathcal{O}(p^3)$ operations.

In section 2, we present some canonical applications and use them to highlight some of the specific difficulties that one encounters when attempting to use a high order scheme to approximate their solutions. The above developments mean one can tackle each of these problems by using the Bernstein basis at an overall complexity of $\mathcal{O}(p^3)$ operations. Finally, in section 5 we return to the canonical examples described in section 2, and illustrate the performance of the above procedures when applied to these cases.

6.2 Model Problems, Finite Element Formulations, and Computational Challenges

We consider three prototypical problems which theory suggests should be amendable to high order FEM approximations yet each problem exhibits features which present challenges in terms of the efficient implementation of a high order scheme.

In each case $\Omega \subset \mathbb{R}^2$ is a polygonal domain which is partitioned into the union $\mathcal{T}$ of non-overlapping triangular elements with the standard assumptions that the nonempty intersection of any two distinct elements from $\mathcal{T}$ is either a common vertex or a single common edge. More generally, we consider a family of partitions which is assumed to be shape regular in that there exists a number $c > 0$ such that for all partitions, each triangle T contains an incircle with radius $r \geq h_T/c$ where h_T is the diameter of T.

A version of this chapter have been previously published in [9].

Let $\mathbb{P}_p(T) = \text{span}\{x^\alpha y^\beta : 0 \leq \alpha, \beta, \alpha + \beta \leq p\}$ denote the space of polynomials of total degree p on $T \in \mathcal{T}$. Define the standard H^1-conforming finite element space $X = \{u \in H^1(\Omega) : u|_T \in \mathbb{P}_p(T), \forall T \in \mathcal{T}\}$ and the H_0^1-conforming space $X_0 = X \cap H_0^1(\Omega)$. Let $\{\varphi_i\}_{i=1}^N$ be a basis for X, so that any $u \in X$ or X_0 can be written as $u = \vec{u}^T \vec{\varphi}$ for $\vec{u} \in \mathbb{R}^N$, where $\vec{\varphi}$ is the vector whose components are the basis functions. Let $\mathbf{M}$ and $\mathbf{S}$ be the associated mass and stiffness matrices.

6.2.1 Sine-Gordon Equation

The sine-Gordon equation arises in a range of applications, including differential geometry [78] and modeling the dislocation of crystals [14,26], and consists of seeking u such that

$$\frac{\partial^2 u}{\partial t^2} = \Delta u - \sin u, \quad (x, y) \in (-7, 7)^2, t > 0 \tag{6.1}$$

subject to initial conditions given, for example, by [16]

$$\begin{aligned} u(x, y, 0) = u_0(x, y) &= 4 \arctan \exp(x + 1 - 2\,\text{sech}(y + 7) - 2\,\text{sech}(y - 7)) \\ \tfrac{\partial}{\partial t} u(x, y, 0) = w_0(x, y) &= 0 \end{aligned}$$

along with homogeneous Neumann boundary conditions. The variational form of the problem consists of seeking $u(t) \in H^1(\Omega), t > 0$ such that

$$\frac{\partial^2}{\partial t^2}(u, v) = -(\nabla u, \nabla v) - (\sin u, v) \qquad \forall v \in H^1(\Omega) \tag{6.2}$$

where $u(0) = u_0$ and $\frac{\partial}{\partial t} u(0) = w_0$. The solution is smooth (see Figure 6.1), and thus should be amenable to approximation using higher order methods [67].

Let $u_p(t) \in X$ be the Galerkin approximation to Equation (6.2) subject to the

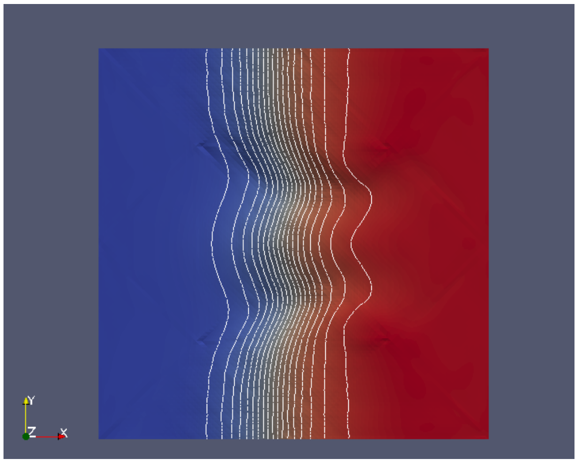


Figure 6.1: Contour plot of the solution of the sine-Gordon equation with the initial conditions from Section 6.2.1 at $t = 5$.

initial conditions $u_p(0) = u_{0p}$ and $\frac{\partial}{\partial t} u_p(0) = w_{0p}$ where $u_{0p}, w_{0p} \in X$ satisfies

$$\begin{aligned} (u_{0p}, v) &= (u_0, v) \qquad \forall v \in X \\ (w_{0p}, v) &= (w_0, v) \qquad \forall v \in X. \end{aligned} \tag{6.3}$$

Writing $u_p(t) = \vec{u}(t)^T \vec{\varphi}$ for $\vec{u}(t) \in \mathbb{R}^N$, the semi-discrete problem takes the form

$$\mathbf{M} \tfrac{\mathrm{d}^2}{\mathrm{d}t^2} \vec{u}(t) = -\mathbf{S}\vec{u}(t) - (\sin u_p(t), \vec{\varphi}).$$

A fully discrete scheme can be obtained by using a Nyström method [40, p. 285] to discretize the temporal derivative:

$$
\begin{aligned}
z &= \vec{u}^n \\
\mathbf{M}\vec{u}_1^{n+1} &= -\mathbf{S}\vec{w} - (\sin z, \vec{\varphi}) \\
z &= \vec{u}^n + (\Delta t)\vec{u}_t^n/2 + (\Delta t)^2\vec{u}_1^{n+1}/8 \\
\mathbf{M}\vec{u}_2^{n+1} &= -\mathbf{S}\vec{w} - (\sin z, \vec{\varphi}) \\
z &= \vec{u}^n + (\Delta t)\vec{u}_t^n + (\Delta t)^2\vec{u}_2^{n+1}/2 \\
\mathbf{M}\vec{u}_3^{n+1} &= -\mathbf{S}\vec{w} - (\sin z, \vec{\varphi}) \\
\vec{u}^{n+1} &= \vec{u}^n + (\Delta t)\vec{u}_t^n + (\Delta t)^2(\vec{u}_1^{n+1}/6 + \vec{u}_1^{n+2}/3) \\
\vec{u}_t^{n+1} &= \vec{u}_t^n + \Delta t(\vec{u}_1^{n+1}/6 + 2\vec{u}_2^{n+1}/3 + \vec{u}_3^{n+1}/6)
\end{aligned}
\tag{6.4}
$$

where $\vec{u}^n = \vec{u}(n\Delta t)$, $\vec{\varphi}^T\vec{u}^0 = u_{0p}$, and $\vec{\varphi}^T\vec{u}_t^0 = w_{0p}$. Implicit time-stepping schemes will be considered later.

The first difficulty encountered in the implementation of Equation (6.4) is the computation of the nonlinear moment $(\sin z, \vec{\varphi})$ whose efficiency is essential as it has to be evaluated at *every* sub-step. A straightforward treatment of the vector $(\sin z, \vec{\varphi})$ would entail using a quadrature rule with $\mathcal{O}(p^2)$ quadrature points for each of the $\mathcal{O}(p^2)$ entries incurring a cost of $\mathcal{O}(p^4)$ in basis function evaluations [3]. For most hierarchical bases, function evaluation involves evaluations of univariate Jacobi polynomials using a recursion at a cost of $\mathcal{O}(p)$ operations per point [1, 69]. It is possible to use precomputed arrays, in which the values of the basis functions at quadrature points are cached, but the current computing platforms lean towards the view that memory access is costlier than CPU cycles [65].

The second difficulty is that one needs to solve *three* systems involving the mass matrix $\mathbf{M}$ at each time-step. The mass matrices obtained using hierarchical bases have condition numbers which grow as $\mathcal{O}(p^4)$, or faster, even for tensor product elements [5, 52, 56]. In [5, 52], it was shown that applying diagonal scaling as a preconditioner for the mass matrix results in a reduction of the condition number to $\mathcal{O}(p^2)$. Nevertheless, there is a significant cost involved in solving systems involving the mass matrix. Fortunately an Additive Scharz Method (ASM) preconditioner was recently developed for the mass matrix which results in a uniformly bounded condition number independent of p Chapter 2. We shall pursue this further in Section 6.4 when we consider how to address the efficient inversion of the mass matrix.

Quite apart from issues of conditioning, efficient iterative methods also require fast matrix-vector multiplication. There are two ways to compute matrix-vector products. The first is the explicit construction of the mass and stiffness matrices, which will incur a cost of $\mathcal{O}(p^6)$ basis function evaluations if performed in a naive fashion [3], and to then compute the matrix-vector products directly. The second way is to use a matrix-free approach which enables the computation of the matrix-vector product in $\mathcal{O}(p^3)$ provided that a tensorial basis is used on the triangle [47].

6.2.2 Brusselator

The Brusselator system is a model of an autocatalytic chemical reaction [40, p. 248] and consists of seeking $(u(t), v(t)), t > 0$ such that

$$\begin{aligned} \frac{\partial u}{\partial t} &= 1 + u^2 v - 4.4u + 0.002\Delta u \\ \frac{\partial v}{\partial t} &= 3.4u - u^2 v + 0.002\Delta v \end{aligned} \qquad (x, y) \in (-1, 1)^2, \tag{6.5}$$

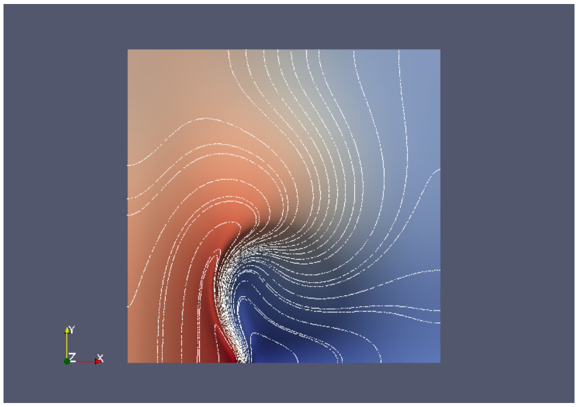


Figure 6.2: Contour plot of the solution of the v component of the Brusselator equation with the initial conditions from Section 6.2.2 at $t = 10$.

subject to homogeneous Neumann boundary conditions, and initial conditions given, for example, by

$$u(x, y, 0) = u_0(x, y) = 0.5 + y$$
$$v(x, y, 0) = v_0(x, y) = 1 + 5x.$$

The corresponding variational formulation is to seek $u(t), v(t) \in H^1(\Omega), t > 0$ such that

$$\begin{aligned} \frac{\partial}{\partial t}(u, w) &= (1, w) + (u^2 v, w) - 4.4(u, w) - 0.002(\nabla u, \nabla w) \\ \frac{\partial}{\partial t}(v, w) &= 3.4(u, w) - (u^2 v, w) - 0.002(\nabla v, \nabla w) \end{aligned} \tag{6.6}$$

for all $w \in H^1(\Omega)$. Although the solution is smooth (see Figure 6.2), it does exhibits steep interior layers whose location changes as the solution evolves.

Let $u_p(t), v_p(t) \in X$ be the Galerkin approximations to Equation (6.6) for u, v

respectively, subject to initial conditions satisfying

$$\begin{aligned}(u_p(0), w) &= (u_0, w) \qquad \forall w \in X\\ (v_p(0), w) &= (v_0, w) \qquad \forall w \in X.\end{aligned}$$

Let $\vec{u}(t) \in \mathbb{R}^N$ be the vector such that $u_p(t) = \vec{u}(t)^T\vec{\varphi}$ and likewise for $\vec{v}(t) \in \mathbb{R}^N$, then the semi-discrete problem is

$$\begin{aligned}\mathbf{M}\tfrac{\partial}{\partial t}\vec{u}(t) &= (1, \vec{\varphi}) + (u_p^2(t)v_p(t), \vec{\varphi}) - 4.4\mathbf{M}\vec{u}(t) - 0.002\mathbf{S}\vec{u}(t)\\ \mathbf{M}\tfrac{\partial}{\partial t}\vec{v}(t) &= 3.4\mathbf{M}\vec{u}(t) - (u_p^2(t)v_p(t), \vec{\varphi}) - 0.002\mathbf{S}\vec{v}(t)\end{aligned}.$$

To arrive at the fully discrete scheme, we use an IMEX scheme [66] for the time discretization as follows:

$$\begin{aligned}\frac{\mathbf{M}\vec{u}^{n+1} - \mathbf{M}\vec{u}^{n}}{\Delta t} &= (1, \vec{\varphi}) + (u_n^2 v_n, \vec{\varphi}) - 4.4\mathbf{M}\vec{u}^{n+1} - \frac{0.002}{2}(\mathbf{S}\vec{u}^{n+1} + \mathbf{S}\vec{u}^{n})\\ \frac{\mathbf{M}\vec{v}^{n+1} - \mathbf{M}\vec{v}^{n}}{\Delta t} &= 3.4\mathbf{M}\vec{u}^{n} - (u_n^2 v_n, \vec{\varphi}) - \frac{0.002}{2}(\mathbf{S}\vec{v}^{n+1} + \mathbf{S}\vec{v}^{n})\end{aligned} \tag{6.7}$$

where $\vec{u}^n$ is the approximation at $n\Delta t$, and $(u_n^2 v_n, \vec{\varphi})$ is shorthand for the nonlinear term $(u_p^2(t)v_p(t), \vec{\varphi})$ at time $t = n\Delta t$. Observe that if we were to use a fully explicit scheme, the CFL condition for stability is $\Delta t \leq C\frac{h^2}{p^4}$ which, owing to the rapid decrease with p, is generally regarded as being overly restrictive for practical computations. Instead, one typically sees $\Delta t \sim \frac{h^2}{p^2}$ being used in practice in conjunction with an implicit scheme.

The efficient application of a high order scheme to the solution of the Brusselator system encounters all of the difficulties which we noted for the sine-Gordon equation. In addition, the Brusselator system involves the repeated inversion of the matrices $\mathbf{M}+0.001\Delta t\mathbf{S}$ and $5.4\mathbf{M}+0.001\Delta t\mathbf{S}$, as opposed to the pure mass matrix. Previously, we alluded the availability of an ASM preconditioner for the mass matrix. Can this preconditioner for the pure mass matrix play a useful role in the case of implicit schemes?

The two dimensional version of Schmidt's inequality [23] implies there is a constant c, independent of h and p, such that $0 \leq \mathbf{S} \leq c\frac{p^4}{h^2}\mathbf{M}$, hence we have

$$\mathbf{M} \leq \mathbf{M} + 0.001\Delta t\mathbf{S} \leq \left(1 + c\frac{p^4\Delta t}{h^2}\right)\mathbf{M}.$$

Let $\mathbf{P}^{-1}$ denote the uniform preconditioner for the mass matrix described in Chapter 2. Then, using $\mathbf{P}^{-1}$ to precondition the implicit scheme gives a condition number satisfying

$$\kappa(\mathbf{P}^{-1}(\mathbf{M} + 0.001\Delta t\mathbf{S})) \leq C\frac{p^4\Delta t}{h^2}.$$

Observe that if one uses a time step which satisfies the CFL condition for the explicit scheme (i.e. $\Delta t \sim \frac{h^2}{p^4}$), then the condition number will be uniformly bounded. Alternatively, taking a step size of $\Delta t \sim \frac{h^2}{p^2}$ results in the condition number of the operator growing as $\mathcal{O}(p^2)$. Therefore a preconditioner for the mass matrix also provides a useful preconditioner for the systems arising from an implicit time stepping scheme.

Finally, a difficulty (pertinent also to the case of the sine-Gordon equation) which often remains unacknowledged in high order finite elements analysis is the cost of post-processing and visualization of the resulting finite element solution. A straightforward approach to visualization based on evaluating the solution at sufficiently many points and using a standard graphics package, would require the evaluation of the solution at $\mathcal{O}(p^2)$ points. At each of those points, we need to evaluate the

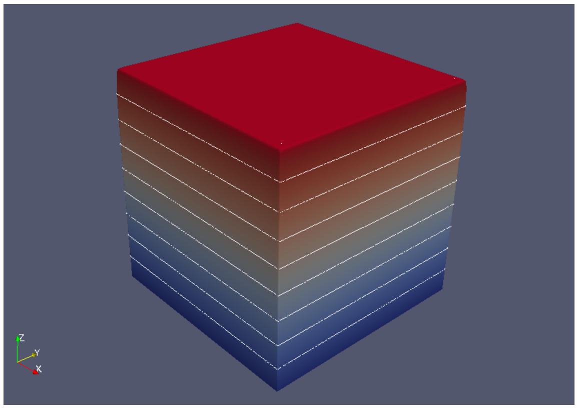


Figure 6.3: Contour plot of the solution of the singularly perturbed problem with $\varepsilon^2 = 10^{-3}$ and $f = 1$.

solution (a vector with with $\mathcal{O}(p^2)$ entries) meaning a total of $\mathcal{O}(p^4)$ Jacobi polynomial evaluations are needed to evaluate u. The same costs apply if one wishes to visualize a component of the gradient etc. We discuss the issue of visualization and post-processing in Section 6.3.

6.2.3 Problems Exhibiting Boundary Layers

Let $0 < \varepsilon \ll 1$ be a parameter, and consider the problem on $\Omega = (0, 1)^2$

$$\begin{aligned} u - \varepsilon^2 \Delta u = f \qquad & x \in \Omega \\ u = 0 \qquad & x \in \partial\Omega \end{aligned}$$

where $f \in L^2(\Omega)$. This is an example of a singularly perturbed problem in which the solution exhibits steep layers of width $\mathcal{O}(\varepsilon)$ in the neighborhood of the boundary [54]; see Figure 6.3 for a plot of the solution for $\varepsilon^2 = 10^{-3}$ with $f = 1$. This problem serves as a prototype for a large class of problems arising in mechanics including, for example, the linear elastic response of thin bodies [38].

The variational form consists of seeking $u \in H_0^1(\Omega)$ such that

$$(u, v) + \varepsilon^2(\nabla u, \nabla v) = (f, v) \qquad \forall v \in H^1(\Omega).$$

In fully discrete form, we arrive at the linear system

$$(\mathbf{M} + \varepsilon^2\mathbf{S})\vec{u} = \vec{f} \tag{6.8}$$

where $\vec{f} = (f, \vec{\varphi})$. Whilst the operator $\mathbf{M}+\varepsilon^2\mathbf{S}$ has, at first glance, the same structure as the operators which arose in the Brusselator example, viz $\mathbf{M} + c\Delta t\mathbf{S}$, the present case poses an additional layer of difficulty which we shall now explain.

The anisotropic behavior of the solution in the neighborhood of the boundary means that, in order to obtain a robust scheme in ε, anisotropic or stretched elements should be used at the boundary in conjunction with regular elements on the interior [11]. Moreover, whilst the solution has boundary layers, it is analytic and as such high order methods can exhibit exponential rates of convergence provided that the anisotropy of the elements is properly combined with the polynomial order p [68].

The correct combination of anisotropic and p consists of using anisotropic elements of width $\mathcal{O}(p\varepsilon)$ along the boundary as illustrated in Figure 6.4 [68]. This approach gives *robust* exponential convergence with respect to ε and, as such, will outperform a pure h-version or pure p-version method [54]. Of course, using a single layer of anisotropic elements around the boundary means that we drop our earlier assumption that the family of partitions is shape uniform.

The fresh computational issue that arises is that the aspect ratio of anisotropic elements has a detrimental effect on the conditioning of the stiffness matrix $\mathbf{S}$ [48] resulting in issues with iterative solvers [54]. Existing preconditioners for anisotropic

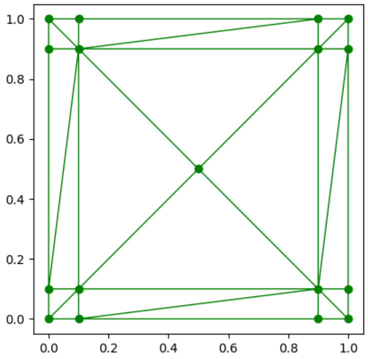


Figure 6.4: Plot of the mesh to approximate the boundary layer problem Section 6.2.3. The needle elements around the boundaries have thickness of $p\varepsilon$ in order to resolve the rapid changes [54, 68].

elements are either inapplicable to the meshes from [54, 68] described above [74] or give condition numbers dependent on the factor ε [53].

The above difficulties notwithstanding, we again propose to simply use the mass matrix preconditioner $\mathbf{P}^{-1}$ from Chapter 2 to precondition the systems arising from meshes such as the one shown in Figure 6.4. A scaling argument applied to the usual two dimensional Schmidt's inequality [23] on isotropic elements can be used to deduce that

$$\mathbf{M} \leq \mathbf{M} + \varepsilon^2 \mathbf{S} \leq \left(1 + c\varepsilon^2 \frac{p^4}{p^2\varepsilon^2}\right)\mathbf{M} \tag{6.9}$$

$$\leq (1 + cp^2)\mathbf{M}. \tag{6.10}$$

Consequently, using the mass preconditioner $\mathbf{P}^{-1}$ from Chapter 2, we have $\kappa(\mathbf{P}^{-1}(\mathbf{M}+\varepsilon^2\mathbf{S})) \leq Cp^2$ with C independent of p, ε and the number of elements. The key advantage of using the mass matrix is that the condition number is *independent of* ε whereas alternative approaches result in a condition number depending on $\varepsilon^{-1} \gg 1$.

6.2.4 Summary

In summary, applying high order methods to tackle the above prototypical problems encounters the following challenges:

1. Calculation of the nonlinear moments, such as $(u_n^2 v_n, \vec{\varphi})$ and $(\sin z, \vec{\varphi})$, is potentially inefficient. As discussed previously, bases which can utilize the sum factorization technique are adept at computing moments and matrix-vector products. In Section 6.3.5, we will briefly discuss how the Bernstein polynomials can use algorithms presented in [3] to calculate the nonlinear moments and the residuals in $\mathcal{O}(p^3)$ operations.

2. Transient problems will require the use of a time stepping scheme, which results in the need to invert either the mass matrix or a perturbation thereof. We propose to solve such systems efficiently by implementing the ASM preconditioner developed in Chapter 2 using the Bernstein basis in Section 6.4. Furthermore, the foregoing discussion showed for the treatment of the matrices arising in implicit time stepping schemes and from problems where anisotropic elements are used, preconditioning the mass matrix can be an effective approach.

3. Finally, once the simulation is complete, one typically wishes to either visualize the solution or carry out post-processing to calculate quantities of interest. We will exposit algorithms which can easily visualize and post-process the solutions obtained using the Bernstein basis in Section 6.3.

6.3 Visualization and Post-Processing

Bernstein-Bézier polynomials have played a fundamental role in the development of computer graphics, splines, PS/TT fonts and computer-aided geometric design (CAGD), resulting in a wealth of elegant and effective algorithms for the visualization and graphical post-processing of polynomials written in Bernstein form [29,30]. In this section, we formally introduce the Bernstein polynomials and give a brief overview of efficient $\mathcal{O}(p^3)$ algorithms for the implementation of post-processing procedures which are pertinent to finite element analysis (e.g. point evaluation, visualization, and evaluations of quantities of interest).

6.3.1 Bernstein Polynomials

Let T be a non-degenerate triangle in $\mathbb{R}^2$ with vertices v_1, v_2, v_3. For a fixed integer $p \geq 3$, we define the domain points as

$$\mathcal{D}^p(T) = \left\{ \frac{1}{p} \left(\alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 \right) : (\alpha_1, \alpha_2, \alpha_3) \in \mathcal{I}^p \right\}$$

where the index set $\mathcal{I}^p = \{\alpha := (\alpha_1, \alpha_2, \alpha_3) \in \mathbb{Z}^3_+ : \sum_{k=1}^3 \alpha_k = p\}$. It is natural to classify the domain points into vertices, edges or interior points. The interior domain points are those associated with $\alpha \in \mathcal{I}^p$ with strictly positive components. The vertex domain points are associated with the indices $(p, 0, 0)$, $(0, p, 0)$ and $(0, 0, p)$. Finally, the edge domain points are the remaining domain points; see Figure 6.5.

The barycentric coordinates $\lambda_i \in \mathbb{P}_1(T), i \in \{1, 2, 3\}$ of T are affine functions such that $\lambda_i(v_j) = \delta_{ij}$ for $i, j \in \{1, 2, 3\}$. The bivariate Bernstein polynomials of degree p associated with triangle T are then defined by

$$B^p_\alpha = \frac{p!}{\alpha_{1!}\alpha_{2!}\alpha_{3!}}\lambda_1^{\alpha_1}\lambda_2^{\alpha_2}\lambda_3^{\alpha_3}, \qquad \alpha \in \mathcal{I}^p.$$

There is a natural one-to-one correspondence between Bernstein polynomials, domain points and the index set $\mathcal{I}^p$. Every Bernstein polynomial on a triangle can be readily classified as an interior, an edge or a vertex polynomial in much the same way as domain points. We denote by B^p_V, B^p_E, B^p_I as the sets of all vertex, edge and interior Bernstein polynomials; see Figure 6.5.

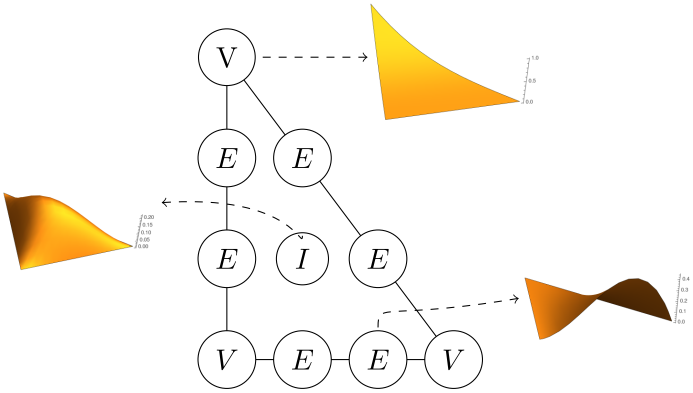


Figure 6.5: Figure showing domain points for degree $p = 3$ along with some plots of a typical Bernstein polynomial corresponding to the domain points. V stands for vertex, E stands for edge and I stands for interior.

Every polynomial $u \in \mathbb{P}_p(T)$ can be expressed in terms of degree p Bernstein polynomials:

$$u = \sum_{\alpha\in\mathcal{I}^p} c^p_\alpha B^p_\alpha$$

the so-called B-form of u. The coefficients c^p_α are usually referred to as the B-net or control points by the graphics community [29].

Likewise, one can define the univariate Bernstein polynomials: let λ_1, λ_2 be the barycentric coordinates of a point on the interval $[a, b]$, then the univariate Bernstein polynomials of degree p on the interval are defined as

$$B_i^p = \binom{p}{i} \lambda_1^i \lambda_2^{p-i}, \qquad i = 0, \dots, p.$$

6.3.2 Point Evaluation using de Casteljau Algorithm

The de Casteljau algorithm is an elegant and stable recursive scheme for the evaluation of a polynomial written in terms of Bernstein polynomials [29]. Given the control points $\{c_\alpha^p\}$ of a polynomial u in B-form, we fix a point $P \in T$ at which we want to evaluate u, and let $\lambda_1, \lambda_2, \lambda_3$ be the values of the barycentric coordinates of P. The de Casteljau algorithm consists of recursively defining points $\{c_\beta^k\}$ for $k \in [0, \dots, p-1]$ and $\beta \in \mathcal{I}^k$ by

$$c^k_{(\beta_1,\beta_2,\beta_3)} := \lambda_1 c^{k+1}_{(\beta_1+1,\beta_2,\beta_3)} + \lambda_2 c^{k+1}_{(\beta_1,\beta_2+1,\beta_3)} + \lambda_3 c^{k+1}_{(\beta_1,\beta_2,\beta_3+1)}.$$

The recursion terminates with a single coefficient c^0_{000} at a cost of $\mathcal{O}(p^3)$ operations which, remarkably, coincides $u(P)$; see Figure 6.6 for an example in the case $p = 3$.

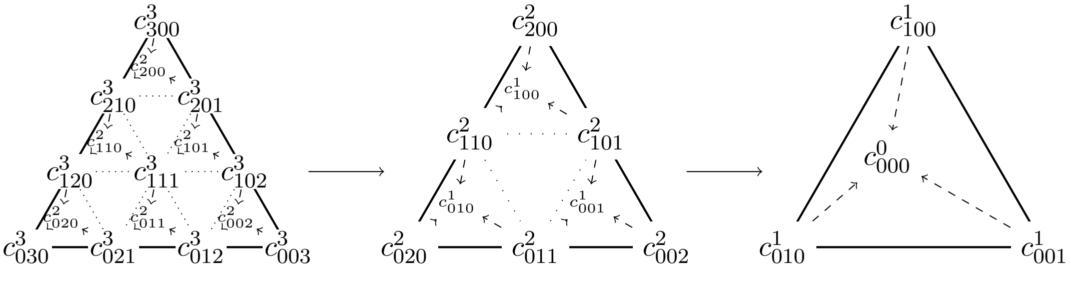


Figure 6.6: Example of applying de Casteljau algorithm to $p = 3$ case

The de Castlejau algorithm is the archetypal example of a *pyramid* algorithm [34]. Specifically, if we stack the coefficients appearing in Figure 6.6 as shown in Figure 6.7, in which each layer corresponds to the recursion level, we obtain a pyramid of coefficients with c_α^p on the bottom and c^0_{000} at the summit.

A key property of the de Castlejau algorithm is that the coefficients which emerge on its three vertical faces of the pyramid satisfies the *blossoming* property. In order to explain what this means, we first label the vertices corresponding to domain points $c^3_{300}, c^3_{030}, c^3_{003}$ (i.e. the vertices of the triangle) as A, B, C respectively and again fix the point $P \in T$ corresponding to barycentric coordinates $\lambda_1, \lambda_2, \lambda_3$ as before (recall $u(P) = c^0_{000}$).

Blossoming is the property whereby the B-form polynomial defined by the coefficients laid out in Triangle 1 in Figure 6.8 (the left face in Figure 6.7) equals the restriction of u to the region $\triangle ABP$, i.e.

$$u_1|_{\triangle ABP} = u|_{\triangle ABP}.$$

The same property holds true for $\triangle BCP$ with coefficients as in Triangle 2, and for $\triangle ACP$ with coefficients from Triangle 3. In other words, we have

$$u(x)|_{\triangle ABC} = \begin{cases} u_1(x) & x \in \triangle ABP \\ u_2(x) & x \in \triangle BCP \\ u_3(x) & x \in \triangle ACP \end{cases}.$$

A pseudo-code implementation of de Casteljau algorithm with the blossoming coefficients stored can be seen in Algorithm 3. Note that the blossoms for the faces are a natural by-product of applying the de Casteljau algorithm.

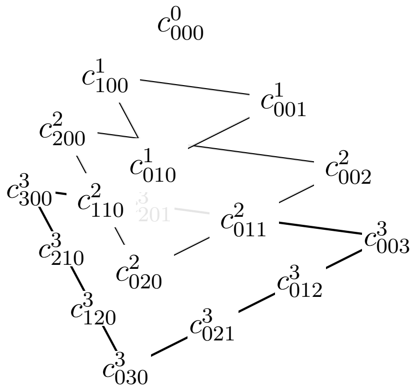
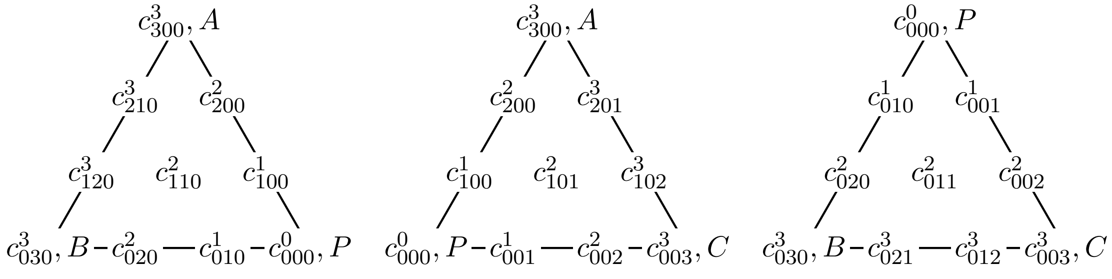


Figure 6.7: Rearranging the de Casteljau algorithm to a pyramid in the $p = 3$ case. The interior coefficients (such as c^3_{111}) are left out for clarity.



Triangle 1: $u_1(x)$ Triangle 2: $u_2(x)$ Triangle 3: $u_3(x)$

Figure 6.8: Example of blossoming

6.3.3 Visualization

While the de Casteljau algorithm is stable, using it to evaluate large numbers of points for plotting is not an efficient strategy (e.g $\mathcal{O}(p^3)$ operations are required for reach of the $\mathcal{O}(p^2)$ points netting an overall cost of $\mathcal{O}(p^5)$). Consequently, a standard technique to the rendering of Bernstein-Bézier surfaces in the computer graphics community consists of plotting the surface obtained by linearly interpolating the coefficients $\{c^p_\alpha\}$ of the Bernstein polynomial [29]. This B-net is a convex hull for the polynomial, and approximates the surface. If higher resolution is needed than provided by the original B-net of the solution, then the *subdivision* algorithm can be invoked to create a finer net which then can be rendered in the same fashion.

Algorithm 3 de Casteljau Algorithm (with blossoming coefficients)

```
Require: $\mathbf{abc}$ 2D array of the B-net of the polynomial of degree $p$
function DECAST($\lambda_1, \lambda_2, \lambda_3, \mathbf{abc}$)
    $\mathbf{apc}, \mathbf{abp} := \text{zeros}((p+1, p+1))$    ▷ Stores blossoming data
    $\mathbf{apc} = \mathbf{abc}[:, 0]$    ▷ Store first rows before overwriting
    $\mathbf{abp} = \mathbf{abc}[0, :]$
    for $k = 0, \ldots, p-1$ do
        for $i = 0, \ldots, p-k-1$ do
            for $j = 0, \ldots, i$ do    ▷ de Casteljau step
                $\mathbf{abc}[j, i-j] = \lambda_1 \mathbf{abc}[j, i-j] + \lambda_2 \mathbf{abc}[j+1, i-j] + \lambda_3 \mathbf{abc}[j, i-j+1]$
            end for
        end for
        $\mathbf{apc}[0 : p-k, k+1] = \mathbf{abc}[0 : p-k, 0]$    ▷ Store the blossoming coefficients before progressing to the next level
        $\mathbf{abp}[k+1, 0 : p-k] = \mathbf{abc}[0, 0 : p-k]$
    end for
    ▷ $\mathbf{apc}$ and $\mathbf{abp}$ contains the B-net for triangles $APC$ and $ABP$ respectively. See Figure 6.8.
    return $\mathbf{abc}, \mathbf{apc}, \mathbf{abp}$    ▷ $\mathbf{abc}$ contains the coefficients for $\mathbf{pbc}$
end function
```

The subdivision algorithm consists of dividing the original triangle into four triangles representing the same polynomial[1]; see Figure 6.9. Although the original function remains unchanged, the B-net representing u now contains roughly four times as many B-net points obtained at the cost of just four applications of the de Casteljau algorithm with storage of the blossoming coefficients (see Algorithm 4) [33, §8.1]; see Table 6.1 for a comparison in the cost of visualization in terms of the number of operations needed per point when using de Casteljau algorithm versus the subdivision algorithm.

The subdivision algorithm converges quadratically in the number of subdivision levels ℓ, in the sense that for a given triangle T with diameter h_T, the error at the

[1] We note that the de Casteljau algorithm with blossoming coefficients divides the triangle into three triangles representing the same polynomial (assuming the point lies in the interior of the triangle).

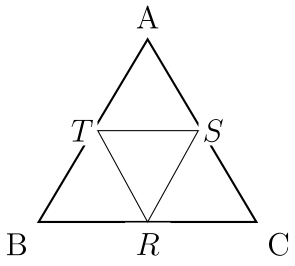


Figure 6.9: The subdivision algorithm: given the control points on $\triangle ABC$, we divide it into four triangles $\triangle ATS$, $\triangle TRS$, $\triangle TBR$, and $\triangle SRC$ whose control points equals the same polynomial.

Table 6.1: Table to illustrate the benefit of using subdivision algorithm by displaying the cost per point of visualization assuming $\mathcal{O}(1)$ number of subdivisions; typically, only two or three subdivisions are needed for visual fidelity.

Method	**Number of Points**	**Cost per Point**	**Total Cost**
de Casteljau	$\mathcal{O}(p^2)$	$\mathcal{O}(p^3)$	$\mathcal{O}(p^5)$
Subdivision	$\mathcal{O}(p^2)$	$\mathcal{O}(p)$	$\mathcal{O}(p^3)$

ℓth level of subdivision is

$$\|u - \bar{u}_\ell\|_\infty \leq \frac{C}{2^\ell}\|\Delta u\|_\infty$$

for C independent of h, where $u \in \mathbb{P}_p(T)$ and $\bar{u}_\ell$ is the linear interpolation of the ℓth level subdivided B-net [20, 21]. In Figure 6.10, we plot a B-net and its subdivisions; we observe that two or three subdivisions is usually more than enough for visual fidelity.

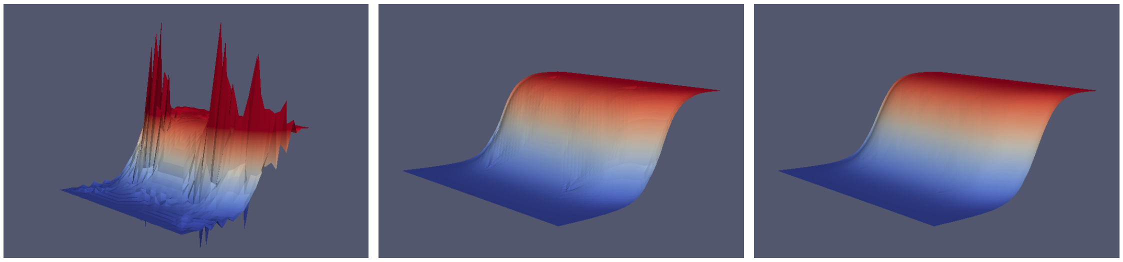
Once the subdivision step is completed, one can save the resulting B-net and the domain points as VTK files in order that sophisticated visualization software akin to Paraview or VisIt can easily process it. From here, robust algorithm in those software packages can post-process the approximation including plotting contour lines.

The efficient rendering of the B-net can be accomplished using OpenGL “evaluators” (see `glEvalMesh2` in [70]). These OpenGL evaluators are defined on a rect-

Algorithm 4 Subdivision Algorithm on a single triangle

```
Require: abc array of the coefficients of B-form polynomial of degree $p$
function SUBDIVISION(c)
    $\_, \mathbf{abr}, \mathbf{arc} = \texttt{decast}(0, .5, .5, \mathbf{abc})$ ▷ We use the blossoming coefficients from Algorithm 3; the _ notation means we do not use the result.
    $\mathbf{tbr}, \_, \_ = \texttt{decast}(.5, .5, 0, \mathbf{abr})$ ▷ Obtain triangle $TBR$ from Figure 6.9
    $\mathbf{src}, \mathbf{ars}, \_ = \texttt{decast}(.5, 0, .5, \mathbf{arc})$ ▷ Obtain triangle $SRC$ from Figure 6.9
    $\mathbf{trs}, \_, \mathbf{ats} = \texttt{decast}(1, 1, -1, \mathbf{ars})$ ▷ point $T$ is outside of $\triangle ARS$; this step gives us the last two triangles of the subdivision.
    return ats, trs, tbr, src
end function
```

Figure 6.10: Figures showing the refinements of the subdivision algorithm for $p = 12$ and 16 elements on the square for the initial condition of the sine-Gordon example. In general, the number of refinements needed is only 2 or 3 depending on the order and size of the elements.



(a) 0 subdivisions (b) 1 subdivision (c) 2 subdivisions

angular patch but a simple transformation of the coefficients on a triangle to the rectangle can be employed [44, 81]. Unfortunately, in many implementations, there is vendor and hardware dependent constant, namely `GL_MAX_EVAL_ORDER`, which sets the maximum order that OpenGL can plot, which is often set to just $p < 8$.

6.3.4 Computation and Visualization of Gradients and Higher Derivatives of the Solution

We now describe how to easily compute and visualize the *gradient* and *higher order derivatives* from a B-form polynomial. The gradient of a Bernstein polynomial can

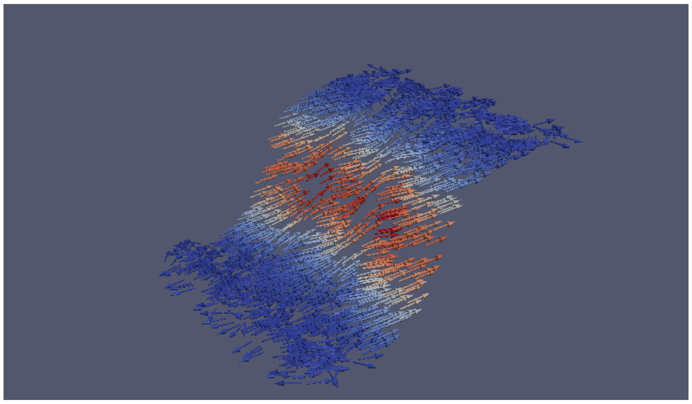


Figure 6.11: Example vector plot of gradients for the sine-Gordon example at $t = 5$ with 2 subdivisions. Note that the gradient glyphs are superimposed over the plot of the sine-Gordon solution.

be expressed as a sum of Bernstein polynomials

$$\nabla B_\alpha^p = p \sum_{k=1}^{3} B_{\alpha - e_k}^p \nabla \lambda_k \tag{6.11}$$

where the sum is over when $\alpha - e_k$ is a valid multi-index [4]. Let u be a given polynomial expressed in B-form, then by Equation (6.11)

$$\nabla u = \sum_{\alpha \in \mathcal{I}^p} c_\alpha^p \left(p \sum_{k=1}^{3} B_{\alpha - e_k}^p \nabla \lambda_k \right) = \sum_{\beta \in \mathcal{I}^{p-1}} \vec{c}_\beta^{\,p-1} B_\beta^{p-1}$$

where

$$\vec{c}_\beta^{\,p-1} = p \sum_{k=1}^{3} c_{\beta + e_k} \nabla \lambda_k \qquad \forall \beta \in \mathcal{I}^{p-1}.$$

Hence, to compute the gradient, we compute $\vec{c}_\beta^{\,p-1}$ from c_α^p at a cost of $\mathcal{O}(p^2)$ operations. One can then use the subdivision algorithm on $\vec{c}_\beta^{\,p-1}$ componentwise to plot the gradient. This results in a smaller B-net than by plotting the function values (i.e. $\alpha \in \mathcal{I}^p$ but $\beta \in \mathcal{I}^{p-1}$).

In order to obtain the B-net of the gradient on the same set of control points as the original approximate u, one would apply the Bernstein 2D degree raising algorithm (Algorithm 16) component-wise which allows one to express $\vec{c}_{\beta}^{p-1}$ for $\beta \in \mathcal{I}^{p-1}$ as $\vec{c}_{\alpha}^{p}$ for $\alpha \in \mathcal{I}^{p}$; the cost of degree raising is $\mathcal{O}(p^2)$. See Figure 6.11 for an example of superimposing the gradients over the solution, and Algorithm 5 for the general visualization algorithm. In fact this procedure can be generalized to arbitrary order derivatives; for example, to visualize the Hessian and superimpose it on the solution, one would have to first calculate the coefficients $\mathbf{c}_{\beta}^{p-2}$ appropriately, degree raise twice, then use the same subdivision algorithm component-wise.

Algorithm 5 Function and Gradient Plotting Algorithm

```
Require: $c_{\alpha}^{p}$ coefficients of Bernstein polynomials
function VISUALIZE($c_{\alpha}^{p}$)
    Calculate $\vec{c}_{\beta}^{p-1}$ from $c_{\alpha}^{p}$
    $\vec{c}_{\alpha}^{p}$ = DegreeRaise2D($\vec{c}_{\beta}^{p-1}$)        ▷ Perform degree raising component-wise
    Apply subdivision to $c_{\alpha}^{p}$ and component-wise to $\vec{c}_{\alpha}^{p}$
    Plot the resulting Bézier net from the subdivision algorithms
end function
```

6.3.5 Evaluation of Quantities of Interest and Nonlinear Moments

In many practical problems, the quantity of interest is not point values but rather some integral quantity of the solution u such as the L^2 energy $\int_{\Omega} u^2\,dx$, H^1 energy $\int_{\Omega}(u^2 + |\nabla u|^2)\,dx$, the average displacement $\frac{1}{|\Omega|}\int_{\Omega} u\,dx$ etc [13]. In general, quantities of interest can be expressed as

$$Q[u] = \int_{\Omega} \eta(u, \nabla u)\,dx$$

for η a given, possibly non-linear, function.

The quantity $Q[u]$ can be computed efficiently by exploiting the tensorial nature of the Bernstein basis (Lemma 1 of [3]). Given the control points c_p^α of the approximate u, we can apply algorithm 1 and eq (3.6) of [3] to directly compute $Q[u]$ for each element at a cost of $\mathcal{O}(p^3)$ operations. Furthermore, the tensorial property allows one to compute the residual and matrix-vector products in $\mathcal{O}(p^3)$ also. The following theorem from [3] is the key:

Theorem 6.3.1. *In two dimensions, the nonlinear moments*

$$\vec{\mu}_T(u, f) = \int_T B_\alpha^p(x) f(x, u, \nabla u)\, dx \qquad \forall \alpha \in \mathcal{I}^p$$

where T is a simplex and f is an arbitrary nonlinear function can be computed with a cost of $\mathcal{O}(p^3)$.

Here, we want to emphasize that Theorem 6.3.1 allows us to calculate the nonlinear evaluation such as $(\sin z, \vec{\varphi})$ or $(u_n^2 v_n, \vec{\varphi})$ from Section 6.2 in $\mathcal{O}(p^3)$. It is a straightforward application of the algorithm in Corollary 3 of [3].

Furthermore, we can calculate matrix-vector multiplication using μ_T; for example, the mass matrix product can be calculated as

$$\vec{\mu}_T(u, 1) = \int_T B_\alpha^p(x) u(x)\, dx = \int_T B_\alpha^p\Big(\sum_{\beta \in \mathcal{I}^p} c_\beta^p B_\beta^p\Big)\, dx = (\mathbf{M}\vec{u})_\alpha \qquad \forall \alpha \in \mathcal{I}^p$$

where $(\mathbf{M}\vec{u})_\alpha$ is the column corresponding to row α. We refer to [3] for efficient techniques for the evaluation of matrix-vector products against the stiffness matrix etc.

6.4 Linear Solver and Preconditioning

In Section 6.3, we discussed computation of the residual and visualization in $\mathcal{O}(p^3)$ using the Bernstein basis; all that remains is inverting the mass matrix (or a small perturbation thereof) in order to time-step. An unfortunate fact of the Bernstein basis is that its mass matrix condition number is $\mathcal{O}(2^{2p}p^{-1/2})$ [51]; an iterative solver will struggle, and direct solvers will lose many digits of accuracy. In this section, we present the implementation of a *uniform* preconditioner in both h and p Chapter 2 for the Bernstein basis mass matrix with a cost of $\mathcal{O}(p^3)$ operations.

We claim that we can simulate and post-process transient problems using an explicit time-stepper (e.g. Section 6.2.1) in $\mathcal{O}(p^3)$ operations. Recall the error at iteration n for conjugate gradient is bounded by

$$\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^n \|\vec{e}_0\|$$

where κ is the condition number of the preconditioned system and $\vec{e}_0$ is the initial residual vector [35, p. 636]. As the condition number κ of the preconditioned mass matrix is bounded uniformly, the number of iterations needed by conjugate gradient to converge to a given tolerance ε is also bounded uniformly by a constant K independent of p and h

$$K \leq \log \frac{\varepsilon}{\|\vec{e}_0\|} / \log \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$$

Hence, to time-step up to time T using an ℓ step explicit time-stepper with Δt would require

$$\frac{T}{\Delta t}\ell N \cdot \mathcal{O}(p^3) \to \mathcal{O}(p^3)$$

operations *total*, including post-processing procedures.

6.4.1 Jacobi Polynomials

We use the standard definition of 1D Jacobi polynomial [1] for $P_p^{(a,b)}$ where p is the order and $a, b > -1$ are the weights. The orthogonality property is such that

$$\int_{-1}^{1} (1-x)^\alpha (1+x)^\beta P_m^{(\alpha,\beta)}(x) P_n^{(\alpha,\beta)}(x)\, dx \\ = \frac{2^{\alpha+\beta+1}}{2n+\alpha+\beta+1} \frac{(n+\alpha)!(n+\beta)!}{(n+\alpha+\beta)!n!} \delta_{nm}.$$

A key identity relating Jacobi polynomials and 1D Bernstein polynomials together is

$$P_p^{(a,b)} = \sum_{i=0}^{n} \frac{\binom{p+a}{i}\binom{p+b}{p-i}}{(-1)^{p-i}\binom{p}{i}} B_i^p. \tag{6.12}$$

6.4.2 Additive Schwarz Preconditioner for Mass Matrix

We present a short review of the additive Schwarz preconditioner presented in Chapter 2 for the mass matrix on a triangulation $\mathcal{T}$ of the domain Ω. Let $X = \{u \in H^1(\Omega) : u|_K \in \mathbb{P}_p(K), \forall K \in \mathcal{T}\}$. Define $X_{K,I} := \mathbb{P}_p(K) \cap H_0^1(K)$ which is the space of polynomial bubble functions on element K, and let $X_I = \cup_{K \in \mathcal{T}} X_{K,I}$. We note that the interior Bernstein polynomials B_I^p is a basis for X_I.

For each edge $e \in \mathcal{T}$, let K_i be the elements such that $e \in \partial K_i$. Let

$$X_e := \operatorname{span}\{B_\alpha^p : \alpha \text{ domain points strictly in } e\}$$

and define the edge spaces

$$\widetilde{X}_e := \left\{u \in X_e : (u, w) = 0 \; \forall w \in X_I\right\}.$$

A key property is that each element in $\widetilde{X}_e$ and X_e can be *uniquely determined by its value restricted to* e (see Lemma 5.1 of Chapter 2).

For $x \in [-1, 1]$, let

$$\nu(x) = \frac{(-1)^{\lfloor p/2 \rfloor + 1}}{\lfloor p/2 \rfloor} \frac{1-x}{2} P^{(1,1)}_{\lfloor p/2 \rfloor - 1}(x). \tag{6.13}$$

For each vertex $v \in \mathcal{T}$, let K_i be the elements such that $v \in \partial K_i$, let λ_i be the barycentric coordinate of K_i such that $\lambda_i(v) = 1$. Define

$$\varphi_v(x) = \begin{cases} \nu(1 - 2\lambda_i) & x \in \cup_i K_i \\ 0 & \text{else} \end{cases}$$

which has the property that $\varphi_v(v) = 1$, and $\varphi_v(x) = 0$ for all $x \in \Omega \setminus \cup_i K_i$. Furthermore, $\varphi_v(x)$ on the edges on which it is supported is a scaling of ν. Now define

$$X_V := \operatorname{span}\left\{\varphi_v : v \in \mathcal{T}\right\}$$

and

$$\widetilde{X}_V := \left\{u \in X_V : (u, w) = 0 \; \forall w \in X_I\right\}$$

Similar to $\widetilde{X}_e$, the space $\widetilde{X}_V$ is uniquely determined by the values on the vertices.

We can decompose X as

$$X = X_I \oplus \widetilde{X}_V \oplus \bigoplus_{e \in \mathcal{T}} \widetilde{X}_e.$$

We now define the bilinear form on the subspaces in the decomposition:

- Interior space X_I:

$$a_I(u, w) := (u, w), \qquad u, w \in X_I.$$

- Vertex space $\widetilde{X}_V$:

$$a_V(\widetilde{u}, \widetilde{w}) := \frac{1}{p^4} \sum_{v \in \mathcal{T}} c_v \widetilde{u}(v) \widetilde{w}(v), \qquad \widetilde{u}, \widetilde{w} \in \widetilde{X}_V.$$

  where $c_v = \sum_{K_i} \frac{\text{area}(K_i)}{2}$ where K_i are the elements such that $v \in \partial K_i$.

- Edge spaces $\widetilde{X}_e$ for all $e \in \mathcal{T}$:

$$a_e(\widetilde{u}, \widetilde{w}) := c_e \sum_{n=0}^{p-2} q_n \mu_n(\widetilde{u}) \mu_n(\widetilde{w}), \qquad \widetilde{u}, \widetilde{w} \in \widetilde{X}_e$$

  where $c_e = \sum_{K_i} \frac{\text{area}(K_i)}{2}$ where K_i are the elements such that $e \in \partial K_i$,

$$\begin{aligned} q_n &:= \frac{2}{(p+4+n)(p-n+1)} \int_{-1}^{1} (1-x^2)^2 P_n^{(2,2)}(x)^2 \, dx \\ &= \frac{64(n+1)(n+2)}{(p+4+n)(p-n+1)(2n+5)(n+3)(n+4)} \end{aligned} \tag{6.14}$$

  and μ_n is the weighted moment given by

$$\mu_n(u) := \frac{(2n+5)(n+3)(n+4)}{32(n+1)(n+2)} \int_{-1}^{1} (1-x^2) P_n^{(2,2)}(x) u(x) \, dx$$

where we use a linear parametrization such that $e = [-1, 1]$.

Given $f \in X$, the additive Schwarz method from Chapter 2 is:

(i) $u_I \in X_I : a_I(u_I, v_I) = (f, v_I) \quad \forall v_I \in X_I$.

(ii) $u_V \in X_V : a_V(\widetilde{u}_V, \widetilde{v}_V) = (f, \widetilde{v}_V) \quad \forall \widetilde{v}_V \in \widetilde{X}_V$.

(iii) For all edges e in $\mathcal{T}$, $\widetilde{u}_e \in \widetilde{X}_e : a_e(\widetilde{u}_e, \widetilde{v}_e) = (f, \widetilde{v}_e) \quad \forall \widetilde{v}_e \in \widetilde{X}_e$.

(iv) $u := u_I + \widetilde{u}_V + \sum_{e \in \mathcal{T}} \widetilde{u}_e$ is our solution.

The key result regarding the condition number is the following:

Theorem 6.4.1. *The condition number of the above additive Schwarz method is bounded by a constant C independent of h and p (Theorem 3.1 from Chapter 2).*

In the following sections, we discuss the implementation of each of the steps of the ASM preconditioner using a Bernstein basis. Let $\vec{B}^p_V, \vec{B}^p_E, \vec{B}^p_I$ be respectively the vectors such that its entries are the vertex, edge and interior Bernstein polynomials on the mesh. We enumerate the basis analogous to Chapter 2:

1. the vertex functions $\vec{B}^p_V$ in any order
2. the edge functions grouped by edges, and ordered by the multi-indices
3. the interior functions grouped by the element which they are supported on

We can construct the mass matrix for the Bernstein basis on $\mathcal{T}$ in the following block form

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{VV} & \mathbf{M}_{VE} & \mathbf{M}_{VI} \\ \mathbf{M}_{EV} & \mathbf{M}_{EE} & \mathbf{M}_{EI} \\ \mathbf{M}_{IV} & \mathbf{M}_{IE} & \mathbf{M}_{II} \end{bmatrix}$$

where the subscripts indicate the interaction between vertices (V), edges (E) or interiors (I); the residual against the Bernstein polynomials $\vec{f}$ and solution $\vec{x}$ vector can be blocked in a similar way as

$$\bar{f} = \begin{bmatrix} \vec{f}_V \\ \vec{f}_E \\ \vec{f}_I \end{bmatrix} \text{ and } \bar{x} = \begin{bmatrix} \vec{x}_V \\ \vec{x}_E \\ \vec{x}_I \end{bmatrix}.$$

6.4.3 Interior Spaces

In this section, we give an efficient algorithm which was communicated in [10] to solve

$$a_I(u, w) = (f, w) \qquad \forall w \in X_I. \tag{6.15}$$

As X_I is the direct sum of bubble functions on each individual element, we can simply discuss the implementation on the reference element. For the sake of conciseness, we leave all proofs in this section to the appendix.

We recall an orthogonal basis for $\mathbb{P}^p(K) \cap H_0^1(K)$ is given by (see §2.1 of Chapter 2)

$$\psi_{ij}(x, y) = \frac{1-s}{2}\frac{1+s}{2}P_{i-1}^{(2,2)}(s)\left(\frac{1-t}{2}\right)^{i+1}\frac{1+t}{2}P_{j-1}^{(2i+3,2)}(t)$$

for $1 \leq i, j, i+j \leq p-1$, where

$$s = \frac{\lambda_2 - \lambda_1}{1 - \lambda_3}, \quad t = 2\lambda_3 - 1$$

and $\lambda_1, \lambda_2, \lambda_3$ are the barycentric coordinates of T. If we let

$$u(x, y) = \sum_{i=1}^{p-1} \sum_{j=1}^{p-1-i} u_{ij} \psi_{ij}(x, y)$$

for coefficients u_{ij}, then plugging $u(x, y)$ into Equation (6.15) with the test functions $w = \psi_{lm}(x, y)$, we see that $u_{ij} = \frac{(f, \psi_{ij})}{\|\psi_{ij}\|^2}$, hence the solution to Equation (6.15) is simply

$$u(x, y) = \sum_{i=1}^{p-1} \sum_{j=1}^{p-1-i} \frac{(f, \psi_{ij})}{\|\psi_{ij}\|^2} \psi_{ij}(x, y) = \sum_{|\alpha|=p} c_\alpha^p B_\alpha^p. \tag{6.16}$$

Since we are working with the Bernstein polynomials, the question is now a matter of converting from the ψ_{ij} basis to the Bernstein basis.

First, we rewrite the basis functions ψ_{ij} as a multiple of $\lambda_1\lambda_2\lambda_3$, and make a change of variables on the indices obtaining

$$\psi_{ij}|_{r=i-1,\, m-r=j-1} = \lambda_1\, \lambda_2\, \lambda_3\, P_r^{(2,2)}(s) \left(\frac{1-t}{2}\right)^r P_{m-r}^{(2r+5,2)}(t),$$

for $0 \leq r \leq m$ and $0 \leq m \leq p-3$. The next lemma gives allows one to rewrite the interior basis functions as a sum of Bernstein polynomials.

Lemma 6.4.2. *Let $0 \leq r \leq m$ and $0 \leq m \leq p-3$. Then, it holds*

$$P_r^{(2,2)}(s) \left(\frac{1-t}{2}\right)^r P_{m-r}^{(2r+5,2)}(t) = \sum_{|\alpha|=m} a_\alpha^{mr} B_\alpha^m(x, y),$$

where, for $|\alpha| = m$

$$a_\alpha^{mr} = \begin{cases} \nu_{\alpha_3}^{mr} \gamma_{\alpha_2}^{r,m-\alpha_3}, & \textit{for } \alpha_3 \le m-r, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\begin{aligned} \nu_{\alpha_3}^{mr} &= (-1)^{m-r-\alpha_3} \frac{\binom{m+r+5}{\alpha_3}\binom{m-r+2}{m-r-\alpha_3}}{\binom{m}{\alpha_3}}, \\ \gamma_{\alpha_2}^{r,m-\alpha_3} &= \sum_{l=0}^{m-r-\alpha_3} \gamma_{\alpha_2-l}^{r} \frac{\binom{m-r-\alpha_3}{l}\binom{r}{\alpha_2-l}}{\binom{m-\alpha_3}{\alpha_2}}, \\ \gamma_j^r &= (-1)^{r-j} \frac{\binom{r+2}{j}\binom{r+2}{r-j}}{\binom{r}{j}}, \end{aligned}$$

for $j = 0, \ldots, r$*. Note that* γ_j^r *are the Bernstein-Bézier coefficients of the one-dimensional Jacobi polynomial* $P_r^{(2,2)}$*.*

To obtain the Bernstein-Bézier coefficients c_α^p of $u(x, y)$, we apply Lemma 6.4.2 to Equation (6.16), obtaining

$$u(x,y) = \lambda_1\lambda_2\lambda_3 \sum_{m=0}^{p-3} \sum_{r=0}^{m} \sum_{|\alpha|=m} a_\alpha^m \frac{(f, \lambda_1\lambda_2\lambda_3 B_\alpha^m)}{\|\psi_{mr}\|^2} \sum_{|\beta|=m} a_\beta^{mr} B_\beta^m(x,y). \tag{6.17}$$

We remark that the form given as above is the sum of Bernstein polynomials of *different orders*; hence care must be taken to ensure that we express $u(x, y)$ as a sum of pth order Bernstein polynomials.

Considering that we are given the Bernstein moments $f_\alpha^p = (f, B_\alpha^p)$ of degree p of a function f, we break down the calculations into 5 steps:

Step 1. Compute moments

$$\tilde{f}_\alpha^{p-3} = (f, \lambda_1\lambda_2\lambda_3 B_\alpha^{p-3}),$$

for the data f^p.

Step 2. Compute

$$S^{mr} = \sum_{|\alpha|=m} a_\alpha^{mr} \frac{\tilde{f}_\alpha^m}{\|\psi_{mr}\|^2}, \quad \text{for } r = 0, \ldots, m,\ m = 0, \ldots, p-3.$$

Step 3. Compute

$$T_\beta^m = \sum_{r=0}^m S^{mr} a_\beta^{mr}, \quad \text{for } |\beta| = m,\ m = 0, \ldots, p-3.$$

Step 4. Compute coefficients c_α^m by raising the coefficients c_α^{m-1} (if $m > 0$) to degree m and adding them to T^m.

Step 5. Compute coefficients c_α^p from coefficients c_α^{p-3} by multiplying by the interior bubble function, i.e.

$$\sum_{|\alpha|=m} c_\alpha^p B_\alpha^p = \lambda_1\lambda_2\lambda_3 \sum_{|\alpha|=p-3} c_\alpha^{p-3} B_\alpha^{p-3}.$$

We will observe in the following sections that the costs of Steps 1, and 5 are of $\mathcal{O}(p^2)$, and Step 4 is of $\mathcal{O}(p^3)$. If we compute the sums in Steps 2 and 3 naively, we end up with a cost of $\mathcal{O}(p^4)$. This is of course not optimal in the sense that it does not match the computational complexity of other algorithms in this paper. In the following subsections we present algorithms computing Steps 1-5, in particular the algorithms for Steps 2 and 3 use recurrence relations in the computations that allow us to achieve an optimal order. In summary, we obtain an algorithm for computing the Bernstein-Bézier coefficients c_α^p of $u(x, y)$ of computational complexity of $\mathcal{O}(p^3)$.

6.4.3.1 Computation of Step 1.

Since the residual vector $\vec{f_I} = (f, B^p_\alpha)$ for α the index set corresponding to the interior domain points, we then note that $\tilde{f}^{p-3}_\alpha$ is obtained by

$$(f, \lambda_1\lambda_2\,\lambda_3 B^{p-3}_\alpha) = \frac{(\alpha_1+1)(\alpha_2+1)(\alpha_3+1)}{(p-2)(p-1)p} f^p_{\alpha+(1,1,1)}.$$

for $|\alpha| = p-3$ as

$$\lambda_1\lambda_2\lambda_3 B^{p-3}_\alpha = \lambda_1\lambda_2\lambda_3\frac{(p-3)!}{\alpha!}\lambda_1^{\alpha_1}\lambda_2^{\alpha_2}\lambda_3^{\alpha_3} = \frac{(\alpha_1+1)(\alpha_2+1)(\alpha_3+1)}{(p-2)(p-1)p} B^p_{\alpha+(1,1,1)}. \tag{6.18}$$

6.4.3.2 Computation of Step 2

We assume that $\tilde{f}^{p-3}_\alpha$ is given from Step 1, then we compute S^{mr} for $r = 0, \dots, m$ and $m = 0, \dots, p-3$. Plugging in the definition of a^{mr}_α, we have

$$\begin{aligned} S^{m,r} &= \frac{1}{\|\psi_{mr}\|^2}\sum_{|\alpha|=m} a^{mr}_\alpha \tilde{f}^m_\alpha = \frac{1}{\|\psi_{mr}\|^2}\sum_{\alpha_3=0}^{m-r}\nu^{mr}_{\alpha_3}\sum_{\alpha_1+\alpha_2=m-\alpha_3}\gamma^{r,m-\alpha_3}_{\alpha_2}\tilde{f}^m_\alpha \\ &:= \frac{1}{\|\psi_{mr}\|^2}\sum_{\alpha_3=0}^{m-r}\nu^{mr}_{\alpha_3} Q^{mr}_{\alpha_3}. \end{aligned}$$

The key to decreasing the number of operations is to use recurrence relations of the coefficients and moments. We claim that we can compute $Q^{mr}_{\alpha_3}$ and S^{mr} with the following strategy:

1. Compute $Q^{mr}_{\alpha_3}$ and S^{mr} for $\alpha_3 = 0, \dots, m-r$, $r = 0, \dots, m$ and $m = p-3$.

   This has a cost of $\mathcal{O}(p^3)$ as we have to iterate over α_3 and r for each $Q^{mr}_{\alpha_3}$, then compute a sum of $\mathcal{O}(p)$. We remark that while we can precompute and store the coefficients $\gamma_{\alpha_2}^{r,m-\alpha_3}$, we can compute these coefficients using a 1D degree raising algorithm (see Lemma 6.7.2).

2. Compute recursively for $m = (p-3)-1, \ldots, 0$:

$$\begin{aligned} Q_{\alpha_3}^{mr} &= \frac{\alpha_3+1}{m+1} Q_{\alpha_3+1}^{m+1r} + \frac{m+1-\alpha_3}{m+1} Q_{\alpha_3}^{m+1r}, \quad \alpha_3 = 0, \ldots, m-r, \\ S^{mr} &= \frac{1}{\|\psi_{mr}\|^2} \sum_{\alpha_3=0}^{m} \nu_{\alpha_3}^{mr} Q_{\alpha_3}^{mr}, \end{aligned}$$

for $r = 0, \ldots, m$.

This step also has a cost of $\mathcal{O}(p^3)$ as we have to loop over m, r, α_3 to calculate $Q_{\alpha_3}^{mr}$; S^{mr} requires us to loop over m, r and sum over m.

We prove this recurrence relation in the appendix.

6.4.3.3 Computation of Step 3.

Similar to Step 2, we compute the terms T_β^m for each m using the following strategy:

1. Compute T_β^m for $|\beta| = m$ and $\beta_3 = 0$.

   Since we are looping over $|\beta| = m$ such that $\beta_3 = 0$, this loop is of $\mathcal{O}(p)$. We also have to calculate the sum for T_β^m at each loop, which incurs a cost of $\mathcal{O}(p)$. Finally, we would have to do this for each m, meaning the final cost is of $\mathcal{O}(p^3)$.

2. Compute for $\beta_3 = 0, \ldots, m-1$

$$T_{\beta+e_3}^m = -\left(\frac{\beta_1+3}{\beta_3+3} T_{\beta+e_1}^m + \frac{\beta_2+3}{\beta_3+3} T_{\beta+e_2}^m \right),$$

for $\beta_2 = 0, \ldots, m - \beta_3 - 1$.

The cost here is clearly $\mathcal{O}(p^3)$ as we need to loop over m, then over β_2, β_3.

6.4.3.4 Computation of Step 4.

We compute the coefficients c^m by adding T^m and c_α^{m-1}. We observe that c_α^{m-1} corresponds to coefficients of degree $m-1$, then we use two dimensional degree raising to carry out the operation (Algorithm 16). The 2D degree raise algorithm has a cost of $\mathcal{O}(p^2)$; we need to do this for all $m < p-3$, hence a cost of $\mathcal{O}(p^3)$.

6.4.3.5 Computation of Step 5.

Similarly to Step 1, we obtain the coefficients c_α^p by using Equation (6.18)

$$c_{\beta+1}^p = \frac{(\beta_1+1)(\beta_2+1)(\beta_3+1)}{(p-2)(p-1)p} c_\beta^{p-3},$$

for $|\beta| = p-3$.

We outline the procedure for computing the coefficients in algorithms 6 and 7.

Algorithm 6 Inversion of interior Bernstein mass matrix

```
Require: Interior Bernstein moments $f_\alpha^p = (f, B_\alpha^p)$, for $|\alpha| = p$, and $0 < \alpha < p$
function $\mathbf{M}_{II}^{-1}(f_\alpha^p)$
    for $|\alpha| = p-3$ do                                                    ▷ Step 1
        $\tilde{f}_\alpha^{p-3} = \frac{(\alpha_1+1)(\alpha_2+1)(\alpha_3+1)}{(p-2)(p-1)p} f_{\alpha+1}^p$
    end for
    $c_\alpha^{p-3} = \texttt{InteriorInverse}\ (\tilde{f}_\alpha^{p-3})$
    for $|\alpha| = p-3$ do                                                    ▷ Step 5
        $c_{\alpha+1}^p = \frac{(\alpha_1+1)(\alpha_2+1)(\alpha_3+1)}{(p-2)(p-1)p} c_\alpha^{p-3}$
    end for
    return $c_\alpha^p$
end function
```

Algorithm 7 Computing interior coefficients

```
function InteriorInverse($\tilde{f}^{p-3}_{\alpha}$)
    $n = p - 3$                                              ▷ For convenience
    for $r = 0, \dots, n$ do                                  ▷ Step 2: Initialize $Q^{nr}$ and $S^{nr}$
        $\gamma^{r,r}$ = Jacobi (2,2)-Bernstein coefficients of degree $r$
        $\alpha_3 = n - r$
        $Q^{n,r}_{\alpha_3} = \gamma^{r,r} \cdot \tilde{f}^{n}_{\cdot,\alpha_3}$
        for $\alpha_3 = n - r - 1, \dots, 0$ do
            $\gamma^{r,n-\alpha_3} = \texttt{DegreeRaise}(\gamma^{r,n-\alpha_3-1})$
            $Q^{n,r}_{\alpha_3} = \gamma^{r,n-\alpha_3} \cdot \tilde{f}^{n}_{\cdot,\alpha_3}$
        end for
        $S^{n,r} = \sum_{\alpha_3=0}^{n-r} \nu^{n,r}_{\alpha_3} Q^{n,r}_{\alpha_3} / \|\psi_{nr}\|^2$
    end for
    for $m = n - 1, \dots, 0$ do                              ▷ Step 2: Recursive portion
        for $r = 0, \dots, m$ do
            for $\alpha_3 = 0, \dots, m - r$ do
                $Q^{m,r}_{\alpha_3} = \frac{(\alpha_3+1)}{m+1} Q^{m+1,r}_{\alpha_3+1} + \frac{(m+1-\alpha_3)}{(m+1)} Q^{m+1,r}_{\alpha_3}$
            end for
            $S^{m,r} = \sum_{\alpha_3=0}^{m-r} \nu^{m,r}_{\alpha_3} Q^{m,r}_{\alpha_3} / \|\psi_{nr}\|^2$
        end for
    end for
    for $m = 0, n$ do
        $T^{m}_{\cdot,0} = \sum_{r=0}^{m} S^{m,r} \nu^{m,r}_{0} \gamma^{r,m}$                  ▷ Step 3: Initialize
        for $\beta_3 = 0, m - 1$ do
            for $\beta_1 = 0, m - \beta_3 - 1$ do
                $\beta_2 = m - (\beta_3 + 1) - \beta_1$                ▷ Step 3: Recursive portion
                $T^{m}_{\beta+e_3} = -\frac{(3+\beta_1)}{(3+\beta_3)} T^{m}_{\beta+e_1} - \frac{(3+\beta_2)}{(3+\beta_3)} T^{m}_{\beta+e_2}$
            end for
        end for
        $c^m + = T^m$                                          ▷ Step 4
        if $m < n$ then
            $c^{m+1} = \texttt{DegreeRaise2D}(c^m)$
        end if
    end for
    return $c^{p-3}_{\alpha}$
end function
```

6.4.4 Edge Spaces

In this section, we give efficient algorithms to solve $a_e(\widetilde{u}, \widetilde{v}) = (f, \widetilde{v})$ for all $\widetilde{v} \in \widetilde{X_e}$ for a given edge e. Without loss of generality, we assume that $e = [-1, 1]$ and that $c_e = 1$. The key to the efficient solver on the edge space is to note that the bilinear form a_e only depends on the value restricted to e which allows us to reformulate the problem over the space X_e rather than $\widetilde{X_e}$.

We break down the edge solver into four distinct steps

Step 1. Reduce the variational form to X_e

Step 2. Compute the residual

Step 3. Compute $a_e(\vec{\varphi}, \vec{\varphi})$ by using a change of basis

Step 4. Find the corresponding solution in $\widetilde{X_e}$.

6.4.4.1 Step 1.

Decompose $\widetilde{u}_e, \widetilde{v}_e \in \widetilde{X_e}$ into edge functions and bubble functions

$$\widetilde{u}_e = u_e + u_b \qquad\qquad \widetilde{v}_e = v_e + v_b$$

where $u_e, v_e \in X_e$, $u_b, v_b \in X_I$ and $(u_b, w_I) = -(u_e, w_I)$ for all $w_I \in X_I$, and analogously for v_b.

On the right hand side

$$\begin{aligned}(f, \widetilde{v}_e) &= (f, v_e) + (f, v_b)\\ &= (f, v_e) + (u_I, v_b)\\ &= (f, v_e) - (u_I, v_e) + (u_I, \widetilde{v}_e) = (f, v_e) - (u_I, v_e).\end{aligned}$$

where u_I is the solution to $(u_I, w_I) = (r, w_I)$ for all $w_I \in X_I$ (i.e. the solution to the interior problem in Section 6.4.3). We also used the fact that $(w_I, \widetilde{v}_e) = 0$ for all $w_I \in X_I$ by definition of $\widetilde{X}_e$.

For the bilinear form, we note that

$$a_e(\widetilde{u}_e, \widetilde{v}_e) = a_e(u_e, v_e) = a_e(u_e|_e, v_e|_e) \qquad \forall \widetilde{v}_e \in \widetilde{X}_e$$

where $u_e|_e$ is the restriction onto e. Hence, we can first find $u_e \in X_e$ such that

$$a_e(u_e, v_e) = (f, v_e) - (u_I, v_e) \qquad \forall v_e \in X_e, \tag{6.19}$$

then use the orthogonality property to find $\widetilde{u}_e \in \widetilde{X}_e$.

6.4.4.2 Computation of Step 2.

Let $\vec{B}_e^p$ be the Bernstein edge polynomials corresponding to the domain points on e (i.e. a basis for X_e), then we see that the right hand side of Equation (6.19) is

$$(f, \vec{B}_e^p) - (u_I, \vec{B}_e^p) = \vec{f}_e - \mathbf{M}_{eI}\vec{u}_I$$

where $\mathbf{M}_{eI}$ the the mass matrix block corresponding to the interaction between $\vec{B}_e^p$ and the interior Bernstein basis, and $\vec{u}_I$ is vector B-form of the solution to $(u_I, w_I) = (f, w_I)$ for all $w_I \in X_I$ (Section 6.4.3). We incur a cost of $\mathcal{O}(p^3)$ here due to the matrix multiply of $\mathbf{M}_{eI}\vec{u}_I$.

6.4.4.3 Computation of Step 3.

Let

$$\vec{\varphi}_e = \begin{bmatrix} (1-x^2)P_0^{(2,2)} \\ \vdots \\ (1-x^2)P_{p-2}^{(2,2)} \end{bmatrix};$$

$\vec{\varphi}_e$ spans the same space as the univariate "interior" Bernstein polynomials (i.e. space spanned by $\{B_1^p, \ldots, B_{p-1}^p\}$). Due to orthogonality of Jacobi polynomials,

$$a_e(\vec{\varphi}_e, \vec{\varphi}_e^T) := \mathbf{D}_{ee} = \operatorname{diag}(q_n)$$

for $0 \leq n \leq p-2$ where q_n is from Equation (6.14) (see §5.1 of Chapter 2).

We use the crucial fact that bivariate Bernstein polynomials restricted to the boundary are simply the univariate Bernstein polynomials [29]; hence X_e restricted to e is simply the span of univariate interior Bernstein polynomials $\{B_1^p, B_2^p, \ldots, B_{p-1}^p\}$. Let us introduce the change of basis matrix $\mathbf{\Gamma}_e$ such that $\vec{\varphi}_e = \mathbf{\Gamma}_e(\vec{B}_e|_e)$ on e, then

$$\mathbf{D}_{ee} = a_e(\vec{\varphi}_e, \vec{\varphi}_e^T) = a_e(\mathbf{\Gamma}_e(\vec{B}_e|_e), (\mathbf{\Gamma}_e(\vec{B}_e|_e))^T) = \mathbf{\Gamma}_e a_e(\vec{B}_e, \vec{B}_e^T)\mathbf{\Gamma}_e^T.$$

The bilinear form under the Bernstein basis $a_e(\vec{B}_e, \vec{B}_e)$ corresponds to $\mathbf{\Gamma}_e^{-1}\mathbf{D}_{ee}\mathbf{\Gamma}_e^{-T}$.

In order to invert this, we need efficient ways to compute $\mathbf{\Gamma}_e$ and $\mathbf{\Gamma}_e^T$.

Rather than store the matrix $\mathbf{\Gamma}_e$, we present algorithms which can compute their actions. For the action of $\mathbf{\Gamma}_e$, we note that given a function f on e

$$\vec{\varphi}_e = \mathbf{\Gamma}_e(\vec{B}_e|_e) \implies (f, \vec{\varphi}_e) = \mathbf{\Gamma}_e(f, (\vec{B}_e|_e)),$$

hence the operator $\mathbf{\Gamma}_e$ converts the residual from the 1D Bernstein basis to the residual with respect to $\vec{\varphi}_e$ basis:

$$\mathbf{\Gamma}_e : \begin{bmatrix} (f, B_1^p) \\ \vdots \\ (f, B_{p-1}^p) \end{bmatrix} \to \begin{bmatrix} (f, (1-x^2)P_0^{(2,2)}) \\ \vdots \\ (f, (1-x^2)P_{p-2}^{(2,2)}) \end{bmatrix}$$

and likewise $\mathbf{\Gamma}_e^T$ is the operator which converts the coefficients of a polynomial expanded with $\vec{\varphi}_e$ into the B-form coefficients as follows:

$$\mathbf{\Gamma}_e^T : (1-x^2)\sum_{j=0}^{p-2} w_j P_j^{(2,2)}(x) \to \sum_{j=1}^{p-1} c_j B_j^p(x).$$

The key identity to an efficient implementation of $\mathbf{\Gamma}_e$ is

$$(1-x^2)P_n^{(2,2)} = 4\sum_{i=0}^{n} \frac{\binom{n+2}{i}\binom{n+2}{n-i}}{(-1)^{n-i}\binom{n}{i}} \frac{i+1}{n+1} \frac{n-i+1}{n+2} B_{i+1}^{n+2}, \tag{6.20}$$

which is obtained from Equation (6.12) and Equation (6.25), hence

$$(f, (1-x^2)P_n^{(2,2)}) = 4\sum_{i=0}^{n} \frac{\binom{n+2}{i}\binom{n+2}{n-i}}{(-1)^{n-i}\binom{n}{i}} \frac{i+1}{n+1} \frac{n-i+1}{n+2} (f, B_{i+1}^{n+2}).$$

Thus, knowing $(f, B_1^p), \ldots, (f, B_{p-1}^p)$ allows us to calculate $(f, (1-x^2)P_{p-2}^{(2,2)})$. A degree lowering operation (see Algorithm 15) can then be used to obtain $(f, B_1^{p-1}), \dots, (f, B_{p-2}^{p-1})$ which allows us to calculate $(f, (1-x^2)P_{p-3}^{(2,2)})$. We can recursively do this to figure out the rest of the residuals. The following function performs $\mathbf{\Gamma}_e$:

Algorithm 8 Γ_e: Converts (f, B_i^p) into $(f, (1-x^2)P_i^{(2,2)})$

```
Require: $\vec{b}$, a vector of length $p-1$
function Gamma(b)
    for $i = p-2$ to $0$ do
        for $j = 0, \dots, i$ do
            $o[i] = o[i] + 4\frac{\binom{i+2}{j}\binom{i+2}{i-j}}{(-1)^{i-j}\binom{i}{j}}\frac{j+1}{i+1}\frac{i-j+1}{i+2}b[j]$
        end for
        $\vec{b} :=$ DegreeLower($\vec{b}$)        ▷ Degree lower moments; cost of $\mathcal{O}(p)$
    end for
    return $\vec{o}$
end function
```

We note that `Gamma` clearly has a cost of $\mathcal{O}(p^2)$.[2]

The key to computing $\mathbf{\Gamma}_e^T$ is to use Equation (6.20) again. Starting with w_0, we can find the coefficients with respect to B_1^2. We perform a degree raising operation on the B_1^2 coefficient to obtain the coefficients in B_1^3, B_2^3. Now, we can use w_1 to find the coefficient with respect to B_1^3, B_2^3 and sum. We keep on degree raising, and using Equation (6.20) to obtain the following algorithm for $\mathbf{\Gamma}_e^T$:

[2] The coefficients can be computed with little cost by noting that

$$4\frac{\binom{i+2}{j}\binom{i+2}{i-j}}{(-1)^{i-j}\binom{i}{j}}\frac{j+1}{i+1}\frac{i-j+1}{i+2} = 4\binom{i+2}{j}\frac{i-j+1}{(j+2)(-1)^{i-j}}$$

hence one can either pre-computing the binomial coefficients up to order p, or updating the binomial coefficients in the for loop for i on the fly while calculating `Gamma`

Algorithm 9 $\boldsymbol{\Gamma}_e^T$: Converts the coefficients of a Jacobi polynomial to a B-form coefficients

```
Require: $w$, a vector of length $p-1$
function GammaTran(w)
    Initialize $o$ of length 1
    for $i = 0, \dots, p-1$ do
        for $j = 0, \dots, i$ do
            $o[i] = o[i] + 4\frac{\binom{i+2}{j}\binom{i+2}{i-j}}{(-1)^{i-j}\binom{i}{j}}\frac{j+1}{i+1}\frac{i-j+1}{i+2}w[j]$
        end for
        $o =$ DegreeRaise($o$)                    ▷ Degree raise B-net; cost of $\mathcal{O}(p)$
    end for
    return $o$
end function
```

Again, we see that `Gammatran` also have a cost of $\mathcal{O}(p^2)$ as the coefficients can be calculated as before. Hence, we can easily compute $\boldsymbol{\Gamma}_e^T \mathbf{D}_{ee}^{-1} \boldsymbol{\Gamma}_e$ with cost of $\mathcal{O}(p^2)$ and efficiently compute the solution to the variational problem Equation (6.19)

$$\vec{u}_e := \boldsymbol{\Gamma}_e^T \mathbf{D}_{ee}^{-1} \boldsymbol{\Gamma}_e (\vec{f}_e - \mathbf{M}_{eI} \vec{u}_I).$$

6.4.4.4 Step 4.

Finally, we recall $\vec{u}_e$ from above corresponds to

$$u_e \in X_e : a_e(u_e, v_e) = (f, v_e) - (u_I, v_e) \qquad \forall v_e \in X_e$$

but the solution we need is $\widetilde{u}_e$ in $\widetilde{X_e}$ with $\widetilde{u}_e = u_e + u_b$. Recall that $(u_b, w_I) = -(u_e, w_I)$ for all $w_I \in X_I$, hence the interior correction can be computed by

$$\vec{u}_b = -\mathbf{M}_{II}^{-1} \mathbf{M}_{Ie} \vec{u}_e.$$

This has a cost of $\mathcal{O}(p^3)$ if we use Algorithm 6.

6.4.5 Vertex Spaces

In this section, we discus how to solve the variational problem $a_V(\widetilde{u}_v, \widetilde{w}_v) = (f, \widetilde{w}_v)$ for $\forall \widetilde{w}_v \in \widetilde{X}_V$. We proceed similarly to the edge solves.

Step 1. Reduce the variational form to X_V

Step 2. Perform change of basis, and calculate the residual

Step 3. Compute the bilinear form

Step 4. Find the corresponding solution in $\widetilde{X}_V$

6.4.5.1 Step 1.

Decompose $\widetilde{u}_v = u_v + u_b$ where $u_v \in X_V, u_b \in X_I$ with a similar decomposition for the test function $\widetilde{w}_v$. By the orthogonality property of $\widetilde{X}_V$, we have that

$$(u_v, w_b) = -(u_b, w_b) \qquad \forall w_b \in X_I. \tag{6.21}$$

and we also recall that

$$(u_I, w_b) = (f, w_b) \qquad \forall w_b \in X_I. \tag{6.22}$$

For the right hand side, we have again

$$\begin{aligned}(f, \widetilde{w}_v) &= (f, w_v) + (f, w_b)\\ &= (f, w_v) + (u_I, \widetilde{w}_v) - (u_I, w_v)\\ &= (f, w_v) - (u_I, w_v).\end{aligned}$$

As $a_V(\widetilde{u}_v, \widetilde{w}_v) = a_V(u_v, w_v)$, we can find $u_v \in X_V$

$$a_V(u_v, w_v) = (f, w_v) - (u_I, w_v) \qquad \forall w_v \in X_V \tag{6.23}$$

then use orthogonality properties to find $\widetilde{u}_v \in \widetilde{X}_V$.

6.4.5.2 Step 2.

Unfortunately, X_V is not simply the span of the Bernstein polynomials. For an arbitrary vertex $v \in \mathcal{T}$, we note that we can rewrite the basis function φ_v as a linear combination of Bernstein polynomials

$$\varphi_v = B_v^p + \vec{\phi}_v^T \vec{B}_E^p + \vec{\chi}_v^T \vec{B}_I^p \tag{6.24}$$

where B_v^p is the Bernstein vertex basis at vertex v, $\vec{\phi}_v$ and $\vec{\chi}_v$ are vectors of appropriate coefficients. We will see that we need to compute $\vec{\phi}_v$, but not $\vec{\chi}_v$.

On the right hand side, using Equation (6.22) for an arbitrary vertex $v \in \mathcal{T}$ and the fact that $(u_I, w_I) = (f, w_I)$ for all $w_I \in X_I$,

$$\begin{aligned}(f, \varphi_v) - (u_I, \varphi_v) &= (f, B_v^p + \vec{\phi}_v^T \vec{B}_E^p + \vec{\chi}_v^T \vec{B}_I^p) - (u_I, B_v^p + \vec{\phi}_v^T \vec{B}_E^p + \vec{\chi}_v^T \vec{B}_I^p) \\ &= (f, B_v^p) - (u_I, B_v^p) + (f, \vec{\phi}_v^T \vec{B}_E^p) - (u_I, \vec{\phi}_v^T \vec{B}_E^p) \\ &= (\vec{f}_V)_v - (\mathbf{M}_{VI} u_I)_v + \vec{\phi}_v^T (\vec{f}_E - \mathbf{M}_{EI} u_I)\end{aligned}$$

where $(\vec{f}_V)_v$ and $(\mathbf{M}_{VI} u_I)_v$ is the row corresponding to B_v^p. The key here is that the interior component does not matter in the computation.

We need to compute $\vec{\phi}_v^T$ for all vertices v which are the coefficients such that $B_v^p + \vec{\phi}_v^T B_E^p$ on the edges equals φ_v. Without loss of generality, given a vertex v, assume an edge e from v is parametrized to be $[-1, 1]$. We recall that φ_v restricted to the edge is Equation (6.13), hence using Equation (6.12), and factoring in the $(1 - x)/2$ term,

$$\begin{aligned}\varphi_v(x)|_e &= \frac{(-1)^{\lfloor p/2 \rfloor + 1}}{\lfloor p/2 \rfloor} \left(\frac{1-x}{2}\right) P_{\lfloor p/2 \rfloor - 1}^{(1,1)}(x) \\ &= \frac{1}{\lfloor p/2 \rfloor} \left(\sum_{j=0}^{\lfloor p/2 \rfloor - 1} \binom{\lfloor p/2 \rfloor}{\lfloor p/2 \rfloor - 1 - j} (-1)^j B_j^{\lfloor p/2 \rfloor}(x) \right) \\ &= B_0^p(x) + \sum_{j=1}^{p-1} \widetilde{c}_j B_j^p(x).\end{aligned}$$

Hence the coefficients we want are $\widetilde{c}_j$, which are the result of using the degree raising formula on $\binom{\lfloor p/2 \rfloor}{\lfloor p/2 \rfloor - 1 - j} (-1)^j$.

Let $\boldsymbol{\phi}$ be the matrix with columns the vector $\vec{\phi}_v$; Algorithm 10 calculates $\boldsymbol{\phi}$ by first computing the coefficients for $B_j^{\lfloor p/2 \rfloor}$, degree raising it to the appropriate order B_j^p, then place the coefficients in the appropriate degrees of freedom in $\boldsymbol{\phi}$. We note that in line 9, we remove the first and last term as that corresponds to the vertex terms. $\boldsymbol{\phi}$ can be precomputed.

Algorithm 10 Computing the values of $\boldsymbol{\phi}$

```
$\boldsymbol{\phi}$ = zeros(numbers of dofs on edges, number of vertices)    ▷ Initialize Matrix
$q := \lfloor p/2 \rfloor$
for $i = 0$ to $q$ do
    $\vec{c}[i] = \frac{(-1.0)^i}{q}\binom{q}{q-1-i}$    ▷ Generate lower-order coefficients
end for
for $i = 0, \dots, p-q-1$ do
    $\vec{c}$ = DegreeRaise1D($\vec{c}$)
end for
$\vec{c} = \vec{c}[1 : p-1]$    ▷ Remove first and last term; length of $p-1$
for $K \in \mathcal{T}$ do
    for vertex $v_i$ in $K$ do
        Let $v_j, v_k$ be the two other vertices of $K$
        $\vec{d_1}$ := DOFs on edge from vertex $v_i$ to $v_j$
        $\vec{d_2}$ := DOFs on edge from vertex $v_i$ to $v_k$
        $\boldsymbol{\phi}[\vec{d_1}, \text{dof of } v_i] = \vec{c}$    ▷ Set an array equal to another array
        $\boldsymbol{\phi}[\vec{d_2}, \text{dof of } v_i] = \vec{c}$
    end for
end for
```

6.4.5.3 Step 3.

The matrix form of the bilinear form is trivial as

$$a_V(\widetilde{u}_v, \widetilde{w}_v) = a_V(u_v, w_v) = \frac{1}{p^4}\mathbf{c}_v$$

where $\mathbf{c}_v$ is a diagonal matrix with c_v as its entries. With the matrix $\boldsymbol{\phi}$ computed, we can solve for Equation (6.23) with the following

$$\frac{1}{p^4}\mathbf{c}_v\vec{\varphi}_v = \vec{f_V} - \mathbf{M}_{VI}\vec{u}_I + \boldsymbol{\phi}^T(\vec{f_E} - \mathbf{M}_{EI}\vec{u}_I).$$

The cost to compute $\vec{\varphi}_v$ is dependent on the number of vertices, but the main cost is $\mathcal{O}(p^3)$ due to the matrix-vector multiply of $\mathbf{M}_{EI}\vec{u}_I$.

6.4.5.4 Step 4.

Finally, the solution vector $\vec{\varphi}_v$ is under the φ_v basis of X_V so we have to manipulate this solution in order to find the corresponding solution in $\widetilde{X}_V$ expanded with the Bernstein basis.

Using Equation (6.24), the coefficient $\vec{u}_v$ for B_V^p is simply $\vec{\varphi}_v$, and the coefficients for the edge Bernstein polynomials are $\vec{u}_E = \boldsymbol{\phi}\vec{\varphi}_v$.

As for the interior bubble functions, we recall the orthogonality condition Equation (6.21). Hence, we do not need to compute $\vec{\chi}_v^T$, but only the following variational problem for all $w_b \in X_I$

$$(\vec{u}_v^T B_V^p + \vec{u}_E^T B_E^p, w_b) = -(\vec{u}_b^T B_I^p, w_b) \implies \mathbf{M}_{IV}\vec{u}_v + \mathbf{M}_{IE}\vec{u}_E = -\mathbf{M}_{II}\vec{u}_b$$

and and hence $\vec{u}_b = -\mathbf{M}_{II}^{-1}\mathbf{M}_{IV}\vec{\varphi}_v - \mathbf{M}_{II}^{-1}\mathbf{M}_{IE}\boldsymbol{\phi}\vec{\varphi}_v$. The cost to compute this is $\mathcal{O}(p^3)$ if we use Algorithm 6.

6.4.6 Matrix Formulation

Collecting the algorithms above, we can finally display the algorithm to precondition the mass matrix of the Bernstein basis. Let the local assembly matrix $\mathbf{\Lambda}_K$ be written in block form

$$\mathbf{\Lambda}_K = \begin{bmatrix} \mathbf{\Lambda}_{K,V} \\ \mathbf{\Lambda}_{K,E} \\ \mathbf{\Lambda}_{K,I} \end{bmatrix}$$

where the blocks correspond to the vertex, edge and interior basis functions on element K, then let the matrices $\mathbf{D}_{EE}$ and $\mathbf{D}_{VV}$ be diagonal matrices defined as

$$\mathbf{D}_{VV} = \sum_{K \in \mathcal{T}} \frac{|K|}{2p^4} \mathbf{\Lambda}_{K,V} \mathbf{\Lambda}_{K,V}^T \text{ and } \mathbf{D}_{EE} = \sum_{K \in \mathcal{T}} \frac{|K|}{2} \mathbf{\Lambda}_{K,E} \hat{\mathbf{D}}_{EE} \mathbf{\Lambda}_{K,E}^T$$

where

$$\hat{\mathbf{D}}_{EE} = \text{block diag}(\hat{\mathbf{D}}_{EE}^{(1)}, \hat{\mathbf{D}}_{EE}^{(2)}, \hat{\mathbf{D}}_{EE}^{(3)})$$

for $\hat{\mathbf{D}}_{EE}^{(i)}, i = 1, 2, 3$ is the diagonal matrix $\hat{\mathbf{D}}_{EE}^{(i)} = \text{diag}(q_j)$, with q_j defined from Equation (6.14) for $j = 0, \ldots, p-2$. We let $\mathbf{\Gamma}_{EE}$ and $\mathbf{\Gamma}_{EE}^T$ simply be the applications of algorithms $\mathbf{\Gamma}_e, \mathbf{\Gamma}_e^T$ repeatedly for each edge.

Then, we can formulate the additive Schwarz preconditioner as

Algorithm 11 P: Preconditioner for the Bernstein Basis Mass Matrix

```
Require: $\mathbf{M}$ global mass matrix, $\vec{f}$ residual vector
function
    $\vec{x}_I := \mathbf{M}_{II}^{-1} \vec{f}_I$    ▷ Interior solve using Section 6.4.3
    $\vec{x}_E := \mathbf{\Gamma}_{EE}^{-T} \mathbf{D}_{EE}^{-1} \mathbf{\Gamma}_{EE}^{-1} \left( \vec{f}_E - \mathbf{M}_{EI} \vec{x}_I \right)$    ▷ Edges solve
    $\vec{x}_V := \mathbf{D}_{VV}^{-1} \left( (\vec{f}_V - \mathbf{M}_{VI} \vec{x}_I) + \boldsymbol{\phi}^T \left( \vec{f}_E - \mathbf{M}_{EI} \vec{x}_I \right) \right)$    ▷ Vertices solve
    $\vec{x}_E := \vec{x}_E + \boldsymbol{\phi} \vec{x}_V$
    $\vec{x}_I := \vec{x}_I - \mathbf{M}_{II}^{-1} \mathbf{M}_{IV} \vec{x}_V - \mathbf{M}_{II}^{-1} \mathbf{M}_{IE} \vec{x}_E$    ▷ Interior correction
    return $x := [\vec{x}_V; \vec{x}_E; \vec{x}_I]$
end function
```

6.4.7 Schur Preconditioner

In this subsection, we present a variation of the above algorithm which is more suited for explicit time-stepping such as the Nyström method or an explicit Runge-Kutta method. We first let $\mathbf{M}$ be the global mass matrix with the Bernstein basis, and block the matrix as follows

$$\mathbf{A} = \begin{bmatrix} \mathbf{M}_{VV} & \mathbf{M}_{VE} \\ \mathbf{M}_{EV} & \mathbf{M}_{EE} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \mathbf{M}_{VI} \\ \mathbf{M}_{EI} \end{bmatrix}, \text{ and } \mathbf{C} = \begin{bmatrix} \mathbf{M}_{II} \end{bmatrix}.$$

One way of solving $\mathbf{M}\vec{x} = \vec{f}$ is to use the Schur complement method (otherwise known as static condensation [80]) by first solving the boundary values:

$$\ddot{\mathbf{S}} \begin{bmatrix} \vec{x}_V \\ \vec{x}_E \end{bmatrix} = \begin{bmatrix} \vec{f}_V \\ \vec{f}_E \end{bmatrix} - \mathbf{B}\mathbf{C}^{-1}\vec{f}_I = \begin{bmatrix} \vec{f}_V - \mathbf{M}_{VI}\mathbf{M}_{II}^{-1}\vec{f}_I \\ \vec{f}_E - \mathbf{M}_{EI}\mathbf{M}_{II}^{-1}\vec{f}_I \end{bmatrix}$$

where $\ddot{\mathbf{S}} = \mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T$, then substitute the solution back to solve for the interior $\vec{x}_I$.

In the case of an explicit time-stepping scheme, we are able to solve for the right-hand side (e.g. $\vec{f}_V - \mathbf{M}_{VI}\mathbf{M}_{II}^{-1}\vec{f}_I$) exactly using Section 6.4.3 and work with the exact Schur complement of the mass matrix.[3] Hence, rather than using conjugate gradient over the mass matrix, *we can simply iterate (and precondition)* on the smaller Schur complement then substitute back into the interior dofs. This idea was first mentioned in Remark 2.7 of [15].

The Schur complement preconditioner is the "middle" portion of Algorithm 11 and is presented in Algorithm 12 independently. Here, we again emphasize that using the Bernstein basis allows for matrix-free computation of the matrix-vector product [3], which coupled with the inversion of the interior blocks Section 6.4.3 allows for matrix-free Schur complement products. Finally, the preconditioner for the whole mass matrix based on preconditioning the Schur complement is presented

[3] This is contrasted against an implicit time-stepping scheme where the right hand side in the Schur complement method will requires $(\mathbf{M}_{II} + c\mathbf{S}_{II})^{-1}$ which cannot be as easily computed exactly as $\mathbf{M}_{II}$.

in Algorithm 13.

Algorithm 12 $\ddot{\mathbf{P}}^{-1}$: Preconditioner for Schur Complement

```
Require: $\vec{f}$ residual vector
function
    $\vec{x}_E := \boldsymbol{\Gamma}_{EE}^{-T}\mathbf{D}_{EE}^{-1}\boldsymbol{\Gamma}_{EE}^{-1}\left(\vec{f}_E\right)$    ▷ Edges solve
    $\vec{x}_V := \mathbf{D}_{VV}^{-1}\left(\vec{f}_V + \boldsymbol{\phi}\vec{f}_E\right)$    ▷ Vertices solve
    $\vec{x}_E := \vec{x}_E + \boldsymbol{\phi}\vec{x}_V$
    return $x := [x_V; x_E]$
end function
```

Algorithm 13 $\tilde{\mathbf{P}}$: Preconditioner for Mass Matrix using $\ddot{\mathbf{P}}^{-1}$

```
Require: $\mathbf{M}$ global Bernstein mass matrix, $\ddot{\mathbf{S}}$ Schur complement of Bernstein basis,
    $\vec{f}$ residual vector
function
    $\vec{x}_I := \mathbf{M}_{II}^{-1}\vec{f}_I$    ▷ Interior solve
    $\widetilde{f}_V = \vec{f}_V - \mathbf{M}_{VI}\mathbf{M}_{II}^{-1}\bar{f}_I$    ▷ Find right-hand sides for Schur complement
    $\widetilde{f}_E = \vec{f}_E - \mathbf{M}_{EI}\mathbf{M}_{II}^{-1}\bar{f}_I$
    $[\vec{x}_V; \vec{x}_E] := \texttt{pcg}(\ddot{\mathbf{S}}, [\widetilde{f}_V; \widetilde{f}_E], \text{Preconditioner} = \ddot{\mathbf{P}}^{-1})$    ▷ Iterate the boundaries
    $\vec{x}_I := \vec{x}_I - \mathbf{M}_{II}^{-1}\mathbf{M}_{IV}\vec{x}_V - \mathbf{M}_{II}^{-1}\mathbf{M}_{IE}\vec{x}_E$    ▷ Interior correction
    return $x := x_I + x_E + x_V$
end function
```

6.5 Illustrative Numerical Examples

6.5.1 Brusselator and Implicit Time-Stepping

We now illustrate the use of the preconditioner in the numerical solution of the Brusselator system. Let $u(x, y, t)$ and $v(x, y, t)$ be the solution to the Brusselator system with initial conditions and time-stepping scheme as described in Section 6.2.2. The spatial discretization is a uniform triangulation of the square with 256 elements.

In Table 6.2, we show the [min, median, max] of the iteration counts of the preconditioned conjugate gradient (PCG) method required to solve for both $u(x,y,t)$ and $v(x,y,t)$ separately. We note that while we are preconditioning a perturbation of the mass matrix, the choice of $\Delta t \sim \frac{h^2}{p^2}$ and a good initial iterate seems to allow us to have *non-increasing* iteration counts as opposed to the $\mathcal{O}(p^2)$ growth shown in Section 6.2.2. This is partly due to the fact that the diffusion coefficient is so small, and the fact that we are using the previous time-step as the initial iterate for PCG.

We also will use this case study to showcase the advantages of using the Bernstein basis in calculating the critical nonlinear moments at each time-step as mentioned in Section 6.3.5. In Figure 6.12, we plot the average number of milliseconds required to calculate the nonlinear moment $(u_n^2 v_n, \vec{\varphi})$ at each time step.[4] We note that while [3] indicated that the asymptotic cost is $\mathcal{O}(p^3)$, in the range of $p \in [3, \ldots, 20]$, we instead see a better cost growth of only $\mathcal{O}(p^2)$.

Table 6.2: Table to illustrate the performance of the preconditioned iterative method to the matrix resulting from a IMEX scheme by displaying the [min, median, max] iteration count of the PCG solves for the variable u and v in a period of 10 seconds on 256 elements for the Brusselator in a uniformly triangulated square. Our scaling for Δt is such that $\Delta t \sim \frac{1}{p^2}$.

p	Δt	Iteration count u	Iteration count v
4	1/10	[22, 25, 28]	[24, 27, 31]
8	1/40	[19, 22, 26]	[20, 23, 27]
12	1/90	[18, 21, 24]	[20, 22, 26]
16	1/160	[18, 21, 26]	[19, 23, 26]
20	1/250	[18, 22, 27]	[20, 23, 27]

[4] Timings were done using Python 3 with the key kernels from [3] written in Cython on an Ryzen 5 1600 processor and 16GB of Ram

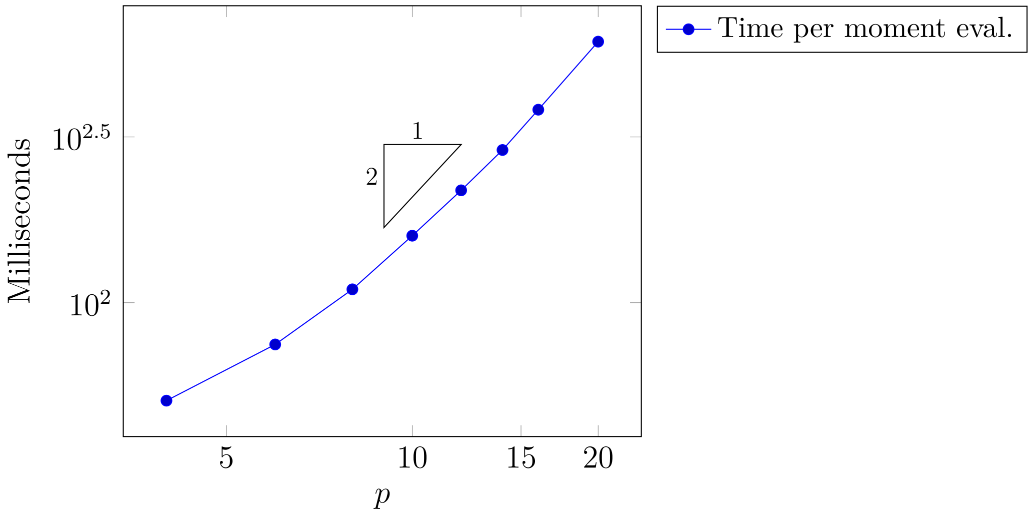
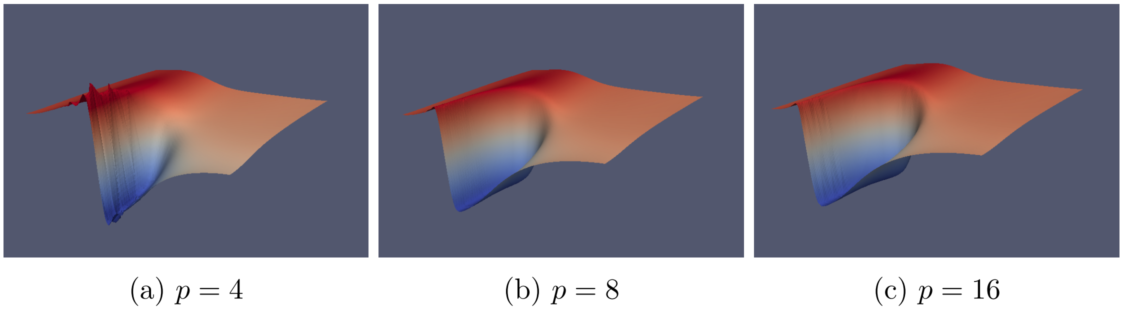


Figure 6.12: The average time required to compute the non-linearity in the Brusselator system is plotted on a log-log axis. We note that for the orders we are examining, the growth is $\mathcal{O}(p^2)$ rather than the asymptotic growth of $\mathcal{O}(p^3)$ from [3]. The asymptotic growth is observed for $p > 30$ (see [3]).



(a) $p = 4$

(b) $p = 8$

(c) $p = 16$

Figure 6.13: Plots of the solution to the Brusselator example at $t = 5$ with the z-axis scaled by a factor of .1 and time-steps as in Table 6.2.

6.5.2 Sine-Gordon and Explicit Time-Stepping

We now illustrate the use of the preconditioner in the numerical solution of the sine-Gordon equation, using an explicit time-stepping scheme which requires the inversion of the exact mass matrix at each step. Let $u(x, y, t)$ be the solution to the sine-Gordon equation using the fourth order Nyström method [40, p. 285] as described in Section 6.2.1. We use a uniform triangulation of the square $[-7, 7] \times [-7, 7]$ in the spatial dimension.

In order to time-step, we use PCG with the initial iterate to be the previous time step (or sub-step). In Table 6.3, we display the [min, median, max] iteration count for all 3000 PCG calls required to time step 10 seconds. As in Chapter 2, we expect the number of iterations to not increase as we refine the mesh or increase p. Indeed, we see that the median iteration counts in Table 6.3 is *the same* as the iteration counts as in the linear wave equation considered in Chapter 2; this is not an unexpected result as only the residuals have changed from the linear heat equation.

While the above result is certainly favorable, the case of explicit time-stepping allows for the use of the preconditioner of just the Schur complement as described in Section 6.4.7. In Table 6.4, we display the iteration count of solving the Schur complement (i.e. the iteration counts of line 5 of Algorithm 13) in the period of 10 seconds for solving the sine-Gordon equation. We note that the iteration count does not increase as we refine h or p which we prove in Section 6.7.1.

Finally, we will use the Sine-Gordon example to demonstrate that PCG is achieving the required accuracy. In Section 6.5.2, we plot the residual of each iteration from PCG of the first linear solve at $t = 0$ for 64 elements; the residual decreases quite nicely and we achieve a tolerance of 10^{-9} easily. In fact, we note that the number of iterations decreases as p increases which matches Table 6.3 and Table 6.4.

Table 6.3: Table illustrates the performance of the preconditioned iterative method of the mass matrix at each time step by displaying the [min, median, max] iteration count of all 3000 PCG solves from using the Nyström method for a period of 10 seconds with a $\Delta t = .01$ in a uniformly triangulated square for the sine-Gordon equation.

Order	16 Elements	64 Elements	256 Elements
4	[21, 26, 32]	[20, 25, 34]	[17, 23, 31]
8	[17, 23, 29]	[16, 21, 30]	[16, 21, 26]
12	[17, 22, 27]	[16, 18, 26]	[17, 17, 24]
16	[16, 18, 25]	[15, 18, 24]	[15, 15, 22]
20	[16, 18, 24]	[15, 15, 23]	

Table 6.4: Table illustrates the performance of the preconditioner *based on the Schur complement* of the mass matrix at each time step by displaying the [min, median, max] iteration count of the Schur complement solve (line 5 of Algorithm 13) from using the Nyström method for a period of 10 seconds with a $\Delta t = .01$ in a uniformly triangulated square for the sine-Gordon equation.

Order	16 Elements	64 Elements	256 Elements
4	[22, 27, 33]	[21, 26, 35]	[18, 24, 32]
8	[18, 24, 30]	[17, 22, 31]	[17, 22, 27]
12	[18, 23, 28]	[17, 19, 27]	[17, 18, 25]
16	[17, 19, 26]	[1, 19, 25]	[16 ,16, 23]
20	[1, 18, 25]	[1, 16, 24]	

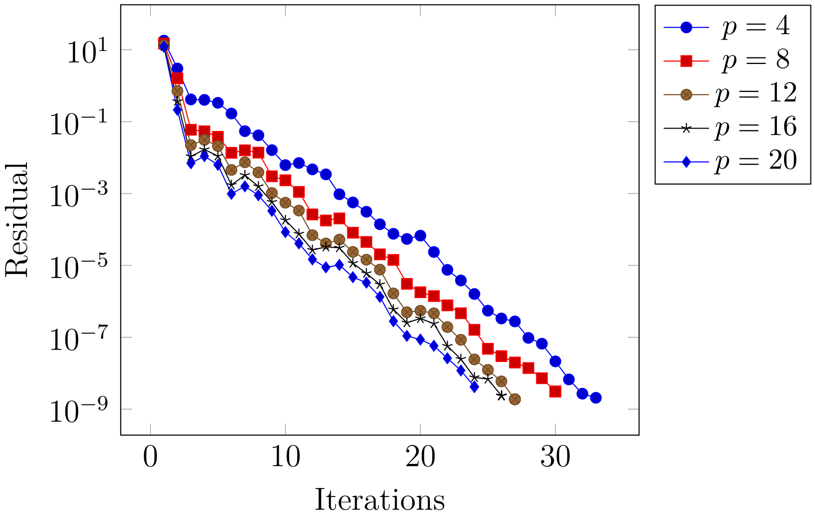


Figure 6.14: Plot of the residuals resulting from the preconditioned conjugate gradient method applied to the Sine-Gordon example at $t = 0$ and 64 elements.

6.5.3 Boundary Layer Problems

We now illustrate the use of the preconditioner in the numerical solution of a boundary layer problem. Let $u(x, y)$ be the solution to the problem as described in Section 6.2.3 with $f = 1$. In Chapter 2, we remarked that our mass preconditioner allows for needle elements, hence we showcase this capability by using the mesh as shown in Figure 6.4.

In Figure 6.15, we plot the condition number of the preconditioned system

$\mathbf{P}^{-1/2}(\mathbf{M} + \varepsilon^2\mathbf{S})\mathbf{P}^{-1/2}$. We observe that the growth of the condition numbers grows as p^2 as Equation (6.9) suggests, and that for ε small enough, that the condition numbers do not depend on ε.

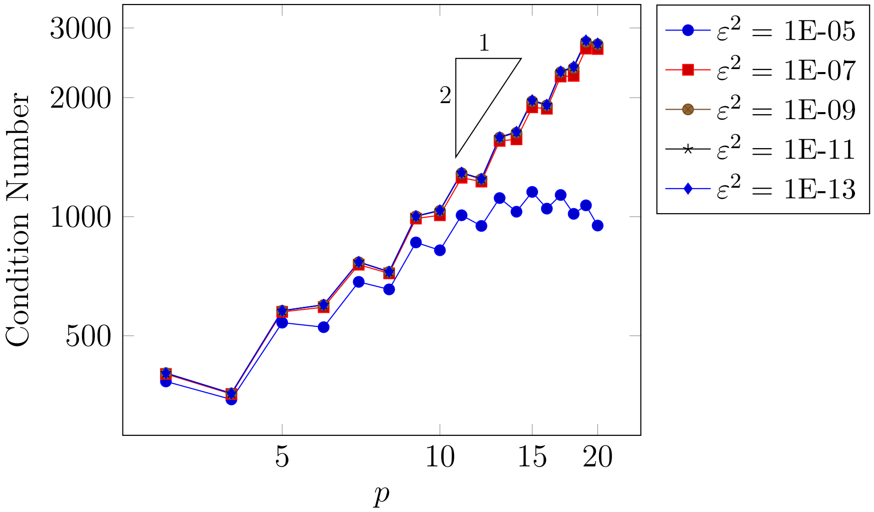


Figure 6.15: The condition numbers of the preconditioned system for the boundary layer problem using the mesh in Figure 6.4 are displayed for varying ε and p in a log-log scale. We note that the growth of the condition number satisfies the Schmidt's inequality estimate, and that as we decrease ε, the condition number seems to converge to a curve.

6.6 Conclusions

The current work described the efficient implementation of a p-version mass matrix preconditioner using the Bernstein basis, alongside useful post-processing procedures such as visualization and gradient evaluations. Of particular note is an algorithm to invert the interior blocks of the mass matrix in $\mathcal{O}(p^3)$ operations. This allowed us to perform the preconditioning step with a total cost of $\mathcal{O}(p^3)$, hence, combined with the results from [3], allows for one to construct the mass and stiffness matrices, time step, and perform post-processing of nonlinear transient problems all with a cost of $\mathcal{O}(p^3)$. While preconditioning the mass matrix will offer no advantages for problems where only the stiffness matrix is present, we also showed that certain challenging elliptic problems such as the singularly perturbed problem can be handled by a preconditioner for the mass matrix. Some of the algorithms does extend naturally to tetrahedrons such as the de Casteljau algorithm, and the Bernstein basis matrix construction and multiplies from [3]. Unfortunately, the interior inversion algorithm does not extend as easily to 3D and will be the subject of a forthcoming work.

6.7 Appendix

6.7.1 Schur Complement Preconditioner

In this section, we present a short proof that the preconditioner for the Schur complement (Algorithm 12) has bounded condition numbers. Like in Chapter 2, it suffices to show the result on the reference triangle. Let $\widetilde{X}_B, \widetilde{X}_V$ and $\widetilde{X}_{E_i}, i = 1, 2, 3$ be the minimal L^2 extension space as defined in §5 of Chapter 2, with the inner-products as $a_V(\cdot, \cdot)$ and $a_{E_i}(\cdot, \cdot)$ from the same section.

The additive Schwarz method preconditioner which arises is given $\widetilde{f} \in \widetilde{X}_B$, find u as follows:

1. $u_V \in \widetilde{X}_V : a_V(u_V, v_V) = (\widetilde{f}, v_V) \quad \forall v_V \in \widetilde{X}_V$.
2. For $i = 1, 2, 3$, $u_{E_i} \in \widetilde{X}_{E_i} : a_{E_i}(u_{E_i}, v_{E_i}) = (\widetilde{f}, v_{E_i}) \quad \forall v_{E_i} \in \widetilde{X}_{E_i}$.
3. $u := u_V + \sum_{i=1}^{3} u_{E_i}$ is our solution.

Note that it is simply what we had in Chapter 2, except the interior solve is not there; this leads to a simple corollary.

Corollary 6.7.1. *The abstract additive Schwarz method defined above corresponds to Algorithm 12 under the Bernstein basis. Furthermore, there exists a constant C independent of h, p such that cond$(\ddot{\mathbf{P}}^{-1}\ddot{\mathbf{S}}) \leq C$.*

Proof. Let us first prove that the abstract ASM has uniform condition number. We see that Lemma 5.3, Lemma 5.4 and Theorem 5.5 can be easily modified to reflect the ASM method above by removing the interior portions from each of the statements; hence this is simply a consequence of Theorem 2.7 of [77].

Finally, applying the exact same techniques from Section 6.4.4 and Section 6.4.5, keeping in mind that we are given $\widetilde{f} \in \widetilde{X}_B$, we see that the ASM method corresponds to Algorithm 12.

□

6.7.2 Dirichlet boundary condition

The enforcement of Dirichlet boundary conditions is trivial to implement for the preconditioner. In Algorithm 11, before the interior correction term (line 6), simply set the degrees of freedom in $\vec{x}_V, \vec{x}_E$ corresponding to the Dirichlet boundary condition equal to the appropriate Bernstein basis values; in our case for the boundary layer case study, this was simply 0.

The more mathematically accurate way would be to modify the diagonal scaling matrices $\mathbf{D}_{VV}, \mathbf{D}_{EE}$ to be 1 at the Dirichlet boundary condition dofs and also use the modified mass matrix (zeroing out the rows and columns and leaving a one on the diagonal) for Dirichlet boundary conditions, the fact that the edge solve and vertex solve are diagonal allows us to use the procedure above.

6.7.3 Degree Raising Algorithms

The degree raising formula for the 1D Bernstein polynomials is easily derived:

$$B_i^p(x) = (\lambda_1 + \lambda_2) B_i^p(x) = \frac{p+1-i}{p+1} B_i^{p+1}(x) + \frac{i+1}{p+1} B_{i+1}^{p+1}(x). \tag{6.25}$$

This allows us to express a Bernstein basis polynomial of degree p as one of degree $p+1$ as such

$$\sum_{i=0}^{p} c_i^p B_i^p(x) = \sum_{i=0}^{p+1} c_i^{p+1} B_i^{p+1}(x).$$

The following subroutine computes c_i^{p+1} in $\mathcal{O}(p)$:

Algorithm 14 Degree Raising Operator

```
Require: $\vec{c}$ corresponding to the B-net of the polynomial of degree $p$
function DegreeRaise1D($\vec{c}$)
    $\vec{o} = \text{zeros}(p+2)$                    ▷ Coefficients for degree $p+1$
    for $i = 0, \dots, p$ do
        $o[i] += (p+1-i)c[i]/(p+1)$
        $o[i+1] += (i+1)c[i]/(p+1)$
    end for
    return $\vec{o}$
end function
```

An equally useful operation is the "degree-lowering operation," which is the opposite of the degree raising operation. This is only used when we are working with inner-products; for example, for a function g, we can deduce (g, B_i^3) for $i = 0, \dots, 3$ given (g, B_j^4) for $j = 0, \dots, 4$. The degree lowering operator also has a cost of $\mathcal{O}(p)$

as it is simply the degree raising operator backwards.

Algorithm 15 Degree Lowering Operator

```
Require: $\vec{c}$ corresponding to the inner-products $(f, B_i^{p+1})$ of Bernstein polynomial
    of degree $p+1$
function DegreeLower($\vec{c}$)
    $\vec{o} = \text{zeros}(p+1)$                          ▷ Coefficients for degree $p$
    for $i = 0, \ldots, p$ do
        $o[i] = ((p+1-i)c[i] + (i+1)c[i+1])/(p+1)$
    end for
    return $\vec{o}$
end function
```

In two dimensions, the Bernstein polynomials also satisfy a degree raising operation. Let $e_k \in \mathbb{R}^3$ be one at the kth index, and zero elsewhere. We have that

$$B_\alpha^p = (\lambda_1 + \lambda_2 + \lambda_3) B_\alpha^p = \frac{\alpha_1 + 1}{p+1} B_{\alpha+e_1}^{p+1} + \frac{\alpha_2 + 1}{p+1} B_{\alpha+e_2}^{p+1} + \frac{\alpha_3 + 1}{p+1} B_{\alpha+e_3}^{p+1}.$$

If we store the control points $\{c_\alpha^p\}$ in a 2D array, then the following algorithm performs degree raising in $\mathcal{O}(p^2)$:

Algorithm 16 2D Degree Raising Operator

```
Require: $\mathbf{c}$ array of the B-net of the polynomial of degree $p$
function DegreeRaise2D($\mathbf{c}$)
    $\mathbf{o} = \text{zeros}((p+2, p+2))$                ▷ Coefficients for degree $p+1$
    for $i = 0, \ldots, p$ do
        for $j = 0, \ldots, p-i$ do
            $k = p - i - j$
            $\mathbf{o}[i,j] += (k+1)/(p+1) * \mathbf{c}[i,j]$
            $\mathbf{o}[i+1,j] += (i+1)/(p+1) * \mathbf{c}[i,j]$
            $\mathbf{o}[i,j+1] += (j+1)/(p+1) * \mathbf{c}[i,j]$
        end for
    end for
    return $\vec{o}$
end function
```

There are many more mathematical and computational properties of Bernstein polynomials which we do not not need; a general reference can be found in [29].

6.7.4 Proofs for Section 6.4.3

In this subsection, we prove the lemmas used in Section 6.4.3. We first prove Lemma 6.4.2.

Proof of Lemma 6.4.2. We begin observing that Equation (6.12) gives

$$\begin{aligned} P_{m-r}^{2r+5,2}(t) &= \sum_{\alpha_3=0}^{m-r} (-1)^{m-r-\alpha_3} \frac{\binom{m+r+5}{\alpha_3}\binom{m-r+2}{m-r-\alpha_3}}{\binom{m-r}{\alpha_3}} B_{\alpha_3}^{m-r}(t) \\ &= \sum_{\alpha_3=0}^{m-r} \nu_{\alpha_3}^{mr} \binom{m}{\alpha_3} (\lambda_1+\lambda_2)^{m-r-\alpha_3} \lambda_3^{\alpha_3}, \end{aligned}$$

and

$$\begin{aligned} P_r^{(2,2)}(s)\left(\frac{1-t}{2}\right)^r &= \sum_{\alpha_2=0}^{r} (-1)^{r-\alpha_2} \frac{\binom{r+2}{\alpha_2}\binom{r+2}{r-\alpha_2}}{\binom{r}{\alpha_2}} B_{\alpha_2}^{r}(s)\left(\frac{1-t}{2}\right)^r \\ &= \sum_{\alpha_2=0}^{r} \gamma_{\alpha_2}^{r} \binom{r}{\alpha_2} \lambda_1^{r-\alpha_2}\lambda_2^{\alpha_2}. \end{aligned}$$

Using the binomial formula and with the convention $\gamma_i^r = 0$ for $i < 0$ and $i > r$, we can write

$$\begin{aligned} &(\lambda_1+\lambda_2)^{m-r-\alpha_3} P_r^{(2,2)}(s)\left(\frac{1-t}{2}\right)^r \\ &\qquad = \sum_{l=0}^{m-r-\alpha_3} \binom{m-r-\alpha_3}{l} \lambda_1^{m-r-\alpha_3-l}\lambda_2^{l} \sum_{\alpha_2=0}^{r} \gamma_{\alpha_2}^{r}\binom{r}{\alpha_2}\lambda_1^{r-\alpha_2}\lambda_2^{\alpha_2} \\ &\qquad = \sum_{l=0}^{m-r-\alpha_3} \binom{m-r-\alpha_3}{l} \sum_{\alpha_2=l}^{r+l} \gamma_{\alpha_2-l}^{r}\binom{r}{\alpha_2-l}\lambda_1^{m-\alpha_3-\alpha_2}\lambda_2^{\alpha_2} \\ &\qquad = \sum_{\alpha_2=0}^{m-\alpha_3} \left(\sum_{l=0}^{m-r-\alpha_3} \gamma_{\alpha_2-l}^{r} \frac{\binom{m-r-\alpha_3}{l}\binom{r}{\alpha_2-l}}{\binom{m-\alpha_3}{\alpha_2}} \right) \binom{m-\alpha_3}{\alpha_2} \lambda_1^{m-\alpha_3-\alpha_2}\lambda_2^{\alpha_2}. \end{aligned}$$

Therefore,

$$\begin{aligned}
&P_r^{(2,2)}(s)\left(\frac{1-t}{2}\right)^r P_{m-r}^{2r+5,2}(t)\\
&\quad=\sum_{\alpha_3=0}^{m-r}\nu_{\alpha_3}^{mr}\sum_{\alpha_2=0}^{m-\alpha_3}\gamma_{\alpha_2}^{r,m-\alpha_3}\binom{m-\alpha_3}{\alpha_2}\binom{m}{\alpha_3}\lambda_1^{m-\alpha_3-\alpha_2}\lambda_2^{\alpha_2}\lambda_3^{\alpha_3}\\
&\quad=\sum_{\alpha_3=0}^{m-r}\nu_{\alpha_3}^{mr}\sum_{\alpha_2=0}^{m-\alpha_3}\gamma_{\alpha_2}^{r,m-\alpha_3}B_\alpha^m(x,y),
\end{aligned}$$

which proves the identity. □

6.7.4.1 Proofs for the recursive computation of S^{mr} (and $Q_{\alpha_3}^{mr}$)

We first show an auxiliary result concerning the coefficients $\gamma_{\alpha_2}^{rj}$ from Lemma 6.4.2.

Lemma 6.7.2. *Consider $\gamma_{\alpha_2}^{rj}$, $j=r,\ldots,m$ and $r=0,\ldots,m$ introduced in Lemma 6.4.2. Then,*

$$\gamma_{\alpha_2}^{r,j+1}=\frac{\alpha_2}{j+1}\gamma_{\alpha_2-1}^{r,j}+\frac{j+1-\alpha_2}{j+1}\gamma_{\alpha_2}^{r,j},\quad \textit{for } \alpha_2=0,\ldots,j+1,$$

i.e. $\vec{\gamma}^{r,j+1}=\mathcal{R}(\vec{\gamma}^{r,j})$ for $j=r,\ldots,m-1.$, where $\mathcal{R}$ denotes the degree raising operator in one dimension (Algorithm 14).

Proof. By the properties of the binomial coefficients, we have that

$$\begin{aligned}
\frac{\alpha_2}{j+1}\gamma_{\alpha_2-1}^{r,j}+\frac{j+1-\alpha_2}{j+1}\gamma_{\alpha_2}^{r,j}&=\sum_{l=0}^{j-r}\gamma_{\alpha_2-1-l}^{r}\frac{\binom{j-r}{l-1}\binom{r}{\alpha_2-1-l}}{\binom{j+1}{\alpha_2}}+\sum_{l=0}^{j-r}\gamma_{\alpha_2-l}^{r}\frac{\binom{j-r}{l}\binom{r}{\alpha_2-l}}{\binom{j+1}{\alpha_2}}\\
&=\sum_{l=1}^{j+1-r}\gamma_{\alpha_2-l}^{r}\frac{\binom{j-r}{l-1}\binom{r}{\alpha_2-l}}{\binom{j+1}{\alpha_2}}+\sum_{l=0}^{j-r}\gamma_{\alpha_2-l}^{r}\frac{\binom{j-r}{l}\binom{r}{\alpha_2-l}}{\binom{j+1}{\alpha_2}}\\
&=\sum_{l=0}^{j+1-r}\gamma_{\alpha_2-l}^{r}\frac{\binom{r}{\alpha_2-l}}{\binom{j+1}{\alpha_2}}\left(\binom{j-r}{l-1}+\binom{j-r}{l}\right)=\gamma_{\alpha_2}^{r,j+1}
\end{aligned}$$

which completes the proof. □

Lemma 6.7.3.

$$Q^{mr}_{\alpha_3} = \frac{\alpha_3+1}{m+1} Q^{m+1r}_{\alpha_3+1} + \frac{m+1-\alpha_3}{m+1} Q^{m+1r}_{\alpha_3},$$

for $\alpha_3 = 0, \ldots, m$ and $r = 0, \ldots, m$.

Proof. We use the two dimensional degree raise operator on the coefficients $\tilde{f}^m_\alpha$ and and Lemma 6.7.2

$$\begin{aligned}
Q^{mr}_{\alpha_3} &= \sum_{\alpha_1+\alpha_2=m-\alpha_3} \gamma^{r,m-\alpha_3}_{\alpha_2} \tilde{f}^m_\alpha \\
&= \sum_{\alpha_1+\alpha_2=m-\alpha_3} \gamma^{r,m-\alpha_3}_{\alpha_2} \left(\frac{\alpha_1+1}{m+1} \tilde{f}^{m+1}_{\alpha+e_1} + \frac{\alpha_2+1}{m+1} \tilde{f}^{m+1}_{\alpha+e_2} + \frac{\alpha_3+1}{m+1} \tilde{f}^{m+1}_{\alpha+e_3} \right) \\
&= \frac{m+1-\alpha_3}{m+1} \sum_{\alpha_1+\alpha_2=m+1-\alpha_3} \left(\frac{\alpha_1}{m+1-\alpha_3} \gamma^{r,m-\alpha_3}_{\alpha_2} + \frac{\alpha_2}{m+1-\alpha_3} \gamma^{r,m-\alpha_3}_{\alpha_2-1} \right) \tilde{f}^m_\alpha \\
&\quad + \frac{\alpha_3+1}{m+1} Q^{m+1\,r}_{\alpha_3+1} \\
&= \frac{m+1-\alpha_3}{m+1} \sum_{\alpha_1+\alpha_2=m+1-\alpha_3} \gamma^{r,m+1-\alpha_3}_{\alpha_2} \tilde{f}^{m+1}_\alpha + \frac{\alpha_3+1}{m+1} Q^{m+1\,r}_{\alpha_3+1} \\
&= \frac{m+1-\alpha_3}{m+1} Q^{m+1\,r}_{\alpha_3} + \frac{\alpha_3+1}{m+1} Q^{m+1\,r}_{\alpha_3+1}.
\end{aligned}$$

□

6.7.4.2 Proofs for the recursive computation of T^m_β

We first need to prove a fact for the coefficients a^{mr}_α.

Lemma 6.7.4. *Consider the coefficients a^{mr}_α, $r = 0, \ldots, m$. Then, the following* *identity holds*

$$(\alpha_1+3)a^{mr}_{\alpha+e_1}+(\alpha_2+3)a^{mr}_{\alpha+e_2}+(\alpha_3+3)a^{mr}_{\alpha+e_3}=0,$$

for $|\alpha|=m-1$.

Proof. We observe that the statement is indeed equivalent to

$$\begin{aligned}(m-\alpha_3-\alpha_2+2)\gamma^{r,m-\alpha_3}_{\alpha_2}+(\alpha_2+3)\gamma^{r,m-\alpha_3}_{\alpha_2+1} &= -(\alpha_3+3)\frac{\nu^{mr}_{\alpha_3+1}}{\nu^{mr}_{\alpha_3}}\gamma^{r,m-\alpha_3-1}_{\alpha_2}\\ &= \frac{(m-\alpha_3+r+5)(m-\alpha_3-r)}{m-\alpha_3}\gamma^{r,m-\alpha_3-1}_{\alpha_2},\end{aligned}$$

which we now prove.

For any $0\le\alpha_2\le m-1$, we proceed by induction on $\alpha_3=0,\ldots,m-1-r$ (equivalently $m-\alpha_3=r+1,\ldots,m$) as $a^{mr}_{\alpha}=0$ for $\alpha_3>m-1-r$. We first prove the statement for $m-\alpha_3=r+1$, i.e., we prove the identity

$$(r+3-\alpha_2)\gamma^{r,r+1}_{\alpha_2}+(\alpha_2+3)\gamma^{r,r+1}_{\alpha_2+1}=\frac{2(r+3)}{r+1}\gamma^{r,r}_{\alpha_2}.$$

We first note that $\gamma^{r,r}_{\alpha_2}=\gamma^{r}_{\alpha_2}$. We note that

$$\gamma^{r}_{\alpha_2-1}=-\frac{(\alpha_2+2)}{(r+3-\alpha_2)}\gamma^{r}_{\alpha_2},\quad\text{for }\alpha_2=1,\ldots,r+1,$$

and by Lemma 6.7.2

$$\gamma^{r,r+1}_{\alpha_2}=\frac{\alpha_2}{r+1}\gamma^{r}_{\alpha_2-1}+\frac{r+1-\alpha_2}{r+1}\gamma^{r}_{\alpha_2},\quad\text{for }\alpha_2=0,\ldots,r+1.$$

Thus, it follows

$$
\begin{aligned}
&(r+3-\alpha_2)\gamma^{r,r+1}_{\alpha_2} + (\alpha_2+3)\gamma^{r,r+1}_{\alpha_2+1} \\
&\qquad = (r+3-\alpha_2)\left(\frac{\alpha_2}{r+1}\gamma^{r}_{\alpha_2-1} + \frac{r+1-\alpha_2}{r+1}\gamma^{r}_{\alpha_2}\right) + (\alpha_2+3)\left(\frac{\alpha_2+1}{r+1}\gamma^{r}_{\alpha_2} + \frac{r-\alpha_2}{r+1}\gamma^{r}_{\alpha_2+1}\right) \\
&\qquad = \frac{2(r+3)}{r+1}\gamma^{r}_{\alpha_2}.
\end{aligned}
$$

We now assume the statement is true for $n > r$, i.e.,

$$(n+2-\alpha_2)\gamma^{r,n}_{\alpha_2} + (\alpha_2+3)\gamma^{r,n}_{\alpha_2+1} = \frac{(n+r+5)(n-r)}{n}\gamma^{r,n-1}_{\alpha_2},$$

and we prove it for $n+1$, i.e., we prove the identity

$$(n+3-\alpha_2)\gamma^{r,n+1}_{\alpha_2} + (\alpha_2+3)\gamma^{r,n+1}_{\alpha_2+1} = \frac{(n+r+6)(n+1-r)}{n+1}\gamma^{r,n}_{\alpha_2}.$$

Arrangement of the inductive hypothesis gives

$$
\begin{aligned}
\gamma^{r,n}_{\alpha_2-1} &= \frac{1}{(n+3-\alpha_2)}\left(\frac{(n+r+5)(n-r)}{n}\gamma^{r,n-1}_{\alpha_2-1} - (\alpha_2+2)\gamma^{r,n}_{\alpha_2}\right), \\
\gamma^{r,n}_{\alpha_2} &= \frac{1}{(\alpha_2+3)}\left(\frac{(n+r+5)(n-r)}{n}\gamma^{r,n-1}_{\alpha_2} - (n+2-\alpha_2)\gamma^{r,n}_{\alpha_2}\right).
\end{aligned}
$$

Thus using Lemma 6.7.2 repeatedly, we have

$$\begin{aligned}
&(n+3-\alpha_2)\gamma^{r,n+1}_{\alpha_2}+(\alpha_2+3)\gamma^{r,n+1}_{\alpha_2+1}\\
=&\frac{1}{n+1}\left((n+3-\alpha_2)\left(\alpha_2\gamma^{r,n}_{\alpha_2-1}+(n+1-\alpha_2)\gamma^{r,n}_{\alpha_2}\right)+(\alpha_2+3)\left((\alpha_2+1)\gamma^{r,n}_{\alpha_2}+(n-\alpha_2)\gamma^{r,n}_{\alpha_2+1}\right)\right)\\
=&\frac{1}{n+1}\left(\alpha_2\Big(\frac{(n+r+5)(n-r)}{n}\gamma^{r,n-1}_{\alpha_2-1}-(\alpha_2+2)\gamma^{r,n}_{\alpha_2}\Big)+(n+3-\alpha_2)(n+1-\alpha_2)\gamma^{r,n}_{\alpha_2}\right.\\
&\left.+(\alpha_2+3)(\alpha_2+1)\gamma^{r,n}_{\alpha_2}+(n-\alpha_2)\Big(\frac{(n+r+5)(n-r)}{n}\gamma^{r,n-1}_{\alpha_2}-(n+2-\alpha_2)\gamma^{r,n}_{\alpha_2}\Big)\right)\\
=&\frac{1}{n+1}\left((n+r+5)(n-r)\Big(\frac{\alpha_2}{n}\gamma^{r,n-1}_{\alpha_2-1}+\frac{n-\alpha_2}{n}\gamma^{r,n-1}_{\alpha_2}\Big)+\gamma^{r,n}_{\alpha_2}(2n+6)\right)\\
=&\frac{1}{n+1}\left((n+r+5)(n-r)\gamma^{r,n}_{\alpha_2}+\gamma^{r,n}_{\alpha_2}(2n+6)\right)\\
=&\frac{(n+r+6)(n+1-r)}{n+1}\gamma^{r,n}_{\alpha_2},
\end{aligned}$$

which completes the proof. □

Lemma 6.7.5. *The following identity hold for* $|\beta|=m-1$

$$T^m_{\beta+e_3}=-\left(\frac{\beta_1+3}{\beta_3+3}T^m_{\beta+e_1}+\frac{\beta_2+3}{\beta_3+3}T^m_{\beta+e_2}\right).$$

Proof. By definition of T^m_β for $|\beta|=m-1$ and applying Lemma 6.7.4, it follows

$$\begin{aligned}
T^m_{\beta+e_3}&=\sum_{r=0}^m S^{mr}a^{mr}_{\beta+e_3}\\
&=-\sum_{r=0}^m S^{mr}\left(\frac{\beta_1+3}{\beta_3+3}a^{mr}_{\beta+e_1}+\frac{\beta_2+3}{\beta_3+3}a^{mr}_{\beta+e_2}\right)\\
&=-\left(\frac{\beta_1+3}{\beta_3+3}\sum_{r=0}^m S^{mr}a^{mr}_{\beta+e_1}+\frac{\beta_2+3}{\beta_3+3}\sum_{r=0}^m S^{mr}a^{mr}_{\beta+e_2}\right).
\end{aligned}$$

□

Bibliography

[1] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, vol. 55 of National Bureau of Standards Applied Mathematics Series, For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C., 1964.

[2] M. Ainsworth, *A hierarchical domain decomposition preconditioner for h-p finite element approximation on locally refined meshes*, SIAM J. Sci. Comput., 17 (1996), pp. 1395–1413.

[3] M. Ainsworth, G. Andriamaro, and O. Davydov, *Bernstein-Bézier finite elements of arbitrary order and optimal assembly procedures*, SIAM J. Sci. Comput., 33 (2011), pp. 3087–3109.

[4] ——, *A Bernstein-Bézier basis for arbitrary order Raviart-Thomas finite elements*, Constr. Approx., 41 (2015), pp. 1–22.

[5] M. Ainsworth and J. Coyle, *Conditioning of hierarchic p-version Nédélec elements on meshes of curvilinear quadrilaterals and hexahedra*, SIAM J. Numer. Anal., 41 (2003), pp. 731–750.

[6] ——, *Hierarchic finite element bases on unstructured tetrahedral meshes*, International journal for numerical methods in engineering, 58 (2003), pp. 2103–2130.

[7] M. Ainsworth and B. Guo, *An additive Schwarz preconditioner for p-version boundary element approximation of the hypersingular operator in three dimensions*, Numer. Math., 85 (2000), pp. 343–366.

[8] M. Ainsworth and S. Jiang, *Preconditioning the mass matrix for high order finite element approximation on triangles*, SIAM J. Numer. Anal., 57 (2019), pp. 355–377.

[9] M. Ainsworth, S. Jiang, and M. A. Sanchéz, *An $\mathcal{O}(p^3)$ hp-version fem in two dimensions: Preconditioning and post-processing*, Computer Methods in Applied Mechanics and Engineering, 350 (2019), pp. 766–802.

[10] M. Ainsworth and M. A. Sanchéz. Personal Communication.

[11] T. Apel, *Anisotropic finite elements: local estimates and applications*, Advances in Numerical Mathematics, B. G. Teubner, Stuttgart, 1999.

[12] I. Babuška, A. Craig, J. Mandel, and J. Pitkäranta, *Efficient preconditioning for the p-version finite element method in two dimensions*, SIAM Journal on Numerical Analysis, 28 (1991), pp. 624–661.

[13] I. Babuška and A. Miller, *The post-processing approach in the finite element methodâĂŤ Part 1: calculation of displacements, stresses and other higher derivatives of the displacements*, International Journal for numerical methods in engineering, 20 (1984), pp. 1085–1109.

[14] A. Barone, F. Esposito, C. Magee, and A. Scott, *Theory and applications of the sine-Gordon equation*, La Rivista del Nuovo Cimento (1971-1977), 1 (1971), pp. 227–267.

[15] J. H. Bramble, J. E. Pasciak, and A. H. Schatz, *The construction of preconditioners for elliptic problems by substructuring. I*, Math. Comp., 47 (1986), pp. 103–134.

[16] A. G. Bratsos, *The solution of the two-dimensional sine-Gordon equation using the method of lines*, J. Comput. Appl. Math., 206 (2007), pp. 251–277.

[17] S. Brenner and R. Scott, *The mathematical theory of finite element methods*, vol. 15, Springer Science & Business Media, 2007.

[18] M. A. Casarin, *Quasi-optimal Schwarz methods for the conforming spectral element discretization*, SIAM J. Numer. Anal., 34 (1997), pp. 2482–2502.

[19] T. F. Chan and T. P. Mathew, *Domain decomposition algorithms*, in Acta numerica, 1994, Acta Numer., Cambridge Univ. Press, Cambridge, 1994, pp. 61–143.

[20] E. Cohen and L. L. Schumaker, *Rates of convergence of control polygons*, Comput. Aided Geom. Design, 2 (1985), pp. 229–235. Surfaces in CAGD '84 (Oberwolfach, 1984).

[21] W. Dahmen, *Subdivision algorithms converge quadratically*, J. Comput. Appl. Math., 16 (1986), pp. 145–158.

[22] L. Demkowicz, *Computing with hp-adaptive finite elements. Vol. 1*, Chapman & Hall/CRC Applied Mathematics and Nonlinear Science Series, Chapman & Hall/CRC, Boca Raton, FL, 2007. One and two dimensional elliptic and Maxwell problems, With 1 CD-ROM (UNIX).

[23] Z. Ditzian, *Multivariate Bernstein and Markov inequalities*, J. Approx. Theory, 70 (1992), pp. 273–283.

[24] *NIST Digital Library of Mathematical Functions*. http://dlmf.nist.gov/, Release 1.0.25 of 2019-12-15. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds.

[25] R. H. Dodds Jr and L. Lopez, *Substructuring in linear and nonlinear analysis*, International Journal for Numerical Methods in Engineering, 15 (1980), pp. 583–597.

[26] P. G. Drazin and R. S. Johnson, *Solitons: an introduction*, Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, 1989.

[27] M. Dryja, B. F. Smith, and O. B. Widlund, *Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions*, SIAM journal on numerical analysis, 31 (1994), pp. 1662–1694.

[28] M. Dubiner, *Spectral methods on triangles and other domains*, J. Sci. Comput., 6 (1991), pp. 345–390.

[29] G. E. Farin, *Curves and Surfaces for CAGD: A Practical Guide.*, vol. 5th ed of The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling, Morgan Kaufmann, 2002.

[30] R. T. Farouki, *The Bernstein polynomial basis: a centennial retrospective*, Comput. Aided Geom. Design, 29 (2012), pp. 379–419.

[31] R. T. Farouki and V. T. Rajan, *Algorithms for polynomials in Bernstein form*, Comput. Aided Geom. Design, 5 (1988), pp. 1–26.

[32] M. Feischl and C. Schwab, *Exponential convergence in H^1 of hp-FEM for Gevrey regularity with isotropic singularities*, Numer. Math., 144 (2020), pp. 323–346.

[33] J. Gallier, *Curves and Surfaces in Geometric Modeling: Theory and Algorithms*, Morgan Kaufmann, 2000.

[34] R. Goldman, *Pyramid algorithms: A dynamic programming approach to curves and surfaces for geometric modeling*, Elsevier, 2002.

[35] G. H. Golub and C. F. Van Loan, *Matrix computations*, vol. 3, JHU Press, 2012.

[36] P. Gray and S. Scott, *Autocatalytic reactions in the isothermal, continuous stirred tank reactor: Oscillations and instabilities in the system a + 2b - 3b; b - c*, Chemical Engineering Science, 39 (1984), pp. 1087 − 1097.

[37] P. Gray and S. Scott, *Autocatalytic reactions in the isothermal, continuous stirred tank reactor: Oscillations and instabilities in the system a+ 2bâĘŠ 3b; bâĘŠ c*, Chemical Engineering Science, 39 (1984), pp. 1087–1097.

[38] A. E. Green and W. Zerna, *Theoretical elasticity*, Dover Publications, Inc., New York, second ed., 1992.

[39] B. Guo and W. Cao, *An additive Schwarz method for the h-p version of the finite element method in three dimensions*, SIAM J. Numer. Anal., 35 (1998), pp. 632–654.

[40] E. Hairer, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer, 2011.

[41] Y. Haralambous, *Parametrization of postscript fonts through metafont: an alternative to adobe multiple master fonts*, Electronic Publishing, 6 (1993), pp. 145–157.

[42] N. Heuer and F. Leydecker, *An extension theorem for polynomials on triangles*, Calcolo, 45 (2008), pp. 69–85.

[43] E. Hille, G. Szegö, J. Tamarkin, et al., *On some generalizations of a theorem of a. markoff*, Duke Mathematical Journal, 3 (1937), pp. 729–739.

[44] S.-M. Hu, *Conversion between triangular and rectangular Bézier patches*, Comput. Aided Geom. Design, 18 (2001), pp. 667–671. Special issue Pierre Bézier.

[45] M. S. Jensen, *High convergence order finite elements with lumped mass matrix*, International Journal for Numerical Methods in Engineering, 39 (1996), pp. 1879–1888.

[46] S. Jund and S. Salmon, *Arbitrary high-order finite element schemes and high-order mass lumping*, Int. J. Appl. Math. Comput. Sci., 17 (2007), pp. 375–393.

[47] G. E. Karniadakis and S. J. Sherwin, *Spectral/hp element methods for computational fluid dynamics*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, second ed., 2005.

[48] B. N. Khoromskij and G. Wittum, *Numerical solution of elliptic differential equations by reduction to the interface*, vol. 36 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin, 2004.

[49] R. C. Kirby, *Fast inversion of the simplicial Bernstein mass matrix*, Numer. Math., 135 (2017), pp. 73–95.

[50] A. Kufner, L. Maligranda, and L.-E. Persson, *The Hardy inequality*, Vydavatelský Servis, Plzeň, 2007. About its history and some related results.

[51] T. Lyche and K. Scherer, *On the p-norm condition number of the multivariate triangular Bernstein basis*, J. Comput. Appl. Math., 119 (2000), pp. 259–273. Dedicated to Professor Larry L. Schumaker on the occasion of his 60th birthday.

[52] J.-F. Maitre and O. Pourquier, *Conditionnements et préconditionnements diagonaux pour la p-version des méthodes d'éléments finis pour des problèmes elliptiques du second ordre*, C. R. Acad. Sci. Paris Sér. I Math., 318 (1994), pp. 583–586.

[53] J. Mandel and G. S. Lett, *Domain decomposition preconditioning for p-version finite elements with high aspect ratios*, Appl. Numer. Math., 8 (1991), pp. 411–425.

[54] J. M. Melenk, *hp-finite element methods for singular perturbations*, vol. 1796 of Lecture Notes in Mathematics, Springer-Verlag, Berlin, 2002.

[55] R. Muñoz Sola, *Polynomial liftings on a tetrahedron and applications to the h-p version of the finite element method in three dimensions*, SIAM J. Numer. Anal., 34 (1997), pp. 282–314.

[56] E. T. Olsen and J. Douglas, Jr., *Bounds on spectral condition numbers of matrices arising in the p-version of the finite element method*, Numer. Math., 69 (1995), pp. 333–352.

[57] S. A. Orszag, *Spectral methods for problems in complex geometries*, J. Comput. Phys., 37 (1980), pp. 70–92.

[58] R. B. W. T. Ozisik, Sevtap, *On the constants in inverse inequalities in L^2*, Technical Report CAAM TR10-19, Rice University, (2010).

[59] A. T. Patera, *A spectral element method for fluid dynamics: laminar flow in a channel expansion*, Journal of computational Physics, 54 (1984), pp. 468–488.

[60] L. F. Pavarino, *Additive Schwarz methods for the p-version finite element method*, Numer. Math., 66 (1994), pp. 493–515.

[61] J. E. Pearson, *Complex patterns in a simple system*, Science, 261 (1993), pp. 189–192.

[62] H. Prautzsch, W. Boehm, and M. Paluszny, *Bézier and B-spline techniques*, Mathematics and Visualization, Springer-Verlag, Berlin, 2002.

[63] A. Quarteroni, C. Canuto, M. Hussaini, and T. Zang, *Spectral methods: Fundamentals in single domains*, Springer Verlag, 4 (2006), p. 16.

[64] J.-F. Remacle, N. Chevaugeon, E. Marchandise, and C. Geuzaine, *Efficient visualization of high-order finite elements*, Internat. J. Numer. Methods Engrg., 69 (2007), pp. 750–771.

[65] M. Rossow and I. Katz, *Hierarchal finite elements and precomputed arrays*, International Journal for Numerical Methods in Engineering, 12 (1978), pp. 977–999.

[66] S. J. Ruuth, *Implicit-explicit methods for reaction-diffusion problems in pattern formation*, J. Math. Biol., 34 (1995), pp. 148–176.

[67] C. Schwab, *p- and hp-finite element methods*, Numerical Mathematics and Scientific Computation, The Clarendon Press, Oxford University Press, New York, 1998. Theory and applications in solid and fluid mechanics.

[68] C. Schwab and M. Suri, *The p and hp versions of the finite element method for problems with boundary layers*, Math. Comp., 65 (1996), pp. 1403–1429.

[69] S. J. Sherwin and G. E. Karniadakis, *A new triangular and tetrahedral basis for high-order (hp) finite element methods*, Internat. J. Numer. Methods Engrg., 38 (1995), pp. 3775–3802.

[70] D. Shreiner, *OpenGL reference manual: The official reference document to OpenGL, version 1.2*, Addison-Wesley Longman Publishing Co., Inc., 1999.

[71] B. Smith, P. Bjorstad, and W. Gropp, *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*, Cambridge university press, 2004.

[72] J. M. Smith, *Efficient domain decomposition preconditioning for the p-version finite element method: the mass matrix.* `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.44.1649`.

[73] B. Szabó and I. Babuška, *Finite element analysis*, A Wiley-Interscience Publication, John Wiley & Sons, Inc., New York, 1991.

[74] A. Toselli and X. Vasseur, *A numerical study on Neumann-Neumann and FETI methods for hp approximations on geometrically refined boundary layer meshes in two dimensions*, Comput. Methods Appl. Mech. Engrg., 192 (2003), pp. 4551–4579.

[75] ——, *Domain decomposition preconditioners of Neumann-Neumann type for hp-approximations on boundary layer meshes in three dimensions*, IMA J. Numer. Anal., 24 (2004), pp. 123–156.

[76] ——, *A numerical study on Neumann-Neumann methods for hp approximations on geometrically refined boundary layer meshes. II. Three-dimensional problems*, M2AN Math. Model. Numer. Anal., 40 (2006), pp. 99–122.

[77] A. Toselli and O. Widlund, *Domain decomposition methods—algorithms and theory*, vol. 34 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2005.

[78] M. Wadati, H. Sanuki, and K. Konno, *Relationships among inverse method, Bäcklund transformation and an infinite number of conservation laws*, Progr. Theoret. Phys., 53 (1975), pp. 419–436.

[79] T. Warburton and J. S. Hesthaven, *On the constants in hp-finite element trace inverse inequalities*, Comput. Methods Appl. Mech. Engrg., 192 (2003), pp. 2765–2773.

[80] E. L. Wilson, *The static condensation algorithm*, International Journal for Numerical Methods in Engineering, 8 (1974), pp. 198–203.

[81] L. Yan, X. Han, and J. Liang, *Conversion between triangular Bézier patches and rectangular Bézier patches*, Applied Mathematics and Computation, 232 (2014), pp. 469–478.