SAND2004-3742
Printed July 2004

Exponential Integrators for the Incompressible Navier–Stokes Equations

Christopher K. Newman

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy
in
Mathematics

Christopher Beattie, Chair
Richard B. Lehoucq
Jeffrey T. Borggaard
Slimane Adjerid
Tao Lin

Date: September 17, 2003
Blacksburg, Virginia

Keywords: Incompressible Navier–Stokes, projection method,
finite elements, Krylov subspace, matrix exponential

Exponential Integrators for the Incompressible Navier–Stokes Equations

Christopher K. Newman

Abstract

We provide an algorithm and analysis of a high order projection scheme for time integration of the incompressible Navier–Stokes equations (NSE). The method is based on a projection onto the subspace of divergence–free (incompressible) functions interleaved with a Krylov–based exponential time integration (KBEI). These time integration methods provide a high order accurate, stable approach with many of the advantages of explicit methods, and can reduce the computational resources over conventional methods. The method is scalable in the sense that the computational costs grow linearly with problem size.

Exponential integrators, used typically to solve systems of ODEs, utilize matrix vector products of the exponential of the Jacobian on a vector. For large systems, this product can be approximated efficiently by Krylov subspace methods. However, in contrast to explicit methods, KBEIs are not restricted by the time step. While implicit methods require a solution of a linear system with the Jacobian, KBEIs only require matrix vector products of the Jacobian. Furthermore, these methods are based on linearization, so there is no non–linear system solve at each time step.

Differential–algebraic equations (DAEs) are ordinary differential equations (ODEs) subject to algebraic constraints. The discretized NSE constitute a system of DAEs, where the incompressibility condition is the algebraic constraint. Exponential integrators can be extended to DAEs with linear constraints imposed via a projection onto the constraint manifold. This results in a projected ODE that is integrated by a KBEI. In this approach, the Krylov subspace satisfies the constraint, hence the solution at the advanced time step automatically satisfies the constraint as well. For the NSE, the pro-

jection onto the constraint is typically achieved by a projection induced by the L^2 inner product. We examine this L^2 projection and an H^1 projection induced by the H^1 semi-inner product. The H^1 projection has an advantage over the L^2 projection in that it retains tangential Dirichlet boundary conditions for the flow. Both the H^1 and L^2 projections are solutions to saddle point problems that are efficiently solved by a preconditioned Uzawa algorithm.

Acknowledgements

I would like to express my sincere gratitude to Dr. Richard Lehoucq, for his support and guidance which made this work possible. I would also like to thank my advisor and committee chair Dr. Christopher Beattie for contributions to my research, constructive comments and new ideas.

Thanks also to the members of my committee: Dr. Jeff Borggard, Dr. Slimane Adjerid and Dr. Tao Lin for their efforts in reviewing my work.

Finally thanks to my wife Chelsea for her capacity for enduring inconvenience without complaint.

I also consider myself very fortunate to have worked with the Computational Mathematics and Algorithms department at Sandia National Laboratories. This research was funded by the ASCI Algorithms project at Sandia National Laboratories[†].

[†]Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-AL85000.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Computer simulations of incompressible flows play an increasingly important role in scientific investigations. As problem sizes increase, computational resources grow. The increase in problem size is often the result of a requirement of greater resolution or accuracy, or a result of increasingly complex geometries. The development of scalable algorithms is crucial to cope with the increase in problem size. The main result of this thesis is the development of a scalable, high order time accurate unsteady solver for the incompressible Navier–Stokes equations based on a marriage of Krylov–based exponential integration and projection methods. Scalability often refers to the ability to use additional computational resources effectively to solve increasingly larger problems. One example of this is through the use of multiple processors. However, we refer to scalability as a description of how the total computational work requirements grow with problem size, which can be discussed independent of the computing platform. We refer to a method as scalable if the computational costs grow linearly with the problem size. The method we describe is constructed of scalable building blocks that are relatively easy to synthesize. Scalability of the method hinges on the following key requirements:

1. Scalable time integration of the momentum equation.

2. Scalable computation of a divergence–free projection.

In the first part of this thesis we address scalable time integration. We provide a detailed introduction to exponential integration methods and Krylov–based exponential integration methods in Chapter 2. Krylov–based exponential integration methods project a problem onto a smaller dimensional subspace where the critical computations are performed, then extends the solution to the larger space. Thus for our method to be scalable, it is crucial that the dimension of the smaller subspace only grow modestly with problem size. We discuss scalable computation of a divergence–free projection in the second part of the thesis. Since a projection onto the space of divergence–free functions is performed for each dimension of the Krylov subspace at each time step, it is essential that we implement this step in an efficient, scalable way.

Finite element discretizations of the incompressible Navier–Stokes equations lead to what are known as differential–algebraic equations rather than ordinary differential equations. Chapters 3 and 4 provide the framework necessary to discuss the incompressible Navier–Stokes equations and a discussion of divergence–free projections. The thesis concludes with results of a Krylov–based exponential integrator on a test problem for the incompressible Navier–Stokes equations (Chapters 6).

One of the advantages of exponential integrators is that the time step restriction due to the von Neumann condition is avoided. The von Neumann condition describes the maximum stable time step size that can be used in an explicit method [51, 68]. PDEs often possess a diffusive term that renders the discretized ODE inherently stiff. The effect of stiffness becomes larger as the mesh size decreases. Thus for stiff systems, the von Neumann condition restricts the step size severely. In these cases, an alternative to explicit methods are implicit methods, which retain stability with larger time steps. Implicit methods typically require the solution of a non–linear system with the Jacobian at each time step. If Krylov iterative methods are employed for the linear solve, this stiffness can cause the convergence of the iterative method to be slow. Unless a suitable preconditioner or multigrid methods are available, a large number of iterations can be required. Exponential integration methods provide an alternative to explicit methods in the sense that they allow a stable time integration with a time step that is not restricted by the von Neumann condition.

Exponential integration methods rely on computation of the matrix exponential operator of the Jacobian. For small systems of ODEs this method has been used with success [41, 42, 79]. For large systems of ODEs, computing

the matrix exponential of a large, non-symmetric Jacobian matrix is infeasible. We refer to [57] for a comprehensive review of methods for computing the matrix exponential operator.

The use of Krylov subspace projection methods in the 1980's provided a solution to this problem for large matrices. The Krylov subspace projection method is based on projecting a matrix onto a smaller dimension Krylov subspace. The exponential function is evaluated with the smaller projected matrix as the argument, then the result is lifted back to the original space. Thus the exponentiation of a large matrix is avoided however, exponentiation of a small matrix is still required. An overview of Krylov subspace projection methods can be found in [70, Chap. 6, 7].

The first use of Krylov subspace methods in ODEs [13, 29] utilized Krylov–based iterative methods to solve systems arising from Newton's method in implicit multistep methods. A high order efficient method is presented in Hochbruck et al. [42]. This method is based on a Runge–Kutta framework and provides an error analysis based on the theory of Krylov subspace approximations to functions of matrices found in earlier work [41]. The methods of [42] were applied to partial differential equations that arise in magnetohydrodynamic modeling of solar magnetic arcades by Tokman [79].

We provide a derivation and discuss in detail how Krylov–based exponential integration methods are constructed in Chapter 2. We discuss the theory of [41] where error estimates and numerical analysis for approximating the matrix exponential operator of a large matrix using Krylov subspace projection techniques are developed. We also discuss why Krylov subspace approximations to the matrix exponential operator can converge faster than Krylov based iteration methods for solving the linear system that arises in implicit methods. In addition we discuss time integration methods for ordinary differential equations. We discuss implicit and explicit methods for ordinary differential equations that arise from spatial discretization of time dependent partial differential equations. We solve the heat equation using a high order Krylov–based exponential integration method and the Crank Nicolson method. Our results provide an insight into the advantages and disadvantages of both methods.

The second part of this thesis discusses the application of Krylov based exponential integration methods to the differential–algebraic equations that arise from finite element discretizations of the incompressible Navier–Stokes equations. We focus on a projection–based scheme in conjunction with a high order exponential integrator. Projection–based schemes precede a time step

with a projection onto the constraint space. In the case of the incompressible Navier–Stokes equations, the constraint space is the incompressibility or divergence–free subspace. The advantages of projection methods is the decoupling of the velocity field from the pressure field.

Projection–based schemes date from the late 1960's to the fractional step projection method of Chorin [15, 16] and Temam [77]. This method is the most frequently employed technique for the numerical solution of the Navier–Stokes equations. These methods are also commonly referred to as semi–implicit or Chorin methods. The method is based on a time discretization where a projection to the divergence–free subspace is utilized to decouple the velocity and pressure calculations. The original scheme was heuristically motivated in [15, 16] via a Taylor series analysis to provide a first order accurate in time approximation for the velocity. The theoretical convergence properties of the method were studied in [37, 60, 71, 74, 73]. A higher order in time method presented in [38, 60] utilizes improved pressure boundary conditions to achieve higher accuracy. An analysis of several second order projection schemes is given in [72]. The fractional step methods have been speculated to be at most second order accurate (see [35]) although this remains unproven.

In Chapter 3 we present the incompressible Navier–Stokes equations in primitive variable form. In addition, we provide an overview of the two common projections onto the divergence–free subspace. The common fractional step projection methods utilize a projection induced by the L^2 inner product. Another viable but seldom utilized projection is induced by the H^1 semi inner product. An overview of the divergence–free L^2 and H^1 projections is given in Glowinski [32] and Gresho and Sani [35]. We provide derivations and discuss in detail the implementations of each of these projections. We discuss the advantages and disadvantages of each projection.

In Chapter 4 we provide insight on the temporal convergence of fractional step projection methods, and discuss advantages and disadvantages over explicit and implicit methods. We outline in detail the Krylov–based exponential integration method for the incompressible Navier–Stokes equations and contrast the method with the fractional step projection methods. The key differences are that the fractional step methods approximate a divergence–free projection of the momentum equation via an initial approximation for the pressure. This produces an intermediate velocity to which a divergence–free projection must be applied. The Krylov–based exponential integration method avoids the approximation by applying a divergence–free projection directly to the momentum equation, eliminating the intermediate velocity

altogether. The projected momentum equation is integrated in time by a high–order Krylov–based exponential integrator.

In Chapter 5 we discuss implementation issues for a scalable projection–based high–order time integration method for the incompressible Navier–Stokes equations. We discuss in detail the stability in the sense of the Ladyzhenskaya–Babuška–Brezzi condition for mixed finite element realizations of the L^2 projection. We provide experimental results that extend the work of Mardal et al. [54]. The work of [54] demonstrates that that mixed finite elements that are stable for the Stokes problem may not be stable for the L^2 projection.

In Chapter 6 we present results of numerical experiments that address convergence and scalability of the method applied to the Stokes and Navier–Stokes problems. We draw conclusions and discuss the contributions from this work and outline future work in Chapter 7.

1.2 Notation

We use $\boldsymbol{x}$ to denote spatial coordinates in $\mathbb{R}^d$ with $d = 2$ or 3. We denote the components of $\boldsymbol{x}$ by x_i, $1 \leq i \leq d$. When $d = 1$ we simply denote $\boldsymbol{x}$ by x. We use bold font to denote vector functions in $\mathbb{R}^d$ with $d = 2$ or 3 and normal font for functions in $\mathbb{R}$. For example we denote the vector velocity by $\boldsymbol{u} : \mathbb{R}^d \to \mathbb{R}^d$ with components u_i, $1 \leq i \leq d$ or denote the scalar quantity pressure by p. When we make use of matrix theory, we use bold non-italicized font to denote vectors and upper case for matrices. To illustrate this notation, consider the linear system $A\mathbf{x} = \mathbf{b}$. In addition to the bracket notation, we use non-italicized fonts for the components of a vector, and subscripts for elements of a matrix. For example, x_i and $[\mathbf{x}]_i$ denote the ith component of $\mathbf{x}$, while A_{ij} and $[A]_{ij}$ denote the element in row i and column j of the matrix A. We denote by $\{\mathbf{e}_i\}_{i=1}^n$ the canonical basis of $\mathbb{R}^n$. We also make use of upper case unambiguously for linear operators and for vector spaces. We define additional notation when needed.

Chapter 2

Exponential integration methods

Development of our scalable solver relies on two crucial building blocks: an efficient and scalable means for time integration, and an efficient and scalable means to apply a divergence–free projection. This chapter focuses on time integration with the development and implementation of high order accurate Krylov–based exponential time integrators. We construct exponential integrators for general large systems of ordinary differential equations. Next we describe how these integrators can be efficiently computed in a Krylov subspace using the Arnoldi method and an efficient high order approximation to the matrix exponential. Finally we provide a numerical experiment that highlights efficiency and scalability of the method over a traditional implicit method. We implement this method for the incompressible Navier–Stokes equations in Chapter 4.

Krylov methods were used to evaluate the exponential of Hamiltonian operators in [58, 59], and later in [61]. The idea of evaluating arbitrary functions of matrices using Krylov subspace methods was mentioned by van der Vorst in [82]. Saad [69] provides a rigorous overview of Krylov subspace approximations to the matrix exponential operator. Krylov–based exponential integration methods for large systems of ordinary differential equations were presented for the first time by Friesner et al. [27]. This work was extended to partial differential equations in an application to the incompressible Navier–Stokes equations in [25]. Methods for parabolic partial differential equations with nonlinear forcing term were presented by Gallopoulos and Saad [28]. This work included results on accuracy and stability of the method. Related

work includes that of [48, 75, 76] in which systems of ordinary differential equations are solved by projecting onto a Krylov subspace and integrated in time with polynomial methods based on Chebyshev expansion. Recently, Pini and Gambolati [63] introduced a method in which a system of ordinary differential equations is projected onto a Krylov subspace and integrated by the Crank Nicolson method.

2.1 Exponential integrators

Consider the ODE

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}), \tag{2.1}$$

$$\mathbf{y}(t_0) = \mathbf{y}_0. \tag{2.2}$$

Implicit methods for (2.1)–(2.2) usually lead to solving linear systems of the form

$$(I - \tau A)\mathbf{x} = \mathbf{v} \tag{2.3}$$

at each time step, where A is the Jacobian (or an approximation) of $\mathbf{f}(\mathbf{y})$ evaluated at some fixed $\mathbf{y}$ and τ is related to the step size. The system (2.3) arises through the use of Rosenbrock–Wanner methods, which are based on linearization, or as a step in a non–linear solution process. Exponential integration methods approximate the action of an exponential function of the Jacobian on a vector, rather than solving non-linear systems at time steps. Gallopoulos and Saad [28] speculated that Krylov approximations to the product $\exp(\tau A)\mathbf{v}$ converge substantially faster than those for the solution of $(I - \tau A)\mathbf{x} = \mathbf{v}$.

In this section we investigate exponential methods for the integration of (2.1)–(2.2) involving matrix vector products of $\varphi(\tau A)\mathbf{v}$, where $\varphi(z)$ is a function involving the exponential of z. Convergence of approximations to $\varphi(\tau A)\mathbf{v}$ [41] compare to those for $\exp(\tau A)\mathbf{v}$. A well known method is the exponentially fitted Euler method given in [42]. Given $\mathbf{y}_n$ at the current time step, the approximate solution, $\mathbf{y}_{n+1}$, at the next time step is

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \varphi(\Delta t A)\mathbf{f}(\mathbf{y}_n), \tag{2.4}$$

where Δt is the step size, A is the Jacobian of $\mathbf{f}$ evaluated at $\mathbf{y}_n$ and

$$\varphi(z) = \frac{1}{z}\left(\mathrm{e}^z - 1\right). \tag{2.5}$$

The exponentially fitted Euler method is derived from a Taylor series approximation. The first two terms of the Taylor series for $\mathbf{f}(\mathbf{y})$ centered at $\mathbf{y}_0$ gives the approximation

$$\mathbf{f}(\mathbf{y}) \approx \mathbf{f}(\mathbf{y}_0) + A\,(\mathbf{y}(t) - \mathbf{y}_0), \tag{2.6}$$

where A is the Jacobian of $\mathbf{f}(\mathbf{y})$ evaluated at $\mathbf{y}_0$. Let $\mathbf{x}(t) = \mathbf{y}(t) - \mathbf{y}_0$ so that $\dot{\mathbf{x}}(t) = \dot{\mathbf{y}}(t)$ and $\mathbf{x}(0) = 0$. In light of this change of variable and the Taylor series approximation, an approximation for $\dot{\mathbf{y}}$ is given by

$$\dot{\mathbf{y}}(t) \approx \mathbf{f}(\mathbf{y}_0) + A\,\mathbf{x}.$$

Now consider the ODE

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{y}_0) + A\,\mathbf{x}, \qquad \mathbf{x}(0) = 0. \tag{2.7}$$

Equation (2.7) can be solved using the integrating factor $\mathrm{e}^{-A\,t}$, thus

$$\begin{aligned}
\mathrm{e}^{-A\,t}\,(\dot{\mathbf{x}}(t) - A\,\mathbf{x}(t)) &= \mathrm{e}^{-A\,t}\mathbf{f}(\mathbf{y}_0), \\
\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathrm{e}^{-A\,t}\,\mathbf{x}(t)\right) &= \mathrm{e}^{-A\,t}\mathbf{f}(\mathbf{y}_0). \qquad (2.8)
\end{aligned}$$

Integration of (2.8) over a time step Δt yields,

$$\begin{aligned}
\int_0^{\Delta t} \frac{\mathrm{d}}{\mathrm{d}t}\left(\mathrm{e}^{-A\,t}\,\mathbf{x}(t)\right)\,\mathrm{d}t &= \int_0^{\Delta t} \mathrm{e}^{-A\,t}\,\mathbf{f}(\mathbf{y}_0)\,\mathrm{d}t, \\
\mathrm{e}^{-A\,t}\,\mathbf{x}(t)\big|_0^{\Delta t} &= \left(\int_0^{\Delta t} \mathrm{e}^{-A\,t}\,\mathrm{d}t\right)\mathbf{f}(\mathbf{y}_0), \\
\mathrm{e}^{-\Delta t\,A}\mathbf{x}(\Delta t) - \mathbf{x}(0) &= \left(\int_0^{\Delta t} \mathrm{e}^{-A\,t}\,\mathrm{d}t\right)\mathbf{f}(\mathbf{y}_0), \\
\mathbf{x}(\Delta t) &= \mathrm{e}^{\Delta t\,A}\left(\int_0^{\Delta t} \mathrm{e}^{-A\,t}\,\mathrm{d}t\right)\mathbf{f}(\mathbf{y}_0), \\
\mathbf{x}(\Delta t) &= \left(\int_0^{\Delta t} \mathrm{e}^{A\,(\Delta t - t)}\,\mathrm{d}t\right)\mathbf{f}(\mathbf{y}_0).
\end{aligned}$$

By the power series definition of $\mathrm{e}^{(\Delta t - t)\,A}$,

$$\begin{aligned}
\mathbf{x}(\Delta t) &= \left(\int_0^{\Delta t} \sum_{i=0}^{\infty} \frac{A^i(\Delta t - t)^i}{i!}\,\mathrm{d}t\right)\mathbf{f}(\mathbf{y}_0), \\
&= \left(\sum_{i=0}^{\infty} \int_0^{\Delta t} \frac{A^i(\Delta t - t)^i}{i!}\,\mathrm{d}t\right)\mathbf{f}(\mathbf{y}_0),
\end{aligned}$$

where the interchange of integration and summation is justified by uniform convergence of the sum for $t \in [0, \Delta t]$. Hence,

$$\begin{aligned}
\mathbf{x}(\Delta t) &= \left(\sum_{i=0}^{\infty} \frac{-A^i(\Delta t - t)^{(i+1)}}{(i+1)!} \Big|_{t=0}^{\Delta t} \right) \mathbf{f}(\mathbf{y}_0), \\
&= \left(\sum_{i=0}^{\infty} \frac{A^i \Delta t^{(i+1)}}{(i+1)!} \right) \mathbf{f}(\mathbf{y}_0), \\
&= \Delta t \left(\sum_{i=0}^{\infty} \frac{(\Delta t A)^i}{(i+1)!} \right) \mathbf{f}(\mathbf{y}_0), \\
&\equiv \Delta t \varphi(\Delta t A) \mathbf{f}(\mathbf{y}_0). \qquad (2.9)
\end{aligned}$$

Substitution of $\mathbf{x}(\Delta t) = \mathbf{y}(\Delta t) - \mathbf{y}_0$ in (2.9) yields the exponentially fitted Euler step (2.4). The exponentially fitted Euler method is locally second order and is exact when $\mathbf{f}(\mathbf{y})$ is linear. This method is usually implemented when $\mathbf{f}(\mathbf{y})$ is nonlinear.

When $\mathbf{f}(\mathbf{y})$ is linear and homogeneous (2.1) becomes $\dot{\mathbf{y}} = A\mathbf{y}$. In this case the exponentially fitted Euler method (2.4) can then be written as, given $\mathbf{y}_n$ at the current time step, the approximate solution, $\mathbf{y}_{n+1}$, at the next time step is

$$\mathbf{y}_{n+1} = \exp(\Delta t A)\mathbf{y}_n, \qquad (2.10)$$

where Δt is the step size.

More general methods are multistage methods given by

$$\mathbf{k}_i = \varphi(\gamma \Delta t A) \left(\mathbf{f}(\mathbf{u}_i) + \Delta t A \sum_{j=1}^{i-1} \gamma_{ij} \mathbf{k}_j \right), \quad i = 1, \ldots, s, \qquad (2.11)$$

$$\mathbf{u}_i = \mathbf{y}_n + \Delta t \sum_{j=1}^{i-1} \alpha_{ij} \mathbf{k}_j, \qquad (2.12)$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \sum_{i=1}^{s} b_i \mathbf{k}_i. \qquad (2.13)$$

Here A is the Jacobian of $\mathbf{f}(\mathbf{y})$ evaluated at $\mathbf{y} = \mathbf{y}_n$, and γ, γ_{ij}, α_{ij}, b_i, with $\gamma_{ij} = \alpha_{ij} = 0$ for $i \leq j$, are coefficients that determine the method. This method is locally fourth order accurate. Note that for $\varphi(z) \equiv 1$ and $\gamma_{ij} = 0$, this is an explicit Runge-Kutta method, and the exponentially fitted Euler

method for $s = 1$. See [42] for a discussion of these methods. In particular we can write the fourth order method (2.11)–(2.13) as follows [42, p. 1563]:

$$\mathbf{k}_1 = \varphi(\tfrac{1}{3}\Delta t A)\mathbf{f}(\mathbf{y}_0), \tag{2.14}$$
$$\mathbf{k}_2 = \varphi(\tfrac{2}{3}\Delta t A)\mathbf{f}(\mathbf{y}_0), \tag{2.15}$$
$$\mathbf{k}_3 = \varphi(\Delta t A)\mathbf{f}(\mathbf{y}_0), \tag{2.16}$$
$$\mathbf{w}_4 = -\frac{7}{300}\mathbf{k}_1 + \frac{97}{150}\mathbf{k}_2 - \frac{37}{300}\mathbf{k}_3, \tag{2.17}$$
$$\mathbf{u}_4 = \mathbf{y}_0 + \Delta t \mathbf{w}_4, \tag{2.18}$$
$$\mathbf{d}_4 = \mathbf{f}(\mathbf{u}_4) - \mathbf{f}(\mathbf{y}_0) - \Delta t A \mathbf{w}_4, \tag{2.19}$$
$$\mathbf{k}_4 = \varphi(\tfrac{1}{3}\Delta t A)\mathbf{d}_4, \tag{2.20}$$
$$\mathbf{k}_5 = \varphi(\tfrac{2}{3}\Delta t A)\mathbf{d}_4, \tag{2.21}$$
$$\mathbf{k}_6 = \varphi(\Delta t A)\mathbf{d}_4, \tag{2.22}$$
$$\mathbf{w}_7 = \frac{59}{300}\mathbf{k}_1 - \frac{7}{75}\mathbf{k}_2 + \frac{269}{300}\mathbf{k}_3 + \frac{2}{3}\left(\mathbf{k}_4 + \mathbf{k}_5 + \mathbf{k}_6\right), \tag{2.23}$$
$$\mathbf{u}_7 = \mathbf{y}_0 + \Delta t \mathbf{w}_7, \tag{2.24}$$
$$\mathbf{k}_7 = \varphi(\tfrac{1}{3}\Delta t A)\mathbf{d}_7, \tag{2.25}$$
$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \left(\mathbf{k}_3 + \mathbf{k}_4 - \frac{4}{3}\mathbf{k}_5 + \mathbf{k}_6 + \frac{1}{6}\mathbf{k}_7\right). \tag{2.26}$$

We note that we have combined the coefficients and intermediate vectors in (2.11)–(2.13) to write a complete time step of the fourth order method as (2.14)–(2.26).

We remark that (2.1)–(2.2) constitute an *autonomous* system of ODEs. As pointed out in [42] the methods mentioned in this section apply to the *nonautonomous* system

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, t), \tag{2.27}$$
$$\mathbf{y}(t_0) = \mathbf{y}_0, \tag{2.28}$$

because we can write the nonautonomous system (2.27)–(2.28) as the autonomous system

$$\begin{bmatrix}\dot{\mathbf{y}} \\ \dot{t}\end{bmatrix} = \begin{bmatrix}\mathbf{f}(\mathbf{y}, t) \\ 1\end{bmatrix},$$
$$\mathbf{y}(t_0) = \mathbf{y}_0.$$

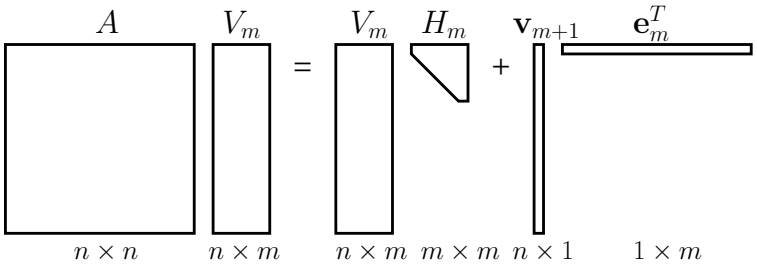

Figure 2.1: Krylov approximation of the matrix A.

2.2 Krylov approximation to the matrix exponential operator

2.2.1 Krylov subspace

For a matrix A of dimension n and a unit vector $\mathbf{v} \in \mathbb{C}^n$, the Arnoldi process generates an orthonormal basis $V_m = [\mathbf{v}_1, \ldots, \mathbf{v}_m]$ for the *Krylov subspace* K_m,

$$K_m = \text{span}\{\mathbf{v}, A\mathbf{v}, \ldots, A^{m-1}\mathbf{v}\}, \tag{2.29}$$

and an upper Hessenberg matrix H_m of dimension m such that

$$AV_m = V_m H_m + h_{m+1,m}\mathbf{v}_{m+1}\mathbf{e}_m^T,$$

where $\mathbf{e}_i$ is the ith standard basis vector for $\mathbb{R}^m$. Gallopoulos and Saad [28] show that for analytic functions f,

$$f(A)\mathbf{v} \approx V_m f(H_m)\,\mathbf{e}_1. \tag{2.30}$$

For $m \ll n$, $V_m f(H_m)\,\mathbf{e}_1$ is usually much easier to compute than $f(A)\,\mathbf{v}$. We can use (2.30) to approximate the product

$$\varphi(\gamma\Delta t A)\,\mathbf{v} \approx V_m \varphi(\gamma\Delta t H_m)\,\mathbf{e}_1. \tag{2.31}$$

2.3 Implementation

In the present section we discuss practical implementation of the Krylov approximation (2.31). In Section 2.3.1 we discuss the use of the Arnoldi algorithm to generate a basis for the Krylov subspace of A. Section 2.3.3 discusses

a high–order accurate method to compute $\exp(\gamma \Delta t H_m)\, \mathbf{e}_1$ and $\varphi(\gamma \Delta t H_m)\, \mathbf{e}_1$ where

$$\varphi(z) = \frac{\mathrm{e}^z - 1}{z}.$$

2.3.1 Arnoldi method

Arnoldi's method is an orthogonal projection of A onto K_m, given by (2.29) when A is non–Hermitian. When A is Hermitian, K_m is typically found via the Lanczos method. In this work we will only encounter non–Hermitian matrices, hence the Lanczos method will not be discussed. The basic algorithm is given in [70, Algorithm 6.1, p. 146].

Algorithm 2.1 (Arnoldi's method).

1: Given $\mathbf{v}_1$ with $\|\mathbf{v}_1\| = 1$
2: **for** $j = 1, 2, \ldots, m$ **do**
3: **for** $i = 1, 2, \ldots j$ **do**
4: $h_{ij} = (A\mathbf{v}_j, \mathbf{v}_i)$
5: **end for**
6: $\mathbf{w}_j = A\mathbf{v}_j - \sum_{i=1}^{j} h_{ij}\mathbf{v}_i$
7: $h_{j+1,j} = \|\mathbf{w}_j\|$
8: **if** $h_{j+1,j} = 0$ **then**
9: **stop**
10: **end if**
11: $\mathbf{v}_{j+1} = \mathbf{w}_j / h_{j+1,j}$
12: **end for**

At each step, the algorithm applies a matrix–vector product with $A\mathbf{v}_j$ and orthonormalizes $\mathbf{w}_j$ against $\mathbf{v}_i$, $i = 1, 2, \ldots, j$ by the Gram–Schmidt process. We denote by V_m the $n \times m$ matrix whose columns are the vectors $\mathbf{v}_1, \ldots \mathbf{v}_m$ found in Algorithm 2.1. We denote by H_m the $m \times m$ upper Hessenberg matrix whose nonzero entries h_{ij} are found in Algorithm 2.1. We state the following results [70, Propositions 6.4–6.5 pp. 146–147].

Theorem 2.1. *If Algorithm 2.1 does not stop prior to step m, then the vectors $\{\mathbf{v}_1, \ldots \mathbf{v}_m\}$ form a basis for K_m.*

Theorem 2.2. *The following relations hold:*

$$AV_m = V_m H_m + \mathbf{w}_m \mathbf{e}_m^T \tag{2.32}$$

and

$$V_m^T A V_m = H_m. \tag{2.33}$$

We often write (2.32) as

$$AV_m = V_m H_m + h_{j+1,m} \mathbf{v}_m \mathbf{e}_m^T.$$

We note that Algorithm 2.1 will break down at step $j < m$ if $h_{j+1,j} = 0$. In this case we have the following result [70, Proposition 6.6 p.147].

Theorem 2.3. *If Algorithm 2.1 breaks down at step $j < m$, then the subspace K_j is invariant under A.*

Algorithm 2.1 assumes exact arithmetic is used. In most cases round off error and cancellation can be severe in the orthogonalization steps. In particular the classical Gram–Schmidt method has poor numerical properties, and a severe loss of orthogonality is exhibited between each $\mathbf{v}_i$. Significant improvement comes from double orthogonalization [52, p. 50].

Since $\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ forms an orthonormal set, we readily see that $V_m^T V_m = I$. We can define the orthogonal projection operator $P : \mathbb{C}^n \to K_m$ by $P = V_m V_m^T$. Thus we can project the product $A\mathbf{v}$ by

$$A\mathbf{v} \approx PAP\mathbf{v} = V_m V_m^T A V_m V_m^T \mathbf{v} = V_m H_m V_m^T \mathbf{v}.$$

We can write $\mathbf{v} = \|\mathbf{v}\|\mathbf{v}_1$ and $\mathbf{v}_1 = V_m \mathbf{e}_1$,

$$A\mathbf{v} \approx \|\mathbf{v}\| V_m H_m \mathbf{e}_1.$$

2.3.2 Error in the approximation

In this section we provide an analysis for the error in approximation of $\varphi(z)$ by Krylov subspace methods. Error estimates were derived in [28, 69] based on an argument that approximating a function of a matrix in a Krylov subspace is equivalent to interpolating the function by a polynomial. The authors derive error estimates based on the remainder of the interpolating polynomial. The authors mention that the error bounds are not sharp enough to explain the error observed in numerical experiments. More recently a theory [41, 42] has been presented that gives much sharper estimates. This theory provides an insight into the relationship between the error, the spectrum of the matrix

and the function that is to be approximated. In this section we present error estimates based on the approach of [41, 42].

We define the *numerical range*

$$\mathcal{F}(A) = \{\mathbf{x}^* A\mathbf{x} : \mathbf{x} \in \mathbb{C}^n, \|\mathbf{x}\| = 1\}.$$

Using (2.33) we have $\mathcal{F}(H_m) \subset \mathcal{F}(A)$. We make use of the Cauchy's integral formula [2, p. 119]

Theorem 2.4 (Cauchy's integral formula). *Let $f(z)$ be analytic in an open disk Ω. and let Γ be a closed curve in Ω. Then for any point a with $a \notin \Gamma$,*

$$f(a) = \frac{1}{2\pi \mathrm{i}} \int_\Gamma \frac{f(z)}{z-a}\,\mathrm{d}z. \tag{2.34}$$

We can treat a as a variable, and hence we can rewrite (2.34) as

$$f(z) = \frac{1}{2\pi \mathrm{i}} \int_\Gamma \frac{f(\xi)}{\xi - z}\,\mathrm{d}\xi, \tag{2.35}$$

Let f be analytic in a neighborhood of $\mathcal{F}(A)$. Then we can use (2.35) to write

$$f(A)\mathbf{v} = \frac{1}{2\pi \mathrm{i}} \int_\Gamma f(\xi)\,(\xi I - A)^{-1}\,\mathbf{v}\,\mathrm{d}\xi, \tag{2.36}$$

where Γ is a closed curve that encloses $\mathcal{F}(A)$. Using (2.36) we have

$$V_m f(H_m)\mathbf{e}_1 = \frac{1}{2\pi \mathrm{i}} \int_\Gamma f(\xi) V_m\,(\xi I - H_m)^{-1}\,\mathbf{e}_1\,\mathrm{d}\xi. \tag{2.37}$$

When $\xi \notin \mathcal{F}(A)$, ξ is not an eigenvalue of A or H_m, thus neither $(\xi I - A)^{-1}$ nor $(\xi I - H_m)^{-1}$ are singular and (2.36)–(2.37) are valid. We denote the error in the Arnoldi approximation as

$$\begin{aligned}\epsilon_m &= f(A)\mathbf{v} - V_m f(H_m)\mathbf{e}_1, \\ &= \frac{1}{2\pi \mathrm{i}} \int_\Gamma f(\xi)\left[(\xi I - A)^{-1}\,\mathbf{v} - V_m\,(\xi I - H_m)^{-1}\,\mathbf{e}_1\right]\,\mathrm{d}\xi.\end{aligned}$$

Thus we can obtain error bounds by estimating the error in $(\xi I - A)^{-1}\,\mathbf{v} - V_m\,(\xi I - H_m)^{-1}\,\mathbf{e}_1$, and multiply by $|f(\xi)|$. We integrate the result along an appropriate contour Γ. We make use of the following [41, Lemma 1]

Theorem 2.5. *Let E be a convex, closed bounded set in the complex plane, $\mathbb{C}$, with $\mathcal{F}(A) \subset E$. Let $\phi(z)$ be the conformal mapping that carries $\mathbb{C} \setminus E$ onto the exterior of the unit circle ($|w| > 1$), with $\phi(z) = z/\rho + \mathcal{O}(1)$ as $z \to \infty$ for $\rho > 0$. Let $f(z)$ be analytic. Let Γ be a piecewise smooth contour that contains E. Let the matrices V_m and H_m be determined by (2.32). Then for every polynomial q_{m-1} of degree at most $m-1$*

$$\|f(A)\mathbf{v} - V_m f(H_m)\mathbf{e}_1\| \leq \frac{M}{2\pi} \int_\Gamma |f(\xi) - q_{m-1}(\xi)|\, |\phi(\xi)|^{-m}\, |\mathrm{d}\xi|, \qquad (2.38)$$

with $M = l(\partial E)/\left(d(\partial E) d(\Gamma)\right)$, where $l(\partial E)$ is the length of the boundary curve ∂E, and $d(S)$ is the distance between $\mathcal{F}(A)$ and a subset S of $\mathbb{C}$. If Γ is a line segment or a disk, then (2.38) holds with $M = 6/d(\Gamma)$.

The following results from [41] specify the error bounds in Theorem 2.5 for the function $f(A) = \mathrm{e}^{\tau A}$ and for special classes of matrices A.

Corollary 2.1. *Let A be a Hermitian negative semidefinite matrix with eigenvalues in the interval $[-4\rho, 0]$, then the error in the Arnoldi approximation of $\exp(\tau A)\mathbf{v}$ is*

$$\|\epsilon_m\| \leq 10\mathrm{e}^{-m^2/(5\rho\tau)} \qquad \sqrt{4\rho\tau} \leq m \leq 2\rho\tau, \qquad (2.39)$$

$$\|\epsilon_m\| \leq 10(\rho\tau)^{-1}\mathrm{e}^{-\rho\tau}\left(\frac{\mathrm{e}\rho\tau}{m}\right)^m \qquad m \geq 2\rho\tau. \qquad (2.40)$$

Corollary 2.2. *Let A be a matrix with $\mathcal{F}(A)$ contained in the disk $|z+\rho| \leq \rho$, then the error in the Arnoldi approximation of $\exp(\tau A)\mathbf{v}$ is*

$$\|\epsilon_m\| \leq 12\mathrm{e}^{-\rho\tau}\left(\frac{\mathrm{e}\rho\tau}{m}\right)^m \qquad m \geq 2\rho\tau.$$

The error estimates in Corollary 2.1 provide some insight on how exponential methods compare with implicit methods. Suppose that A is symmetric negative definite with eigenvalues in the interval $[-4\rho, 0]$, and we wish to solve (2.3) by the conjugate gradient method. Then application of the conjugate gradient method to solve $(I - \tau A)\,\mathbf{x} = \mathbf{v}$ is equivalent to using the Arnoldi method to approximate

$$\mathbf{x} = f(\tau A)\mathbf{v} = (I - \tau A)^{-1}\,\mathbf{v} \qquad (2.41)$$

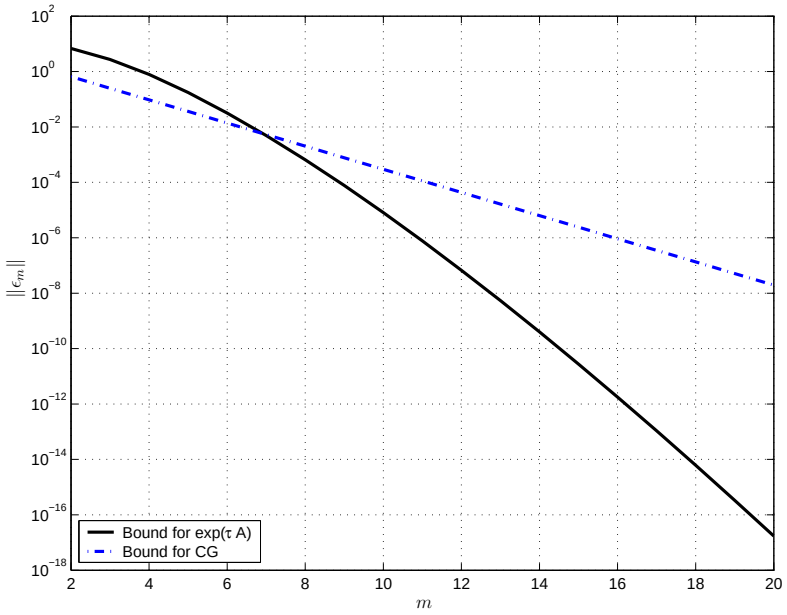

Figure 2.2: Error bounds for the Krylov approximation (2.40) (black) and for conjugate gradient (2.42) (blue) versus m for $\rho = 10$ and $\tau = 10^{-1}$.

in a Krylov subspace. We apply Theorem 2.5 with $f(\tau A)\mathbf{v}$ defined by (2.41) to obtain an error bound for the conjugate gradient method. Theorem 2.5 yields the error bound (see [41, p. 1916])

$$\|\mathbf{x} - \mathbf{x}_m\| \leq 2\sqrt{1+4\rho\tau}\left(1 - \frac{2}{\sqrt{1+4\rho\tau}+1}\right)^m \tag{2.42}$$

for the conjugate gradient method. Comparison of (2.42) to (2.39)–(2.40) suggests that the error in approximating the matrix exponential will be reduced faster than the error in the conjugate gradient method. This phenomena is illustrated in several examples in [41], and confirmed in computational experiments in Section 2.4. Figure 2.2 shows the bounds (2.40) and (2.42) versus m for $\rho = 10$ and $\tau = 10^{-1}$

Hochbruck and Lubich [41, p. 1916] state that the same bounds as in Theorem 2.5 and Corollaries 2.1–2.2 are valid for Krylov subspace approximations of $\varphi(\tau A)\,\mathbf{v}$ with $\varphi(z) = (z-1)/z$.

2.3.3 Computation of the matrix exponential operator

In this section we discuss practical computation of the products $\mathrm{e}^{H_m}\mathbf{e}_1$ and $\varphi(H_m)\mathbf{e}_1$. We make use of a rational approximation to the matrix exponential by partial fraction expansion given in [28]. The rational approximation to the exponential has the form

$$\mathrm{e}^{-z} \approx R_{k_1,k_2}(z) = \frac{p_{k_1}(z)}{q_{k_2}(z)},$$

where $p_{k_1}(z)$ and $q_{k_2}(z)$ are polynomials of degree k_1 and k_2 with leading coefficients ρ_k and σ_k respectively, that is

$$p_k(z) = \sum_{i=1}^{k} \rho_i z^i, \quad \text{and} \quad q_k(z) = \sum_{i=1}^{k} \sigma_i z^i.$$

Approximations of this type are commonly referred to as Padé approximations. Padé approximations are local, and are very accurate near the origin, however may suffer in accuracy away from the origin. We are mainly interested in the integration of ordinary differential equations that arise from finite element discretizations of parabolic partial differential equations. These discretizations possess eigenvalues with ratios in the interval $[0, \mathcal{O}\left(h^{-d}\right)]$, where h is related to the spatial mesh size and d is the spatial dimension. We also wish to utilize large time steps. Thus we seek an approximation that has greater accuracy throughout the interval $[0, \infty)$. We utilize a rational Chebyshev approximation to e^{-z} as introduced in [84]. In the rational Chebyshev approximation the coefficients ρ_k, σ_k are found such that

$$\max |R_{k_1,k_2}(z) - \mathrm{e}^{-z}|$$

is minimized on the interval $[0, \infty)$. The rational Chebyshev approximation will allow us to use a time step as large as our Krylov subspace allows.

We shall restrict ourselves to the use of diagonal rational approximations where $k_1 = k_2$. We use the notation R_k for $R_{k,k}$. To evaluate the approximation to $\mathrm{e}^{-H_m}\mathbf{e}_1$, we compute the vector $\mathbf{y}$ with

$$\mathbf{y} = p_k(H_m)q_k(H_m)^{-1}\mathbf{e}_1 = q_k(H_m)^{-1}p_k(H_m)\mathbf{e}_1. \tag{2.43}$$

We compute (2.43) via its partial fraction expansion

$$R_k(z) = \alpha_0 + \sum_{i=1}^{k} \frac{\alpha_i}{z - \beta_i},$$

where

$$\alpha_0 = \frac{\rho_k}{\sigma_k}, \quad \alpha_i = \frac{p_k(\beta_i)}{q_k'(\beta_i)}, \quad i = 1, 2, \ldots k,$$

and β_i, $i = 1, 2, \ldots k$ are the roots of $q_k(z)$. The coefficients α_i and β_i for $k = 10$ and 14 are given in [28, p. 1261]. The vector $\mathbf{y}$ is computed using the following

Algorithm 2.2 (Approximation of $\mathbf{y} = \mathrm{e}^{-H_m}\mathbf{e}_1$).

1: **for** i=1,2,...,k **do**
2: solve $(H_m - \beta_i I)\,\mathbf{y}_i = \mathbf{e}_1$
3: **end for**
4: $\mathbf{y} = \alpha_0\mathbf{e}_1 + \sum_{i=1}^{k} \alpha_i \mathbf{y}_i$

We make use of the relation [69, (17) p. 214] to approximate $\varphi(-z)$ as follows:

$$\varphi(-z) \approx Q_k(z) = \sum_{i=1}^{k} \frac{\alpha_i}{\beta_i\,(z - \beta_i)}.$$

Similar to Algorithm 2.2 we the following

Algorithm 2.3 (Approximation of $\mathbf{y} = \varphi(-H_m)\mathbf{e}_1$).

1: **for** i=1,2,...,k **do**
2: solve $(H_m - \beta_i I)\,\mathbf{y}_i = \mathbf{e}_1/\beta_i$
3: **end for**
4: $\mathbf{y} = \sum_{i=1}^{k} \alpha_i \mathbf{y}_i$

We remark step 2 in Algorithms 2.2–2.3 involve a linear solution of an upper Hessenberg matrix. Hence this step is handled efficiently by a direct method. Algorithms 2.2–2.3 have worked well in practice, for a discussion of stability properties and accuracy see [26, 43, 44, 85].

As a final note, when we make the approximation

$$\varphi(\gamma\Delta t A)\,\mathbf{v} \approx V_m \varphi(\gamma\Delta t H_m)\,\mathbf{e}_1 \tag{2.44}$$

in the exponential method defined by (2.11)–(2.13) or (2.14)–(2.15) we refer to those methods as *Krylov–based exponential methods*.

2.4 Heat Equation

In this section we investigate exponential integrators for large systems of ordinary differential equations. An exponential integration method is applied to an ODE obtained from discretizing the heat equation via the finite element method in one spatial dimension. The performance of this method is compared with the Crank Nicolson method.

2.4.1 Model problem

Consider the time dependent heat equation in one dimension,

$$u_t(x,t) = \Delta u(x,t), \tag{2.45}$$
$$u(0,x) = u(1,x) = 0, \tag{2.46}$$
$$u(x,0) = g(x). \tag{2.47}$$

With initial condition given by $g(x) = x(1-x)\sinh \pi x$, (2.45)–(2.46) has an exact solution that can be determined by Fourier series.

The problem is spatially discretized using the finite element method, where the solution $u(x,t)$ is approximated by $u^h(x,t)$, with

$$u^h(x,t) = \sum_{i=1}^{n} \mathrm{u}_i(t)\varphi_i(x).$$

Each $\varphi_i(x)$ is a basis function for the finite element space and each $\mathrm{u}_i(t)$ is a time dependent coefficient to be determined. Problem (2.45)–(2.47) was solved using piecewise linear ($p = 1$), piecewise quadratic ($p = 2$), and piecewise cubic ($p = 3$) basis functions. Each discretization yields a finite dimensional system of ordinary differential equations given by

$$M\dot{\mathbf{u}}(t) = -K\mathbf{u}(t), \tag{2.48}$$
$$\mathbf{u}(0) = \mathbf{u}_0, \tag{2.49}$$

where M is the *mass* matrix, K is the *stiffness* matrix, $\mathbf{u}(t)$ is a vector containing the coefficients $\mathrm{u}_i(t)$, and $\mathbf{u}_0$ is a vector containing the coefficients of the Lagrange interpolant of the initial condition $g(x)$. For this example, the spatial discretization is chosen to be $h = 1/128 = 0.0078125$. The matrices M and K are square and of dimension $n = 129$ for $p = 1$, $n = 257$ for $p = 2$ and $n = 385$ for $p = 3$. The vectors $\mathbf{u}$ and $\mathbf{u}(t)$ are also of dimension n.

Finite element theory [5] states that the spatial discretization obeys

$$\|u(x,t) - u^h(x,t)\|_0 = \mathcal{O}(h^{p+1}), \tag{2.50}$$

where $\|\cdot\|_0$ is the L^2 norm [64, p. 99]. The bound (2.50) also holds for the Lagrange interpolant of the initial condition [64, p. 91]. The error in the initial condition in $\|u(x,0) - u^h(x,0)\|_0$ is 1.50714×10^{-4} for $p =$ 1,4.7363×10^{-7} for $p = 2$ and 1.08518×10^{-9} for $p = 3$.

In order to preserve this spatial accuracy, the time step Δt must be chosen such that the accuracy of the time integration method is as accurate as the spatial discretization. Rather than examining the L^2 error in the continuous problem given by (2.50), we will examine the relative l^2 error in computing the coefficient vector $\mathbf{u}(t)$ in the discrete problem (2.48)–(2.49). The relative l^2 error is given by

$$\epsilon(t) = \frac{\|\mathbf{u}(t) - \mathbf{u}_e(t)\|}{\|\mathbf{u}_e(t)\|},$$

where $\mathbf{u}(t)$ is the computed solution to (2.48)–(2.49) at time t, $\mathbf{u}_e(t)$ is a Lagrange interpolant to the exact solution at time t and $\|\cdot\|$ is the norm in l^2.

2.4.2 Exponentially fitted Euler Method for the Heat Equation

The discrete heat equation (2.48)–(2.49), can be rewritten as

$$\dot{\mathbf{u}}(t) = -M^{-1}K\mathbf{u}(t), \tag{2.51}$$

$$\mathbf{u}(0) = \mathbf{u}_0. \tag{2.52}$$

The Jacobian of equation (2.51) is given by $-M^{-1}K$. Applying the exponentially fitted Euler method (2.10) to (2.51)–(2.52) yields,

$$\mathbf{u}_{n+1} = \varphi(-\Delta t M^{-1}K)\mathbf{u}_n, \tag{2.53}$$

where Δt is the step size, $\varphi(z) = e^z$ and $\mathbf{u}_n = \mathbf{u}(t_n)$. The task is to compute the Krylov subspace $K_m = \text{span}\{\mathbf{u}_n, (M^{-1}K)\,\mathbf{u}_n, \ldots, (M^{-1}K)^{m-1}\,\mathbf{u}_n\}$, for each $\mathbf{u}_n$ (at each time step), via the Arnoldi process, and to approximate

$$\varphi(-\Delta t M^{-1}K)\mathbf{u}_n$$

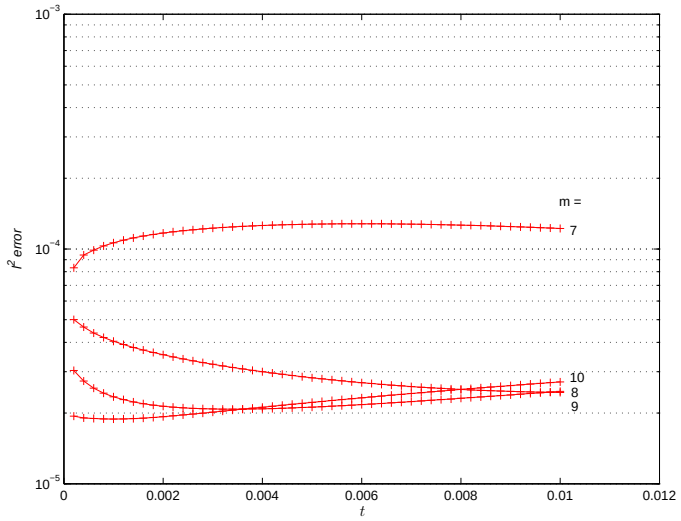

Figure 2.3: Relative l^2 error at each time step for the exponentially fitted Euler method corresponding to $\Delta t = 2 \cdot 10^{-4}$, for $p = 1$ and selected values of m.

by

$$V_m \varphi(-\Delta t H_m)\mathbf{e}_1$$

in (2.53). We are interested in integrating (2.51)–(2.52) from $t = 0$ to $t = 1 \cdot 10^{-2}$ in such a way that the l^2 error at $t = 1 \cdot 10^{-2}$ is as small as the L^2 spatial discretization error with as little computational costs as possible.

Figures 2.3–2.4 show relative l^2 errors at each time step for the exponentially fitted Euler method for $p = 1$ with time steps $\Delta t = 2 \cdot 10^{-4}$ and $\Delta t = 1 \cdot 10^{-4}$ and various values of m. Observe that as m increases, the error decreases, and as Δt decreases the error decreases.

As the degree of the spatial approximation, p, increases, so does the spatial accuracy and matrix dimension. Thus a more accurate approximation to a larger dimension matrix is required. Figures 2.5–2.6 show relative l^2 errors for $p = 2$ and $p = 3$ with time steps $\Delta t = 10^{-4}$.

To reduce the error using the exponentially fitted Euler method with Krylov approximation to the exponential, the time step can be decreased, the dimension of the Krylov subspace can be increased, or a combination of the two can be employed.

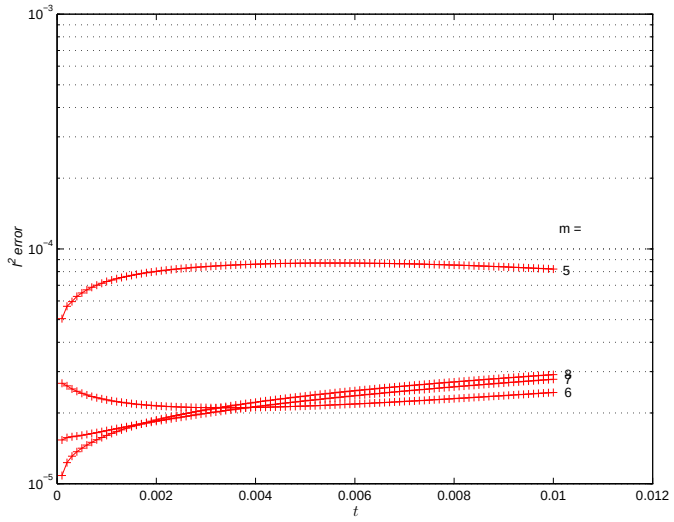

Figure 2.4: Relative l^2 error at each time step for the exponentially fitted Euler method corresponding to $\Delta t = 1 \cdot 10^{-4}$, for $p = 1$ and selected values of m.

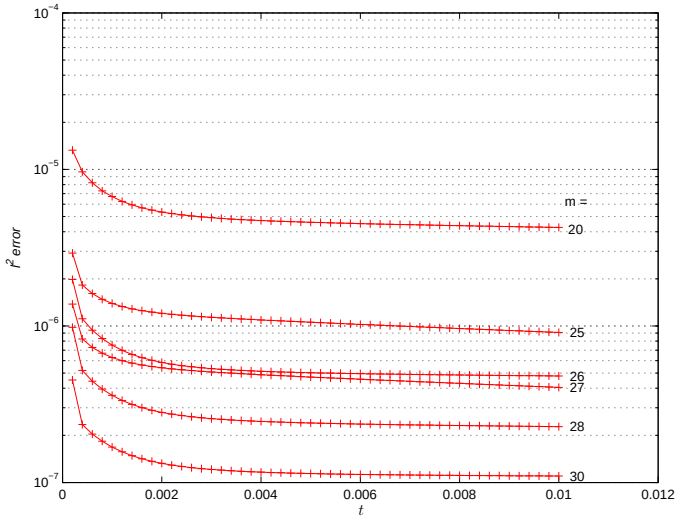

Figure 2.5: Relative l^2 error at each time step for the exponentially fitted Euler method corresponding to $\Delta t = 2 \cdot 10^{-4}$, for $p = 2$ and selected values of m.

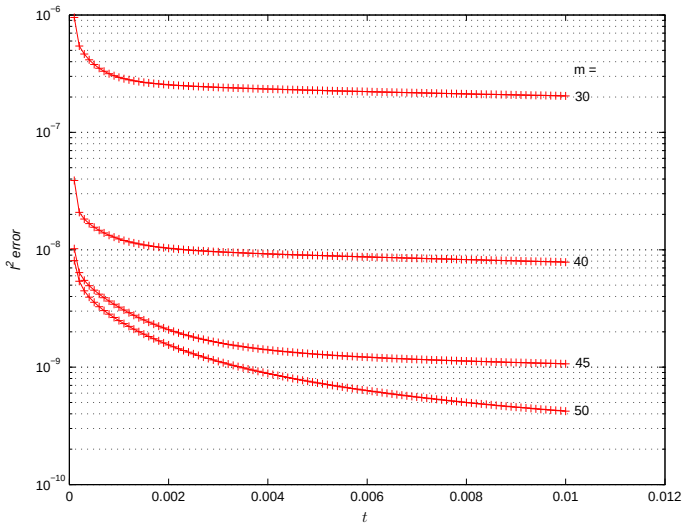

Figure 2.6: Relative l^2 error at each time step for the exponentially fitted Euler method corresponding to $\Delta t = 1 \cdot 10^{-4}$, for $p = 3$ and selected values of m.

2.4.3 Crank Nicolson Method for the Heat Equation

The second order implicit Crank Nicolson method applied to (2.48)–(2.49) yields

$$\left(M + \frac{\Delta t}{2}K\right)\mathbf{u}_{n+1} = \left(M - \frac{\Delta t}{2}K\right)\mathbf{u}_n. \tag{2.54}$$

Crank Nicolson has local truncation error $\mathcal{O}(\Delta t^3)$, and global error $\mathcal{O}(\Delta t^2)$.

In order to reduce the error using Crank Nicolson, the only option is to reduce the time step size. Figures 2.7–2.9 show relative l^2 errors for Crank Nicolson with time steps $\Delta t = 1 \cdot 10^{-3}$, $\Delta t = 1 \cdot 10^{-4}$ and $\Delta t = 1 \cdot 10^{-5}$ for $p = 1$, 2 and 3. As expected, the relative error decreases as the step size decreases. Note that for $p = 3$, a step size of $\Delta t = 1 \cdot 10^{-5}$ is not sufficient to achieve an error in time integration that balances the error in spatial discretization. If we wish to achieve a balance in temporal and spatial accuracy we require that $(\Delta t)^2 \approx h^{p+1}$. In this case, the step size Δt should be chosen *smaller than* $\Delta t = 1 \cdot 10^{-5}$. As can be seen in Figure 2.9, high order accuracy can require many small time steps.

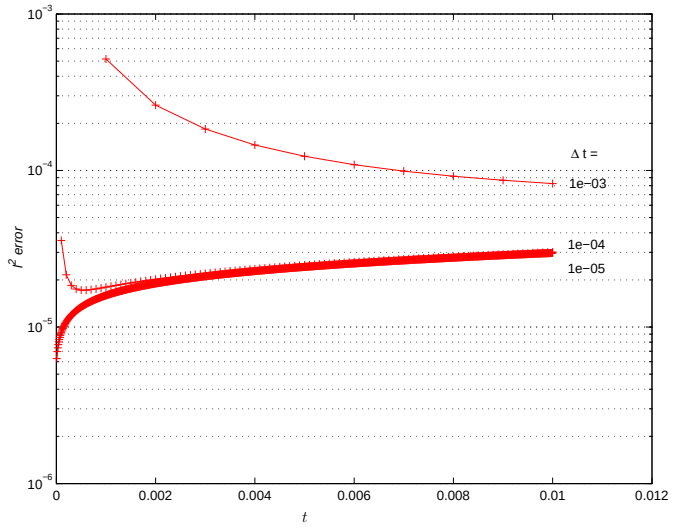

Figure 2.7: Relative l^2 error at each time step for the Crank Nicolson method for $p = 1$ and $\Delta t = 1 \cdot 10^{-3}, 1 \cdot 10^{-4}, 1 \cdot 10^{-5}$.

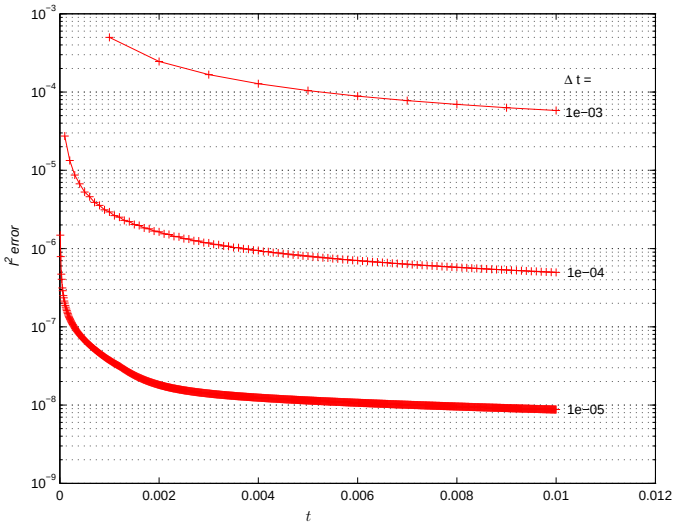

Figure 2.8: Relative l^2 error at each time step for the Crank Nicolson method for $p = 2$ and $\Delta t = 1 \cdot 10^{-3}, 1 \cdot 10^{-4}, 1 \cdot 10^{-5}$.

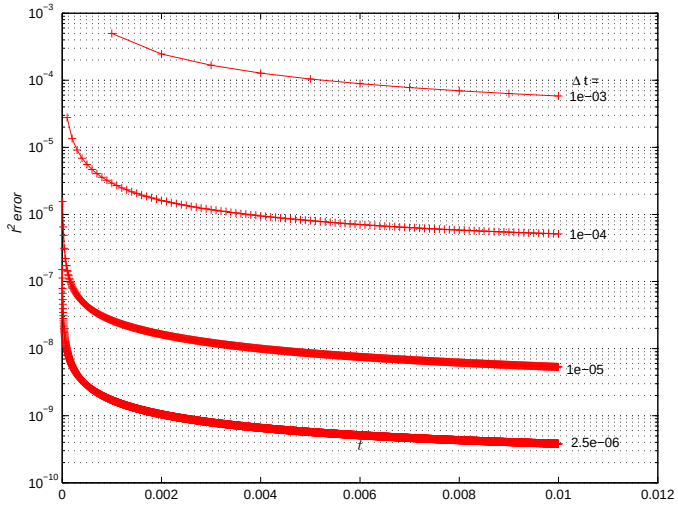

Figure 2.9: Relative l^2 error at each time step for the Crank Nicolson method for $p = 3$ and $\Delta t = 1 \cdot 10^{-3}, 1 \cdot 10^{-4}, 1 \cdot 10^{-5}, 2.5 \cdot 10^{-6}$.

2.4.4 Computational Costs

Figure 2.10 shows relative l^2 errors for the exponentially fitted Euler method with $p = 3$ and $\Delta t = 1 \cdot 10^{-4}$ for $m = 40$, $m = 45$ and $m = 50$; and for the Crank Nicolson method with $p = 3$ and $\Delta t = 1 \cdot 10^{-5}$, $\Delta t = 2.5 \cdot 10^{-6}$. In particular, it shows that the Crank Nicolson method with step size $\Delta t = 1 \cdot 10^{-5}$ is comparable to the exponentially fitted Euler method with $\Delta t = 1 \cdot 10^{-4}$ and $m = 40$ and that the Crank Nicolson method with step size $\Delta t = 2.5 \cdot 10^{-6}$ is comparable to the exponentially fitted Euler method with $\Delta t = 1 \cdot 10^{-4}$ and $m = 50$. It also shows that better accuracy can be attained with $m > 40$.

Unless mass lumping is used or the operator $M^{-1}K$ can be computed, the computation of one Krylov vector requires the application of M^{-1}. Recall that the condition number of M is $\mathcal{O}(1)$. Each application of M^{-1} by the conjugate gradient method with tolerance $1 \cdot 10^{-6}$ requires about 8 iterations. Hence about 3.2×10^4 iterations of conjugate gradient are required to reach $t = 1 \cdot 10^{-2}$ with the exponentially fitted Euler method for $m = 40$ and about 4.0×10^4 iterations are required for $m = 50$.

Crank Nicolson requires an application of $\left(M + \frac{\Delta t}{2} K\right)^{-1}$ to the vector

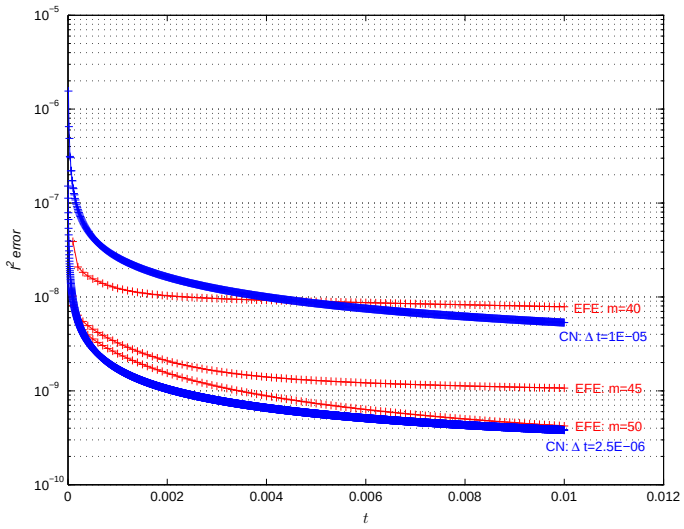

Figure 2.10: Relative l^2 error at each time step for the exponentially fitted Euler method for $p = 3$ with $\Delta t = 1 \cdot 10^{-4}$, $m = 40$, $m = 45$ and $m = 50$ (red) and for the Crank Nicolson method for $p = 3$ with $\Delta t = 1 \cdot 10^{-5}$ and $\Delta t = 2.5 \cdot 10^{-6}$ (blue).

$\left(M - \frac{\Delta t}{2}K\right)\mathbf{u}_n$ at each time step. Some rearranging yields,

$$\begin{aligned}
\mathbf{u}_{n+1} &= \left(M + \frac{\Delta t}{2}K\right)^{-1}\left(M - \frac{\Delta t}{2}K\right)\mathbf{u}_n, \\
&= \left(I + \frac{\Delta t}{2}M^{-1}K\right)^{-1} M^{-1}M\left(I - \frac{\Delta t}{2}M^{-1}K\right)\mathbf{u}_n, \\
&= \left(I + \frac{\Delta t}{2}M^{-1}K\right)^{-1}\left(I - \frac{\Delta t}{2}M^{-1}K\right)\mathbf{u}_n.
\end{aligned}$$

The condition number of K is $\mathcal{O}(h^{-2})$, hence the condition number of $M^{-1}K$ is $\mathcal{O}(h^{-2})$. When Δt is small, in particular when $\Delta t = \mathcal{O}(h^2)$ the condition number of $I + \Delta t M^{-1}K$ is small and the application of $(I + \Delta t M^{-1}K)^{-1}$ can be achieved via an iterative process with very few iterations. However, $(I + \Delta t M^{-1}K)^{-1}$ needs to be applied at every time step. In this example the conjugate gradient method with tolerance $1 \cdot 10^{-8}$ was used. In practice, this tolerance is a little excessive, however it assures an accurate representation of the temporal error. Integration to $t = 1 \cdot 10^{-2}$ required about $1.2 \cdot 10^4$ iterations of conjugate gradient for $\Delta t = 1 \cdot 10^{-5}$ and about $3.0 \cdot 10^4$ iterations

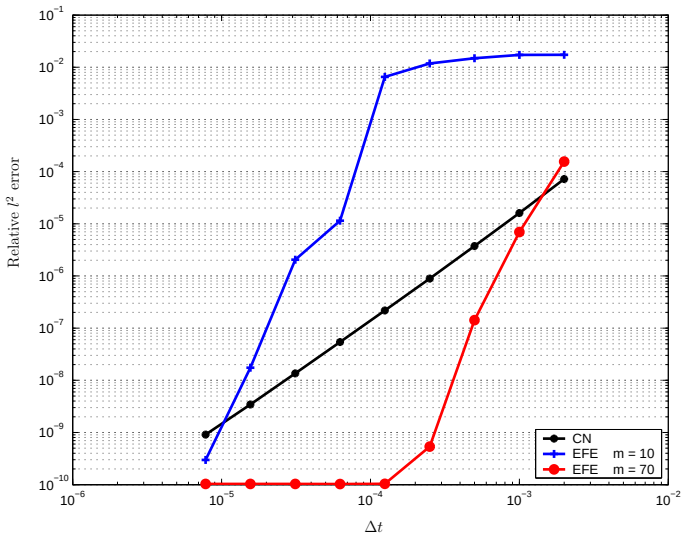

Figure 2.11: Relative l^2 error at $\Delta t = 2 \cdot 10^{-3}, 1 \cdot 10^{-3}, 5 \cdot 10^{-4}, 2.5 \cdot 10^{-4}, 1.25 \cdot 10^{-4}, 6.25 \cdot 10^{-5}, 3.125 \cdot 10^{-5}, 1.563 \cdot 10^{-5}, 7.813 \cdot 10^{-5}$ for the exponentially fitted Euler method for $p = 3$ with $m = 10$ (blue), and $m = 70$ (red) and for the Crank Nicolson method for $p = 3$ (black).

of conjugate gradient for $\Delta t = 2.5 \cdot 10^{-6}$. When Δt is large, considerably more iterations of conjugate gradient are needed to apply $\left(M + \frac{\Delta t}{2} K\right)^{-1}$

We note that increasing the number of Krylov vectors from $m = 40$ to $m = 50$ requires about 20% increase in computational cost. However, decreasing the step size from $\Delta t = 1 \cdot 10^{-5}$ to $\Delta t = 2.5 \cdot 10^{-6}$ requires over 200% increase in computational cost.

These results suggest that the exponentially fitted Euler method and the Crank Nicolson method can be used to integrate the example problem with comparable computational cost and accuracy.

Figure 2.11 shows relative l^2 error versus time step size for the exponentially fitted Euler method with $m = 10$ and $m = 70$ and the Crank Nicolson method for $p = 3$. The figure reveals extremely high convergence rates for the exponentially fitted Euler method. However, for $m = 10$ a small time step is needed to attain a high convergence rate, while large time steps attain higher order convergence rates for $m = 70$. In particular for the exponentially fitted Euler method with $\Delta t = 5 \times 10^{-4}$ and $m = 70$ corresponds to an error on the order of $l^2 = 10^{-7}$. For the Crank Nicolson method, a time

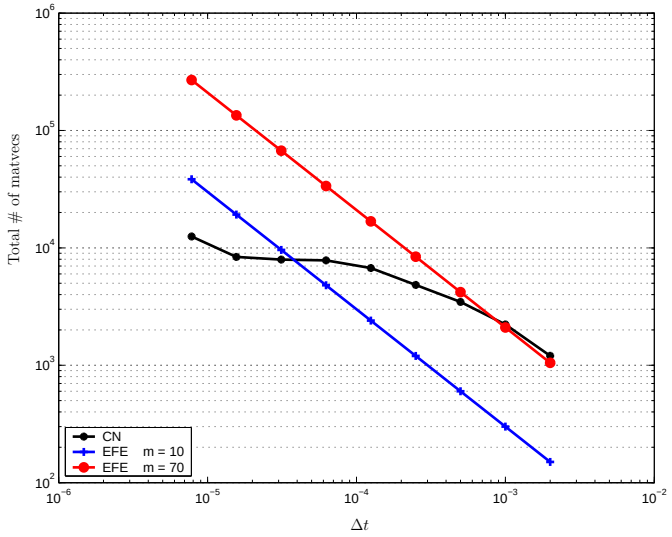

Figure 2.12: Cost measured in total number of matrix–vector products at $\Delta t = 2{\cdot}10^{-3}, 1{\cdot}10^{-3}, 5{\cdot}10^{-4}, 2.5{\cdot}10^{-4}, 1.25{\cdot}10^{-4}, 6.25{\cdot}10^{-5}, 3.125{\cdot}10^{-5}, 1.563{\cdot}10^{-5}, 7.813 \cdot 10^{-5}$ for the exponentially fitted Euler method for $p = 3$ with $m = 10$ (blue), and $m = 70$ (red) and for the Crank Nicolson method for $p = 3$ (black).

step of $\Delta t < 1.25 \times 10^{-4}$ is needed to achieve this error.

The relative costs of each method are shown in Figure 2.12. The cost is measured in total number of matrix–vector products required for integration to $t = 10^{-2}$. The cost grows linearly for the exponentially fitted Euler method, while the cost grows sub-linear for the Crank Nicolson method. We note for the exponentially fitted Euler method with $\Delta t = 5 \times 10^{-4}$ and $m = 70$, the cost is roughly 2 orders of magnitude less than that of the Crank Nicolson method with $\Delta t < 1.25 \times 10^{-4}$. In this case the exponentially fitted Euler method can take a time step that is four times longer, with greater accuracy and less cost that the Crank Nicolson method.

We remark that the exponentially fitted Euler method and the Crank Nicolson method are comparable in this example. This problem is "easy" for Crank Nicolson in the sense that preconditioning is not necessary for the solution of (2.54) when time steps are sufficiently small. In a more complicated problem preconditioning Crank Nicolson may be difficult. Higher spatial dimension, complicated geometry, an unstructured mesh, or the introduction

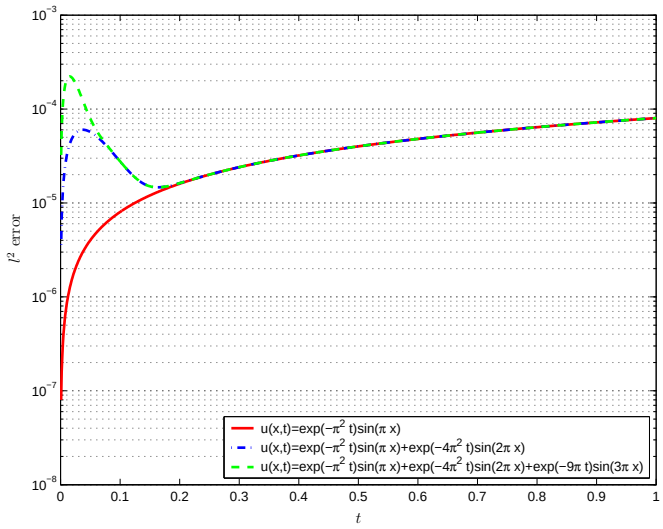

Figure 2.13: Relative l^2 error at each time step for the Crank Nicolson method for $p = 3$ with $\Delta t = 1 \cdot 10^{-3}$, when the exact solution to (2.45)–(2.47) is $u(x,t) = \mathrm{e}^{-\pi^2 t} \sin \pi x$ (red), $u(x,t) = \mathrm{e}^{-\pi^2 t} \sin \pi x + \mathrm{e}^{-4\pi^2 t} \sin 2\pi x$ (blue), and $u(x,t) = \mathrm{e}^{-\pi^2 t} \sin \pi x + \mathrm{e}^{-4\pi^2 t} \sin 2\pi x + \mathrm{e}^{-9\pi^2 t} \sin 3\pi x$ (green).

of constraints may also lead to difficult preconditioning. If preconditioning is difficult or nonexistent, the use of exponential integrators may prove to be beneficial.

We also remark that Figures 2.3–2.10 show that temporal errors decrease as t increases. This seems counter intuitive, as errors in time integration should accumulate. However, the exact solution of (2.51)–(2.52) is given by the Fourier series

$$u(x,t) = \sum_{n=1}^{\infty} c_n \mathrm{e}^{-n^2\pi^2 t} \sin(n\pi x),$$

where $c_n = 2\int_0^1 g(x) \sin(n\pi x)$. Thus $u(x,t)$ is a linear combination of sine waves $\sin(n\pi x)$ with frequencies $n\pi$ and amplitudes $c_n e^{-n^2\pi^2 t}$. The solution $u(x,t)$ has an initial phase, called the *initial transient* for small t where derivatives are large but decrease as t increases. The large derivatives in the initial transient cause the error to be amplified. For a discussion of this effect see [45, pp. 147–149]. Figures 2.3–2.10 show errors for t inside this initial transient.

This phenomenon is demonstrated in Figure 2.13. Initial conditions are

chosen such that the exact solution to (2.45)–(2.47) is $u(x,t) = \mathrm{e}^{-\pi^2 t}\sin(\pi x)$, $u(x,t) = \mathrm{e}^{-\pi^2 t}\sin(\pi x) + \mathrm{e}^{-4\pi^2 t}\sin(2\pi x)$, and $u(x,t) = \mathrm{e}^{-\pi^2 t}\sin(\pi x) + \mathrm{e}^{-4\pi^2 t}\sin(2\pi x) + \mathrm{e}^{-9\pi^2 t}\sin(3\pi x)$. The Crank Nicolson method is applied to (2.48)–(2.49) for $t \in [0,1]$ with $\Delta t = 1 \cdot 10^{-3}$ for each initial condition. In each case $\mathbf{u}_0$ was chosen as the Lagrange interpolant of $u(x,0)$. Figure 2.13 shows l^2 errors in each case. Note that when the exact solution contains terms with large derivatives in the initial transient, the error is amplified in that region.

Chapter 3

Navier–Stokes equations

Our second building block requires an efficient, scalable application of a projection onto a divergence–free subspace. In this chapter, we address this issue by deriving the two common divergence–free projections. The divergence–free L^2 projection is the most commonly used projection. This projection stems from the L^2 inner product. The less common divergence–free H^1 projection is induced by the H^1 seminorm. Since there are close connections between the incompressible Navier–Stokes equations and these projections, we provide a background for these equations as well. Practical implementation issues regarding both the L^2 and H^1 divergence–free projections will be discussed in Chapter 4.

3.1 Function Spaces

In this section we introduce some function spaces that are important for the theory of the Navier–Stokes equations.

Let H be a Hilbert space. We define the *projection operator* $P : H \to H$ by

1. $PPu = Pu, \quad \forall u \in H.$

2. P is a bounded operator.

P is an *orthogonal projection operator* if $P = P^*$, otherwise P is an *oblique projection operator*.

Let Ω be an open, bounded domain in $\mathbb{R}^d$, $d = 2$ or 3 with a Lipschitz continuous boundary Γ, that is Γ can be parameterized by Lipschitz continuous functions. A particular family of Hilbert spaces is defined as measurable functions u such that

$$\int_\Omega |u|^2 \, \mathrm{d}x < \infty.$$

This function space is denoted by $L^2(\Omega)$ and is endowed with the norm

$$\|u\| = \left(\int_\Omega |u|^2 \, \mathrm{d}x \right)^{1/2},$$

and inner product

$$(u, v) = \int_\Omega uv \, \mathrm{d}x.$$

We use the following notation for partial derivatives:

$$D^\alpha u = \frac{\partial^{|\alpha|} u}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \cdots \partial x_n^{\alpha_n}},$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ is a *multi–index* whose components are non–negative integers. For any multi–index we make the definition $|\alpha| = \sum_{i=1}^n \alpha_i$. For k a non–negative integer we define the *Sobolev space*

$$H^k(\Omega) = \left\{ u \in L^2(\Omega) : D^\alpha u \in L^2(\Omega), \forall \ |\alpha| \leq k \right\}.$$

$H^k(\Omega)$ is a Hilbert space with norm

$$\|u\|_k = \left(\sum_{|\alpha| \leq k} \|D^\alpha u\|^2 \right)^{1/2},$$

and corresponding seminorm

$$|u|_k = \left(\sum_{|\alpha| = k} \|D^\alpha u\|^2 \right)^{1/2},$$

and inner product

$$(u, v)_k = \sum_{|\alpha| \leq k} (D^\alpha u, D^\alpha v).$$

We identify $H^0(\Omega)$ with $L^2(\Omega)$.

The trace theorem given in [64, Theorem 1.3.1 p. 10] allows us to consider how functions in Sobolev spaces can be restricted to the boundary Γ of Ω.

Theorem 3.1 (Trace Theorem). *Let Ω be a bounded open set of $\mathbb{R}^d$ with Lipschitz continuous boundary Γ and let $s > 1/2$.*

1. *There exists a unique continuous trace operator $\gamma_0 : H^s(\Omega) \to H^{s-1/2}(\Gamma)$ such that $\gamma_0 v = v|_\Gamma$ for each $v \in H^s(\Omega) \cap C^0(\bar{\Omega})$.*

2. *There exists a continuous operator $R_0 : H^{s-1/2}(\Gamma) \to H^s(\Omega)$ such that $\gamma_0 R_0 \varphi = \varphi$ for each $\varphi \in H^{s-1/2}(\Gamma)$.*

For $g \in H^{s-1/2}(\Gamma)$ we define the boundary norm

$$\|g\|_{s-1/2} = \inf_{\substack{u \in H^s(\Omega) \\ \gamma_0 u = g}} \|u\|_s.$$

The existence of the trace operator enables us to characterize functions in $H_0^1(\Omega)$ in terms of a boundary condition. We define the space $H_0^k = (\Omega)\{u \in H^k(\Omega) : \gamma_0 u = 0\}$ and denote the dual space of $H_0^k(\Omega)$ by $H^{-k}(\Omega)$ Analogous results hold for the trace operator γ_Σ on a Lipschitz continuous subset Σ of Γ. For example, we can define the space

$$H^1(\Omega, \Sigma) = \left\{u \in H^1(\Omega) : \gamma_\Sigma u = 0\right\},$$

of functions which vanish on Σ.

When analyzing functions of both space and time, we utilize the space

$$L^2(0, T; \Omega) = \left\{u : (0, T) \to H^k(\Omega) : \int_0^T \|u(t)\|_k^2 \, \mathrm{d}t < \infty\right\},$$

with norm

$$\|u\|_{L^2(0,T;\Omega)} = \left(\int_0^T \|u(t)\|_k^2 \, \mathrm{d}t\right)^{1/2}.$$

We define the following differential operators. For scalar fields $q : \Omega \to \mathbb{R}$ we denote the *gradient* by ∇q. The gradient is a vector with

$$[\nabla q]_i = \frac{\partial q}{\partial x_i} \quad 1 < i \leq d.$$

For vector fields $\boldsymbol{u} : \Omega \to \mathbb{R}^d$ the gradient is a matrix defined as

$$[\nabla \boldsymbol{u}]_{ij} = \frac{\partial u_j}{\partial x_i} \quad 1 < i, j \leq d.$$

We denote the *divergence* by $\nabla \cdot \boldsymbol{u}$ with

$$\nabla \cdot \boldsymbol{u} = \sum_{i=1}^{d} \frac{\partial u_i}{\partial x_i}.$$

For matrix fields $A : \Omega \to \mathbb{R}^{d\times d}$, the divergence is a vector defined by

$$[\nabla \cdot A]_i = \sum_{j=1}^{d} \frac{\partial A_{ji}}{\partial x_j} \quad 1 < i \leq d.$$

When $d = 2$ and $q : \Omega \to \mathbb{R}$ we denote the *curl* by $\nabla \times q$ with

$$\nabla \times q = \begin{bmatrix} \partial q/\partial x_2 \\ -\partial q/\partial x_1 \end{bmatrix},$$

and for $\boldsymbol{u} : \Omega \to \mathbb{R}^2$ we define

$$\nabla \times \boldsymbol{u} = \frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_2}.$$

When $d = 3$ and $\boldsymbol{u} : \Omega \to \mathbb{R}^3$ we define the curl as

$$\nabla \times \boldsymbol{u} = \begin{bmatrix} \partial u_3/\partial x_2 - \partial u_2/\partial x_3 \\ \partial u_1/\partial x_3 - \partial u_3/\partial x_1 \\ \partial u_2/\partial x_1 - \partial u_1/\partial x_2 \end{bmatrix}.$$

We define the *Laplacian* operator as $\Delta = \nabla \cdot \nabla$, and note the Laplacian makes sense when applied to both scalar and vector fields. For matrices A, $B : \Omega \to \mathbb{R}^{d\times d}$, we will also use the notation

$$A : B = \sum_{i,j=1}^{d} A_{ij} B_{ij}.$$

For vector valued functions $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ we denote the *Jacobian* of $\mathbf{f}$ by $\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}\mathbf{u}}$ with

$$\left[\frac{\mathrm{d}\mathbf{f}(\mathbf{u})}{\mathrm{d}\mathbf{u}}\right]_{ij} = \frac{\mathrm{df}_i(\mathbf{u})}{\mathrm{du}_j} \quad 1 \leq i \leq m,\ 1 \leq j \leq n. \tag{3.1}$$

Sobolev spaces extend to vector-valued functions if we require that each component belongs to the Sobolev space. For a vector $\boldsymbol{u}$ we will use the notation $\gamma_0 \boldsymbol{u}$ to denote the trace of $\boldsymbol{u}$ by

$$[\gamma_0 \boldsymbol{u}]_i = \gamma_0 u_i \quad 1 \leq i \leq d.$$

We denote the *exterior unit normal* vector by $\boldsymbol{n}$, with components n_i, $1 \leq i \leq d$, and the *unit tangent* vector by $\boldsymbol{\tau}$, with components τ_i, $1 \leq i \leq d$.

We will make use of additional Sobolev spaces of vector-valued functions defined below. When considering incompressible flows we utilize the subspace of $L^2(\Omega)^d$ whose divergence is in $L^2(\Omega)$,

$$H^{\mathrm{div}}(\Omega) = \left\{ \boldsymbol{u} \in L^2(\Omega)^d : \nabla \cdot \boldsymbol{u} \in L^2(\Omega) \right\}.$$

The space $H^{\mathrm{div}}(\Omega)$ is a Hilbert space with inner product

$$(\boldsymbol{u}, \boldsymbol{v})_{\mathrm{div}} = (\boldsymbol{u}, \boldsymbol{v}) + (\nabla \cdot \boldsymbol{u}, \nabla \cdot \boldsymbol{v}),$$

and norm

$$\|\boldsymbol{u}\|_{\mathrm{div}} = (\boldsymbol{u}, \boldsymbol{u})_{\mathrm{div}}^{1/2}.$$

In addition to Theorem 3.1, there is a theorem [64, Theorem 1.3.2 p. 10] that allows us to discuss normal components of vectors restricted to boundaries.

Theorem 3.2 (Trace Theorem for vector functions). *Let Ω be a bounded open set of $\mathbb{R}^d$ with Lipschitz continuous boundary Γ.*

1. *There exists a unique continuous trace operator $\gamma_{\boldsymbol{n}} : H^{\mathrm{div}}(\Omega) \to H^{-1/2}(\Gamma)$ such that $\gamma_{\boldsymbol{n}} \boldsymbol{v} = (\boldsymbol{v} \cdot \boldsymbol{n})|_\Gamma$ for each $v \in H^{\mathrm{div}}(\Omega) \cap C^0\left(\bar{\Omega}\right)$.*

2. *There exists a continuous operator $R_{\boldsymbol{n}} : H^{-1/2}(\Gamma) \to H^{\mathrm{div}}(\Omega)$ such that $\gamma_{\boldsymbol{n}} R_{\boldsymbol{n}} \varphi = \varphi$ for each $\varphi \in H^{-1/2}(\Gamma)$.*

In addition to $H^{\mathrm{div}}(\Omega)$ we define

$$H_0^{\mathrm{div}}(\Omega) = \left\{ \boldsymbol{u} \in H^{\mathrm{div}}(\Omega) : \gamma_{\boldsymbol{n}} \boldsymbol{u} = 0 \right\}.$$

The *divergence–free* subspaces of $L^2(\Omega)^d$ and $H^1(\Omega)^d$ respectively are

$$H(\Omega) = \left\{ \boldsymbol{u} \in L^2(\Omega)^d : \nabla \cdot \boldsymbol{u} = 0,\ \gamma_{\boldsymbol{n}} \boldsymbol{u} = 0 \right\}, \tag{3.2}$$

$$J(\Omega) = \left\{ \boldsymbol{u} \in H^{\mathrm{div}}(\Omega) : \nabla \cdot \boldsymbol{u} = 0 \right\},$$

and

$$V(\Omega) = \left\{ \boldsymbol{u} \in H_0^1(\Omega)^d : \nabla \cdot \boldsymbol{u} = 0 \right\}.$$

We also consider the space of *curl–free* functions

$$G(\Omega) = \left\{ \boldsymbol{u} \in L^2(\Omega)^d : \boldsymbol{u} = \nabla q,\ q \in H^1(\Omega) \right\}. \tag{3.3}$$

When considering approximations to pressure fields, we will utilize the space of *zero–mean* functions

$$L_0^2(\Omega) = \left\{ q \in L^2(\Omega) : \int_\Omega q \, \mathrm{d}\boldsymbol{x} = 0 \right\}.$$

We will also utilize some integration formulas. For u, $v \in H^1(\Omega)$ we have *Green's formula*:

$$\int_\Omega v \frac{\partial u}{\partial x_i} \, \mathrm{d}\boldsymbol{x} = -\int_\Omega u \frac{\partial v}{\partial x_i} \, \mathrm{d}\boldsymbol{x} + \int_\Gamma (\gamma_0 u)(\gamma_0 v) \, n_i \, \mathrm{d}s \quad 1 \le i \le d. \tag{3.4}$$

In particular we have *Stokes' formula*, for $\boldsymbol{u} \in H^{\mathrm{div}}(\Omega)$ and $q \in H^1(\Omega)$,

$$(q, \nabla \cdot \boldsymbol{u}) = -(\boldsymbol{u}, \nabla q) + \langle \gamma_{\boldsymbol{n}} \boldsymbol{u}, \gamma_0 q \rangle_\Gamma. \tag{3.5}$$

For $u \in H^2(\Omega)$ and $v \in H^1(\Omega)$ we have

$$(\nabla u, \nabla v) = -(v, \Delta u) + \langle \gamma_0 v, \gamma_{\boldsymbol{n}} \nabla u \rangle_\Gamma.$$

For $\boldsymbol{u} \in H^2(\Omega)^d$ and $\boldsymbol{v} \in H^1(\Omega)^d$ we have

$$\int_\Omega \nabla \boldsymbol{u} : \nabla \boldsymbol{v} \, \mathrm{d}x = -(\boldsymbol{v}, \Delta \boldsymbol{u}) + \sum_{i=1}^d \langle \gamma_0 v_i, \gamma_{\boldsymbol{n}} \nabla u_i \rangle_\Gamma. \tag{3.6}$$

We will sometimes use the notation

$$\sum_{i=1}^d \langle \gamma_0 v_i, \gamma_{\boldsymbol{n}} \nabla u_i \rangle_\Gamma = \int_\Gamma \boldsymbol{v} \cdot \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} \, \mathrm{d}s,$$

where

$$\left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} \right]_i = \frac{\partial u_i}{\partial \boldsymbol{n}}. \tag{3.7}$$

The following two theorems characterize orthogonal decompositions of $L^2(\Omega)^d$. See [78, Theorem 1.4 p. 15] and [78, Theorem 1.5 p. 16].

Theorem 3.3 (Helmholtz Decomposition). *Let Ω be a bounded open set of $\mathbb{R}^d$ with Lipschitz continuous boundary Γ. Then*

$$L^2(\Omega)^d = H(\Omega) \oplus G(\Omega), \tag{3.8}$$

where $H(\Omega)$ and $G(\Omega)$ given by (3.2) and (3.3) respectively, are mutually orthogonal spaces.

Theorem 3.4 (Decomposition of $G(\Omega)$). *Let Ω be a bounded open set of $\mathbb{R}^d$ with Lipschitz continuous boundary Γ. Then*

$$G(\Omega) = G_1(\Omega) \oplus G_2(\Omega), \tag{3.9}$$

where $G_1(\Omega)$ and $G_2(\Omega)$ are mutually orthogonal spaces

$$G_1(\Omega) = \left\{ \boldsymbol{u} \in L^2(\Omega)^d : \boldsymbol{u} = \nabla q,\ q \in H^1(\Omega),\ \Delta q = 0 \right\} \tag{3.10}$$

$$G_2(\Omega) = \left\{ \boldsymbol{u} \in L^2(\Omega)^d : \boldsymbol{u} = \nabla q,\ q \in H_0^1(\Omega) \right\}. \tag{3.11}$$

We note that $G_1(\Omega)$ is the subspace of $L^2(\Omega)^d$ containing functions that are both divergence–free and curl–free. These important results will be addressed in Section 3.5.2. Similarly the space $H_0^1(\Omega)^d$ has the orthogonal decomposition [30, Theorem 3.8 p. 36]:

Theorem 3.5 (Decomposition of $H_0^1(\Omega)^d$). *Let Ω be a bounded open set of $\mathbb{R}^d$ with Lipschitz continuous boundary Γ. The space $H_0^1(\Omega)^d$ has the decomposition:*

$$H_0^1(\Omega)^d = V(\Omega) \oplus V^{\perp}(\Omega),$$

where

$$V^{\perp}(\Omega) = \left\{ \boldsymbol{v} \in H_0^1(\Omega) : -\Delta \boldsymbol{v} = \nabla q,\ q \in L^2(\Omega) \right\}.$$

We can easily verify that the spaces V and $V^{\perp}$ are orthogonal with respect to the bilinear form $a(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \nabla \boldsymbol{u} : \nabla \boldsymbol{v}\, \mathrm{d}x$, which induces the seminorm in $H_0^1(\Omega)^d$. By the Poincaré–Friedrich's inequality (see [30, Theorem 1.1 p. 3]), $|\cdot|_1$ is a norm equivalent to $\|\cdot\|_1$ on $H_0^1(\Omega)^d$. Let $\boldsymbol{u} \in V^{\perp}$ and $\boldsymbol{v} \in V$. By (3.6) and since $\boldsymbol{v} \in H_0^1(\Omega)^d$ we have

$$\begin{aligned} a(\boldsymbol{u}, \boldsymbol{v}) &= -\langle \Delta \boldsymbol{u}, \boldsymbol{v} \rangle, \\ &= \langle \nabla q, \boldsymbol{v} \rangle, \end{aligned}$$

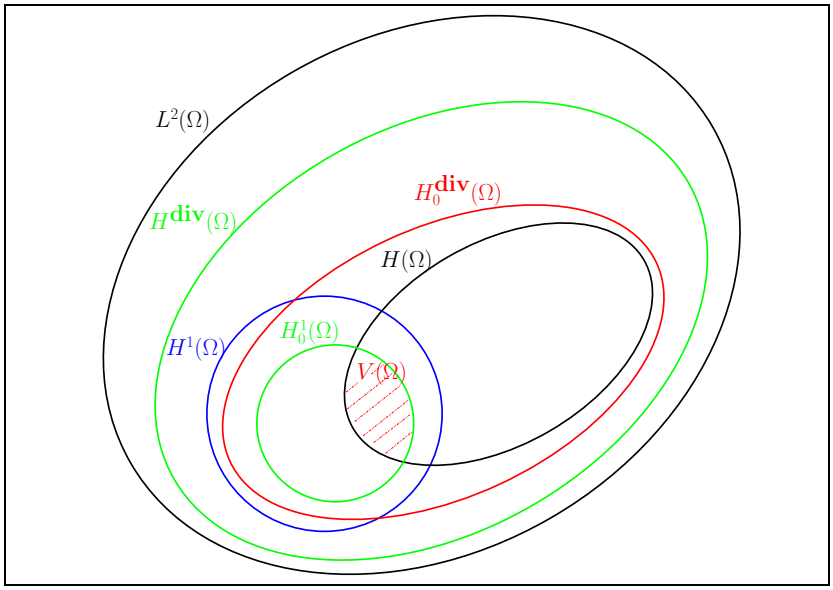

Figure 3.1: Venn diagram characterizing the Sobolev spaces $L^2(\Omega)$, $H^{\mathrm{div}}(\Omega)$, $H_0^{\mathrm{div}}(\Omega)$, $H^1(\Omega)$, $H_0^1(\Omega)$, $H(\Omega)$ and $V(\Omega)$.

for some $q \in L^2(\Omega)$,

$$\begin{aligned} &= \langle q, \nabla \cdot \boldsymbol{v} \rangle, \\ &= 0, \end{aligned}$$

which follows by the assumption $\nabla \cdot \boldsymbol{v} = 0$. This theorem will be addressed in Section 3.5.3.

We will introduce additional function spaces as needed. For a more comprehensive survey of Sobolev spaces we refer to [55, 64, 67, 78]. For a more elaborate discussion of the decompositions of $L^2(\Omega)^d$ we refer to [20, p. 312]. For a general background on functional analysis we refer to [49, 66]. We conclude this section with some examples.

Example 3.1 (Illustration of the common function spaces). Figure 3.1 shows a Venn diagram characterizing the spaces mentioned above. This diagram together with some properties of these spaces will be made clear in the following examples. □

Example 3.2 (The space $H^k(\Omega)$). Let $\Omega = (0,1)$ and $\varphi(x) = x^\alpha$ with $\alpha \in \mathbb{R}$. Since

$$\int \varphi(x)^2 \, \mathrm{d}x = \frac{x^{1+2\alpha}}{1+2\alpha},$$

we have $\varphi(x) \in L^2(\Omega)$ for $\alpha > -1/2$. The k^{th} derivative of $\varphi(x)$ is given by

$$\varphi^{(k)}(x) = x^{-k+\alpha} \prod_{i=0}^{k-1} (-i+\alpha). \tag{3.12}$$

Using (3.12) we can show that $\varphi(x) \in H^k(\Omega)$ for $\alpha > (2k-1)/2$. □

Example 3.3 ($H^1(\Omega)$ is a subset of $H^{\text{div}}(\Omega)$). Let $\Omega = (0,1) \times (0,1)$. To show that $H^1(\Omega) \subset H^{\text{div}}(\Omega)$, we let $\boldsymbol{u} \in H^1(\Omega)$, thus

$$\frac{\partial u_i}{\partial x_j} \in L^2(\Omega) \quad \text{for } i,j = 1,2. \tag{3.13}$$

In particular (3.13) holds for $i = j$. We use the triangle inequality:

$$\begin{aligned} \|\nabla \cdot \boldsymbol{u}\| &= \left\| \sum_{i=1}^{d} \frac{\partial u_i}{\partial x_i} \right\|, \\ &\leq \sum_{i=1}^{d} \left\| \frac{\partial u_i}{\partial x_i} \right\|. \end{aligned}$$

Thus $\boldsymbol{u} \in H^{\text{div}}(\Omega)$. Now to show $H^{\text{div}}(\Omega) \not\subset H^1(\Omega)$, let $\varphi(x)$ as in Example 3.2 with $-1/2 < \alpha < 1/2$, for this choice of α, $\varphi(x) \in L^2(0,1)$ but $\varphi(x) \notin H^1(0,1)$. Consider $\boldsymbol{u} = (u_1, u_2)^T = (x_1\varphi(x_2), x_2)^T$. We see that $\nabla \cdot \boldsymbol{u} = \varphi(x_2) + 1 \in L^2(\Omega)$, thus $\boldsymbol{u} \in H^{\text{div}}(\Omega)$. However, for $\boldsymbol{u} \in H^1(\Omega)$ we require that (3.13) holds. In particular we have

$$\frac{\partial u_1}{\partial x_2} = x_1 \varphi'(x_2),$$

with $\varphi'(x_2) \notin L^2(0,1)$. □

Example 3.4 (The space $H_0^k(\Omega)$). Let $\Omega = (0,1)$ and $\varphi(x) = x(x^\alpha - 1)$ with $\alpha \in \mathbb{R}$. We note that $\varphi(0) = \varphi(1) = 0$. Since

$$\int \varphi(x)^2 \, \mathrm{d}x = \frac{x^3}{3} - \frac{2x^{3+\alpha}}{3+\alpha} + \frac{x^{3+2\alpha}}{3+2\alpha},$$

we have $\varphi(x) \in L^2(\Omega)$ for $\alpha > -3/2$. The k^{th} derivative of $\varphi(x)$ is given by

$$\varphi^{(k)}(x) = (1+\alpha)x^{-k+\alpha+1} \prod_{i=0}^{k-2} (-i+\alpha). \tag{3.14}$$

Using (3.14) we can show that $\varphi(x) \in H_0^k(\Omega)$ for $\alpha > (2k-3)/2$. □

Example 3.5 (Decomposition of $L^2(\Omega)^2$.). Let $\Omega = (0,1) \times (0,1)$ and

$$\boldsymbol{u} = \begin{bmatrix} \sin \pi x_1 \cos \pi x_2 + \cos \pi x_1 \sin \pi x_2 + \cos \pi x_1 \sinh \pi x_2 \\ -\cos \pi x_1 \sin \pi x_2 + \sin \pi x_1 \cos \pi x_2 + \sin \pi x_1 \cosh \pi x_2 \end{bmatrix}.$$

$\boldsymbol{u}$ is clearly an element of $L^2(\Omega)^2$ and can be written as

$$\boldsymbol{u} = \boldsymbol{v} + \nabla q_1 + \nabla q_2,$$

with

$$\boldsymbol{v} = \nabla \times \frac{1}{\pi} \sin \pi x_1 \sin \pi x_2 = \begin{bmatrix} \sin \pi x_1 \cos \pi x_2 \\ -\cos \pi x_1 \sin \pi x_2 \end{bmatrix},$$

$$q_1 = \frac{1}{\pi} \sin \pi x_1 \sinh \pi x_2, \quad \nabla q_1 = \begin{bmatrix} \cos \pi x_1 \sinh \pi x_2 \\ \sin \pi x_1 \cosh \pi x_2 \end{bmatrix},$$

$$q_2 = \sin \pi x_1 \sin \pi x_2, \qquad \nabla q_2 = \begin{bmatrix} \pi \cos \pi x_1 \sin \pi x_2 \\ \pi \sin \pi x_1 \cos \pi x_2 \end{bmatrix}.$$

Since $\nabla \cdot \boldsymbol{v} = 0$, and $\boldsymbol{v} \cdot \boldsymbol{n}|_\Gamma = 0$, we have $\boldsymbol{v} \in H(\Omega)$. We have $q_1 \in H^1(\Omega)$ with $\Delta q_1 = 0$, hence $\nabla q_1 \in G_1(\Omega)$; and $q_2 \in H_0^1(\Omega)$, hence $\nabla q_2 \in G_2(\Omega)$. These functions are illustrated in Figure 3.2. Some remarks are in order. $\boldsymbol{u}$ is nonzero on Γ, in particular $u_1|_{x_2=0} \neq 0$, $u_2|_{x_2=0} \neq 0$, $u_2|_{x_1=0} \neq 0$ and $u_2|_{x_1=1} \neq 0$. The projection of $\boldsymbol{u}$ onto the divergence–free subspace of $L^2(\Omega)$ is given by $\boldsymbol{v} + \nabla q_1$. Figure 3.3 illustrates this decomposition. □

Example 3.6 (Relationship to the curl spaces.). In terms of the curl operator we can make the following identifications:

$$\begin{aligned} G(\Omega) &= \left\{ \boldsymbol{u} \in L^2(\Omega)^2 : \nabla \times \boldsymbol{u} = 0 \right\}, \\ G_2(\Omega) &= \left\{ \boldsymbol{u} \in G(\Omega) : \boldsymbol{u} \cdot \boldsymbol{\tau}|_\Gamma = 0 \right\}, \\ H(\Omega) &= \nabla \times H_0^1(\Omega), \\ J(\Omega) &= \nabla \times H^1(\Omega). \end{aligned}$$

Note that the space $G_2(\Omega)$ is composed of functions with zero tangential component on Γ. Figure 3.3 and Example 3.5 illustrate these identifications. We refer to [20, Chap. IX] for an overview of the curl spaces. □

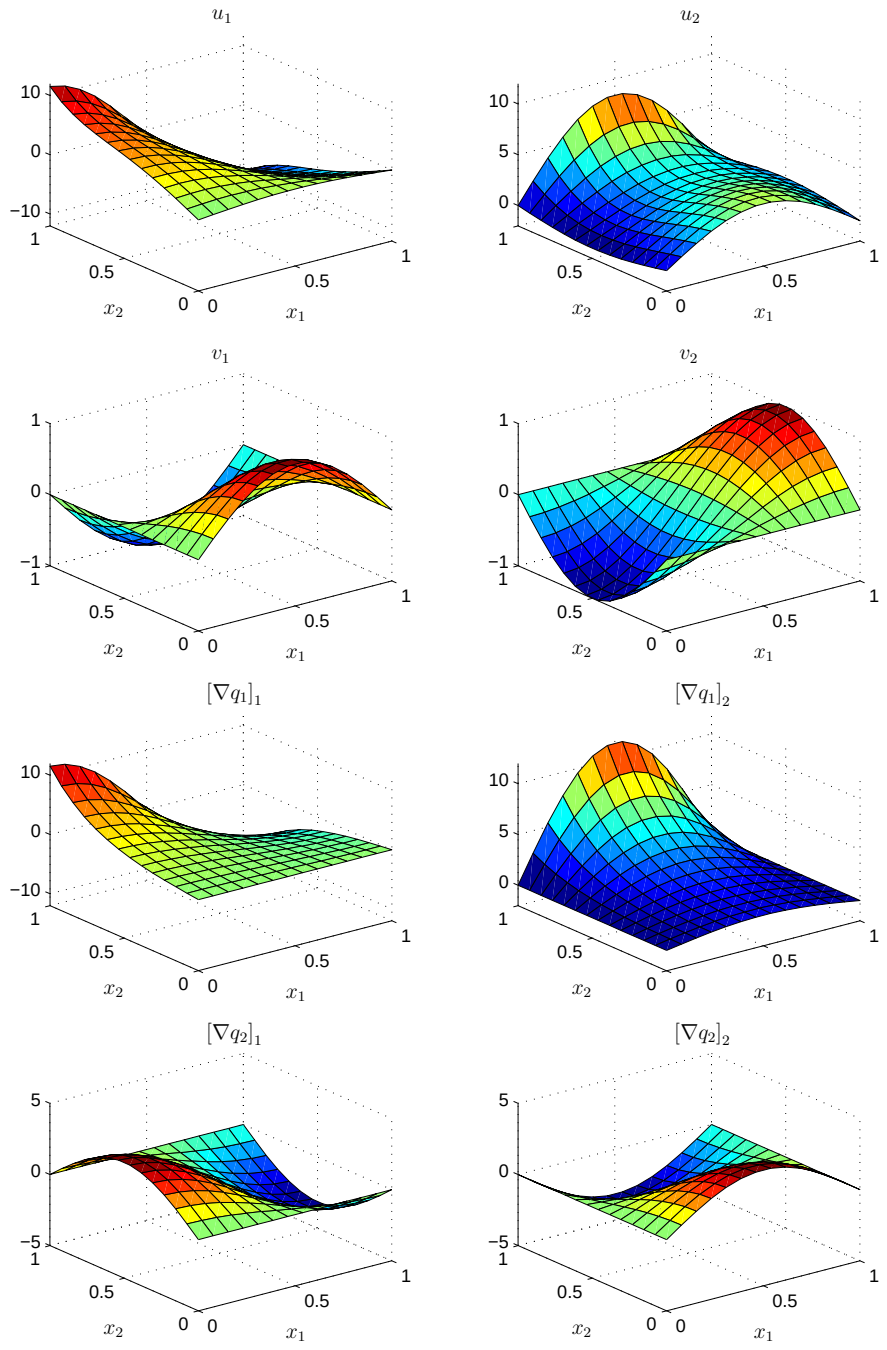

Figure 3.2: Decomposition of $\boldsymbol{u}$ in $L^2(\Omega)$; $\boldsymbol{u}$ given in Example 3.5.

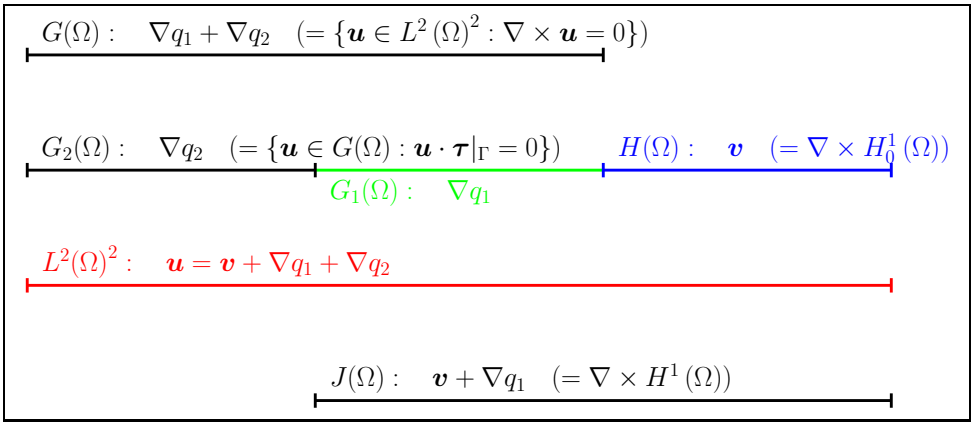

Figure 3.3: Illustration of the decompositions of $\boldsymbol{u}$ in $L^2(\Omega)$; $\boldsymbol{u}$ given in Example 3.5, and identification of curl–free spaces.

3.2 Navier–Stokes equations

Let Ω be an open, bounded domain in $\mathbb{R}^d$, $d = 2$ or 3 with a Lipschitz continuous boundary Γ. The Navier–Stokes equations for incompressible flow with homogeneous Dirichlet boundary condition are given by

$$\frac{\partial \boldsymbol{u}}{\partial t} = \nu \Delta \boldsymbol{u} - (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \boldsymbol{f} - \nabla p \quad \boldsymbol{x} \in \Omega, \tag{3.15}$$

$$\nabla \cdot \boldsymbol{u} = 0 \quad \boldsymbol{x} \in \Omega, \tag{3.16}$$

$$\boldsymbol{u} = 0 \quad \boldsymbol{x} \in \Gamma, \tag{3.17}$$

where $\boldsymbol{u} : \Omega \to \mathbb{R}^d$ is the velocity, $p : \Omega \to \mathbb{R}$ is the pressure, $\boldsymbol{f} : \Omega \to \mathbb{R}^d$ is the body force and $\nu > 0$ is the kinematic viscosity, a constant. Equation (3.15) is the momentum equation and the incompressibility of the fluid is enforced by the continuity equation (3.16). Since (3.15) only determines p to an additive constant, we require $p \in L_0^2(\Omega)$. Additionally (3.15)–(3.16) require an initial condition

$$\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x}) \quad \boldsymbol{x} \in \Omega, \tag{3.18}$$

and appropriate boundary conditions.

The most common boundary conditions for (3.15)–(3.16) are the *no–penetration* boundary condition

$$\boldsymbol{n} \cdot \boldsymbol{u} = 0 \quad \boldsymbol{x} \in \Gamma, \tag{3.19}$$

and the *no–slip* boundary condition

$$\boldsymbol{\tau} \cdot \boldsymbol{u} = 0 \quad \boldsymbol{x} \in \Gamma. \tag{3.20}$$

The no–penetration boundary condition is equivalent to no flow crossing the boundary. For viscous flow ($\nu > 0$) the no–slip boundary condition means that the fluid in direct contact with a solid boundary has the same velocity as the boundary itself. The homogeneous Dirichlet boundary condition (3.17) is a no–penetration, no–slip boundary condition. We could also impose a non–homogeneous Dirichlet boundary condition

$$\boldsymbol{u} = \boldsymbol{w} \quad \boldsymbol{x} \in \Gamma. \tag{3.21}$$

In this case $\boldsymbol{n} \cdot \boldsymbol{u} = w_{\boldsymbol{n}}$ and $\boldsymbol{\tau} \cdot \boldsymbol{u} = w_{\boldsymbol{\tau}}$ for $\boldsymbol{x} \in \Gamma$. Non–homogeneous Dirichlet boundary conditions coincide with *inflow* conditions.

The *outflow* or *non–friction* boundary condition is

$$\sum_{j=1}^{d} \left(-p\delta_{ij} + \nu \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right) \right) n_j = g_i \quad \boldsymbol{x} \in \Gamma \quad 1 < i \leq d, \tag{3.22}$$

where δ_{ij} is the Kronecker delta ($\delta_{ij} = 0$ for $i \neq j, \delta_{ii} = 1$).

We will often partition the boundary as $\Gamma = \Gamma_E \cup \Gamma_N$, with $\Gamma_E \cap \Gamma_N = \emptyset$ and refer to Γ_E as the portion of the boundary with essential boundary conditions and Γ_N as the portion of the boundary with natural boundary conditions. Γ_E will correspond to boundaries where penetration and slip conditions are specified and Γ_N will correspond to boundaries with an outflow condition. In the present section we shall only consider homogeneous Dirichlet boundary conditions.

We will be interested in several forms of (3.15)–(3.16) that are independent of t. When $\frac{\partial \boldsymbol{u}}{\partial t} = 0$ we have the steady Navier–Stokes equations:

$$-\nu \Delta \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla) \boldsymbol{u} + \nabla p = \boldsymbol{f} \quad \boldsymbol{x} \in \Omega, \tag{3.23}$$

$$\nabla \cdot \boldsymbol{u} = 0 \quad \boldsymbol{x} \in \Omega, \tag{3.24}$$

$$\boldsymbol{u} = 0 \quad \boldsymbol{x} \in \Gamma. \tag{3.25}$$

The steady Navier–Stokes equations with the nonlinear term $(\boldsymbol{u} \cdot \nabla) \boldsymbol{u}$ omitted are known as the steady Stokes equations:

$$-\nu \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f} \quad \boldsymbol{x} \in \Omega, \tag{3.26}$$

$$\nabla \cdot \boldsymbol{u} = 0 \quad \boldsymbol{x} \in \Omega, \tag{3.27}$$

$$\boldsymbol{u} = 0 \quad \boldsymbol{x} \in \Gamma. \tag{3.28}$$

Another form is known as Darcy–Stokes flow or the generalized Stokes equations given by

$$\alpha \boldsymbol{u} - \nu \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f} \quad \boldsymbol{x} \in \Omega, \tag{3.29}$$

$$\nabla \cdot \boldsymbol{u} = 0 \quad \boldsymbol{x} \in \Omega, \tag{3.30}$$

$$\boldsymbol{u} = 0 \quad \boldsymbol{x} \in \Gamma, \tag{3.31}$$

where $\alpha > 0$ is a constant. Equation (3.29)–(3.30) arises as part of the solution process of (3.15)–(3.16) using an implicit time integration scheme. In this case α plays the role of an inverse time step. The final form we will consider is the Darcy or Potential flow equations:

$$\boldsymbol{u} + \nabla p = \boldsymbol{f} \quad \boldsymbol{x} \in \Omega, \tag{3.32}$$

$$\nabla \cdot \boldsymbol{u} = 0 \quad \boldsymbol{x} \in \Omega, \tag{3.33}$$

$$\boldsymbol{n} \cdot \boldsymbol{u} = 0 \quad \boldsymbol{x} \in \Gamma. \tag{3.34}$$

The steady Stokes equations (3.26)–(3.27) and Darcy flow equations (3.32)–(3.33) play an important role in divergence–free projections that will be discussed in Section 3.5.

3.3 Weak formulation

Given Hilbert spaces V and Q, a bilinear form $b(\cdot,\cdot)$, is a mapping $V \times Q \to \mathbb{R}$ that is linear in each argument. Let $V = H_0^1(\Omega)^d$ and $Q = L_0^2(\Omega)$. We define the bilinear form $b(\cdot,\cdot) : V \times Q \to \mathbb{R}$ by

$$b(\boldsymbol{v}, q) = -\left(q, \nabla \cdot \boldsymbol{v}\right), \tag{3.35}$$

the bilinear form $a(\cdot,\cdot) : V \times V \to \mathbb{R}$ by

$$a(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \nabla \boldsymbol{u} : \nabla \boldsymbol{v} \, \mathrm{d}\boldsymbol{x}, \tag{3.36}$$

and the trilinear form $c(\cdot;\cdot,\cdot) : V \times V \times V \to \mathbb{R}$ by

$$c(\boldsymbol{w}; \boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \left(\left(\boldsymbol{w} \cdot \nabla\right) \boldsymbol{u}\right) \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} = \sum_{i,j=1}^{d} \left(w_j \frac{\partial u_i}{\partial x_j}, v_i \right).$$

The weak formulation of (3.15)–(3.17) can be stated as: given $\boldsymbol{u}_0 \in H^{\mathrm{div}}(\Omega)$ and $\boldsymbol{f} \in L^2\left(0,T;H^{\mathrm{div}}(\Omega)\right)$ find $\boldsymbol{u}(t) \in V$ and $p(t) \in Q$ such that for all $t \in (0,T)$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{u}(t),\boldsymbol{v}) + \nu a(\boldsymbol{u}(t),\boldsymbol{v}) + c(\boldsymbol{u}(t);\boldsymbol{u}(t),\boldsymbol{v}) & \\ +b(\boldsymbol{v},p(t)) = (\boldsymbol{f}(t),\boldsymbol{v}) \quad \forall \boldsymbol{v} \in V, & \quad (3.37)\\ b(\boldsymbol{u}(t),q) = 0 \qquad\quad \forall q \in Q, & \quad (3.38)\\ \boldsymbol{u}(0) = \boldsymbol{u}_0. & \quad (3.39)\end{aligned}$$

Existence of a solution to (3.15)–(3.17) for $d = 2,3$ is proved in [30, Theorem 1.4, p. 165]. Uniqueness of a solution to (3.15)–(3.17) for $d = 2$ is proved in [30, Theorem 1.5, p. 168], uniqueness for $d = 3$ remains unproven. For further results on existence and uniqueness we refer to [30, Chap. V] and [78, Chap. 3]. To study the existence and uniqueness of problem (3.26)–(3.27), we will make use of the Ladyzhenskaya–Babuška–Brezzi (LBB) or inf–sup condition. For simplicity we set $\nu = 1$, and write the weak form of (3.26)–(3.27) as: find $\boldsymbol{u} \in V$ and $p \in Q$ such that

$$\begin{aligned}a(\boldsymbol{u},\boldsymbol{v}) + b(\boldsymbol{v},p) = (\boldsymbol{f},\boldsymbol{v}) \quad \forall \boldsymbol{v} \in V, & \quad (3.40)\\ b(\boldsymbol{u},q) = 0 \qquad \forall q \in Q. & \quad (3.41)\end{aligned}$$

We introduce the space of *weakly divergence–free* or *solenoidal* functions

$$Z = \{\boldsymbol{v} \in V : b(\boldsymbol{v},q) = 0 \quad \forall q \in Q\}.$$

The next theorem [5, Theorem 4.3 p. 127] gives sufficient conditions to solve (3.40)–(3.41).

Theorem 3.6 (Ladyzhenskaya–Babuška–Brezzi). *Let $a(\cdot,\cdot)$ be continuous on $V \times V$:*

$$|a(\boldsymbol{u},\boldsymbol{v})| \le C_a \|\boldsymbol{u}\|_V \|\boldsymbol{v}\|_V \quad \forall \boldsymbol{u},\boldsymbol{v} \in V,$$

and coercive on $Z \times Z$:

$$a(\boldsymbol{u},\boldsymbol{u}) \ge \gamma_a \|\boldsymbol{u}\|_V^2 \qquad \forall \boldsymbol{u} \in Z,$$

let $b(\cdot,\cdot)$ be continuous on $V \times Q$:

$$|b(\boldsymbol{u},p)| \le C_b \|\boldsymbol{u}\|_V \|p\|_Q \quad \forall \boldsymbol{u} \in V, p \in Q,$$

and let $b(\cdot,\cdot)$ *satisfy the "LBB" or "inf–sup" condition*

$$\sup_{\boldsymbol{u}\in V} \frac{b(\boldsymbol{u},p)}{\|\boldsymbol{u}\|_V} \geq \gamma_b \|p\|_Q \qquad\qquad \forall p \in Q.$$

Then (3.40)–(3.41) *has a unique solution.*

We remark that Theorem 3.6 assumes and "Z–coercivity" of the bilinear form $a(\cdot,\cdot)$, thus Theorem 3.6 can be applied to study stability of the weak forms of (3.29)–(3.30) or (3.32)–(3.33). In particular in Section 5.2 we examine the consequences when coercivity does not hold for equations of the form (3.32)–(3.33).

3.4 Finite element discretization

For simplicity, let Ω be a polygonal domain that can be partitioned into triangles or quadrilaterals. Let $\mathcal{T}_h = \{T_1, T_2, \ldots, T_M\}$ be a partition of Ω. Each T_i is called a *finite element*. We define the *triangulation* of Ω by $\mathcal{T}_h$ if the following properties hold:

1. $\bar{\Omega} = \bigcup_{i=1}^{M} T_i$.
2. If $T_i \bigcap T_j$ consists of exactly one point, then it is a common vertex of T_i and T_j.
3. If for $i \neq j$, $T_i \bigcap T_j$ consists of more than one point, then $T_i \bigcap T_j$ is a common edge of T_i and T_j.

The parameter h is associated with the size of the finite elements. If all the finite elements are the same size, then $\mathcal{T}_h$ is a *uniform* triangulation. A family of triangulations $\{\mathcal{T}_h\}$ is called *regular* if there exists $\sigma > 0$ such that every T in $\mathcal{T}_h$ contains a circle of diameter ρ_T with

$$\frac{h_T}{\rho_T} \leq \sigma,$$

where h_T is the diameter of T.

Let $\mathcal{T}_h$ be a triangulation of Ω. We introduce finite element approximations V^h of V and Q^h of Q, based on the triangulation $\mathcal{T}_h$. The Navier–Stokes

problem (3.37)–(3.39) becomes: find $\boldsymbol{u}^h(t) \in V^h$ and $p^h(t) \in Q^h$ such that for all $t \in (0,T)$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{u}^h(t),\boldsymbol{v}^h\right) + \nu a\big(\boldsymbol{u}^h(t),\boldsymbol{v}^h\big) + c\big(\boldsymbol{u}^h(t);\boldsymbol{u}^h(t),\boldsymbol{v}^h\big) + b\big(\boldsymbol{v}^h,p^h(t)\big) = \big(\boldsymbol{f}(t),\boldsymbol{v}^h\big) \quad \forall \boldsymbol{v}^h \in V^h, \tag{3.42}$$

$$b\big(\boldsymbol{u}^h(t),q^h\big) = 0 \qquad \forall q^h \in Q^h, \tag{3.43}$$

$$\boldsymbol{u}(0) = \boldsymbol{u}_0. \tag{3.44}$$

Let $\{\boldsymbol{\varphi}_k\}_{k=1}^K$ denote the basis of V^h and $\{\psi_j\}_{j=1}^J$ denote the basis of Q^h. Then $\boldsymbol{u}^h(t)$ can be written as

$$\boldsymbol{u}^h(t) = \sum_{k=1}^{K} \mathrm{u}_k(t)\boldsymbol{\varphi}_k,$$

and p^h can be written as

$$p^h(t) = \sum_{j=1}^{J} \mathrm{p}_j(t)\psi_j.$$

The algebraic or discrete form of (3.42)–(3.44) is

$$M\frac{\mathrm{d}\mathbf{u}(t)}{\mathrm{d}t} + K\mathbf{u}(t) + N\left(\mathbf{u}(t)\right)\mathbf{u}(t) + C\mathbf{p}(t) = \mathbf{f}(t), \tag{3.45}$$

$$C^T\mathbf{u}(t) = 0, \tag{3.46}$$

$$\mathbf{u}(0) = \mathbf{u}_0, \tag{3.47}$$

where we make the following definitions:

$$M_{ij} = \left(\boldsymbol{\varphi}_i,\boldsymbol{\varphi}_j\right), \tag{3.48}$$

$$K_{ij} = a\big(\boldsymbol{\varphi}_i,\boldsymbol{\varphi}_j\big), \tag{3.49}$$

$$[N(\mathbf{w})]_{ij} = \sum_{k=1}^{K} \mathrm{w}_k\, c\big(\boldsymbol{\varphi}_k;\boldsymbol{\varphi}_j,\boldsymbol{\varphi}_i\big), \tag{3.50}$$

$$C_{ij} = b\big(\boldsymbol{\varphi}_j,\psi_i\big), \tag{3.51}$$

$$\left[\mathbf{f}(t)\right]_i = \left(\boldsymbol{f}(t),\boldsymbol{\varphi}_i\right), \tag{3.52}$$

$$\left[\mathbf{u}(t)\right]_i = \mathrm{u}_i, \tag{3.53}$$

and

$$[\mathbf{p}(t)]_i = \mathrm{p}_i. \tag{3.54}$$

The matrix M is the mass matrix, K is the stiffness or diffusion matrix, $N(\mathbf{w})$ is the advection matrix, C is the gradient matrix, C^T is the divergence matrix, $\mathbf{f}(t)$ is the load vector and $\mathbf{u}(t)$ and $\mathbf{p}(t)$ are the vectors containing the nodal coefficients for velocity and pressure, respectively.

We define the following polynomial spaces:

$$\begin{aligned} Q_k &= \{\text{polynomials of degree } \leq \mathrm{k} \text{ in each coordinate}\}, \\ \mathbb{Q}_k &= [Q_k]^n = \{\text{n-vectors whose components belong to } Q_k\}. \end{aligned}$$

Let the triangulation $\mathcal{T}_h$ be a partition of Ω into a quadrilateral mesh if $d = 2$ or a hexahedral mesh if $d = 3$. For the $\mathbb{Q}_m Q_n$ element we approximate the velocity and the pressure with polynomials from $\mathbb{Q}_m$ and Q_n on each quadrilateral or hexahedral, such that the approximations are continuous on Ω. The following Lemma characterizes the asymptotic convergence rate in the space Q_k [64, p. 96].

Lemma 3.1. *Let $\mathcal{T}_h$ be a regular triangulation of Ω. Let the number s satisfy $1 \leq s \leq k$ and $m = 0, 1$. Then there exists an element $v^h \in Q_k$ such that*

$$\|v - v^h\|_m \leq C h^{s+1-m} \|v\|_{s+1} \quad \forall v \in H^s(\Omega), \tag{3.55}$$

where C is a positive constant independent of h and v.

We provide a stability result that is a discrete analog of Theorem 3.6. We introduce the space of *weakly discretely divergence–free* functions

$$Z^h(\Omega) = \left\{ \boldsymbol{u}^h \in V^h : b\big(\boldsymbol{u}^h, q^h\big) = 0 \ \forall q_h \in Q^h \right\}. \tag{3.56}$$

The space $Z^h(\Omega)$ corresponds to $\mathcal{N}\left(C^T\right)$. Problem (3.40)–(3.41) is approximated by: find $\boldsymbol{u}^h \in V^h$ and $p^h \in Q^h$ such that

$$a\big(\boldsymbol{u}^h, \boldsymbol{v}^h\big) + b\big(\boldsymbol{v}^h, p^h\big) = \big(\boldsymbol{f}, \boldsymbol{v}^h\big) \quad \forall \boldsymbol{v}^h \in V^h, \tag{3.57}$$

$$b\big(\boldsymbol{u}^h, q^h\big) = 0 \qquad \forall q^h \in Q^h. \tag{3.58}$$

As a consequence, if $(\boldsymbol{u}^h, p^h)$ solves (3.57)–(3.58), then $\boldsymbol{u}^h \in Z^h$ and $\boldsymbol{u}^h$ solves the following: find $\boldsymbol{u}^h \in Z^h$ such that

$$a\big(\boldsymbol{u}^h, \boldsymbol{v}^h\big) = \big(\boldsymbol{f}, \boldsymbol{v}^h\big) \quad \forall \boldsymbol{v}^h \in Z^h. \tag{3.59}$$

The following Theorem provides error estimates for a mixed approximation. [39, pp. 10–16]

Theorem 3.7. *Let $V^h \subset V$ and $Q^h \subset Q$.*

1. *Assume that*

   (a) Z^h is nonempty.

   (b) $a(\boldsymbol{u}^h, \boldsymbol{u}^h) \geq \gamma_a^h \|\boldsymbol{u}^h\|_V^2 \quad \forall \boldsymbol{u}^h \in Z^h$.

   Then (3.59) has a unique solution $\boldsymbol{u}^h \in Z^h$ and

$$\|\boldsymbol{u} - \boldsymbol{u}^h\|_V \leq C_1 \inf_{\boldsymbol{v}^h \in V^h} \|\boldsymbol{u} - \boldsymbol{v}\|_V + C_2 \inf_{q^h \in Q^h} \|p - q^h\|_Q,$$

   where

$$C_1 = 1 + \frac{C_a}{\gamma_a^h} \quad \textit{and} \quad 1 + \frac{C_b}{\gamma_a^h}.$$

2. *Assume in addition, there exists $\gamma_b^h > 0$ such that*

$$\sup_{\boldsymbol{u}^h \in V^h} \frac{b(\boldsymbol{u}^h, p^h)}{\|\boldsymbol{u}^h\|_V} \geq \gamma_b^h \|p^h\|_Q \quad \forall p^h \in Q.$$

   Then there exists a unique $p^h \in Q^h$ such that $(\boldsymbol{u}^h, p^h)$ solves (3.57)–(3.58) and we have the error bounds

$$\|\boldsymbol{u} - \boldsymbol{u}^h\|_V \leq K_{11} \inf_{\boldsymbol{v}^h \in V^h} \|\boldsymbol{u} - \boldsymbol{v}\|_V + K_{12} \inf_{q^h \in Q^h} \|p - q^h\|_Q, \tag{3.60}$$

$$\|p - p^h\|_V \leq K_{21} \inf_{\boldsymbol{v}^h \in V^h} \|\boldsymbol{u} - \boldsymbol{v}\|_V + K_{22} \inf_{q^h \in Q^h} \|p - q^h\|_Q, \tag{3.61}$$

   where

$$K_{11} = 1 + \frac{C_a}{\gamma_a^h} + \frac{C_a C_b}{\gamma_a^h \gamma_b^h}, \quad K_{12} = \frac{C_b}{\gamma_a^h} \delta(Z, Z^h),$$

$$K_{21} = \frac{C_b}{\gamma_b^h}, \quad K_{22} = 1 + \frac{C_b}{\gamma_b^h} \quad \textit{and} \quad \delta(Z, Z^h) = \sup_{\substack{\boldsymbol{z}^h \in Z^h \\ \|\boldsymbol{z}^h\|_V = 1}} \inf_{\boldsymbol{z} \in Z} \|\boldsymbol{z} - \boldsymbol{z}^h\|_V.$$

Note here that if $Z^h \subset Z$, then $\delta(Z, Z^h) = 0$ and the error estimate for the velocity decouples from the pressure error; however the error estimate for the pressure is always coupled to the velocity error. The error bounds in (3.60) and (3.61) contain terms that involve different norms. These terms are usually equilibrated by choosing polynomials for the velocity approximation $\boldsymbol{u}^h$ a degree higher than the polynomials for the pressure approximation p^h.

The present discretization will utilize the quadrilateral or hexahedral Taylor–Hood element $\mathbb{Q}_2Q_1$. The finite element spaces are given by

$$V^h = \{\boldsymbol{v}^h \in \mathbb{Q}_2 : \boldsymbol{v}^h = 0 \text{ on } \Gamma\}$$

and

$$Q^h = \{q^h \in Q_1 : \int_\Omega q^h \, \mathrm{d}\boldsymbol{x} = 0\}.$$

The $\mathbb{Q}_2Q_1$ element consists of piecewise continuous bi-quadratic polynomial approximation for the velocity space and piecewise continuous bilinear polynomials for the pressure space. The $\mathbb{Q}_2Q_1$ element is a conforming element of $V^h \times Q^h$. By Lemma 3.1, we have error estimates given by

$$\inf_{\boldsymbol{v}^h \in V^h} \|\boldsymbol{u} - \boldsymbol{v}^h\|_1 \leq c_1 h^3,$$

and

$$\inf_{q^h \in Q^h} \|p - q^h\|_0 \leq c_1 h^2,$$

the error bounds (5.26) and (5.27) are balanced. The $\mathbb{Q}_2Q_1$ element satisfies the condition 2 of Theorem 3.7 or the inf-sup condition [39, pp. 31–34]. The $\mathbb{Q}_2Q_1$ element provides a stable spatial discretization for the Navier–Stokes equations (3.15)–(3.18) supplemented with boundary conditions (3.21) and (3.22) see [64, pp. 310–311]. In addition to the references above we refer to [30, 62, 78] for discussion of finite element methods for the incompressible Navier–Stokes equations.

3.5 Divergence–free projections

In the present section we discuss divergence–free projections. We provide an abstract framework that will allow us to characterize the divergence–free L^2 and H^1 projections. Section 3.5.2 discusses the L^2 projection from $L^2(\Omega)^d$ onto the divergence–free subspace of $L^2(\Omega)^d$, and issues of practical implementation. Section 3.5.3 discusses the H^1 projection from $H^1(\Omega)^d$ onto the divergence–free subspace of $H^1(\Omega)^d$.

3.5.1 Abstract framework for projections

We now construct an abstract framework that will be useful in our study of divergence–free projections. Suppose X and Y are Hilbert spaces. Assume

the bilinear forms $s(\boldsymbol{u}, \boldsymbol{v})$ and $r(\boldsymbol{v}, p)$ are given so that $s(\cdot, \cdot)$ is continuous on $X \times X$:

$$|s(\boldsymbol{u}, \boldsymbol{v})| \leq C_s \|\boldsymbol{u}\|_X \|\boldsymbol{v}\|_X \quad \forall \boldsymbol{u}, \boldsymbol{v} \in X,$$

$r(\cdot, \cdot)$ is continuous on $X \times Y$:

$$|r(\boldsymbol{u}, p)| \leq C_r \|\boldsymbol{u}\|_X \|p\|_Y \quad \forall \boldsymbol{u} \in X, p \in Y,$$

and an "inf–sup" condition holds:

$$\sup_{\boldsymbol{u} \in X} \frac{r(\boldsymbol{u}, p)}{\|\boldsymbol{u}\|_X} \geq \gamma_r \|p\|_Y \quad \forall p \in Y.$$

Define the kernel of r as

$$Z_r = \{\boldsymbol{v} \in X : r(\boldsymbol{v}, q) = 0 \quad \forall q \in Y\}.$$

with a cokernel defined by

$$Z_r^{\perp} = \{\mathbf{w} \in X : s(\mathbf{w}, \boldsymbol{v}) + r(\boldsymbol{v}, p) = 0 \quad \forall \boldsymbol{v} \in X \text{ and some } p \in Y\}.$$

Suppose further that s is coercive on $Z_r \times Z_r$:

$$s(\boldsymbol{u}, \boldsymbol{u}) \geq \gamma_s \|\boldsymbol{u}\|_X^2 \quad \forall \boldsymbol{u} \in Z_r.$$

For any $\mathbf{x} \in X$, there exist unique $\boldsymbol{u} \in X$ and $p \in Y$ such that

$$s(\boldsymbol{u}, \boldsymbol{v}) + r(\boldsymbol{v}, p) = s(\mathbf{x}, \boldsymbol{v}) \quad \forall \boldsymbol{v} \in X, \tag{3.62}$$

$$r(\boldsymbol{u}, q) = 0 \qquad \forall q \in Y. \tag{3.63}$$

Existence and uniqueness follows from Theorem 3.6. This also compels the conclusion that Z_r and $Z_r^{\perp}$ are complementary subspaces in X: $Z_r \cap Z_r^{\perp} = \{0\}$ and $X = Z_r \oplus Z_r^{\perp}$. Note that (3.62) implies that

$$s(\boldsymbol{u} - \mathbf{x}, \boldsymbol{v}) + r(\boldsymbol{v}, p) = 0 \quad \forall \boldsymbol{v} \in X,$$

so $\boldsymbol{u} - \mathbf{x} \in Z_r^{\perp}$ and $\boldsymbol{u} \in Z_r$ solves the best approximation problem

$$s(\boldsymbol{u} - \mathbf{x}, \boldsymbol{u} - \mathbf{x}) = \min_{\boldsymbol{w} \in Z_r} s(\boldsymbol{w} - \mathbf{x}, \boldsymbol{w} - \mathbf{x}).$$

The implied mapping $P : \mathbf{x} \mapsto \boldsymbol{u}$ that (3.62) determines is a (oblique) projection of X onto Z_r along the cokernel $Z_r^{\perp}$. Indeed, consider $\hat{\boldsymbol{u}} = P^2\mathbf{x}$ as the solution to

$$s(\hat{\boldsymbol{u}}, \boldsymbol{v}) + r(\boldsymbol{v}, \hat{p}) = s(P\mathbf{x}, \boldsymbol{v}) \quad \forall \boldsymbol{v} \in X, \tag{3.64}$$

$$r(\hat{\boldsymbol{u}}, q) = 0 \qquad \forall q \in Y. \tag{3.65}$$

Since $s(P\mathbf{x}, \boldsymbol{v}) + r(\boldsymbol{v}, p) = s(\mathbf{x}, \boldsymbol{v}) \quad \forall \boldsymbol{v} \in X$, we see that $s(\hat{\boldsymbol{u}}, \boldsymbol{v}) + r(\boldsymbol{v}, \hat{p} + p) = s(\mathbf{x}, \boldsymbol{v}) \quad \forall\, \boldsymbol{v} \in X$ which implies that $\hat{\boldsymbol{u}} = P\mathbf{x}$ and so $P^2 = P$.

3.5.2 Divergence–free L^2 projection

In this section we utilize the abstract formulation developed in Section 3.5.1 to discuss the projection of $L^2(\Omega)^d$ onto the divergence–free subspace $H(\Omega)$ given by Theorem 3.3. For the divergence–free L^2 projection we set

$$s(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u}, \boldsymbol{v}), \tag{3.66}$$
$$r(\boldsymbol{u}, p) = b(\boldsymbol{u}, p), \tag{3.67}$$

$X = H^{\mathrm{div}}(\Omega)$, $Y = L^2(\Omega)$ and the bilinear form $b(\cdot, \cdot)$ given by (3.35). Thus we state the weak form of the divergence–free L^2 projection as follows: given $\boldsymbol{u}_0 \in L^2(\Omega)$, find $\boldsymbol{u} \in H^{\mathrm{div}}(\Omega)$ and $p \in L^2(\Omega)$ such that

$$(\boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{v}, p) = (\boldsymbol{u}_0, \boldsymbol{v}) \quad \forall \boldsymbol{v} \in H^{\mathrm{div}}(\Omega), \tag{3.68}$$
$$b(\boldsymbol{u}, q) = 0 \qquad \forall q \in L^2(\Omega). \tag{3.69}$$

The forms (3.66) and (3.67) are continuous. We set

$$Z_r = \{\boldsymbol{v} \in H^{\mathrm{div}}(\Omega) : b(\boldsymbol{v}, q) = 0 \quad \forall q \in L^2(\Omega)\}.$$

Since $\nabla \cdot \boldsymbol{u} = 0$ on Z_r, we have

$$(\boldsymbol{u}, \boldsymbol{u}) = \|\boldsymbol{u}\|^2 = \|\boldsymbol{u}\|^2 + \|\nabla \cdot \boldsymbol{u}\|^2 = \|\boldsymbol{u}\|^2_{\mathrm{div}},$$

which establishes coercivity on Z_r. Satisfaction of the "inf–sup" condition is a consequence of standard results for the mixed formulation of the Poisson equation [5, p. 133]. Thus by Theorem 3.6, (3.68)–(3.69) has a unique solution $\boldsymbol{u} = P_0 \boldsymbol{u}_0$, where we denote the divergence–free L^2 projection operator by P_0. Equations (3.68)–(3.69) constitute a Darcy flow problem, hence we will often refer to (3.68)–(3.69) as the Darcy flow formulation of the divergence–free L^2 projection.

3.5.3 Divergence–free H^1 projection

We proceed as in Section 3.5.2 and utilize the abstract framework and set

$$s(\boldsymbol{u}, \boldsymbol{v}) = a(\boldsymbol{u}, \boldsymbol{v})$$
$$r(\boldsymbol{u}, p) = b(\boldsymbol{u}, p),$$

$X = H_0^1(\Omega)^d$, $Y = L_0^2(\Omega)$ and the bilinear form $a(\cdot,\cdot)$ given by (3.36). We state the divergence–free H^1 projection problem as: given $\boldsymbol{u}_0 \in H^1(\Omega)$, find $\boldsymbol{u} \in H_0^1(\Omega)$ and $p \in L_0^2(\Omega)$ such that

$$a(\boldsymbol{u},\boldsymbol{v}) + b(\boldsymbol{v},p) = a(\boldsymbol{u}_0,\boldsymbol{v}) \quad \forall \boldsymbol{v} \in H_0^1(\Omega), \tag{3.70}$$

$$b(\boldsymbol{u},q) = 0 \quad \forall q \in L_0^2(\Omega). \tag{3.71}$$

We remark that (3.70)–(3.71) corresponds to a Stokes problem with forcing term $-\Delta\boldsymbol{u}_0$. The weak formulation (3.70)–(3.71) satisfies Theorem 3.6. This follows from standard results for the Stokes problem (see [5, pp. 142–146]). Thus (3.70)–(3.71) has a unique solution $\boldsymbol{u} = P_1\boldsymbol{u}_0$.

The divergence–free H^1 projection is equivalent to a Stokes problem, which can be discretized using conforming finite elements that are stable in the sense of Theorem 3.7. In addition this projection allows complete specification of Dirichlet or Neumann boundary conditions.

Chapter 4

Projection based schemes

In this chapter we combine the building blocks to construct the Krylov–based exponential projection method (KBEI projection method). Prior to constructing the method, we provide a brief overview of fractional step projection methods. These methods constitute a class of methods in which a divergence–free L^2 projection is utilized at an intermediate step to produce a divergence–free velocity. The fractional step projection methods are the most typically employed methods for time integration of the incompressible Navier–Stokes equations. We provide a formal and non-rigorous overview of the method due to Chorin [15] in order to contrast this method from the KBEI projection method and to provide an insight on the convergence properties of Chorin's method.

4.1 Semi–implicit projection methods

The semi–implicit projection methods utilize a divergence–free L^2 projection to decouple the diffusive and convective terms. The diffusive terms are treated implicitly to avoid stability restrictions and the convective terms are treated explicitly. The advantages of these methods are the unconditional stability of the diffusion and the decoupling. One disadvantage is that the decoupling is attained at the expense of a projection step, which introduces a spatial error because the projection in general does not recover the tangential component of the boundary conditions. Another disadvantage is that the decoupling introduces a temporal error of at least second order. These methods are sometimes referred to as Chorin projection methods, fractional

step methods, pressure correction or operator splitting methods.

In this section we provide an overview of projection schemes to contrast these schemes with the KBEI projection method. These methods were first proposed by Chorin [15, 16]. Rannacher [65] provided a thorough analysis of the convergence in time of the original method of Chorin and Temam. In practice the fractional step projection method is a time integration method that can be combined with any spatial discretization method. The method and its variants have employed in a finite difference setting in [4]; finite element methods in [1, 33, 34, 36, 37, 53]; and spectral methods in [50]. Van Kan [83] provides a rigorous analysis of the method of [46] and presents a variant that incorporates a pressure correction. The method presented in [4] utilizes an idea similar to [83] to construct a second order method. A recurring difficulty encountered in projection–based methods is the proper choice of boundary conditions for the velocity and/or pressure at the intermediate time steps. Suitable choices of proper boundary conditions are discussed in [12, 19, 23]. Kim and Moin [46] present a second order accurate scheme based on Crank Nicolson and Adams Bashforth schemes and an appropriate boundary condition on the intermediate velocity. Discussion regarding the proper treatment of the intermediate pressure can be found in [17, 56]. In addition to the above references and the references therein we refer to sequence of papers by E and Liu [21, 22, 24] and to [18, 35, 64, 65, 80, 81] for an overview of fractional step projection schemes.

For our analysis we rewrite the Navier–Stokes equations (3.15)–(3.17) as

$$\frac{\partial \boldsymbol{u}}{\partial t} = \nu \Delta \boldsymbol{u} - (\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} + \boldsymbol{f} - \nabla p \quad \boldsymbol{x} \in \Omega, \tag{4.1}$$

$$\equiv \boldsymbol{F}(\boldsymbol{u}) - \nabla p \qquad \boldsymbol{x} \in \Omega, \tag{4.2}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \boldsymbol{x} \in \Omega, \tag{4.3}$$

$$\boldsymbol{u} = 0 \qquad \boldsymbol{x} \in \Gamma. \tag{4.4}$$

We refer to (4.2) as the momentum equation and (4.3) as the continuity equation. We now formally describe fractional step methods. We apply the continuous divergence–free L^2 projection P_0 to (4.2)

$$\frac{\partial \boldsymbol{u}}{\partial t} = P_0 \boldsymbol{F}(\boldsymbol{u}) \quad \boldsymbol{x} \in \Omega, \tag{4.5}$$

$$\boldsymbol{n} \cdot \boldsymbol{u} = 0 \qquad \boldsymbol{x} \in \Gamma. \tag{4.6}$$

By (4.2) and (4.5) we have

$$\frac{\partial \boldsymbol{u}}{\partial t} = P_0 \boldsymbol{F}(\boldsymbol{u}) = \boldsymbol{F}(\boldsymbol{u}) - \nabla p \quad \boldsymbol{x} \in \Omega, \tag{4.7}$$

$$\boldsymbol{n} \cdot \boldsymbol{u} = 0 \qquad \boldsymbol{x} \in \Gamma. \tag{4.8}$$

Generally, the semi-implicit projection proceeds as follows: given $\boldsymbol{u}_0$ satisfying (4.3) at the current time step, approximate ∇p at the current time step and insert in (4.7), and integrate (4.7) over one time step Δt to an intermediate velocity $\tilde{\boldsymbol{u}}$. We note that the projection in (4.7) is not actually applied. The projection is approximated by approximating ∇p. Due to this approximation, the right hand side of (4.7) will not be divergence–free, hence the intermediate velocity $\tilde{\boldsymbol{u}}$ will in general not be divergence–free. The velocity at the end of the time step, $\boldsymbol{u}_1$, is attained by $\boldsymbol{u}_1 = P_0 \tilde{\boldsymbol{u}}$.

This scheme attempts to subtract the curl–free portion, ∇p to obtain the divergence–free portion rather that applying the projection directly to $\boldsymbol{F}(\boldsymbol{u}) - \nabla p$. An alternative would be, given $\boldsymbol{u}_0$ with $\nabla \cdot \boldsymbol{u}_0 = 0$, to apply the projection directly to $\boldsymbol{F}(\boldsymbol{u}) - \nabla p$, which would remove ∇p from the momentum equation. Then (4.5) is integrated one time step to a divergence–free velocity $\boldsymbol{u}_1$. A scheme of this type was proposed in [25], and implemented with a spectral method for spatial discretization.

The original Chorin scheme also referred to as Projection 1 in [35] is defined as follows:

1. Given $\boldsymbol{u}$ with $\nabla \cdot \boldsymbol{u} = 0$ and p find $\tilde{\boldsymbol{u}}(\Delta t)$, with $\tilde{\boldsymbol{u}}(0) = \boldsymbol{u}(0)$ from

$$\frac{\partial \tilde{\boldsymbol{u}}}{\partial t} = \boldsymbol{F}(\tilde{\boldsymbol{u}}) \quad \boldsymbol{x} \in \Omega, \tag{4.9}$$

$$\tilde{\boldsymbol{u}} = t \nabla p \qquad \boldsymbol{x} \in \Gamma. \tag{4.10}$$

2. Find ϕ and $\boldsymbol{u}$ from

$$\Delta \phi = \nabla \cdot \tilde{\boldsymbol{u}}(\Delta t) \qquad \boldsymbol{x} \in \Omega, \tag{4.11}$$

$$\frac{\partial \phi}{\partial \boldsymbol{n}} = \Delta t \frac{\partial p}{\partial \boldsymbol{n}} \qquad \boldsymbol{x} \in \Gamma, \tag{4.12}$$

$$\boldsymbol{u} = \tilde{\boldsymbol{u}}(\Delta t) - \nabla \phi \quad \boldsymbol{x} \in \Omega. \tag{4.13}$$

3. Find p from

$$\Delta p = \nabla \cdot \boldsymbol{F}(\boldsymbol{u}) \quad \boldsymbol{x} \in \Omega, \tag{4.14}$$

$$\frac{\partial p}{\partial \boldsymbol{n}} = \boldsymbol{n} \cdot \boldsymbol{F}(\boldsymbol{u}) \quad \boldsymbol{x} \in \Gamma. \tag{4.15}$$

Numerical experiences [35] suggest this method to be first order accurate in time. We utilize a formal Taylor series analysis to show that this method is first order accurate in time for $\boldsymbol{x} \in \Omega$, and to explain the choice of the boundary conditions (4.10), (4.12) and (4.15). By the Taylor series expansion of $\boldsymbol{u}(\Delta t)$ about $\Delta t = 0$ we have

$$\boldsymbol{u}(\Delta t) = \boldsymbol{u}(0) + \Delta t \dot{\boldsymbol{u}}(0) + \mathcal{O}\left(\Delta t^2\right), \tag{4.16}$$

$$= \boldsymbol{u}(0) + \Delta t \left(\boldsymbol{F}(\boldsymbol{u}(0)) - \nabla p(0)\right) + \mathcal{O}\left(\Delta t^2\right). \tag{4.17}$$

Where we have used (4.2) in the second term of (4.17). Similarly the Taylor series expansion of $\tilde{\boldsymbol{u}}(\Delta t)$ about $\Delta t = 0$ is given by

$$\tilde{\boldsymbol{u}}(\Delta t) = \tilde{\boldsymbol{u}}(0) + \Delta t \dot{\tilde{\boldsymbol{u}}}(0) + \mathcal{O}\left(\Delta t^2\right). \tag{4.18}$$

Since $\tilde{\boldsymbol{u}}(0) = \boldsymbol{u}(0)$ and $\boldsymbol{F}(\boldsymbol{u}(0)) = \boldsymbol{F}(\tilde{\boldsymbol{u}}(0))$ we have

$$\tilde{\boldsymbol{u}}(t) = \boldsymbol{u}(0) + t\boldsymbol{F}(\boldsymbol{u}(0)) + \mathcal{O}\left(t^2\right). \tag{4.19}$$

We subtract (4.18) from (4.19) to obtain

$$\tilde{\boldsymbol{u}}(t) - \boldsymbol{u}(t) = -t\nabla p + \mathcal{O}\left(t^2\right), \tag{4.20}$$

$$= \mathcal{O}(t). \tag{4.21}$$

Thus the final velocity $\boldsymbol{u}(t)$ is first order accurate in time for $\boldsymbol{x} \in \Omega$. To obtain the boundary condition (4.10), we recall that we have specified homogeneous Dirichlet boundary conditions on $\boldsymbol{u}$. Thus evaluation of (4.20) on Γ yields (4.10). The boundary condition (4.12) is obtained by inserting (4.20) into (4.11)

$$\Delta\phi = \nabla \cdot \tilde{\boldsymbol{u}}(\Delta t), \tag{4.22}$$

$$= \nabla \cdot \left(-\Delta t \nabla p(\Delta t) + \boldsymbol{u}(\Delta t) + \mathcal{O}\left(\Delta t^2\right)\right), \tag{4.23}$$

$$= -\Delta t \Delta p(\Delta t) + \mathcal{O}\left(\Delta t^2\right). \tag{4.24}$$

We neglect higher order terms and use (4.24) to determine that $\phi = -\Delta t p(\Delta t)$ thus

$$\left.\frac{\partial \phi}{\partial \boldsymbol{n}}\right|_{\Gamma} = \Delta t \frac{\partial p}{\partial \boldsymbol{n}}.$$

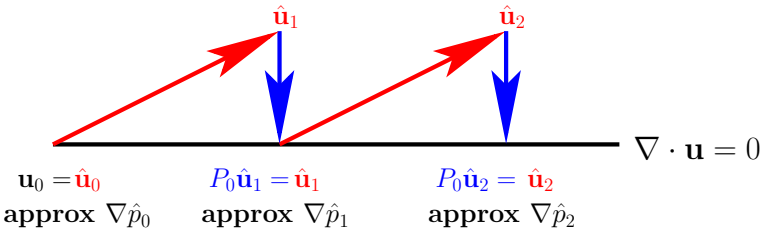

Figure 4.1: A typical fractional step projection method.

Figure 4.2: Krylov–based exponential projection method.

The boundary condition (4.15) is obtained by isolating normal components of (4.7):

$$\begin{aligned} \boldsymbol{n} \cdot \left.\frac{\partial \boldsymbol{u}}{\partial t}\right|_{\Gamma} &= \boldsymbol{n} \cdot \boldsymbol{F}(\boldsymbol{u})|_{\Gamma} - \boldsymbol{n} \cdot \nabla p|_{\Gamma}, \\ &= \boldsymbol{n} \cdot \boldsymbol{F}(\boldsymbol{u})|_{\Gamma} - \left.\frac{\partial p}{\partial \boldsymbol{n}}\right|_{\Gamma}. \end{aligned}$$

Since $\boldsymbol{n} \cdot \boldsymbol{u}|_{\Gamma} = 0$ implies

$$\boldsymbol{n} \cdot \left.\frac{\partial \boldsymbol{u}}{\partial t}\right|_{\Gamma} = 0,$$

we deduce (4.7).

Equations (4.11)–(4.13) define a Poisson formulation of the L^2 projection of the velocity $\boldsymbol{u}$ onto the divergence–free subspace $H = \{\boldsymbol{u} \in L^2(\Omega)^d : \nabla \cdot \boldsymbol{u} = 0, \boldsymbol{u} \cdot \boldsymbol{n}|_{\Gamma} = 0\}$, thus the velocity does not necessarily satisfy the tangential Dirichlet boundary conditions. If the tangential components were then set to zero, a numerical boundary layer results [65].

There are variants of the semi–implicit projection schemes, because there are many valid methods to integrate the momentum equation and to approximate ∇p. We note that a semi–implicit projection scheme that is higher than second order accurate in time has not been developed [35, pp. 860–862]. For construction of second order methods we refer to [46, 65].

4.2 Krylov–based exponential projection methods

In this section, we discuss the steps to construct the Krylov–based exponential projection method in detail. The three steps are as follows:

1. Evaluation of the Jacobian
2. Application of divergence–free projection
3. Application of KBEI

We consider the discretized form of the Navier–Stokes equations given by (3.45) as

$$\dot{\mathbf{u}}(t) = M^{-1}\left(-K\mathbf{u}(t) - N\left(\mathbf{u}(t)\right)\mathbf{u}(t) - C\mathbf{p}(t) + \mathbf{f}(t)\right), \tag{4.25}$$

$$= \mathbf{F}\left(\mathbf{u}(t)\right) + \mathbf{G}\left(\mathbf{p}(t)\right), \tag{4.26}$$

$$C^T\mathbf{u}(t) = 0, \tag{4.27}$$

$$\mathbf{u}(0) = \mathbf{u}_0, \tag{4.28}$$

where we have made the identifications

$$\mathbf{F}\left(\mathbf{u}(t)\right) = M^{-1}\left(-K\mathbf{u}(t) - N\left(\mathbf{u}(t)\right)\mathbf{u}(t) + \mathbf{f}(t)\right) \tag{4.29}$$

and

$$\mathbf{G}\left(\mathbf{p}(t)\right) = -M^{-1}C\mathbf{p}(t). \tag{4.30}$$

We utilize a discretely divergence–free projection that we denote by P^h. Presently we make no distinction between the divergence–free L^2 projection, P_0^h, and the divergence–free H^1 projection, P_1^h. We discuss the application of each projection separately in the paragraphs to follow. We apply P^h to each (4.25)–(4.28), and recall the following property of P^h:

$$C^T P^h \mathbf{u} = 0.$$

This property allows us to eliminate the continuity equation (4.27) and write the discretized Navier–Stokes equations as the following ordinary differential equation

$$\dot{\mathbf{v}}(t) = P^h\mathbf{F}\left(\mathbf{v}(t)\right) + P^h\mathbf{G}\left(\mathbf{p}(t)\right), \tag{4.31}$$

$$\mathbf{v}(0) = P^h\mathbf{v}_0, \tag{4.32}$$

to which we apply the high order Krylov–based exponential integration method given by (2.14)–(2.26). Note that we use $\mathbf{v}$ to denote the velocity in the projected ordinary differential equation (4.31)–(4.32) rather than $\mathbf{u}$ that solves (4.25) because in general $\boldsymbol{v}$ depends on the particular divergence–free projection used.

4.2.1 Evaluation of the Jacobian

In order to implement the high order Krylov–based exponential integration method (2.14)–(2.26), we need to evaluate the product of the Jacobian of $P^h\mathbf{F}(\mathbf{u}) + P^h\mathbf{G}(\mathbf{p})$ with respect to $\mathbf{u}$ on a vector $\mathbf{v}$.

The first case we consider is the autonomous case, where $\mathbf{f}$ is independent of t. We can obtain the Jacobian matrix applied to a vector $\mathbf{v} \in \mathbb{R}^N$ exactly

$$J(\mathbf{u})\mathbf{v} = \frac{\mathrm{d}\left(P^h\mathbf{F}(\mathbf{u})\right)}{\mathrm{d}\mathbf{u}}\mathbf{v}, \tag{4.33}$$

$$= P^h M^{-1}\left(-K\mathbf{v} - N(\mathbf{u})\mathbf{v} - N(\mathbf{v})\mathbf{u}\right), \tag{4.34}$$

$$\equiv P^h\mathbf{F}'(\mathbf{u})\mathbf{v}, \tag{4.35}$$

where we have used the identity

$$\frac{\mathrm{d}N(\mathbf{u})}{\mathrm{d}\mathbf{u}}\mathbf{v} = N(\mathbf{u})\mathbf{v} + N(\mathbf{v})\mathbf{u} \tag{4.36}$$

in (4.34). Equation (4.36) can be derived by considering the ith component of the Jacobian $J(\mathbf{u})$ applied to $\mathbf{v}$

$$\begin{aligned}\left[\frac{\mathrm{d}\left(N(\mathbf{u})\mathbf{u}\right)}{\mathrm{d}\mathbf{u}}\mathbf{v}\right]_i &= \sum_{l=1}^{N}\left[\frac{\mathrm{d}\left(N(\mathbf{u})\mathbf{u}\right)}{\mathrm{d}\mathbf{u}}\right]_{il}\mathrm{v}_l,\\ &= \sum_{l=1}^{N}\frac{\mathrm{d}\left[N(\mathbf{u})\mathbf{u}\right]_i}{\mathrm{d}\mathrm{u}_l}\mathrm{v}_l,\\ &= \sum_{l=1}^{N}\mathrm{v}_l\frac{\mathrm{d}}{\mathrm{d}\mathrm{u}_l}\sum_{j=1}^{N}\sum_{k=1}^{N}\mathrm{u}_k c\big(\boldsymbol{\varphi}_k;\boldsymbol{\varphi}_j,\boldsymbol{\varphi}_i\big)\,\mathrm{u}_j,\end{aligned}$$

where we have used matrix–vector multiplication and the definition of $N(\mathbf{u})\mathbf{u}$ given by (3.50). We differentiate under the summations and use the fact that

$$\frac{\mathrm{d}\mathrm{u}_k}{\mathrm{d}\mathrm{u}_l} = \delta_{kl},$$

where δ_{kl} is the Kronecker delta, to obtain

$$\begin{aligned}\left[\frac{\mathrm{d}\left(N(\mathbf{u})\mathbf{u}\right)}{\mathrm{d}\mathbf{u}}\mathbf{v}\right]_i &= \sum_{l=1}^{N} \mathrm{v}_l \sum_{k=1}^{N} \mathrm{u}_k c(\boldsymbol{\varphi}_k;\boldsymbol{\varphi}_l,\boldsymbol{\varphi}_i) + \sum_{l=1}^{N} \mathrm{v}_l \sum_{j=1}^{N} c\big(\boldsymbol{\varphi}_l;\boldsymbol{\varphi}_j,\boldsymbol{\varphi}_i\big)\,\mathrm{u}_j,\\ &= \left[N(\mathbf{u})\mathbf{v}\right]_i + \left[N(\mathbf{v})\mathbf{u}\right]_i,\end{aligned}$$

from which (4.36) follows.

4.2.2 Application of the divergence–free projections

We recall the L^2 projection from Chapter 5. To find the application of the discrete L^2 projection $\mathbf{u}_0 = P_0^h\mathbf{F}'(\mathbf{u})$ we solve the following system for $\mathbf{u}_0$

$$\begin{bmatrix} M & C \\ C^T & 0 \end{bmatrix}\begin{bmatrix}\mathbf{u}_0 \\ \boldsymbol{\lambda}_0\end{bmatrix} = \begin{bmatrix} M\mathbf{F}'(\mathbf{u}) \\ 0 \end{bmatrix}. \tag{4.37}$$

We note the presence of M in the right hand side of (4.37) cancels the M^{-1} in $\mathbf{F}'(\mathbf{u})$. We also call attention to the property

$$P_0^h\mathbf{G}(\mathbf{p}(t)) = 0.$$

We discuss a scalable method for solution of this system in Section 5.1 and the stability of finite element solutions in Section 5.2.

To apply the discrete H^1 projection $\mathbf{u}_0 = P_1^h\mathbf{F}'(\mathbf{u})$ we solve the following system for $\mathbf{u}_0$

$$\begin{bmatrix} K & C \\ C^T & 0 \end{bmatrix}\begin{bmatrix}\mathbf{u}_0 \\ \boldsymbol{\lambda}_0\end{bmatrix} = \begin{bmatrix} K\mathbf{F}'(\mathbf{u}) \\ 0 \end{bmatrix}. \tag{4.38}$$

We discuss a scalable method for solution of this system in Section 5.1.

4.2.3 Application of KBEI

For sake of illustration, we outline the method for the simple case of the exponentially fitted Euler method (2.4) for the autonomous Navier–Stokes equations, using a divergence–free projection P^h. We apply the Krylov–based exponentially fitted Euler method to (4.31)–(4.32); each step yields the following

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t V_m \|\mathbf{v}\| \varphi(\Delta t H_m)\mathbf{e}_1. \tag{4.39}$$

Here V_m and H_m are the matrices found in (2.32) computed using Algorithm 2.1 for the Arnoldi process for the matrix product $P^h\mathbf{F}'(\mathbf{u}_n)\mathbf{v}$ with

$$\mathbf{v} = P^h\mathbf{F}(\mathbf{u}_n). \tag{4.40}$$

We note that P^h is applied m times at each time step to compute $\mathbf{v}_i$, $i = 1, \ldots, m$, where m is the dimension of the Krylov subspace. We compute the product $\varphi(\Delta t H_m)\mathbf{e}_1$ using Algorithm 2.3. We note that if $P^h = P_0^h$, then we use (4.39) and $\mathbf{v} = P^h\mathbf{F}(\mathbf{u}_n)$.

In practice we utilize the high order method given by (2.14)–(2.26), where the steps (2.14)–(2.16) are computed using Algorithms 2.1 and 2.3 for the product $P^h\mathbf{F}'(\mathbf{u}_n)\mathbf{v}$ with $\mathbf{v}$ given by (4.40), (2.20)–(2.22) are computed using Algorithms 2.1 and 2.3 for the product $P^h\mathbf{F}'(\mathbf{u}_n)\mathbf{d}_4$ and (2.25) is computed similarly for the product $P^h\mathbf{F}'(\mathbf{u}_n)\mathbf{d}_7$. In practice a smaller dimensional Krylov subspace is required for steps (2.20)–(2.22) and (2.25), since the norms of $\mathbf{d}_4$ and $\mathbf{d}_7$ are generally smaller than the norm of $\mathbf{v}$ given by (4.40).

4.2.4 Notes on the non–autonomous case

We consider the non-autonomous case where $\mathbf{f} = \mathbf{f}(t)$. We recall from Section 2.1 the special treatment of non–autonomous systems. In this case the momentum equation (4.31) becomes

$$\begin{bmatrix} \dot{\mathbf{u}}(t) \\ t \end{bmatrix} = \begin{bmatrix} P^h\mathbf{F}\left(\mathbf{u}(t)\right) \\ 1 \end{bmatrix}.$$

We write the Jacobian applied to the vector $\hat{\mathbf{v}} = \begin{bmatrix}\mathbf{v}^T & w\end{bmatrix}^T$ with $\mathbf{v} \in \mathbb{R}^N$ and $w \in \mathbb{R}$ as

$$\begin{bmatrix} P^h\mathbf{F}'(\mathbf{u}(t)) & P^h\dot{\mathbf{f}}(t) \\ 0 & 0 \end{bmatrix}\hat{\mathbf{v}} = \begin{bmatrix} P^h\left(\mathbf{F}'(\mathbf{u}(t))\mathbf{v} + w\mathbf{f}(t)\right) \\ 0 \end{bmatrix}.$$

We define

$$P^h\hat{\mathbf{F}}'(\mathbf{u})\hat{\mathbf{v}} = \begin{bmatrix} P^h\left(\mathbf{F}'(\mathbf{u}(t))\mathbf{v} + w\mathbf{f}(t)\right) \\ 0 \end{bmatrix}. \tag{4.41}$$

The implementation of a Krylov–based exponential integration for the non–autonomous case is essentially the same as the autonomous case, with the exception that we compute the Arnoldi process for the products of the form (4.41), rather than (4.34), where $\hat{\mathbf{v}} \in \mathbb{R}^{N+1}$.

4.2.5 Summary

We have outlined the key differences between the Krylov–based projection method and the traditional fractional step methods. The Krylov–based projection method projects the momentum equation onto a divergence–free subspace, then integrates in time with a high order Krylov–based exponential integrator. By construction, the Krylov subspace consists of weakly discretely divergence–free vectors, hence the vector obtained at the advanced time step is also weakly discretely divergence–free. The fractional step methods approximate a divergence–free momentum equation via a guess for the pressure at the initial time step. This results in a velocity vector at the advanced time step that is not weakly discretely divergence–free. This is remedied by projecting this vector onto a weakly discretely divergence–free subspace via a divergence–free L^2 projection obtained by a Poisson solve. This process is illustrated graphically in Figure 4.1. Figure 4.2 illustrates the projection process for the Krylov–based projection method. One advantage of the Krylov–based projection method is a higher order temporal convergence.

Chapter 5

Implementation of divergence–free projections: scalability and stability

In this chapter we discuss the issues regarding the implementation of a Krylov–based exponential projection method for the time integration of the Navier–Stokes equations. We rewrite the discrete momentum equation (3.45) as

$$\dot{\mathbf{u}}(t) = M^{-1}\left(-K\mathbf{u}(t) - N\left(\mathbf{u}(t)\right)\mathbf{u}(t) - C\mathbf{p}(t) + \mathbf{f}(t)\right), \tag{5.1}$$

$$= \mathbf{F}(\mathbf{u}(t)) + \mathbf{G}(\mathbf{p}(t)), \tag{5.2}$$

$$\mathbf{u}(0) = \mathbf{u}_0. \tag{5.3}$$

We apply a discretely divergence–free L^2 or H^1 projection, which we denote abstractly by P^h, and denote the velocity by $\mathbf{v}$ to (5.1)–(5.3) as follows:

$$\dot{\mathbf{v}}(t) = P^h\mathbf{F}(\mathbf{v}(t)), \tag{5.4}$$

$$\mathbf{v}(0) = P^h\mathbf{v}_0. \tag{5.5}$$

Recall from Chapter 4 that application of P^h on a vector $\mathbf{f}$ is equivalent to solving the following system for $\mathbf{u}$:

$$\begin{bmatrix} A & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}, \tag{5.6}$$

where we have $A = M$ for the divergence–free L^2 projection and $A = K$ for the divergence–free H^1 projection. The final step is to apply a high order

Krylov–based exponential method defined by (2.14)–(2.15) to (5.4)–(5.5). A solution to (5.6) must be computed for each each Krylov vector at each time step. Thus it is critical that we solve (5.6) in an efficient and scalable way. Another critical issue regards stability of finite element solutions to the divergence–free L^2 and H^1 projections when boundary conditions are imposed.

This chapter is dedicated to the following two issues:

1. Efficient and scalable computation of solutions to (5.6).

2. Stability of finite element solutions to the L^2 projection.

We address the first issue in Section 5.1 where we discuss the implementation of the Uzawa method for solution of (5.6). The Uzawa method is coupled with a preconditioning strategy that is shown to be scalable or independent of the mesh size. We address the second issue in Section 5.2, where we discuss the stability theory and provide insight on the stability through a series of numerical experiments. In particular, we show that the divergence–free L^2 projection suffers from suboptimal convergence while the divergence–free H^1 projection converges optimally.

5.1 Uzawa method

The Uzawa method is a common method for solving (5.6) when A is positive definite. Efficient methods for (5.6) based on domain decomposition [7] and block preconditioning are proposed in [6, 8]. A survey of preconditioning the Uzawa method can be found in [9]. These strategies are somewhat limited because they are based on preconditioning the Stokes system under the assumption $A = K$. More useful preconditioning strategies are found in [14, 47], in which preconditioners are sought for discretizations of the generalized Stokes problem (3.29)–(3.30). Preconditioners for (3.29)–(3.30) are more versatile because the matrix A has contributions from both M and K. The preconditioned Uzawa method is an efficient method to solve the discretized saddle point systems for Stokes flow and Darcy flow. In particular we investigate the effect of preconditioning the Uzawa algorithm and a variant that utilizes conjugate directions. The preconditioned Uzawa method, Uzawa method with conjugate directions, and the preconditioned Uzawa method with conjugate directions are compared under a variety of conditions.

We utilize the preconditioner due to Cahouet and Chabard [14] for its ease of implementation and versatility when applied to (5.6) with $A = M$ or $A = K$. We discuss the implementation and versatility in detail below, and provide experimental results that reveal the strengths of the preconditioner. We evaluate the performance of each method.

We consider as a model problem a finite element discretization of Darcy–Stokes flow given by (3.29)–(3.30):

$$\alpha^2 \boldsymbol{u} - \epsilon^2 \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f}, \quad \boldsymbol{x} \in \Omega, \tag{5.7}$$

$$\nabla \cdot \boldsymbol{u} = \boldsymbol{g}, \quad \boldsymbol{x} \in \Omega, \tag{5.8}$$

$$\boldsymbol{u} = 0, \quad \boldsymbol{x} \in \Gamma, \tag{5.9}$$

with $\Omega = (0,1) \times (0,1)$ and boundary $\partial\Omega$. For $\alpha = 0$ and $\epsilon = 1$ (5.7)–(5.9) represents a divergence–free H^1 projection of $\boldsymbol{f}$ or Stokes flow and for $\alpha = 1$ and $\epsilon = 0$ (5.7)–(5.9) represents a divergence–free L^2 projection of $\boldsymbol{f}$ or Darcy flow. Discretization via a standard $\mathbb{Q}_2 Q_1$ elements on a uniform, structured mesh of size h yields a saddle point problem given by

$$\begin{bmatrix} A & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}, \tag{5.10}$$

with $A \in \mathbb{R}^{n \times n}$; $C \in \mathbb{R}^{n \times m}$; $\mathbf{u}, \mathbf{f} \in \mathbb{R}^n$; and $\mathbf{p}, \mathbf{g} \in \mathbb{R}^n$. The matrix A is given by

$$A = \alpha^2 M + \epsilon^2 K,$$

where M is the mass matrix defined in (3.48), K is the stiffness matrix defined in (3.49) and $\mathbf{f}$ is as defined in (3.52). We shall consider solution methods for (5.10) with $\alpha, \epsilon \in [0,1]$. The condition number of a matrix A is given by $\kappa(A) = \|A\| \, \|A^{-1}\|$. We note the condition numbers of the matrices M and K: $\kappa(M) = \mathcal{O}(1)$, $\kappa(K) = \mathcal{O}(h^{-2})$.

Equation (5.10) can be written as

$$A\,\mathbf{u} = \mathbf{f} - C\,\mathbf{p}, \tag{5.11}$$

$$S\,\mathbf{p} = C^T A^{-1}\,\mathbf{f} - \mathbf{g}, \tag{5.12}$$

with S given by $C^T A^{-1} C$. The matrix S is symmetric positive definite. The standard Uzawa method is a steepest descent iteration applied to (5.12). The conjugate gradient method can also be applied to (5.12). This is often referred to as Uzawa with conjugate directions. We recall that C is a finite

element discretization of a first order differential operator ($C \approx \nabla$ and $C^T \approx \nabla\cdot$) and M is a finite element discretization of the identity operator, thus when $\alpha = 1$ and $\epsilon = 0$, can be viewed as a Laplacian operator ($C^T M^{-1} C \approx \nabla \cdot I \nabla \approx \Delta$). Thus the matrix S can have large condition number, notably when $\alpha = 1$ and $\epsilon = 0$, $\kappa(S) = \mathcal{O}(h^{-2})$. Thus preconditioning of the Uzawa method is necessary.

To analyze preconditioning of the Uzawa method, let R be a preconditioner for (5.12). For R to be an effective preconditioner for S we desire

$$R^{-1} S \approx I,$$

where I is the identity matrix, and that the cost of applying R^{-1} be considerably less than that of applying S^{-1}. We transform (5.12) as

$$R^{-1} S \mathbf{p} = R^{-1} \left(C^T A^{-1} \mathbf{f} - \mathbf{g} \right). \tag{5.13}$$

We utilize the Cahouet–Chabard preconditioner given in [14, 31] as follows:

$$R = \left(\epsilon^2 M_p^{-1} + \alpha^2 K_p^{-1} \right)^{-1}, \tag{5.14}$$

where M_p is the mass matrix corresponding to the pressure, and K_p is the stiffness matrix corresponding to the pressure with pure homogeneous Neumann boundary conditions. The matrix R is symmetric positive definite. Note that when $\epsilon^2 = 0$, $\kappa(R^{-1}S) = \mathcal{O}(1)$. Bramble and Pasciak [9] prove that when the Cahouet–Chabard preconditioner is implemented, the number of iterations required in the Uzawa method is independent of the mesh size. This is important because as we increase the problem size, the number of iterations require remains constant.

The Uzawa method consists of three key steps:

1. An *outer iteration* to apply S^{-1} in (5.12). Each outer iteration is an iteration of the conjugate gradient method to apply S^{-1}.

2. For each outer iteration, we require *inner iterations* to apply A^{-1} in (5.11). Each inner iteration is an iteration of the conjugate gradient method to apply A^{-1}.

3. For each outer iteration, *preconditioned iterations* are required to apply R^{-1} in (5.13). Each precondition iteration is an iteration of the conjugate gradient method to apply R^{-1}.

Table 5.1: Number of iterations for $\alpha^2\boldsymbol{u}-\epsilon^2\Delta\boldsymbol{u}+\nabla p=\boldsymbol{f}$, $\nabla\cdot\boldsymbol{u}=0$ with $\mathbb{Q}_2Q_1$ elements, third order quadrature, mesh size h, using Uzawa with conjugate directions (CD), Preconditioned Uzawa with steepest descents (PSD), and Preconditioned Uzawa with conjugate directions (PCD). The left table shows varying ϵ with $\alpha=1$. The right table shows varying α with $\epsilon=1$.

ϵ	h	CD	PSD	PCD
1	2^{-1}	2	46	3
	2^{-2}	8	43	8
	2^{-3}	30	47	8
	2^{-4}	40	50	9
	2^{-5}	51	51	9
	2^{-6}	52	52	9
	2^{-7}	54	53	10
2^{-2}	2^{-1}	2	34	3
	2^{-2}	8	36	8
	2^{-3}	27	42	8
	2^{-4}	38	46	8
	2^{-5}	51	49	9
	2^{-6}	51	50	10
	2^{-7}	51	51	10
2^{-4}	2^{-1}	2	22	3
	2^{-2}	8	23	6
	2^{-3}	33	29	7
	2^{-4}	46	37	8
	2^{-5}	50	41	9
	2^{-6}	50	42	9
	2^{-7}	52	43	9
2^{-6}	2^{-1}	2	18	3
	2^{-2}	8	17	5
	2^{-3}	21	18	6
	2^{-4}	32	24	7
	2^{-5}	43	30	7
	2^{-6}	56	33	8
	2^{-7}	66	35	8
2^{-8}	2^{-1}	2	18	3
	2^{-2}	8	16	5
	2^{-3}	21	16	6
	2^{-4}	37	18	6
	2^{-5}	65	20	6
	2^{-6}	99	23	6
	2^{-7}	139	26	7
0	2^{-1}	2	18	3
	2^{-2}	8	16	5
	2^{-3}	21	16	6
	2^{-4}	38	17	6
	2^{-5}	69	17	6
	2^{-6}	124	16	5
	2^{-7}	236	14	5

α	h	CD	PSD	PCD
1	2^{-1}	5	50	3
	2^{-2}	20	54	6
	2^{-3}	46	55	12
	2^{-4}	55	55	13
	2^{-5}	60	58	14
	2^{-6}	62	54	14
	2^{-7}	68	56	13
2^{-2}	2^{-1}	5	50	3
	2^{-2}	19	56	6
	2^{-3}	46	55	12
	2^{-4}	55	56	13
	2^{-5}	59	53	14
	2^{-6}	62	54	14
	2^{-7}	71	55	12
2^{-4}	2^{-1}	5	50	3
	2^{-2}	19	57	6
	2^{-3}	46	57	11
	2^{-4}	55	56	13
	2^{-5}	59	55	13
	2^{-6}	62	54	13
	2^{-7}	64	55	13
2^{-6}	2^{-1}	5	50	3
	2^{-2}	20	57	6
	2^{-3}	49	57	9
	2^{-4}	55	56	11
	2^{-5}	62	56	12
	2^{-6}	67	55	11
	2^{-7}	70	55	11
2^{-8}	2^{-1}	5	50	3
	2^{-2}	19	57	6
	2^{-3}	48	57	9
	2^{-4}	57	56	10
	2^{-5}	60	55	10
	2^{-6}	81	55	9
	2^{-7}	92	54	9
0	2^{-1}	5	53	3
	2^{-2}	19	51	6
	2^{-3}	53	56	8
	2^{-4}	75	57	8
	2^{-5}	82	55	7
	2^{-6}	92	55	7
	2^{-7}	110	54	7

Table 5.2: $\alpha = 0$, $\epsilon = 1$.

h	# inner	# prec	time (sec)
2^{-1}	34	12	0.01
2^{-2}	236	68	0.01
2^{-3}	522	175	0.09
2^{-4}	971	215	0.73
2^{-5}	1852	217	6.50
2^{-6}	3448	203	52.27
2^{-7}	6803	189	420.79

Table 5.3: $\epsilon = 0$, $\alpha = 1$.

h	# inner	# prec	time (sec)
2^{-1}	33	30	0.01
2^{-2}	119	65	0.01
2^{-3}	147	115	0.03
2^{-4}	147	191	0.13
2^{-5}	127	309	0.71
2^{-6}	121	517	3.48
2^{-7}	112	927	17.58

Since $\kappa(K) = \mathcal{O}(h^{-2})$ and $\kappa(K_p) = \mathcal{O}(h^{-2})$, the applications of A^{-1} and R^{-1} should be preconditioned as well. An enhancement to the algorithm would apply a multigrid method [11] for the applications of A^{-1} and R^{-1}.

Equation (5.7)–(5.9) is discretized using $\mathbb{Q}_2 Q_1$ elements on a uniform structured triangulation of the domain $\Omega = (0,1) \times (0,1)$. The functions $\boldsymbol{f}$ and $\boldsymbol{g}$ are chosen such that $\boldsymbol{u} = \nabla \times \sin^2 \pi x_1 \sin^2 \pi x_2$ and $p = \sin \pi x_1$. Note that this solution satisfies $\nabla \cdot \boldsymbol{u} = \boldsymbol{g} = 0$ on Ω and $\boldsymbol{u} = 0$ on Γ. We solve the resulting saddle point problem (5.10) using the Uzawa method with conjugate directions, preconditioned Uzawa with steepest descents, and preconditioned Uzawa with conjugate directions. Each preconditioned method utilizes the preconditioner R given by (5.14). The goal is to study these different solution techniques with varying ϵ, α and mesh size, h. Each solution technique is applied with a mesh size $h = 2^{-1}$, 2^{-2}, 2^{-3}, 2^{-4}, 2^{-6}, 2^{-7}; and with ϵ, $\alpha = 1$, 2^{-2}, 2^{-4}, 2^{-6}, 2^{-8}, 0.

For each outer iteration a residual tolerance of $1 \cdot 10^{-10}$ was used. For each of the inner and precondition iterations the conjugate gradient method with SSOR preconditioning with relaxation parameter $\omega = 1.2$ and residual tolerance $1 \cdot 10^{-12}$ was used.

Table 5.1 shows the number of iterations required to reach convergence

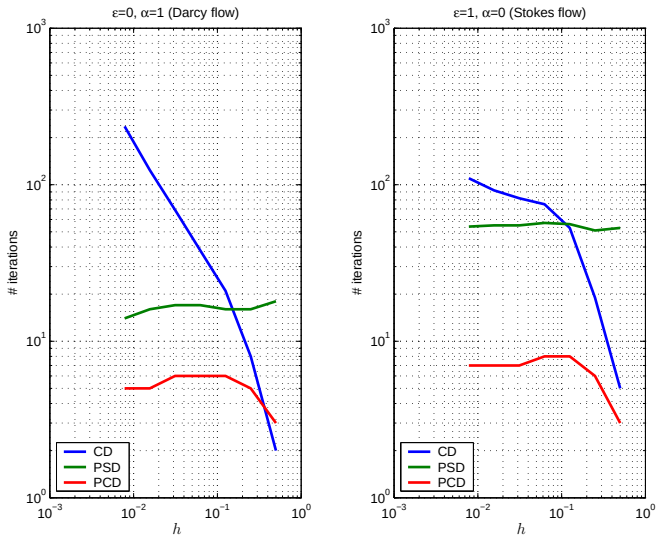

Figure 5.1: Number of outer iterations vs. h for Uzawa method with conjugate directions (blue), preconditioned Uzawa with steepest descent (green) and preconditioned Uzawa with conjugate directions (red) for Darcy flow (left) and Stokes flow (right).

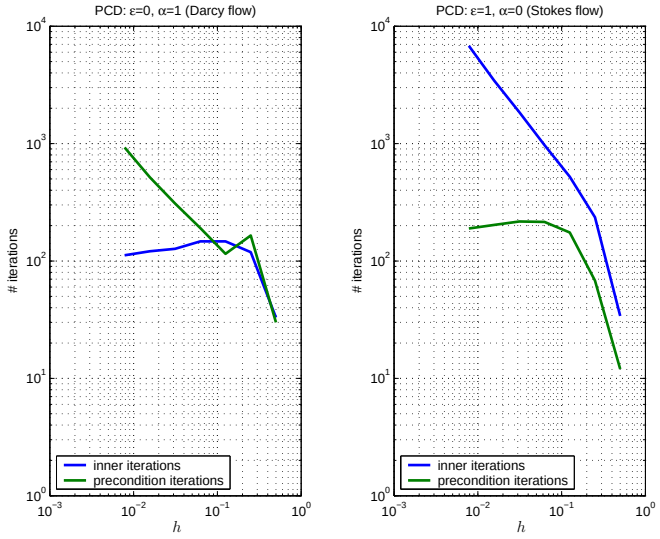

Figure 5.2: Number of inner iterations vs. h (blue) and number of precondition iterations vs. h (green) for preconditioned Uzawa with conjugate directions for Darcy flow (left) and Stokes flow (right).

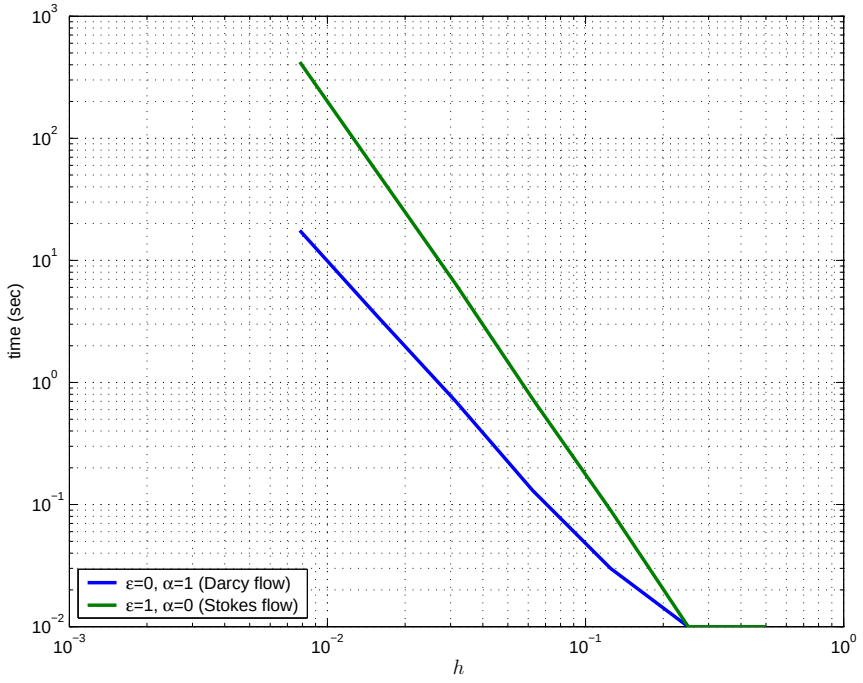

Figure 5.3: CPU time vs. h for Darcy flow (blue) and Stokes flow (green).

for each method with selected values of ϵ and h. We observe that for Uzawa with conjugate directions and no preconditioning the number of iterations increase as h decreases and as ϵ decreases. The increase in the number of iterations as h decreases is due to the poor conditioning of S, while the increase in the number of iterations as ϵ decreases is because S approximates a second order differential operator as $\epsilon \to 0$. For both preconditioned methods, observe that for fixed ϵ, the number of iterations remains constant as h decreases. This is due to the condition of $R^{-1}S$, and demonstrates that this preconditioning is independent of mesh size. We note that the number of iterations decrease as ϵ decreases, since $R^{-1}S$ becomes better conditioned as $\epsilon \to 0$. Another observation is the number of iterations is less for preconditioned Uzawa with conjugate directions than for preconditioned Uzawa with steepest descent.

Figure 5.1 shows the number of outer iterations vs. h for Uzawa method with conjugate directions, preconditioned Uzawa with steepest descent and preconditioned Uzawa with conjugate directions for the case $\alpha = 1$, $\epsilon = 0$ and

the case $\alpha = 0$, $\epsilon = 1$. Both preconditioned methods are independent of mesh size and that the number of iterations is less for preconditioned Uzawa with conjugate directions than for preconditioned Uzawa with steepest descent.

Figure 5.2 shows the number of inner iterations vs. h and number of precondition iterations vs. h for preconditioned Uzawa with conjugate directions for for the case $\alpha = 1$, $\epsilon = 0$ and the case $\alpha = 0$, $\epsilon = 1$. When $\alpha = 1$ and $\epsilon = 0$ the number of inner iterations is independent of mesh size; however the number of iterations to apply the preconditioner increases. When $\alpha = 0$ and $\epsilon = 1$ the number of inner iterations increases while the number of iterations to apply the preconditioner is independent of mesh size.

Table 5.2 shows the number of inner iterations and the number of precondition iterations required to reach convergence and CPU time for the preconditioned Uzawa method with conjugate directions with selected values of h for the case $\alpha = 0$, $\epsilon = 1$. Table 5.3 shows the number of inner iterations and the number of precondition iterations required to reach convergence and CPU time for the preconditioned Uzawa method with conjugate directions with selected values of h for the case $\alpha = 1$, $\epsilon = 0$.

We reiterate that application of the divergence–free L^2 projection is equivalent to solving a Darcy flow problem and application of the divergence–free H^1 projection is equivalent to solving a Stokes problem. The results of this section conclude that we can apply the divergence–free L^2 projection or the divergence–free H^1 projection in a scalable manner that is independent of the mesh size. Furthermore, the application of either projection is easily implemented using a solver for the generalized Stokes problem and choosing α and ϵ accordingly.

5.2 Stability of the divergence–free L^2 projection

In this section we investigate the stability of some common conforming mixed finite elements for Darcy–Stokes flow. Mardal et al. [54] show that stable finite elements for Stokes flow are not necessarily stable for Darcy flow. Despite this lack of stability, the use of Stokes elements for the divergence–free L^2 projection [35, 33, 34] has been employed with some success. We examine the stability of some of these elements at the transition from Stokes flow to Darcy flow both mathematically and experimentally. This transition corre-

sponds to the application of the divergence–free L^2 projection with imposed boundary conditions.

We consider a finite element discretization of Darcy–Stokes flow given by (3.29) with $\alpha = 1$:

$$\left(I - \epsilon^2 \Delta\right) \boldsymbol{u} + \nabla p = \boldsymbol{f} \quad \boldsymbol{x} \in \Omega, \tag{5.15}$$

$$\nabla \cdot \boldsymbol{u} = g \quad \boldsymbol{x} \in \Omega, \tag{5.16}$$

$$\boldsymbol{u} = 0 \quad \boldsymbol{x} \in \Gamma, \tag{5.17}$$

with $\Omega = (0,1) \times (0,1)$ and boundary Γ. When the parameter $\epsilon = 0$, (5.15)–(5.17) represents the divergence–free L^2 projection or Darcy flow, otherwise (5.15)–(5.17) is a Stokes problem with the addition of a lower order term.

When $\epsilon > 0$, a weak formulation of (5.15)–(5.17) is given by: find $\boldsymbol{u} \in H_0^1(\Omega)^2$ and $p \in L_0^2(\Omega)$ such that

$$a_\epsilon(\boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{v}, p) = (\boldsymbol{f}, \boldsymbol{v}) \quad \forall \boldsymbol{v} \in H_0^1(\Omega)^2, \tag{5.18}$$

$$b(\boldsymbol{u}, q) = (g, q) \quad \forall q \in L_0^2(\Omega). \tag{5.19}$$

Where the bilinear form $a_\epsilon(\cdot,\cdot) : H^1(\Omega)^2 \times H^1(\Omega)^2 \mapsto \mathbb{R}$ is given by

$$a_\epsilon(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u}, \boldsymbol{v}) + \epsilon^2 a(\boldsymbol{u}, \boldsymbol{v}), \tag{5.20}$$

with $a(\cdot,\cdot) : H^1(\Omega)^2 \times H^1(\Omega)^2 \mapsto \mathbb{R}$ given by (3.36) and $b(\cdot,\cdot) : H^1 \times L_0^2 \mapsto \mathbb{R}$ given by (3.35).

For $\epsilon \in (0,1]$, (5.18)–(5.19) has a unique solution for $(\boldsymbol{u}, p) \in H_0^1(\Omega)^2 \times L_0^2(\Omega)$. This is due to standard results for the Stokes problem, that is, Theorem 3.6 is satisfied for the space $(\boldsymbol{u}, p) \in H_0^1(\Omega)^2 \times L_0^2(\Omega)$ and the bilinear form $a_\epsilon(\cdot,\cdot)$ defined as in (5.20). For the details the reader is referred to [5]. However, for $\epsilon = 0$, $H_0^1(\Omega)^2 \times L_0^2(\Omega)$ is not a proper space for the solution. In this case proper spaces for the solution are either $H_0^{\mathrm{div}}(\Omega) \times L^2(\Omega)$ or $L^2(\Omega)^2 \times (H^1 \cap L_0^2(\Omega))$. These spaces are proper solution spaces for a standard mixed formulation of the Poisson problem [54].

We introduce the space $\mathcal{S}_\epsilon = H_0^{\mathrm{div}}(\Omega) \cap \epsilon \cdot H_0^1(\Omega)^2$ with norm

$$\|\boldsymbol{v}\|_\epsilon^2 = \|\boldsymbol{v}\|^2 + \|\nabla \cdot \boldsymbol{v}\|^2 + \epsilon^2 \|\nabla \boldsymbol{v}\|^2.$$

Note that because $H_0^1(\Omega) \subset H_0^{\mathrm{div}}(\Omega)$, we have $\mathcal{S}_\epsilon = H_0^1(\Omega)$ for $\epsilon > 0$, and $\mathcal{S}_\epsilon = H_0^{\mathrm{div}}(\Omega)$ for $\epsilon = 0$.

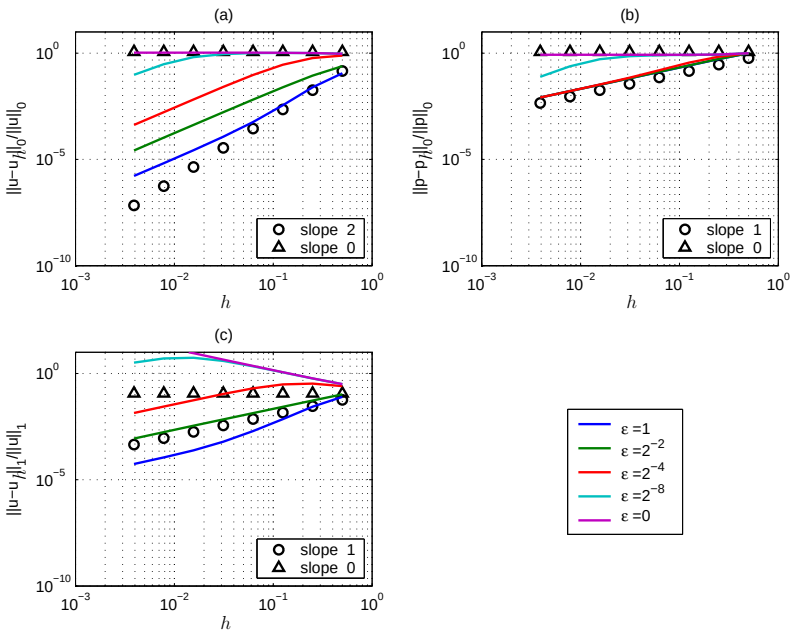

Figure 5.4: Relative errors vs. h for $\mathbb{Q}_2 Q_0$ element.

In this section we review the work of [54] and present further numerical experiments to demonstrate that not all stable finite element discretizations of (5.18)–(5.19) for $\epsilon \in (0, 1]$ are stable finite element discretizations of (5.18)–(5.19) for $\epsilon = 0$.

Let V^h and Q^h be conforming finite element spaces of $H_0^1(\Omega)^2$ and $L_0^2(\Omega)$. Problem (5.18)–(5.19) leads to the finite element discretization, find $\boldsymbol{u}^h \in V^h$ and $p^h \in Q^h$ such that

$$a_\epsilon(\boldsymbol{u}^h, \boldsymbol{v}^h) + b(\boldsymbol{v}^h, p^h) = (\boldsymbol{f}, \boldsymbol{v}^h) \quad \forall \boldsymbol{v}^h \in V^h, \tag{5.21}$$

$$b(\boldsymbol{u}^h, q^h) = (g, q^h) \quad \forall q^h \in Q^h. \tag{5.22}$$

The discretization (5.21)–(5.22) is stable if Theorem 3.7 is satisfied: there exist positive constants c_1 and c_2 independent of h and ϵ such that

$$a_\epsilon(\boldsymbol{v}^h, \boldsymbol{v}^h) \geq c_1 \|\boldsymbol{v}^h\|_\epsilon^2 \quad \forall \boldsymbol{v}^h \in Z^h, \tag{5.23}$$

and

$$\sup_{\boldsymbol{v}^h \in V^h} \frac{b(\boldsymbol{v}^h, q^h)}{\|\boldsymbol{v}\|_\epsilon^2} \geq c_2 \|q^h\|^2 \quad \forall q^h \in Q^h, \tag{5.24}$$

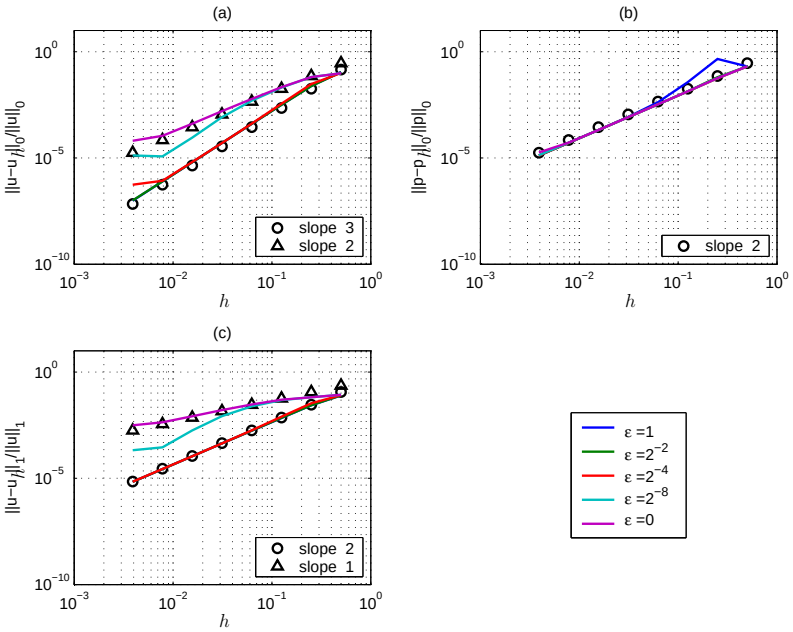

Figure 5.5: Relative errors vs. h for $\mathbb{Q}_2Q_1$ element.

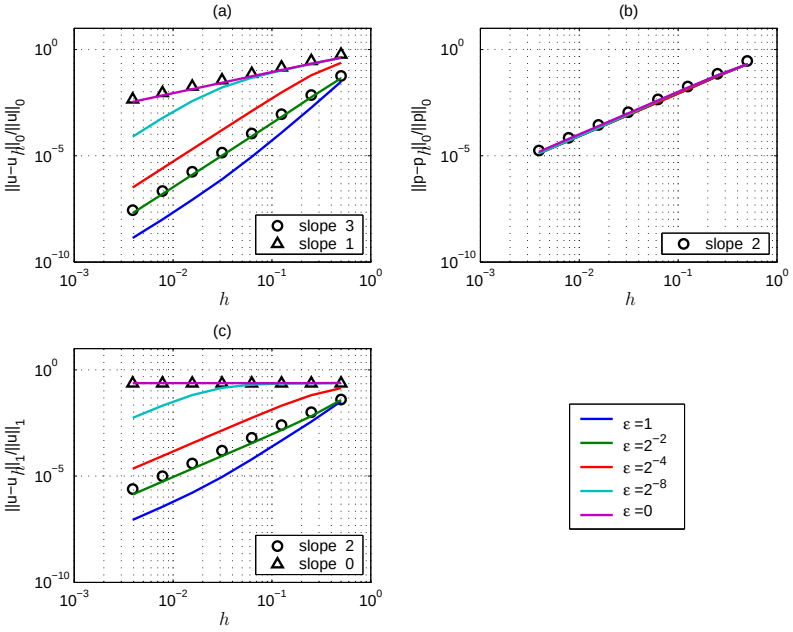

Figure 5.6: Relative errors vs. h for $\mathbb{Q}_3Q_1$ element.

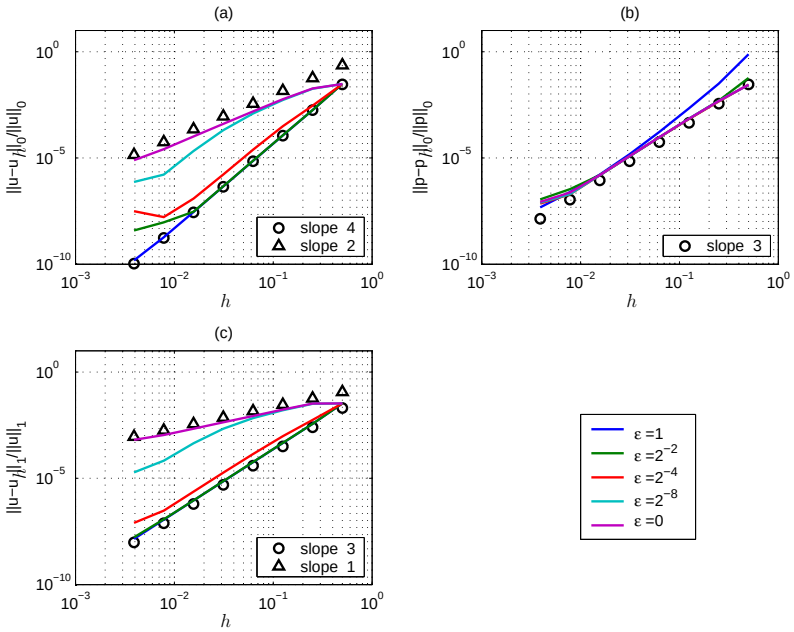

Figure 5.7: Relative errors vs. h for $\mathbb{Q}_3Q_2$ element.

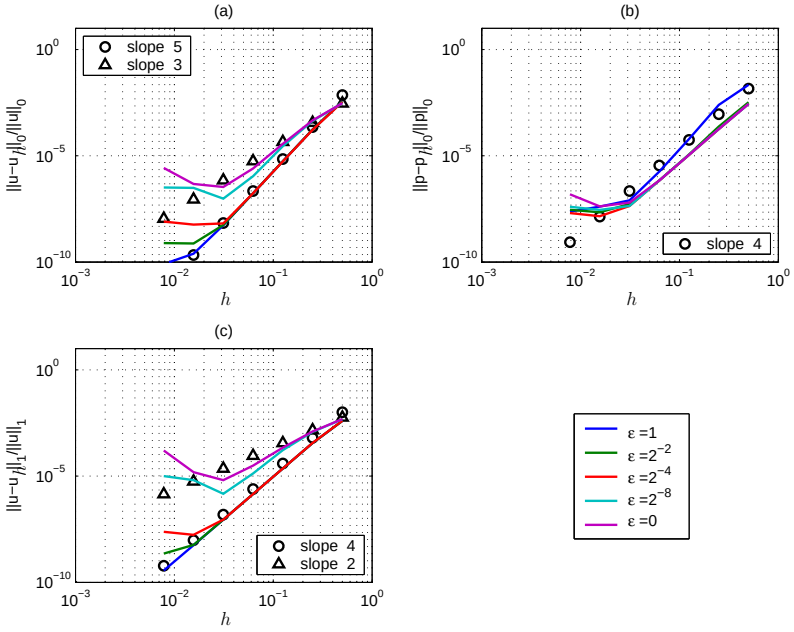

Figure 5.8: Relative errors vs. h for $\mathbb{Q}_4Q_3$ element.

where Z^h is the space of weakly discretely divergence–free functions given by (3.56). For $\epsilon \in (0,1]$, (5.23) is satisfied with $c_1 \leq 1$. This is due to the fact that coerciveness holds on subspaces, that is $a_\epsilon(\boldsymbol{v},\boldsymbol{v}) \geq c_1\|\boldsymbol{v}\|_\epsilon^2$ holds for all $\boldsymbol{v} \in H^1(\Omega)^2$ and $Z^h \subset H_0^1(\Omega)^2$ since $V^h \subset H_0^1(\Omega)^2 \subset H^1(\Omega)^2$. In this case one can choose conforming finite element spaces such that (5.24) is satisfied. The standard $\mathbb{Q}_2Q_1$ element is an example of a stable element. More stable elements can be found in [5].

However, when $\epsilon = 0$ (5.23) may not hold. In fact when $\epsilon = 0$, $a_\epsilon\big(\boldsymbol{v}^h,\boldsymbol{v}^h\big) = \|\boldsymbol{v}^h\|^2$, $\|\boldsymbol{v}^h\|_\epsilon^2 = \|\boldsymbol{v}^h\|^2 + \|\nabla\cdot\boldsymbol{v}^h\|^2$ and (5.23) gives

$$\|\boldsymbol{v}^h\|^2 \geq c_1\left[\|\boldsymbol{v}^h\|^2 + \|\nabla\cdot\boldsymbol{v}^h\|^2\right] \quad \forall \boldsymbol{v}^h \in Z^h.$$

Thus

$$\gamma\|\boldsymbol{v}^h\|^2 \geq \|\nabla\cdot\boldsymbol{v}^h\|^2 \tag{5.25}$$

does not hold for all $\boldsymbol{v}^h \in Z^h$ and $\gamma = (1-c_1)/c_1$ independent of h. Since $H^1(\Omega)^2$ is the smallest function space where coercivity for the bilinear form $a(\cdot,\cdot)$ is guaranteed to hold, and $H^1(\Omega)^2 \subset H^{\mathrm{div}}(\Omega)$, we can not expect (5.25) to hold on the larger space $H_0^{\mathrm{div}}(\Omega)$. Therefore condition (5.25) does not hold for all conforming elements for (5.18)–(5.19) with $\epsilon > 0$.

If the finite element spaces V^h and Q^h satisfy the condition of Theorem 3.7, a unique solution of (5.21)–(5.22) exists and the solution of (5.21)–(5.22) converges to the solution of the continuous Stokes problem (5.15)–(5.17) as the mesh size, $h \to 0$. Theorem 3.7 provides the following error estimates:

$$|\boldsymbol{u}-\boldsymbol{u}^h|_1 \leq c_1 \inf_{\boldsymbol{v}^h\in V^h}|\boldsymbol{u}-\boldsymbol{v}^h|_1 + c_2\Theta \inf_{q^h\in S^h}\|p-q^h\|, \tag{5.26}$$

and

$$|p-p^h|_1 \leq c_3 \inf_{\boldsymbol{v}^h\in V^h}|\boldsymbol{u}-\boldsymbol{v}^h|_1 + c_4 \inf_{q^h\in S^h}\|p-q^h\|, \tag{5.27}$$

where Θ is the angle between Z^h and Z:

$$\Theta = \sup_{\boldsymbol{u}\in Z^h, |\boldsymbol{u}^h|_1=1}\inf_{\boldsymbol{u}^h\in Z}|\boldsymbol{u}-\boldsymbol{u}^h|_1.$$

Equation (5.15)–(5.17) is discretized using $\mathbb{Q}_2Q_0$, $\mathbb{Q}_2Q_1$, $\mathbb{Q}_3Q_1$ and $\mathbb{Q}_3Q_2$ elements on a uniform structured triangulation of the domain $\Omega = (0,1)\times(0,1)$ consisting of quadrilateral elements. The functions $\boldsymbol{f}$ and $\boldsymbol{g}$ were chosen such that $\boldsymbol{u} = \nabla\times\sin^2(\pi x_1)\sin^2(\pi x_2)$ and $p = \sin(\pi x_1)$. This solution satisfies $\nabla\cdot\boldsymbol{u} = \boldsymbol{g} = 0$ on Ω and $\boldsymbol{u} = 0$ on Γ. A quadrature formula is

used for the load vector, mass and stiffness matrices such that the mass and stiffness matrix are integrated exactly. More precisely, for a degree p velocity element, the mass matrix is of degree $2p$ and a $p+1$ order quadrature rule is required to integrate the mass matrix exactly. The initial mesh of size $h = 2^{-1}$ was uniformly refined until $h = 2^{-8}$. The L^2 and H^1 velocity errors and L^2 pressure error were recorded at each refinement. The resulting saddle point problem is solved using Uzawa with conjugate directions with the Cahouet–Chabard preconditioner with a residual tolerance of $1 \cdot 10^{-14}$.

Table 5.4 shows the L^2 error in velocity, $\|\boldsymbol{u} - \boldsymbol{u}^h\|_0$, the L^2 error in pressure, $\|p - p^h\|_0$, and the H^1 error in velocity, $\|\boldsymbol{u} - \boldsymbol{u}^h\|_1$ for the $\mathbb{Q}_2Q_0$ element with different values of ϵ and h. In the tables we define rate as ratio of error on the previous or coarse mesh to the error on the current mesh. Figure 5.4 shows plots of these errors against h for each ϵ. When $\epsilon = 1$, the L^2 error in velocity appears to be quadratic, while L^2 error in pressure and the H^1 error in velocity appear to be linear. Note that as ϵ decreases, the convergence in L^2 error for velocity and pressure deteriorates and when $\epsilon = 0$ there is no convergence. As ϵ decreases, the convergence in H^1 error in velocity also deteriorates, however convergence is lost for some $\epsilon > 0$.

Table 5.5 and Figure 5.5 show errors in the same norms for the $\mathbb{Q}_2Q_1$ element. When $\epsilon = 1$, the L^2 error in velocity appears to be cubic, while L^2 error in pressure and the H^1 error in velocity appear to be quadratic. As ϵ decreases the convergence in velocity deteriorates, however the convergence in pressure does not change. In fact, when $\epsilon = 0$, the convergence in the L^2 error in velocity is quadratic and the convergence in the H^1 error in velocity is linear.

Table 5.6 and Figure 5.6 show the same errors for the $\mathbb{Q}_3Q_1$ element. As expected, for $\epsilon = 1$, the L^2 error in velocity appears to be cubic and deteriorates as ϵ decreases. For $\epsilon = 0$ the convergence in the L^2 error in velocity is linear. The L^2 error in pressure appears be quadratic over the range $\epsilon \in [0, 1]$. For $\epsilon = 1$ the H^1 error in velocity is quadratic and deteriorates as ϵ decreases. The H^1 error in velocity shows no convergence for $\epsilon = 0$.

Figures 5.7–5.8 and Tables 5.7–5.8 show errors for the $\mathbb{Q}_3Q_2$ and $\mathbb{Q}_4Q_3$ elements. We observe a trend for elements $\mathbb{Q}_pQ_{p-1}$ for $p > 2$. For $\epsilon = 1$, the L^2 error in velocity appears to be order $p+1$, and deteriorates to order $p-1$ for $\epsilon = 0$. The L^2 error in pressure appears be order p over the range $\epsilon \in [0, 1]$. The H^1 error in velocity appears to be order p for $\epsilon = 1$, and deteriorates to order $p-2$ for $\epsilon = 0$.

We also observe for higher order elements that as h decreases the errors

Table 5.4: Convergence table for $\boldsymbol{u} - \epsilon^2 \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f}$, $\nabla \cdot \boldsymbol{u} = 0$ with $\mathbb{Q}_2 Q_0$ elements, third order quadrature.

ϵ	h	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_0 / \|\boldsymbol{u}\|_0$	rate	$\|p - p_h\|_0 / \|p\|_0$	rate	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_1 / \|\boldsymbol{u}\|_1$	rate
1	2^{-1}	$1.13e-01$		$1.00e+00$		$7.94e-02$	
	2^{-2}	$2.48e-02$	4.54	$5.27e-01$	1.90	$2.75e-02$	2.89
	2^{-3}	$3.54e-03$	7.02	$2.63e-01$	2.00	$7.13e-03$	3.85
	2^{-4}	$5.74e-04$	6.16	$1.31e-01$	2.01	$1.94e-03$	3.68
	2^{-5}	$1.16e-04$	4.95	$6.53e-02$	2.01	$6.13e-04$	3.16
	2^{-6}	$2.71e-05$	4.28	$3.26e-02$	2.00	$2.43e-04$	2.53
	2^{-7}	$6.66e-06$	4.07	$1.63e-02$	2.00	$1.12e-04$	2.17
	2^{-8}	$1.66e-06$	4.01	$8.14e-03$	2.00	$5.48e-05$	2.04
2^{-2}	2^{-1}	$2.45e-01$		$1.00e+00$		$1.03e-01$	
	2^{-2}	$8.71e-02$	2.81	$5.30e-01$	1.89	$5.37e-02$	1.93
	2^{-3}	$2.48e-02$	3.51	$2.65e-01$	2.00	$2.70e-02$	1.99
	2^{-4}	$6.54e-03$	3.79	$1.32e-01$	2.02	$1.37e-02$	1.97
	2^{-5}	$1.67e-03$	3.92	$6.54e-02$	2.01	$6.91e-03$	1.98
	2^{-6}	$4.21e-04$	3.97	$3.26e-02$	2.01	$3.47e-03$	1.99
	2^{-7}	$1.06e-04$	3.99	$1.63e-02$	2.00	$1.74e-03$	2.00
	2^{-8}	$2.65e-05$	3.99	$8.14e-03$	2.00	$8.70e-04$	2.00
2^{-4}	2^{-1}	$7.74e-01$		$1.00e+00$		$2.53e-01$	
	2^{-2}	$5.90e-01$	1.31	$6.74e-01$	1.48	$3.32e-01$	0.762
	2^{-3}	$2.87e-01$	2.06	$3.59e-01$	1.88	$3.08e-01$	1.08
	2^{-4}	$9.43e-02$	3.04	$1.56e-01$	2.30	$1.99e-01$	1.55
	2^{-5}	$2.58e-02$	3.66	$6.95e-02$	2.25	$1.08e-01$	1.85
	2^{-6}	$6.62e-03$	3.89	$3.32e-02$	2.09	$5.52e-02$	1.95
	2^{-7}	$1.67e-03$	3.97	$1.64e-02$	2.03	$2.78e-02$	1.99
	2^{-8}	$4.18e-04$	3.99	$8.15e-03$	2.01	$1.39e-02$	2.00
2^{-8}	2^{-1}	$9.39e-01$		$1.00e+00$		$3.13e-01$	
	2^{-2}	$1.00e+00$	0.935	$8.55e-01$	1.17	$5.82e-01$	0.538
	2^{-3}	$1.02e+00$	0.981	$8.30e-01$	1.03	$1.13e+00$	0.515
	2^{-4}	$1.00e+00$	1.02	$8.04e-01$	1.03	$2.17e+00$	0.522
	2^{-5}	$9.06e-01$	1.11	$7.22e-01$	1.11	$3.86e+00$	0.561
	2^{-6}	$6.46e-01$	1.40	$5.14e-01$	1.40	$5.47e+00$	0.705
	2^{-7}	$3.01e-01$	2.15	$2.40e-01$	2.15	$5.08e+00$	1.08
	2^{-8}	$9.58e-02$	3.14	$7.67e-02$	3.13	$3.24e+00$	1.57
0	2^{-1}	$9.39e-01$		$1.00e+00$		$3.13e-01$	
	2^{-2}	$1.01e+00$	0.930	$8.57e-01$	1.17	$5.84e-01$	0.536
	2^{-3}	$1.03e+00$	0.974	$8.38e-01$	1.02	$1.14e+00$	0.510
	2^{-4}	$1.05e+00$	0.995	$8.34e-01$	1.00	$2.26e+00$	0.506
	2^{-5}	$1.05e+00$	0.996	$8.34e-01$	1.00	$4.49e+00$	0.504
	2^{-6}	$1.05e+00$	0.998	$8.33e-01$	1.00	$8.94e+00$	0.502
	2^{-7}	$1.05e+00$	0.999	$8.33e-01$	1.00	$1.79e+01$	0.501
	2^{-8}	$1.05e+00$	1.00	$8.33e-01$	1.00	$3.57e+01$	0.500

Table 5.5: Convergence table for $\boldsymbol{u} - \epsilon^2 \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f}$, $\nabla \cdot \boldsymbol{u} = 0$ with $\mathbb{Q}_2 Q_1$ elements, third order quadrature.

ϵ	h	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_0 / \|\boldsymbol{u}\|_0$	rate	$\|p - p_h\|_0 / \|p\|_0$	rate	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_1 / \|\boldsymbol{u}\|_1$	rate
1	2^{-1}	$1.11e-01$		$2.04e-01$		$7.92e-02$	
	2^{-2}	$2.43e-02$	4.59	$4.57e-01$	0.447	$2.74e-02$	2.89
	2^{-3}	$3.18e-03$	7.63	$3.71e-02$	12.3	$6.93e-03$	3.95
	2^{-4}	$4.01e-04$	7.93	$4.17e-03$	8.88	$1.74e-03$	3.99
	2^{-5}	$5.02e-05$	7.98	$8.49e-04$	4.92	$4.35e-04$	4.00
	2^{-6}	$6.28e-06$	8.00	$2.07e-04$	4.10	$1.09e-04$	4.00
	2^{-7}	$7.85e-07$	8.00	$5.17e-05$	4.00	$2.72e-05$	4.00
	2^{-8}	$9.82e-08$	8.00	$1.29e-05$	4.00	$6.80e-06$	4.00
2^{-2}	2^{-1}	$1.09e-01$		$2.04e-01$		$7.93e-02$	
	2^{-2}	$2.40e-02$	4.55	$6.33e-02$	3.22	$2.74e-02$	2.89
	2^{-3}	$3.17e-03$	7.55	$1.36e-02$	4.65	$6.94e-03$	3.95
	2^{-4}	$4.01e-04$	7.91	$3.32e-03$	4.10	$1.74e-03$	3.99
	2^{-5}	$5.02e-05$	7.98	$8.26e-04$	4.02	$4.35e-04$	4.00
	2^{-6}	$6.28e-06$	8.00	$2.06e-04$	4.00	$1.09e-04$	4.00
	2^{-7}	$7.88e-07$	7.97	$5.16e-05$	4.00	$2.72e-05$	4.00
	2^{-8}	$9.96e-08$	7.91	$1.29e-05$	4.00	$6.80e-06$	4.00
2^{-4}	2^{-1}	$9.95e-02$		$2.04e-01$		$8.16e-02$	
	2^{-2}	$3.00e-02$	3.31	$5.58e-02$	3.65	$3.37e-02$	2.42
	2^{-3}	$3.57e-03$	8.42	$1.34e-02$	4.16	$7.91e-03$	4.26
	2^{-4}	$4.15e-04$	8.59	$3.32e-03$	4.05	$1.82e-03$	4.35
	2^{-5}	$5.07e-05$	8.19	$8.26e-04$	4.01	$4.41e-04$	4.13
	2^{-6}	$6.45e-06$	7.86	$2.06e-04$	4.00	$1.09e-04$	4.04
	2^{-7}	$8.50e-07$	7.59	$5.16e-05$	4.00	$2.72e-05$	4.01
	2^{-8}	$5.55e-07$	1.53	$1.30e-05$	3.98	$6.96e-06$	3.92
2^{-8}	2^{-1}	$9.74e-02$		$2.04e-01$		$8.51e-02$	
	2^{-2}	$6.39e-02$	1.52	$5.75e-02$	3.55	$6.46e-02$	1.32
	2^{-3}	$2.05e-02$	3.12	$1.35e-02$	4.25	$4.76e-02$	1.36
	2^{-4}	$4.96e-03$	4.13	$3.32e-03$	4.08	$2.46e-02$	1.94
	2^{-5}	$8.35e-04$	5.94	$8.26e-04$	4.02	$8.45e-03$	2.91
	2^{-6}	$9.08e-05$	9.20	$2.07e-04$	4.00	$1.79e-03$	4.72
	2^{-7}	$1.19e-05$	7.60	$5.17e-05$	4.00	$2.87e-04$	6.24
	2^{-8}	$1.32e-05$	0.90	$1.34e-05$	3.85	$2.10e-04$	1.37
0	2^{-1}	$9.74e-02$		$2.04e-01$		$8.51e-02$	
	2^{-2}	$6.44e-02$	1.51	$5.75e-02$	3.54	$6.50e-02$	1.31
	2^{-3}	$2.14e-02$	3.01	$1.35e-02$	4.25	$4.98e-02$	1.31
	2^{-4}	$6.02e-03$	3.56	$3.32e-03$	4.08	$2.99e-02$	1.67
	2^{-5}	$1.58e-03$	3.80	$8.26e-04$	4.02	$1.61e-02$	1.86
	2^{-6}	$4.06e-04$	3.90	$2.07e-04$	4.00	$8.33e-03$	1.93
	2^{-7}	$1.10e-04$	3.71	$5.29e-05$	3.90	$4.29e-03$	1.94
	2^{-8}	$6.41e-05$	1.71	$1.82e-05$	2.90	$3.09e-03$	1.39

Table 5.6: Convergence table for $\boldsymbol{u} - \epsilon^2 \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f}$, $\nabla \cdot \boldsymbol{u} = 0$ with $\mathbb{Q}_3 Q_1$ elements, fourth order quadrature.

ϵ	h	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_0 / \|\boldsymbol{u}\|_0$	rate	$\|p - p_h\|_0 / \|p\|_0$	rate	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_1 / \|\boldsymbol{u}\|_1$	rate
1	2^{-1}	$3.08e-02$		$2.04e-01$		$3.30e-02$	
	2^{-2}	$1.83e-03$	16.9	$5.58e-02$	3.65	$3.64e-03$	9.07
	2^{-3}	$1.23e-04$	14.9	$1.34e-02$	4.16	$4.68e-04$	7.77
	2^{-4}	$8.88e-06$	13.8	$3.32e-03$	4.04	$6.17e-05$	7.58
	2^{-5}	$7.81e-07$	11.4	$8.26e-04$	4.01	$9.07e-06$	6.80
	2^{-6}	$8.43e-08$	9.26	$2.06e-04$	4.00	$1.65e-06$	5.51
	2^{-7}	$1.01e-08$	8.35	$5.16e-05$	4.00	$3.62e-07$	4.54
	2^{-8}	$1.36e-09$	7.42	$1.29e-05$	4.00	$8.73e-08$	4.15
2^{-2}	2^{-1}	$4.40e-02$		$2.04e-01$		$3.77e-02$	
	2^{-2}	$5.74e-03$	7.67	$5.54e-02$	3.68	$6.58e-03$	5.72
	2^{-3}	$6.66e-04$	8.62	$1.34e-02$	4.13	$1.47e-03$	4.48
	2^{-4}	$8.14e-05$	8.18	$3.32e-03$	4.04	$3.55e-04$	4.14
	2^{-5}	$1.01e-05$	8.02	$8.26e-04$	4.01	$8.82e-05$	4.03
	2^{-6}	$1.27e-06$	7.99	$2.06e-04$	4.00	$2.20e-05$	4.00
	2^{-7}	$1.59e-07$	7.99	$5.16e-05$	4.00	$5.51e-06$	4.00
	2^{-8}	$2.01e-08$	7.91	$1.29e-05$	4.00	$1.38e-06$	4.00
2^{-4}	2^{-1}	$2.35e-01$		$2.04e-01$		$1.38e-01$	
	2^{-2}	$6.13e-02$	3.84	$5.65e-02$	3.61	$6.37e-02$	2.16
	2^{-3}	$9.59e-03$	6.39	$1.34e-02$	4.21	$2.04e-02$	3.11
	2^{-4}	$1.27e-03$	7.56	$3.32e-03$	4.05	$5.48e-03$	3.73
	2^{-5}	$1.61e-04$	7.87	$8.26e-04$	4.01	$1.40e-03$	3.92
	2^{-6}	$2.03e-05$	7.95	$2.06e-04$	4.00	$3.52e-04$	3.98
	2^{-7}	$2.54e-06$	7.98	$5.16e-05$	4.00	$8.81e-05$	3.99
	2^{-8}	$3.20e-07$	7.95	$1.29e-05$	4.00	$2.20e-05$	4.00
2^{-8}	2^{-1}	$4.06e-01$		$2.04e-01$		$2.39e-01$	
	2^{-2}	$2.11e-01$	1.92	$6.17e-02$	3.30	$2.33e-01$	1.02
	2^{-3}	$1.04e-01$	2.03	$1.57e-02$	3.93	$2.28e-01$	1.02
	2^{-4}	$4.67e-02$	2.22	$3.83e-03$	4.10	$2.04e-01$	1.12
	2^{-5}	$1.65e-02$	2.84	$8.93e-04$	4.28	$1.43e-01$	1.42
	2^{-6}	$3.77e-03$	4.37	$2.10e-04$	4.25	$6.53e-02$	2.19
	2^{-7}	$5.94e-04$	6.34	$5.17e-05$	4.07	$2.06e-02$	3.17
	2^{-8}	$7.96e-05$	7.47	$1.29e-05$	4.00	$5.51e-03$	3.74
0	2^{-1}	$4.07e-01$		$2.04e-01$		$2.40e-01$	
	2^{-2}	$2.13e-01$	1.91	$6.18e-02$	3.30	$2.36e-01$	1.02
	2^{-3}	$1.08e-01$	1.97	$1.59e-02$	3.90	$2.38e-01$	0.99
	2^{-4}	$5.45e-02$	1.99	$3.99e-03$	3.98	$2.38e-01$	1.00
	2^{-5}	$2.73e-02$	2.00	$9.98e-04$	4.00	$2.38e-01$	1.00
	2^{-6}	$1.37e-02$	2.00	$2.49e-04$	4.00	$2.37e-01$	1.00
	2^{-7}	$6.84e-03$	2.00	$6.22e-05$	4.01	$2.37e-01$	1.00
	2^{-8}	$3.42e-03$	2.00	$1.54e-05$	4.03	$2.37e-01$	1.00

Table 5.7: Convergence table for $\boldsymbol{u} - \epsilon^2 \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f}$, $\nabla \cdot \boldsymbol{u} = 0$ with $\mathbb{Q}_3 Q_2$ elements, fourth order quadrature.

ϵ	h	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_0 / \|\boldsymbol{u}\|_0$	rate	$\|p - p_h\|_0 / \|p\|_0$	rate	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_1 / \|\boldsymbol{u}\|_1$	rate
1	2^{-1}	$3.11e-02$		$7.57e-01$		$3.32e-02$	
	2^{-2}	$1.78e-03$	17.4	$3.14e-02$	24.1	$3.62e-03$	9.16
	2^{-3}	$1.15e-04$	15.5	$2.17e-03$	14.5	$4.60e-04$	7.88
	2^{-4}	$7.27e-06$	15.8	$1.67e-04$	13.0	$5.77e-05$	7.97
	2^{-5}	$4.56e-07$	15.94	$1.50e-05$	11.1	$7.22e-06$	7.99
	2^{-6}	$2.86e-08$	16.0	$1.65e-06$	9.11	$9.03e-07$	8.00
	2^{-7}	$1.86e-09$	15.4	$2.50e-07$	6.59	$1.13e-07$	8.00
	2^{-8}	$1.51e-10$	12.3	$4.72e-08$	5.30	$1.41e-08$	8.00
2^{-2}	2^{-1}	$3.07e-02$		$5.81e-02$		$3.32e-02$	
	2^{-2}	$1.77e-03$	17.3	$4.94e-03$	11.8	$3.64e-03$	9.13
	2^{-3}	$1.16e-04$	15.3	$6.95e-04$	7.11	$4.63e-04$	7.85
	2^{-4}	$7.41e-06$	15.7	$9.35e-05$	7.43	$5.83e-05$	7.95
	2^{-5}	$4.67e-07$	15.9	$1.21e-05$	7.73	$7.30e-06$	7.98
	2^{-6}	$2.97e-08$	15.7	$1.55e-06$	7.83	$9.13e-07$	7.99
	2^{-7}	$9.36e-09$	3.17	$3.39e-07$	4.57	$1.17e-07$	7.84
	2^{-8}	$3.88e-09$	2.41	$1.13e-07$	3.00	$1.68e-08$	6.95
2^{-4}	2^{-1}	$3.01e-02$		$2.85e-02$		$3.33e-02$	
	2^{-2}	$2.90e-03$	10.4	$4.54e-03$	6.27	$5.55e-03$	6.00
	2^{-3}	$3.10e-04$	9.35	$6.83e-04$	6.64	$9.57e-04$	5.80
	2^{-4}	$2.43e-05$	12.7	$9.31e-05$	7.34	$1.38e-04$	6.91
	2^{-5}	$1.67e-06$	14.6	$1.21e-05$	7.70	$1.84e-05$	7.53
	2^{-6}	$1.22e-07$	13.6	$1.54e-06$	7.85	$2.36e-06$	7.79
	2^{-7}	$1.65e-08$	7.44	$1.97e-07$	7.84	$3.00e-07$	7.86
	2^{-8}	$3.11e-08$	0.529	$7.10e-08$	2.77	$7.96e-08$	3.77
2^{-8}	2^{-1}	$3.00e-02$		$2.79e-02$		$3.33e-02$	
	2^{-2}	$1.82e-02$	1.65	$4.81e-03$	5.79	$3.19e-02$	1.05
	2^{-3}	$5.36e-03$	3.39	$7.07e-04$	6.81	$1.58e-02$	2.01
	2^{-4}	$1.23e-03$	4.35	$9.42e-05$	7.51	$6.75e-03$	2.34
	2^{-5}	$2.03e-04$	6.08	$1.21e-05$	7.77	$2.15e-03$	3.14
	2^{-6}	$2.12e-05$	9.59	$1.54e-06$	7.87	$4.42e-04$	4.86
	2^{-7}	$1.64e-06$	12.9	$1.98e-07$	7.78	$6.67e-05$	6.63
	2^{-8}	$7.38e-07$	2.22	$7.82e-08$	2.53	$1.90e-05$	3.51
0	2^{-1}	$3.00e-02$		$2.79e-02$		$3.33e-02$	
	2^{-2}	$1.87e-02$	1.61	$4.83e-03$	5.77	$3.27e-02$	1.02
	2^{-3}	$5.77e-03$	3.24	$7.11e-04$	6.79	$1.71e-02$	1.91
	2^{-4}	$1.56e-03$	3.71	$9.48e-05$	7.50	$8.58e-03$	1.99
	2^{-5}	$4.03e-04$	3.87	$1.22e-05$	7.77	$4.29e-03$	2.00
	2^{-6}	$1.02e-04$	3.94	$1.56e-06$	7.83	$2.15e-03$	2.00
	2^{-7}	$2.59e-05$	3.95	$2.34e-07$	6.67	$1.08e-03$	1.99
	2^{-8}	$7.90e-06$	3.28	$8.58e-08$	2.72	$6.45e-04$	1.67

Table 5.8: Convergence table for $\boldsymbol{u} - \epsilon^2 \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f}$, $\nabla \cdot \boldsymbol{u} = 0$ with $\mathbb{Q}_4 Q_3$ elements, fifth order quadrature.

ϵ	h	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_0 / \|\boldsymbol{u}\|_0$	rate	$\|p - p_h\|_0 / \|p\|_0$	rate	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_1 / \|\boldsymbol{u}\|_1$	rate
1	2^{-1}	$3.62e-03$		$2.18e-02$		$3.96e-03$	
	2^{-2}	$1.72e-04$	21.0	$2.47e-03$	8.83	$3.59e-04$	11.0
	2^{-3}	$5.49e-06$	31.4	$6.20e-05$	39.8	$2.28e-05$	15.8
	2^{-4}	$1.72e-07$	31.9	$1.79e-06$	34.7	$1.43e-06$	15.9
	2^{-5}	$5.39e-09$	32.0	$7.97e-08$	22.4	$8.93e-08$	16.0
	2^{-6}	$2.45e-10$	22.0	$4.01e-08$	1.99	$5.59e-09$	16.0
	2^{-7}	$8.08e-11$	3.03	$2.35e-08$	1.71	$3.61e-10$	15.5
2^{-2}	2^{-1}	$3.60e-03$		$3.29e-03$		$3.96e-03$	
	2^{-2}	$1.72e-04$	20.9	$2.38e-04$	13.8	$3.59e-04$	11.0
	2^{-3}	$5.48e-06$	31.3	$1.16e-05$	20.5	$2.28e-05$	15.8
	2^{-4}	$1.72e-07$	31.9	$6.89e-07$	16.9	$1.43e-06$	15.9
	2^{-5}	$5.57e-09$	30.9	$5.33e-08$	12.9	$8.94e-08$	16.0
	2^{-6}	$7.41e-10$	7.51	$2.18e-08$	2.45	$5.80e-09$	15.4
	2^{-7}	$7.80e-10$	0.950	$2.87e-08$	0.761	$2.27e-09$	2.55
2^{-4}	2^{-1}	$3.39e-03$		$2.77e-03$		$4.00e-03$	
	2^{-2}	$1.68e-04$	20.2	$1.79e-04$	15.5	$3.66e-04$	10.9
	2^{-3}	$5.44e-06$	30.8	$1.09e-05$	16.3	$2.29e-05$	16.0
	2^{-4}	$1.73e-07$	31.5	$6.81e-07$	16.1	$1.43e-06$	16.0
	2^{-5}	$6.56e-09$	26.3	$4.33e-08$	15.7	$8.99e-08$	15.9
	2^{-6}	$5.76e-09$	1.14	$1.44e-08$	3.01	$1.71e-08$	5.25
	2^{-7}	$7.95e-09$	0.724	$2.01e-08$	0.716	$2.39e-08$	0.716
2^{-8}	2^{-1}	$2.99e-03$		$2.73e-03$		$4.79e-03$	
	2^{-2}	$4.40e-04$	6.80	$1.80e-04$	15.1	$1.19e-03$	4.01
	2^{-3}	$2.81e-05$	15.6	$1.10e-05$	16.4	$1.67e-04$	7.15
	2^{-4}	$1.09e-06$	25.7	$6.80e-07$	16.1	$1.34e-05$	12.5
	2^{-5}	$9.64e-08$	11.4	$4.31e-08$	15.8	$1.47e-06$	9.09
	2^{-6}	$3.07e-07$	0.314	$2.84e-08$	1.52	$6.42e-06$	0.229
	2^{-7}	$3.22e-07$	0.953	$3.96e-08$	0.717	$1.00e-05$	0.642
0	2^{-1}	$2.99e-03$		$2.73e-03$		$4.81e-03$	
	2^{-2}	$4.65e-04$	6.44	$1.81e-04$	15.1	$1.26e-03$	3.82
	2^{-3}	$3.67e-05$	12.7	$1.10e-05$	16.4	$2.18e-04$	5.78
	2^{-4}	$2.52e-06$	14.5	$6.82e-07$	16.1	$3.12e-05$	7.01
	2^{-5}	$3.41e-07$	7.41	$6.25e-08$	10.9	$6.40e-06$	4.88
	2^{-6}	$4.70e-07$	0.725	$4.19e-08$	1.49	$1.54e-05$	0.416
	2^{-7}	$2.66e-06$	0.176	$1.53e-07$	0.274	$1.61e-04$	0.0959

deteriorate for both velocity and pressure. This is because the error is dominated by the error in solving the discrete system. Even though the solution to the discrete system gives a small residual, this contribution to the L^2 error is larger than the high order spatial accuracy. Using the inverse estimate [64, Proposition 6.3.1 p. 194], there exists c_1, $c_2 > 0$ such that for each $\boldsymbol{v}^h \in V^h$ with $\boldsymbol{v}^h = \sum_i \mathrm{v}_i \boldsymbol{\varphi}_i$

$$c_1 h^d |\mathbf{v}|^2 \leq \|\boldsymbol{v}^h\|_0^2 \leq c_2 h^d |\mathbf{v}|^2,$$

we can relate the residual tolerance to the computed error. When $\epsilon = 1$,

$$\|\boldsymbol{u} - \boldsymbol{u}^h\|_0 \leq c h^{p+1} + h^{-1} \epsilon_r,$$

where c is a constant and ϵ_r is the residual tolerance of the discrete system. If ϵ_r is chosen such that $\epsilon_r \ll 1$, then for large h and small p the term $h^{-1}\epsilon_r$ is negligible. However when h is small and p is large, the last term may contribute significantly to the error. Similar results hold for the L^2 error in pressure and H^1 error in velocity.

In conclusion we observe that finite elements that are stable for the Stokes equations are not stable for computation of the application of the divergence–free L^2 projection as $h \to 0$, in particular convergence can be suboptimal. However, reasonable convergence can be retained if the $h \to 0$ cases are avoided. We remark that despite the lack of stability, the divergence–free L^2 projection has been implemented as in Section 5.2 with success in a series of papers by Gresho [35, 33, 34]. In practice we prefer to compute the divergence–free L^2 projection using the approach outlined in this section because it allows us to specify pure Dirichlet boundary conditions for the divergence–free velocity with a minimal loss of stability. We conclude with the reminder that Stokes elements provide a stable discretization for the divergence–free H^1 projection and lead to a discrete system that is efficiently solved using the techniques of Section 5.1. Thus the divergence–free H^1 projection provides an alternative to the divergence–free L^2 projection.

Chapter 6

Numerical experiments

In this chapter we present results of a numerical experiment that show that higher–order temporal convergence can be achieved for the Navier–Stokes equations with lower computational cost than a Crank Nicolson implementation.

6.1 Implementation

Computer code for the numerical experiments in this chapter and Chapter 2 were written in C++. We make extensive use of the `deal.II` finite element library [3]. `deal.II` is a C++ program library targeted at adaptive finite elements and error estimation. It makes extensive use of modern object–oriented programming techniques of the C++ programming language. It provides an interface to the complex data structures and algorithms required for adaptivity, a variety of finite elements in up to three space dimensions, and the capability for time-dependent problems.

6.2 Numerical experiment

In this section we compare accuracy and cost for the Krylov–based exponential projection method and an implementation of the Crank Nicolson method for the full Navier–Stokes equations (3.15)–(3.17).

The experimental problem was chosen such that the exact solution is $\boldsymbol{u}(\boldsymbol{x},t) = \nabla \times \sin^2 \pi x_1 \sin^2 \pi x_2 \sin \frac{\pi}{2} t$ and $p(\boldsymbol{x},t) = \sin \pi x_1 - 2/\pi$. Equations (3.15)–(3.17) were discretized using 1024 $\mathbb{Q}_2 Q_1$ elements on a structured

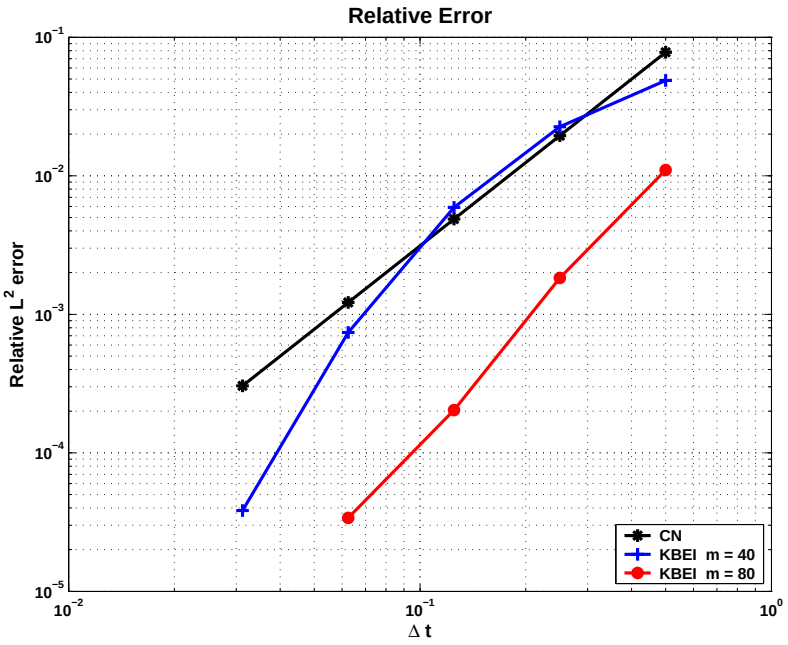

Figure 6.1: Relative L^2 error in velocity at $t = 1$ for Crank Nicolson method (black), KBEI with $m = 40$ (blue) and $m = 80$ (red).

triangulation of $\Omega = (0,1) \times (0,1)$. This discretization yields the system (3.45)–(3.47) with 8450 unknowns for the velocity space and 1089 unknowns for the pressure space. We integrate (3.45)–(3.47) from $t = 0$ to $t = 1$ using the Krylov–based exponential projection method and Crank Nicolson methods with time step lengths $\Delta t = .5, .25, \ldots, .03125$.

Since $\mathbf{f}$ depends on t, the Krylov–based exponential projection method utilizes a nonautonomous formulation of the Jacobian given by (4.41), the Darcy flow formulation of the L^2 projection and the KBEI given by (2.14)–(2.26). Application of the Crank Nicolson method to (3.45)–(3.47) results in a fully coupled nonlinear system that needs to be solved at each time step. The nonlinear system is solved with a Newton method. Each Newton iteration requires a linear solve with an unsymmetric, indefinite system, that is solved using standard GMRES without preconditioning. The residual tolerance for both the Newton and GMRES iterations was chosen based on Δt as given in [35, p. 801]. We note that second order temporal convergence is verified with the tolerances chosen in this manner (see Figure 6.1).

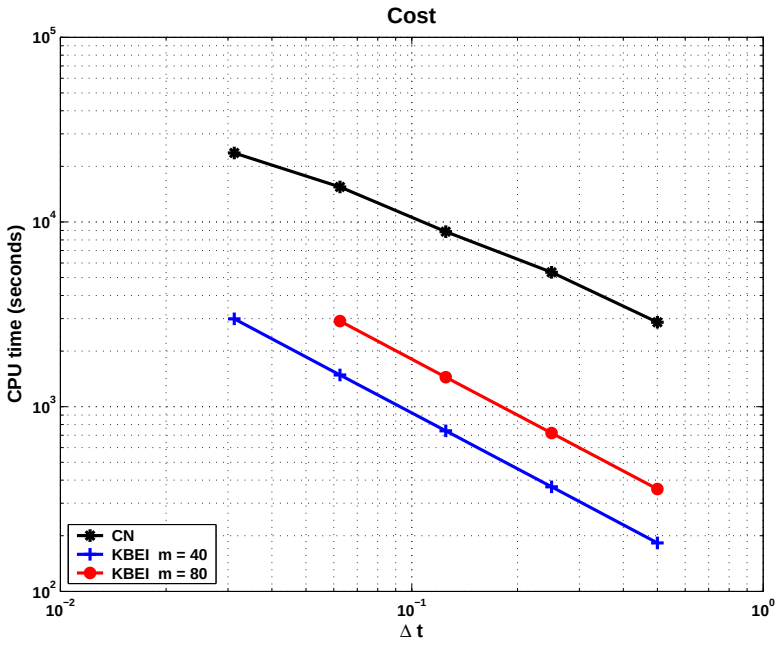

Figure 6.2: Cost in CPU time for Crank Nicolson method (black), KBEI with $m = 40$ (blue) and $m = 80$ (red).

We use Crank Nicolson as a comparison because of its easy implementation, unconditional stability and second order accuracy. We do not consider explicit methods because of their inherent stability limitations; nor do we consider first order methods. Common second order implicit methods that we could have considered for a comparison include the backward difference method and the implicit midpoint rule.

Figure 6.1 shows relative L^2 error in the velocity at $t = 1$ versus time step size for the Krylov–based exponential projection method with $m = 40$, $m = 80$ and the Crank Nicolson method. Second order temporal convergence for the Crank Nicolson method is evident. The convergence rate for the Krylov–based exponential projection method with $m = 40$ is less than second order for $\Delta t > 10^{-1}$ and improves to greater than second order for $\Delta t < 10^{-1}$. For $m = 80$, the Krylov–based exponential projection method exhibits convergence greater that second order for $\Delta t < 5 \times 10^{-1}$. Thus temporal convergence greater than second order can be attained using the Krylov–based exponential projection method.

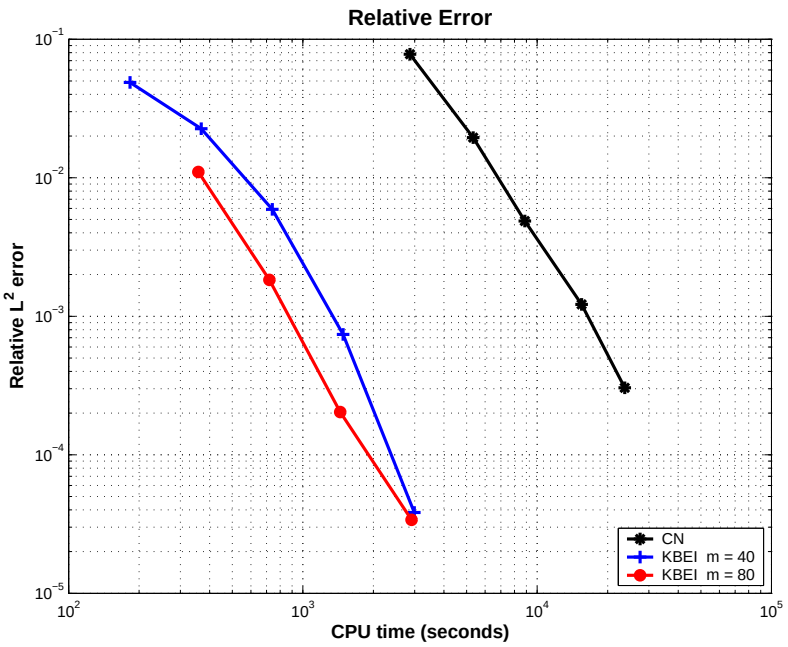

Figure 6.3: Cost versus Relative L^2 error in velocity at $t = 1$ for Crank Nicolson method (black), KBEI with $m = 40$ (blue) and $m = 80$ (red).

The cost measured in CPU seconds of each method versus time step length is displayed in Figure 6.2. For integration to $t = 1$, the Crank Nicolson method requires more CPU time than the Krylov–based exponential projection method for any choice of Δt. Figure 6.3 shows cost versus L^2 error for each of the methods. In particular, we observe that the Krylov–based exponential projection method with $m = 40$ and $m = 80$ achieve a smaller error at a much smaller cost than the Crank Nicolson method. We remark that the large cost in Crank Nicolson is due to the need to solve a nonlinear system at each time step. We recall that no nonlinear solve is required by the Krylov–based exponential projection method.

6.3 Summary

The results suggest that the methods described herein show promise and can provide an alternative to conventional methods. In summary, we note that these experiments and the experiments in Chapter 2 do not show beyond

reasonable doubt that the Krylov–based exponential projection method outperforms conventional time integration methods for the Navier–Stokes equations.

Crank Nicolson possesses some advantages over other higher–order implicit methods. Although the family of higher–order backward difference methods possesses favorable stability properties, it suffers from artificial dissipation. Backward difference methods also require that information from the previous two time steps be retained. This facilitates the need to implement another integration method to reach the first time step. Another high order alternative is the implicit midpoint rule. A drawback of the implicit midpoint rule is that it requires function evaluations at mid time step. Implicit Runge–Kutta methods [10, 40] and Rosenbrock methods [51] provide high order accuracy at the expense of multiple linear solves at each timestep. We note that if proper care is given in the sense that associated stability criterions are satisfied, explicit methods may also be useful. In particular Runge–Kutta methods and Adams–Bashforth methods [35] provide higher order explicit methods.

The results of this chapter are positive and indicate promise. However, one cannot conclude that traditional methods are inferior. Careful evaluations against the methods mentioned in this section must be performed.

Chapter 7

Conclusions

7.1 Summary

In this work we have studied KBEIs and used them to provided an algorithm for time integration of the incompressible Navier–Stokes equations. Our analysis of KBEIs highlighted the important issues in implementation of such a method and demonstrates that Krylov–based exponential integration methods provide an alternative to both implicit and explicit methods.

We have outlined in detail how Krylov–based exponential methods can be applied to the incompressible Navier–Stokes equations. The two key building blocks of our algorithm consist of projection of the momentum equation to a divergence–free subspace and integration of the momentum equation with a KBEI. We have highlighted the theoretical and practical issues for implementation for both of these building blocks. In particular we have described how to project the momentum equation onto a divergence–free subspace using a divergence–free L^2 or H^1 projection and obtain scalable solution of the projection subproblems using a preconditioned Uzawa method.

7.2 Contributions

The main contribution of this work is the identification and development of scalable solver for time integration of the incompressible Navier–Stokes equations. We have also contributed an analysis of the divergence–free L^2 and H^1 projections and have shown that the seldom used divergence–free H^1 projection provides a viable alternative to the divergence–free L^2 projection.

An analysis of a scalable preconditioned solution method for each projection has been outlined. We have also addressed the stability of finite element methods for the projection subproblems.

7.3 Future work

There are four research areas to pursue. The first lies in the area of KBEIs. We plan on providing a theoretical description of scalability for KBEIs applied to not only discretizations the incompressible Navier–Stokes equations but to partial differential equations in general. We would also like to conduct a theoretical study to identify the types of problems in which KBEIs work well. Another area of interest is the integration of KBEIs with methods for solving steady state problems and operator–splitting methods.

The other area of research is incompressible flow. In particular we plan to study implementation of other boundary conditions in our algorithm for the Navier–Stokes equations. In particular we are interested in how the divergence–free projections accommodate boundary conditions other than Dirichlet.

The third area is code development. We plan on implementing the solver for three spatial dimensions. Another improvement to the code would be to use multigrid for the inner and precondition iterations in our Uzawa solver for the divergence–free L^2 or H^1 projections. These two steps are outlined in Section 5.1 and consist of solving a Poisson problem on either the pressure space or the velocity space. This Poisson problem can be solved efficiently using multigrid methods [11]. We also plan to identify components of the code that can be parallelized. The development of parallel code would allow us to run larger problems.

Finally, the method needs to be thoroughly evaluated against other standard approaches including the high order explicit and implicit methods and the semi–implicit projection methods.

Bibliography

[1] Y. Achdou and J. Guermond. Convergence analysis of a finite element projection/Lagrange–Galerkin method for the incompressible Navier–Stokes equations. SIAM J. Numer. Anal., 37(3):799–826, 2000.

[2] L. V. Ahlfors. *Complex Analysis*. McGraw Hill, third edition, 1979.

[3] W. Bangerth, R. Hartmann and G. Kanschat. `deal.II` *Differential Equations Analysis Library, Technical Reference*. IWR. `http://www.dealii.org`.

[4] J. B. Bell, P. Colella and H. M. Glaz. A second order projection method for the incompressible Navier–Stokes equations. J. Comput. Phys., 85:257–283, 1989.

[5] D. Braess. *Finite Elements*. Cambridge University Press, Cambridge, 1997.

[6] J. H. Bramble and J. E. Pasciak. A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems. Math. Comp., 50(181):1 – 17, 1988.

[7] J. H. Bramble and J. E. Pasciak. A domain decomposition technique for Stokes problems. Appl. Numer. Math., 6(4):251 – 261, 1990.

[8] J. H. Bramble and J. E. Pasciak. Iterative techniques for time dependent Stokes problems. Comput. Math. Appl., 33(1-2):13–30, 1997.

[9] J. H. Bramble, J. E. Pasciak and A. T. Vassilev. Analysis of the inexact Uzawa algorithm for saddle point problems. SIAM J. Numer. Anal., 34(3):1072 – 1092, 1997.

[10] K. E. Brenan, S. L. Campbell and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. SIAM, Philadelphia, 1996.

[11] W. L. Briggs, V. E. Henson and S. F. McCormick. *A Multigrid Tutorial*. Society for Industrial and Applied Mathematics, second edition, 2000.

[12] D. L. Brown. Accuracy of projection methods for the incompressible Navier–Stokes equations. Technical report, Lawrence Livermore National Laboratory, 2001.

[13] P. N. Brown and A. C. Hindmarsh. Matrix-free methods for stiff systems of ODEs. SIAM J. Numer. Anal., 23(3):610–638, 1986.

[14] J. Cahouet and J.-P. Chabard. Some fast 3D finite element solvers for the generalized Stokes problem. Int. J. Numer. Meth. Fluids, 8:869–895, 1988.

[15] A. J. Chorin. Numerical solution of the Navier–Stokes equations. Math. Comput., 22(104):745–768, 1968.

[16] A. J. Chorin. On the convergence of discrete approximations to the Navier–Stokes equations. Math. Comput., 23(106):341–353, 1969.

[17] R. Codina. Pressure stability in fractional step finite element methods for incompressible flows. J. Comput. Phys., 170:112–140, 2001.

[18] P. Colella and K. Pao. A projection method for low speed flows. J. Comput. Phys., 149:245–269, 1999.

[19] A. Dagan. Numerical consistency and spurious boundary layer in the projection method. Computers and Fluids, 32(9):1213 – 1232, 2003.

[20] R. Dautray and J. L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology*, volume 3. Springer–Verlag, 1990.

[21] W. E and J. Liu. Projection method I: Convergence and numerical boundary layers. SIAM J. Numer. Anal., 32(4):1017–1057, 1995.

[22] W. E and J. Liu. Projection method II: Godunov–Ryabenki analysis. SIAM J. Numer. Anal., 33(4):1597–1621, 1996.

[23] W. E and J. Liu. Vorticity boundary condition and related issues for finite difference schemes. J. Comput. Phys., 124:368–382, 1996.

[24] W. E and J. Liu. Projection method III: Spatial discretization on the staggered grid. Math. Comp., 71(237):27–47, 2001.

[25] W. S. Edwards, L. S. Tuckerman, R. A. Friesner and D. C. Sorensen. Krylov methods for the incompressible Navier–Stokes equations. J. Comput. Phys., 110:82–102, 1994.

[26] B. L. Ehle. *A*–stable methods and Padé approximations to the exponential. SIAM J. Math. Anal., 4(4):671–680, 1973.

[27] R. A. Friesner, L. S. Tuckerman, B. C. Dornblaser and T. V. Russo. A method for exponential propagation of large systems of stiff nonlinear differential equations. J. Sci. Comput., 4:327–354, 1989.

[28] E. Gallopoulos and Y. Saad. Efficient solution of parabolic equations by Krylov approximation methods. SIAM J. Stat. Comput., 13(5):1236–1264, 1992.

[29] C. W. Gear and Y. Saad. Iterative solution of linear equations in ODE codes. SIAM J. Stat. Comput., 4(4):583–601, 1983.

[30] V. Girault and P. Raviart. *Finite Element Approximation of the Navier–Stokes Equations*, volume 749 of *Lecture Notes in Mathematics*. Springer, 1979.

[31] R. Glowinski. Finite element methods for the numerical solution of incompressible viscous flow. Introduction to the control of the Navier-Stokes equations. In *Lectures in Applied Mathematics*, volume 28, pp. 219–301. American Mathematical Society, Providence, RI, 1991.

[32] R. Glowinski. *Handbook of Numerical Analysis Volume IX: Numerical Methods for Fluids (Part 3)*. North–Holland, 2003.

[33] P. M. Gresho. On the theory of semi–implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 1: Theory. Int. J. Numer. Methods Fluids, 11:587–620, 1990.

[34] P. M. Gresho and S. S. Chan. On the theory of semi–implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 2: Implementation. Int. J. Numer. Methods Fluids, 11:621–659, 1990.

[35] P. M. Gresho and R. Sani. *Incompressible Flow and the Finite Element Method*, volume 2. Wiley, Chichester, 2000.

[36] J. Guermond and L. Quatrapelle. Calculation of incompressible viscous flows by an unconditionally stable projection FEM. J. Comput. Phys., 132:12–33, 1997.

[37] J. Guermond and L. Quatrapelle. On the approximation of the unsteady Navier–Stokes equations by finite element projection methods. Numer. Math., 80:207–238, 1998.

[38] J. L. Guermond and J. Shen. Velocity-correction projection methods for incompressible flows. SIAM J. Numer. Anal., 41(1):112 – 134, 2003.

[39] M. D. Gunzburger. *Finite Element Methods for Viscous Incompressible Flows: A Guide to Theory, Practice and Algorithms*. Academic Press, 1989.

[40] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer-Verlag, Berlin, 1991.

[41] M. Hochbruck and C. Lubich. On Krylov subspace approximations to the matrix exponential operator. SIAM J. Numer. Anal., 34(5):1911–1925, 1997.

[42] M. Hochbruck, C. Lubich and H. Selhofer. Exponential integrators for large systems of differential equations. SIAM J. Sci. Comput., 19(5):1552–1574, 1998.

[43] A. Iserles. Rational interpolation to $\exp(-x)$ with application to certain stiff systems. SIAM J. Numer. Anal., 18(1):1 – 12, 1981.

[44] A. Iserles and M. J. D. Powell. On the A–acceptability of rational approximations that interpolate the exponential function. IMA Journal of Numerical Analysis, 1:241–251, 1981.

[45] C. Johnson. *Numerical solution of partial differential equations by the finite element method*. Cambridge University Press, 1987.

[46] J. Kim and P. Moin. Application of a fractional–step method to incompressible Navier–Stokes equations. J. Comput. Phys., 59:308–323, 1985.

[47] G. M. Kobelkov and M. A. Olshanskii. Effective preconditioning of Uzawa type schemes for a generalized Stokes problem. Numer. Math., 86:443–470, 2000.

[48] R. Kosloff. Propagation methods for quantum molecular–dynamics. Ann. Rev. Phys. Chem., 45:145 – 178, 1994.

[49] E. Kreyszig. *Introductory Functional Analysis with Applications*. Wiley, 1978.

[50] H. Ku, R. S. Hirsch and T. D. Taylor. A pseudospectral method for solution of the three–dimensional incompressible Navier–Stokes equations. J. Comput. Phys., 70:439–462, 1987.

[51] J. D. Lambert. *Numerical Methods for Ordinary Differential Systems*. Wiley, Chichester, 1991.

[52] R. B. Lehoucq, D. C. Sorensen and C. Yang. *ARPACK User's Guide*. SIAM, 1998.

[53] C. H. Liu and D. Y. C. Leung. Development of a finite element solution for the unsteady Navier–Stokes equations using projection method and fractional–θ–scheme. Comput. Methods Appl. Engrg, 190:4301–4317, 2001.

[54] K. A. Mardal, X. Cheng Tai and R. Winther. A robust finite element method for Darcy–Stokes flow. SIAM J. Numer. Anal., 40(5):1605–1631, 2002.

[55] V. G. Maz'ja. *Sobolev Spaces*. Springer, 1985.

[56] P. D. Minev and P. M. Gresho. A remark on pressure correction schemes for transient viscous flow.

[57] C. Moler and C. V. Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. SIAM Review, 45(1):3 – 49, 2003.

[58] A. Nauts and R. E. Wyatt. New approach to many-state quantum dynamics – the recursive–residue–generation method. Phys. Rev. Lett., 51(25):2238 – 2241, 1983.

[59] A. Nauts and R. E. Wyatt. Theory of laser-molecule interaction - the recursive-residue-generation method. Phys. Rev. A, 30(2):872 – 883, 1984.

[60] S. A. Orszag, M. Israeli and M. Deville. Boundary conditions for incompressible flows. J. Sci. Comput., pp. 75–111, 1986.

[61] T. J. Park and J. C. Light. Unitary quantum time evolution by iterative Lanczos reduction. J. Chem. Phys., 85:5870–5876, 1986.

[62] R. Peyret and T. D. Taylor. *Computational Methods for Fluid Flow*. Springer–Verlag, 1983.

[63] G. Pini and G. Gambolati. Arnoldi and Crank–Nicolson methods for integration in time of the transport equation. Int. J. Numer. Meth. Fluids, 35:25–38, 2001.

[64] A. Quateroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer–Verlag, 1997.

[65] R. Rannacher. On Chorin's projection method for the incompressible Navier–Stokes equations. In *The Navier–Stokes equations II–Theory and Numerical Methods*, volume 1530 of *Lecture Notes in Mathematics*, pp. 167–183. Springer, 1992.

[66] M. Reed and B. Simon. *Functional Analysis*, volume I of *Methods of Modern Mathematical Physics*. Academic Press, 1972.

[67] M. Renardy and R. Rogers. *An Introduction to Partial Differential Equations*. Springer, 1993.

[68] R. D. Richtmyer and K. W. Morton. *Difference Methods for Initial–Value Problems*. Krieger, second edition, 1994.

[69] Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. SIAM J. Numer. Anal., 29(1):209 – 228, 1992.

[70] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS, 1996.

[71] J. Shen. On error estimates of projection methods for Navier–Stokes equations: First–order schemes. Siam J. Numer. Anal., 29(1):57–77, 1992.

[72] J. Shen. On error-estimates of some higher-order projection and penalty–projection methods for Navier–Stokes equations. Numer. Math., 62(1):49 – 73, 1992.

[73] J. Shen. On error estimates of the projection methods for the Navier–Stokes equations: First–order schemes. Math. Comp., 65(215):1039–1065, 1996.

[74] J. Shen and R. Temam. A new fractional scheme for the approximation of incompressible flows. Mat. Apli. Comput., 8:3–22, 1989.

[75] H. Tal-Ezer. Spectral methods in time for parabolic problems. SIAM J. Numer. Anal., 26(1):1 – 11, 1989.

[76] H. Tal-Ezer and R. Kosloff. An accurate and efficient scheme for propagating the time-dependent Schrödinger equation. J. Chem. Phys., 81(9):3967 – 3971, 1984.

[77] R. Temam. Sur l'approximation de la solution des équations de Navier–Stokes par la méthode de pas fractionnaires (II). Arch. Rat. Mech. Anal, 33:377–385, 1969.

[78] R. Temam. *Navier–Stokes Equations. Theory and Numerical Analysis*. North–Holland, third edition, 1984.

[79] M. Tokman. *Magnetohydrodynamic Modeling of Solar Magnetic Arcades Using Exponential Propagation Methods*. Ph.D. thesis, California Institute of Technology, 2001.

[80] S. Turek. A comparative study of some time–stepping techniques for the incompressible Navier–Stokes equations: From fully implicit nonlinear schemes to semi–implicit projection methods. Int. J. Numer. Meth. Fluids, 22(10):987–1012, 1996.

[81] S. Turek. On discrete projection methods for the incompressible Navier–Stokes equations: An algorithmical approach. Comput. Meth. Appl. Mech. Eng., 143:271–288, 1997.

[82] H. van der Vorst. An iterative solution method for solving $f(A)x = b$ using Krylov subspace information obtained for the symmetric positive definite matrix A. J. Comput. Appl. Math., 18:249 – 263, 1987.

[83] J. Van Kan. A second–order accurate pressure–correction scheme for viscous incompressible flow. SIAM J. Sci. Stat. Comput., 7(3):870–891, 1986.

[84] R. S. Varga. On higher order stable implicit methods for solving parabolic partial differential equations. J. Math. Phys., 40:220–231, 1961.

[85] G. Wanner. Order stars and stability. In A. Iserles, ed., *The State of the Art in Numerical Analysis*, pp. 451–471. Clarendon Press, 1987.

Vita

Christopher K. Newman

EDUCATION

Virginia Tech **January 1999–December 2003**
Ph.D. in Mathematics.
Advisor: Christopher Beattie

University of New Mexico **September 1995–December 1997**
M.A. in Applied Mathematics.

New Mexico State University **January 1990–July 1992**
B.S. in Mathematics, Minor in Physics.

EXPERIENCE

Sandia National Laboratories, Computational Mathematics and Algorithms
Position: Student Internship Program **May 2001–September 2003**
Mentor: Richard Lehoucq **May 2000–August 2000**
Compute and analyze numerical solutions to incompressible Navier–Stokes equations. Develop tools and software for analysis of time integration methods using C++, MATLAB, and the deal.II finite element library. Provide theoretical analysis and application of exponential integration methods to differential-algebraic equations.

Virginia Tech
Position: Graduate Teaching Assistant **January 2000–May 2001**
Created and implemented course instruction for Elementary Calculus I and II. Duties included writing and delivering lectures, writing and grading exams, and advising students during regular office hours.

Position: Graduate Research Assistant **January 1999–December 1999**
Performed mathematical analysis of a system of nonlinear PDEs that describe heat and mass transfer of air and water within a wood based composite during manufacture. Implemented a semi–discrete finite difference method for PDEs using FORTRAN 90 and Mathematica under DEC Unix.

Position: Supervisor, Math Emporium **August 1998–December 1998**
Supervised a staff of 20–30 graduate and undergraduate students. Maintained a weekly staff work schedule. Maintained a web site for Department of Mathematics course announcements and information.

Balleau Groundwater, Inc.
Position: Hydrologic Technician **April 1997–June 1998**
Performed hydrologic calculations and modeling using FORTRAN 77/90 in Windows NT environment. Performed integration of geographic information system and groundwater modeling using ArcView GIS, Surfer and MODFLOW. Performed statistical analysis of hydrologic data and analysis of surface water and groundwater interaction.

SKILLS

Operating Systems: Linux, Solaris, IRIX,*Programming*: C/C++, Fortran 77/90, make/gmake, bash, ksh, csh, tcsh, PERL, HTML, PVM.
Software: Mathematica, MATLAB, Cubit, GNUPlot, GMV, LaTeX, netCDF, deal.II, Sundance.

PRESENTATIONS

Seventh U.S. National Conference on Computational Mechanics, July 2003.
SIAM Conference on Computational Science and Engineering, February 2003.
Virginia Tech Department of Mathematics NA/PDE Seminar, April 2002.
Copper Mountain Conference on Iterative Methods, March 2002.
Sandia National Laboratories Sixth Annual Student Internship Program Symposium, August 2001.

PROFESSIONAL SOCIETIES

Society for Industrial andApplied Mathematics, since May 2001.
American Mathematical Society, since January 1999.