Monolithic multigrid for higher-order discretizations of poroelasticity

by

© **Selvabavitha Vijendiran**

A thesis submitted to the School of Graduate Studies in partial fulfillment of the requirements for the degree of Master in Science.

Department of Mathematics and Statistics

Memorial University

August 2024

St. John's, Newfoundland and Labrador, Canada

Abstract

Mathematical models of poroelasticity, the study of the behaviour of fluid-saturated porous media, present complex challenges in numerical simulation due to their inherent coupling between fluid and solid phases. In this study, we propose higher-order discretization techniques for poroelasticity problems, that we couple with monolithic multigrid methods to enable efficient high-fidelity simulations. These discretizations are based on higher-order finite elements in space (including reduced quadrature techniques to effectively model nearly incompressible solid phases) and implicit Runge-Kutta methods in time, to ensure robustness and stability in the time-stepping procedure. The monolithic multigrid approach leverages recent work extending Vanka-style relaxation to incompressible flow models, that we adapt to the equations of poroelasticity. Through numerical experiments and comparisons, we demonstrate the effectiveness of our proposed approach in accurately capturing the behaviour of poroelastic models while maintaining computational efficiency.

To my loving parents, husband and sweet kids.

Lay summary

Understanding and modeling real-world phenomena often requires sophisticated mathematical approaches. However, exact solutions are often unattainable, prompting the use of numerical methods to generate accurate approximations. This thesis tackles the complexities of poroelasticity, a field concerned with the behavior of materials that simultaneously deform and allow fluid flow through their pores. By leveraging the power of numerical methods, particularly higher-order discretizations, the aim is to develop accurate approximations of poroelastic phenomena. The primary focus is on the implementation of a monolithic multigrid framework, which optimizes computational efficiency. This involves combining various numerical schemes and integrating specific discretization techniques tailored to the intricacies of poroelastic materials. Through rigorous testing and optimization, the goal is to create a robust numerical solver capable of handling both spatial and temporal discretizations with higher-order accuracy. This solver is crucial for understanding and predicting the behavior of poroelastic materials in diverse real-world scenarios, from geotechnical engineering to biomedical applications.

Acknowledgements

I want to thank everyone who has supported me throughout my journey in graduate studies.

Firstly, I am deeply grateful to my project supervisor, Prof. Scott MacLachlan, a remarkable Professor in Mathematics and Scientific Computing at Memorial University, St. John's, Newfoundland, Labrador, Canada. His guidance, encouragement, and insightful suggestions have been invaluable to me from the start of my academic journey to the completion of this thesis.

I also want to acknowledge my husband for his constant love and support. His encouragement has been a source of strength for me, and I am thankful for his unwavering support in every situation. I am also grateful to my kids for their cooperation, which has enriched my study. Their support and positivity have made this academic endeavor even more fulfilling.

Lastly, I extend my thanks to the examiners of this thesis for their time and feedback.

Statement of contribution

This thesis is a collaborative effort between Selvabavitha Vijendiran and Prof. Scott MacLachlan. Writing the scientific simulation codes used in the research and the thesis was done by Selvabavitha. Supervision, guidance in coding and editing of the thesis was done by Prof. Scott MacLachlan.

Table of contents

Chapter 1

Introduction

Poroelasticity, a field delving into the mechanical behavior of fluid-saturated porous media, traces its roots to Biot's theory of consolidation, initiated to address challenges in soil consolidation. Early contributions by Terzaghi [34], alongside subsequent advancements by Biot [7], laid the groundwork for comprehending the intricate interactions between solid and fluid constituents within porous structures.

Biot's pioneering work expanded the scope of the field by considering both compressible solid and fluid phases, introducing variables such as fluid content. Additionally, he extended the theory to encompass anisotropic elasticity and dynamic responses. In the realm of fluid-infiltrated porous media, comprising a solid skeleton and fluid occupying porous spaces, the material undergoes quasi-static deformations, considering compressibility in both solid and fluid phases. Despite its applicability for high degrees of liquid saturation, researchers have extended the theory to lower saturation levels under specific assumptions.

Central to describing porous media's mechanical behavior is the principle of effective stress, delineating the transmission of internal stresses to the solid skeleton and pore fluid. This principle, coupled with the conservation of mass and Darcy's law governing viscous fluid flow, forms the basis for understanding fluid flow and deformations in poroelastic materials.

Poroelasticity holds paramount importance across diverse scientific and engineering disciplines. In geoscience, the understanding of poroelastic phenomena is essential for deciphering processes like groundwater flow, oil reservoir behavior, and seismic

responses in subsurface formations. Biomedical science benefits from poroelasticity in advancing medical diagnostics and enhancing our understanding of biomechanics, particularly in studying the behavior of biological tissues. In engineering, applications span from geotechnical engineering to the development of biomimetic materials, where poroelastic considerations significantly influence the design and performance of structures [4].

The study of poroelasticity is not without its challenges. The coupled, multiphysics nature of these systems introduces inherent complexities that demand sophisticated computational methodologies and a nuanced understanding of material behavior. Theoretical modeling must navigate the delicate balance between the mechanical response of the solid matrix and the fluid flow dynamics within porous structures. Numerical simulations face difficulties in accurately capturing these complex interactions, often encountering challenges such as spurious oscillations and variations in physical parameters. Spurious oscillations are numerical artifacts that arise due to discretization errors, particularly in simulations involving sharp gradients or interfaces, leading to unphysical fluctuations in the computed solution. Variations in physical parameters, on the other hand, pertain to the sensitivity of the model's behavior to changes in material properties or boundary conditions, which can significantly affect the accuracy and stability of the simulation. These challenges highlight the need for advanced methodologies that can comprehensively model and simulate the intricate behaviors associated with poroelastic materials, underscoring the complexity and difficulty inherent in the study of poroelasticity. Robust numerical methods, such as those discussed in [5] and [19], have been developed to address some of these issues, focusing on enhancing the stability and accuracy of simulations through improved discretization techniques and preconditioners.

The nexus between flow, fluid, and poroelasticity is intricately woven in the simultaneous interactions of fluids within porous media and the resulting deformations of the solid matrix. Poroelasticity, as governed by Biot's theory, encapsulates the interdependence of fluid flow and the deformation of the solid matrix in porous structures. Fluids, whether water, air, or other substances, traverse through interconnected voids within the porous media, inducing deformations in the solid framework. Biot's equations, considering parameters like porosity, pore fluid pressure, and solid matrix deformations, elucidate this coupled behavior. The principle of effective stress further delineates how internal stresses are distributed between the solid skeleton and pore

fluid, influencing both deformations and fluid flow. Robust mathematical models and numerical methods, such as those developed in [5], play a critical role in analyzing and solving these complex interactions.

Poroelasticity, born out of the necessity to understand soil consolidation, has evolved into a critical field influencing various scientific and engineering domains. Its importance lies in unraveling the complexities of fluid-saturated porous media, while its challenges highlight the need for continuous advancements in theoretical modeling and numerical simulations. As we delve into the depths of poroelasticity, we unlock the potential to address real-world problems and pave the way for innovative solutions in an array of applications.

This thesis is structured as follows. In Chapter 2, we include the background information summarizing key theories, models, and prior research related to our topic. In Chapter 3, we present the stabilized finite-element discretization for the three-field formulation of Biot's model. Additionally, we introduce the reduced-quadrature discretization, providing proofs of well-posedness and error estimates and monolithic multigrid, focusing on the choice of Vanka relaxation scheme for solving the implicit Runge-Kutta discretized system and the use of divergence-preserving interpolation operators to ensure robustness in nearly incompressible cases. Chapter 4 is dedicated to presenting numerical results that demonstrate the efficiency of the proposed solvers. Finally, Chapter 5 concludes the thesis, offering remarks and reflections on the findings and their implications.

Chapter 2

Background

In this chapter, we present important theories, models, and prior work that are relevant to this thesis. This background covers three fundamental areas: the finite element method (FEM), the multigrid method, and time integration techniques. The finite element method is a powerful numerical technique used for approximating solutions to complex problems in engineering and physical sciences, particularly in the field of poroelasticity. It allows for the discretization of continuous domains into finite elements, making it possible to solve differential equations that describe the behavior of poroelastic materials. The multigrid method is an efficient algorithm designed to solve large-scale linear systems of equations, which often arise in the discretization of partial differential equations using FEM. By employing a hierarchy of discretizations, the multigrid method accelerates the convergence of solutions, significantly reducing computational time and resources. This is particularly important for higher-order discretizations, where the computational cost can become prohibitively high. Time integration techniques are essential for solving time-dependent problems in poroelasticity, where the behavior of materials evolves over time due to fluid flow and mechanical deformation. Accurate and stable time integration methods ensure that the numerical solutions remain reliable and physically meaningful throughout the simulation. Together, these foundational concepts provide the necessary framework for developing and analyzing monolithic multigrid methods for higher-order discretizations of poroelasticity. Understanding these theories and models is crucial for advancing the state of the art in this field and achieving efficient and accurate numerical solutions.

2.1 Finite-element method

In this section, we discuss the finite-element method (FEM), a powerful numerical technique widely employed in engineering and scientific simulations that discretizes complex continuous systems into simpler, finite elements, allowing for the approximation of solutions to partial differential equations [2, 9, 10, 13, 20]. In the context of poroelasticity, FEM plays a pivotal role in modeling the coupled behavior of the solid and fluid phases within porous media [4, 31].

2.1.1 Finite-element method for the Poisson equation

Finite-element approximation is a robust computational approach for generating numerical approximations of solutions to differential equations. In this section, we offer an initial exploration of the finite-element method as applied to the Poisson equation,

$$-\Delta u = f, \tag{2.1}$$

an elementary and well-known elliptic partial differential equation that serves as a fundamental mathematical model. The source or load function, denoted as f, is defined over a domain Ω in two or three dimensions. A solution u satisfying (2.1) should also satisfy given boundary conditions on the boundary, $\partial\Omega$, of Ω ; for example

$$\alpha u + \beta \frac{\partial u}{\partial n} = g \quad \text{on} \quad \partial\Omega, \tag{2.2}$$

where $\dfrac{\partial u}{\partial n} = n \cdot \nabla u = g$ signifies the directional derivative along the outward normal direction to the boundary $\partial\Omega$. The coefficients, α and β, can be constant or variable. The combination of Equations (2.1) and (2.2) collectively forms a boundary value problem. When the constant β in (2.2) takes on a value of zero, the associated boundary condition is of Dirichlet type, and the resulting boundary value problem is identified as the Dirichlet problem for the Poisson equation. Conversely, when the constant α is zero, we encounter a Neumann boundary condition, thereby constituting a Neumann problem. A third scenario arises when Dirichlet conditions apply to a specific part of the boundary $\partial\Omega_D$, while Neumann conditions are satisfied on the remaining portion $\partial\Omega \setminus \partial\Omega_D$. Applying a constant value of $g = 0$ with $\beta = 0$ across

the entire boundary is termed homogeneous Dirichlet boundary conditions. When non-zero $u = g(x, y)$ is imposed along the entire boundary, it is termed an inhomogeneous Dirichlet boundary condition. Alternatively, it is feasible to specify the value of the solution, $u = c$, for constant value c, or $u = g(x, y)$, along a (continuous) segment of the boundary. On the remaining part of the boundary, flux variation can be specified using Neumann boundary conditions, referred to as mixed boundary conditions. Neumann boundary conditions define the directional derivative of u along a normal vector, denoted as $\dfrac{\partial u}{\partial n}$. The function $g = g(x, y)$ is provided and, ultimately, we have known values of u on some (continuous) portion of the boundary and the directional derivative on another. The Dirichlet boundary condition becomes an essential boundary condition, imposed directly on the function space for u, while the Neumann boundary condition becomes a natural boundary condition, accounted for in the following weak form (2.5).
We next consider how to choose the function space, $\mathcal{V}$. For more details, see [9, 10, 2].

Definition 1 *L^p spaces are defined as follows for $1 \leq p < \infty$:*

$$L^p(\Omega) = \left\{ u \,|\, u \text{ is real and measurable and } \int_\Omega |u|^p \, dx < \infty \right\}.$$

The following L^p-norm defines a norm on this space:

$$||u||_p = \left(\int_\Omega |u(x)|^p \, dx \right)^{1/p}.$$

In the case of $p = 2$, we define the space as

$$L^2(\Omega) = \left\{ u : \int_\Omega |u(x)|^2 \, dx < \infty \right\}.$$

Definition 2 ***Sobolev Spaces*** $H^m(\Omega)$**.** *Given an integer number $m \geq 1$, standard Sobolev spaces read*

$$H^m(\Omega) = \{v \in L^2(\Omega) : D^\alpha v \in L^2(\Omega), |\alpha| \leq m\},$$

with the H^m norm of a function v is defined as:

$$\|v\|_{H^m} = \left(\sum_{|\alpha| \leq m} \int_\Omega |D^\alpha v|^2 \, dx \right)^{1/2}$$

where

$$D^\alpha = \frac{\partial^{|\alpha|}}{\partial_1^{\alpha_1} \partial_2^{\alpha_2} \cdots \partial_n^{\alpha_n}}, \quad |\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_n.$$

We can also consider subspaces of these Sobolev spaces that include boundary conditions.

Definition 3 $H_0^1(\Omega) = \{u \in H^1(\Omega) : u = 0 \text{ on } \partial\Omega\}$.

Definition 4 ***Product Space.*** *So far, we have been thinking of spaces of scalar functions, $u : \Omega \to \mathbb{R}$. However, we can also consider vector-valued functions $\boldsymbol{u} : \Omega \to \mathbb{R}^d$. We will denote a space of vector-valued functions with vector notation, i.e., $\boldsymbol{H^1}(\Omega)$, or we may use $(H^1(\Omega))^d$ to explicitly indicate the dimension.*

The H^k (and $\boldsymbol{H^k}$) spaces concern a function and all of its partial derivatives. However, for vector-valued functions, we might want to only take divergences or curls. Thus, $\boldsymbol{H}(\mathrm{div}, \Omega)$ and $\boldsymbol{H}(\mathrm{curl}, \Omega)$ can be defined as

$$\boldsymbol{H}(\mathrm{div}, \Omega) = \{\boldsymbol{u} \in \boldsymbol{L^2}(\Omega) | \nabla \cdot \boldsymbol{u} \in L^2(\Omega)\},$$
$$\boldsymbol{H}(\mathrm{curl}, \Omega) = \{\boldsymbol{u} \in \boldsymbol{L^2}(\Omega) | \nabla \times \boldsymbol{u} \in \boldsymbol{L^2}(\Omega)\},$$

with corresponding norms,

$$\|\boldsymbol{u}\|_{div}^2 = \|\boldsymbol{u}\|^2 + \|\nabla \cdot \boldsymbol{u}\|^2;$$
$$\|\boldsymbol{u}\|_{curl}^2 = \|\boldsymbol{u}\|^2 + \|\nabla \times \boldsymbol{u}\|^2.$$

For simplicity, we first consider a model problem of the form [2],

$$-\nabla \cdot \nabla u + u = f \quad \text{in } \Omega,$$

with $u = 0$ on $\partial\Omega$, and with $f \in L^2(\Omega)$. The weak form is to find $u \in \mathcal{V} = H_0^1(\Omega)$

such that

$$\int_\Omega \nabla u \cdot \nabla v + \int_\Omega uv = \int_\Omega fv dx \text{ for all } v \in \mathcal{V}. \tag{2.3}$$

We define a bilinear form, $a(\cdot,\cdot)$, and a linear functional, $g(\cdot)$, as follows:

$$\begin{aligned} a(u,v) &= \langle \nabla u, \nabla v \rangle + \langle u, v \rangle, \\ g(v) &= \langle f, v \rangle, \end{aligned}$$

where u and v are in the function space $\mathcal{V} = H_0^1(\Omega)$, ∇ represents the gradient, and $\langle \cdot, \cdot \rangle$ denotes the $L^2(\Omega)$ inner product.

Theorem 1 ***Riesz representation theorem [14].*** *Let $\mathcal{V}$ be a Hilbert space with inner product $\langle \cdot, \cdot \rangle_\mathcal{V}$. Let g be a bounded linear functional on $\mathcal{V}$—i.e., $g \in \mathcal{V}^*$. Then, there exists a unique $u \in \mathcal{V}$ such that*

$$g(v) = \langle u, v \rangle_\mathcal{V} \quad \textit{for all } v \in \mathcal{V}.$$

For the model problem described by Equation (2.3), the bilinear form is equivalent to the natural inner product in $H^1(\Omega)$:

$$a(u,v) = \langle \nabla u, \nabla v \rangle + \langle u, v \rangle = \langle u, v \rangle_1$$

Therefore, the weak formulation of the problem in (2.3) seeks to find $u \in H_0^1(\Omega)$ such that

$$a(u,v) = g(v), \quad \forall v \in H_0^1(\Omega),$$

where $H_0^1(\Omega)$ is chosen to accommodate the Dirichlet boundary conditions. By the Riesz representation theorem, there exists a unique $u \in H_0^1(\Omega)$ that satisfies this equation. For more general equations, we require a generalized approach.

Definition 5 ***Ellipticity.*** *Given a Hilbert Space, $\mathcal{V}$, consider a bilinear form:*

$$a(\cdot,\cdot) : \mathcal{V} \times \mathcal{V} \to \mathbb{R}.$$

The form $a(\cdot,\cdot)$ is coercive if there exists a constant $c_0 > 0$ such that

$$c_0\|u\|_{\mathcal{V}}^2 \leq a(u,u) \quad \textit{for all } u \in \mathcal{V};$$

and $a(\cdot,\cdot)$ is continuous if there exists a constant $c_1 > 0$ such that

$$|a(u,v)| \leq c_1\|u\|_{\mathcal{V}}\|v\|_{\mathcal{V}} \quad \textit{for all } u, v \in \mathcal{V}.$$

If $a(\cdot,\cdot)$ is both coercive and continuous on $\mathcal{V}$, then $a(\cdot,\cdot)$ is said to be $\mathcal{V}$-elliptic.

We next consider the boundary-value problem,

$$-\Delta u = f \quad \text{in} \quad \Omega, \tag{2.4a}$$

$$u = 0 \quad \text{on} \quad \partial\Omega. \tag{2.4b}$$

The problem defined in Equation (2.3) is called the strong formulation of the partial differential equation (PDE). The weak formulation serves as a rephrasing of the original (strong form) PDE, and it is through this reformulation that the final finite-element (FE) approach takes shape. To derive the weak form of the PDE, we multiply both sides of (2.4a) by an arbitrary function, commonly referred to as a test function, denoted as v. If we let v be a smooth function with $v = 0$ on $\partial\Omega$, we can define the bilinear form, $a(u,v)$, and the weak formulation given by finding $u \in \mathcal{V}$ such that

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \, d\Omega, \tag{2.5a}$$

$$= -\int_\Omega \Delta u \cdot v \, d\Omega + \int_{\partial\Omega} \frac{\partial u}{\partial n} v \, dS, \tag{2.5b}$$

$$= \int_\Omega f v \, d\Omega \quad \text{for all } v \in \mathcal{V}. \tag{2.5c}$$

With this, we turn to a key theoretical result, called the Lax-Milgram theorem.

Theorem 2 ***Lax-Milgram theorem [14].*** *Let $\mathcal{V}$ be a Hilbert space with inner product $\langle\cdot,\cdot\rangle_{\mathcal{V}}$. Assume that $a(\cdot,\cdot)$ is a bilinear form that is coercive and continuous on $\mathcal{V}$. In addition, assume that $g(\cdot)$ is a bounded linear functional on $\mathcal{V}$. Then, there exists a unique $u \in \mathcal{V}$ such that*

$$a(u,v) = g(v) \quad \textit{for all } v \in \mathcal{V}.$$

The Lax-Milgram theorem stands as a key in the analysis of finite element methods, and we utilize it to establish the existence and uniqueness of a solution. Again consider the weak formulation in (2.5) to find $u \in \mathcal{V}$ such that

$$a(u, v) = \langle f, v \rangle, \quad \forall v \in \mathcal{V},$$

where $\mathcal{V} = H_0^1(\Omega)$ is chosen to accommodate the Dirichlet boundary conditions. Here, $a(u, v) = \int_\Omega \nabla u \cdot \nabla v \, d\Omega$ is the bilinear form, and $\langle f, v \rangle = \int_\Omega f v \, d\Omega$ is a bounded linear functional on $\mathcal{V}$. By the Lax-Milgram theorem, if $a(\cdot, \cdot)$ is a continuous and coercive bilinear form on $\mathcal{V}$ and $\langle f, \cdot \rangle$ is a bounded linear functional on $\mathcal{V}$, then there exists a unique solution $u \in \mathcal{V}$ that satisfies the weak formulation. The continuity of $a(\cdot, \cdot)$ follows from the Cauchy-Schwarz inequality, ensuring that $|a(u, v)| \leq C\|u\|_{H^1}\|v\|_{H^1}$ for some constant C. The coercivity of $a(\cdot, \cdot)$ is established by the Poincaré inequality [2], implying that $a(u, u) \geq \alpha\|u\|_{H^1}^2$ for some $\alpha > 0$. Thus, the conditions of the Lax-Milgram theorem are satisfied, guaranteeing the existence and uniqueness of the solution to the weak form.
Next, we will discuss the Ritz-Galerkin approximation. This method is a way to approximate the solution of a continuous problem with a finite-dimensional problem that is easier to solve computationally.

Definition 6 ***Ritz-Galerkin approximation****. Let $a : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ be a bilinear form, and let $\mathcal{V}^h$ be a finite-dimensional subspace of $\mathcal{V}$. Consider the weak form restricted to $\mathcal{V}^h$: Find $u^h \in \mathcal{V}^h$ such that*

$$a(u^h, v) = \langle f, v \rangle \quad \forall v \in \mathcal{V}^h. \tag{2.6}$$

Here, u^h is called the Ritz-Galerkin approximation of the weak solution $u \in \mathcal{V}$.

Let u represent the solution to the variational problem, while u^h denotes the solution to the Galerkin approximation problem. Our aim is to evaluate the error $\|u - u^h\|_{\mathcal{V}}$. This estimation is facilitated by the following lemma:

Lemma 1 ***Céa's Lemma.*** *Let $\mathcal{V} \subseteq H$ be a closed subspace of the Hilbert space H. Let $a(\cdot, \cdot)$ be a coercive and continuous bilinear form on $\mathcal{V}$. In addition, for a bounded linear functional $g(\cdot)$ on $\mathcal{V}$, let $u \in \mathcal{V}$ satisfy*

$$a(u, v) = g(v) \quad \textit{for all } v \in \mathcal{V}.$$

Consider a finite-dimensional subspace $\mathcal{V}^h \subset \mathcal{V}$ and $u^h \in \mathcal{V}^h$ that satisfies

$$a(u^h, v^h) = g(v^h) \quad \textit{for all } v^h \in \mathcal{V}^h.$$

Then,

$$\|u - u^h\|_{\mathcal{V}} \leq \frac{c_1}{c_0} \min_{v^h \in \mathcal{V}^h} \|u - v^h\|_{\mathcal{V}},$$

where c_0 and c_1 are the coercivity and continuity constants for $a(\cdot, \cdot)$, respectively.

Remark 1 *Céa's Lemma establishes that u^h is quasi-optimal, indicating that the error $\|u - u^h\|_{\mathcal{V}}$ is close to the best approximation within the subspace $\mathcal{V}^h$.*

Definition 7 ***Polynomial spaces***. *Let T be any triangle, $k > 0$. Let $P_k(T)$ denote the set of all polynomials in two variables of degree $\leq k$ on T, and*

$$P_k(\Omega^h) = \{u \in C^0(\Omega^h) \ : \ u(x)|_T \in P_k(T), \forall T \in \Omega^h\},$$

where $\Omega^h = \{T\}$ is a set of triangles.

In general, there are three factors that significantly influence the accuracy of the approximation. Firstly, the regularity of the solution, as better approximations are achievable when the solution possesses higher degrees of smoothness. Secondly, the mesh quality, which is assessed based on the element size and shape quality. Lastly, the choice of approximation space itself also plays a crucial role. Explaining the quality of Ritz-Galerkin approximations involves complex theory [10], which we briefly summarize here.

Definition 8 ***Diameter of a Set.*** *Given a set $S \subseteq \mathbb{R}^n$, the diameter is given by $diam(S) = \sup_{x,y \in S} \|x - y\|$.*

Definition 9 *For any $T \in \Omega^h$, let B_T be the largest ball contained in T such that for any $x \in T$, the closed convex hull of $\{x\} \cup B_T$ is contained in T. We say that T is star-shaped with respect to B_T. A family of subdivisions, $\{\Omega^h\}$, is said to be* ***non-degenerate*** *or regular if there exists $\rho > 0$ such that for all $T \in \Omega_h$ and for all*

$h \in (0, 1]$,

$$diam\ B_T \geq \rho\ diam\ T. \tag{2.7}$$

Proving Theorem 3 requires concepts like reference elements, affine equivalent elements, and certain theorems outlined in [10], which play a crucial role in deepening our comprehension of this area.

Theorem 3 ***Accuracy of*** $P_k(\Omega^h)$**.** *Let* $\{\Omega^h\}, 0 < h \leq 1$*, be a non-degenerate family of subdivisions of a polyhedral domain* Ω *in* $\mathbb{R}^n$*. Let* $\mathcal{V}^h = P_k(\Omega^h)$ *with* $k + 1 - \frac{n}{2} > 0$ *and a suitable choice of nodes for the degrees of freedom of* $P_k(\Omega^h)$*. Let* I^h *be such that* $I^h v \in \mathcal{V}^h$ *is the interpolant of* $v \in C^0(\Omega)$*. Then, there exists a constant* C *depending on the choice of nodes,* n*,* k*, and* ρ *such that if* $u \in H^{k+1}(\Omega)$*, then*

$$\left(\sum_{T \in \Omega_h} \|u - I^h u\|^p_{p, W^p_s(T)} \right)^{1/p} \leq C h^{k+1-s} |u|_{W^p_{k+1}(\Omega)}, \quad for\ 0 \leq s \leq k+1$$

2.1.2 Mixed Poisson finite-element method

In this section, we introduce finite-element methods for the "mixed Poisson equation". Consider the alternative formulation of the Poisson equation (2.1) given by introducing an additional vector (variable), namely the (negative) flux: $\boldsymbol{\sigma} = \nabla u$. With this definition of $\boldsymbol{\sigma}$, we can rewrite the original PDE as,

$$\boldsymbol{\sigma} - \nabla u = \mathbf{0} \quad \text{in } \Omega, \tag{2.8a}$$

$$-\nabla \cdot \boldsymbol{\sigma} = f \quad \text{in} \quad \Omega, \tag{2.8b}$$

We have at present two unknowns, u and $\boldsymbol{\sigma}$, and must determine suitable finite-element spaces for each of them. Given that the space for $\boldsymbol{\sigma}$ must consist of vector-valued functions, it is reasonable to assume that the two unknowns occupy different spaces. We write $u \in \mathcal{V}$ and $\boldsymbol{\sigma} \in \boldsymbol{\mathcal{W}}$ and multiply Equation (2.8) by test functions $\boldsymbol{w} \in \boldsymbol{\mathcal{W}}$ and $v \in \mathcal{V}$ and integrate over the domain to obtain a weak formulation, to

find $\boldsymbol{\sigma} \in \boldsymbol{\mathcal{W}}$ and $u \in \mathcal{V}$ such that

$$\langle \boldsymbol{\sigma} - \nabla u, \boldsymbol{w} \rangle = 0 \quad \forall \boldsymbol{w} \in \boldsymbol{\mathcal{W}}, \tag{2.9a}$$

$$\langle -\nabla \cdot \boldsymbol{\sigma}, v \rangle = \langle f, v \rangle \quad \forall v \in \mathcal{V}, \tag{2.9b}$$

Now, consider the more general boundary conditions for the Poisson problem, with

$$\begin{aligned} u &= u_0 \quad \text{on } \Gamma_D, \\ \boldsymbol{\sigma} \cdot \boldsymbol{n} &= g \quad \text{on} \quad \Gamma_N. \end{aligned}$$

where $\boldsymbol{n}$ is the outward unit normal vector to the boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$, where Γ_D and Γ_N are disjoint segments on which we impose Dirichlet and Neumann boundary conditions for u, respectively. Now consider the term $\langle \nabla u, \boldsymbol{w} \rangle$ in Equation (2.9a) and integrate by parts on it to obtain

$$\begin{aligned} -\langle \nabla u, \boldsymbol{w} \rangle &= -\int_\Omega \nabla u \cdot \boldsymbol{w} \, dx \\ &= -\int_{\partial\Omega} u\boldsymbol{w} \cdot \boldsymbol{n} \, dS + \int_\Omega u \nabla \cdot \boldsymbol{w} \, dx \\ &= -\int_{\Gamma_D} u_0 \boldsymbol{w} \cdot \boldsymbol{n} \, dS - \int_{\Gamma_N} u\boldsymbol{w} \cdot \boldsymbol{n} \, dS + \int_\Omega u \nabla \cdot \boldsymbol{w} \, dx \\ &= -\int_{\Gamma_D} u_0 \boldsymbol{w} \cdot \boldsymbol{n} \, dS + \int_\Omega u \nabla \cdot \boldsymbol{w} \, dx, \end{aligned}$$

where we restrict $\boldsymbol{w} \cdot \boldsymbol{n} = 0 \quad \text{on} \quad \Gamma_N$. Using the above integration-by-parts, rewrite Equation (2.9) as the first-order system

$$\langle \boldsymbol{\sigma}, \boldsymbol{w} \rangle + \langle u, \nabla \cdot \boldsymbol{w} \rangle = -\int_{\Gamma_D} u_0 \boldsymbol{w} \cdot \boldsymbol{n} \, dS \quad \forall \boldsymbol{w} \in \boldsymbol{\mathcal{W}}, \tag{2.10a}$$

$$\langle -\nabla \cdot \boldsymbol{\sigma}, v \rangle = \langle f, v \rangle \quad \forall v \in \mathcal{V}, \tag{2.10b}$$

In [2], it is observed that when a homogeneous boundary condition of $u_0 = 0$ is adopted, the right-hand side term of Equation (2.10a) becomes zero. From Equation (2.10), we can deduce that the left-hand side can be expressed in terms of two bilinear forms, defined as follows: $a(\boldsymbol{\sigma}, \boldsymbol{w}) = \langle \boldsymbol{\sigma}, \boldsymbol{w} \rangle$ and $b(\boldsymbol{w}, u) = \langle u, \nabla \cdot \boldsymbol{w} \rangle$. To ensure symmetry, the second equation was multiplied by -1. Consequently, the symmetric

saddle point problem can be formulated as finding $\boldsymbol{\sigma} \in \boldsymbol{\mathcal{W}}, u \in \mathcal{V}$ such that

$$a(\boldsymbol{\sigma}, \boldsymbol{w}) + b(\boldsymbol{w}, u) = 0 \quad \forall \boldsymbol{w} \in \boldsymbol{\mathcal{W}}, \tag{2.11a}$$

$$b(\boldsymbol{\sigma}, v) = -\langle f, v \rangle \quad \forall v \in \mathcal{V}. \tag{2.11b}$$

Next, we will discuss continuous inf-sup conditions, highlighting how important it is to have both weak coercivity and continuity in the product space $\boldsymbol{\mathcal{W}} \times \mathcal{V}$. Thus, let us consider a slightly more general mixed formulation than in Equation (2.11) of finding $\boldsymbol{\sigma} \in \boldsymbol{\mathcal{W}}, u \in \mathcal{V}$ such that

$$a(\boldsymbol{\sigma}, \boldsymbol{w}) + b(\boldsymbol{w}, u) = g(\boldsymbol{w}) \quad \forall \boldsymbol{w} \in \boldsymbol{\mathcal{W}}, \tag{2.12a}$$

$$b(\boldsymbol{\sigma}, v) = f(v) \quad \forall v \in \mathcal{V}, \tag{2.12b}$$

where g and f are bounded linear functions on $\boldsymbol{\mathcal{W}}$ and $\mathcal{V}$, respectively, and $\boldsymbol{\mathcal{W}}$ and $\mathcal{V}$ are Hilbert spaces. The first step in any well-posedness result is always to demonstrate the continuity (or boundedness) of the bilinear forms in (2.12), as defined in Definition 6, that there exist constants $c_a > 0$ and $c_b > 0$ such that

$$\|a(\boldsymbol{\sigma}, \boldsymbol{w})\| \leq c_a \|\boldsymbol{\sigma}\|_{\mathcal{W}} \|\boldsymbol{w}\|_{\boldsymbol{\mathcal{W}}}, \quad \forall \boldsymbol{\sigma} \in \mathcal{W}, \boldsymbol{w} \in \boldsymbol{\mathcal{W}},$$
$$\|b(\boldsymbol{\sigma}, v)\| \leq c_b \|\boldsymbol{\sigma}\|_{\mathcal{W}} \|v\|_{\mathcal{V}}, \quad \forall \boldsymbol{\sigma} \in \mathcal{W}, v \in \mathcal{V}.$$

We next discuss coercivity. To keep things simple, we first consider the scenario $f = 0$. Finding solutions to (2.12a) with $\boldsymbol{\sigma} \in \widehat{\boldsymbol{\mathcal{W}}} = \{\boldsymbol{w} \in \mathcal{W} : b(\boldsymbol{w}, v) = 0 \ \forall v \in \mathcal{V}\}$ is the main idea behind the existence and uniqueness results for mixed systems. It should be noted that $\widehat{\boldsymbol{\mathcal{W}}}$ is a closed linear subspace of $\boldsymbol{\mathcal{W}}$. Selecting the test function for (2.12a) from the limited space, $\boldsymbol{w} \in \widehat{\boldsymbol{\mathcal{W}}}$, results in finding $\boldsymbol{\sigma} \in \widehat{\boldsymbol{\mathcal{W}}}$ such that

$$a(\boldsymbol{\sigma}, \boldsymbol{w}) = g(\boldsymbol{w}) \quad \forall \boldsymbol{w} \in \widehat{\boldsymbol{\mathcal{W}}}. \tag{2.13}$$

This simplifies the saddle-point system given in Equation (2.12) to a single (vector) equation as given in Equation (2.13). Consequently, if $a(\cdot, \cdot)$ exhibits coercivity on $\widehat{\boldsymbol{\mathcal{W}}}$, it fulfills the requirements of the Lax-Milgram Theorem. Hence, there exists a

unique solution, denoted as $\boldsymbol{\sigma}^* \in \widehat{\boldsymbol{\mathcal{W}}}$, to this modified problem,

$$a(\boldsymbol{\sigma}, \boldsymbol{w}) = g(\boldsymbol{w}) \quad \text{for all } \boldsymbol{w} \in \widehat{\boldsymbol{\mathcal{W}}}, \tag{2.14}$$

$$b(\boldsymbol{\sigma}, v) = 0 \quad \text{for all } v \in \mathcal{V}, \tag{2.15}$$

provided that $g(\boldsymbol{w})$ is a bounded linear functional on $\boldsymbol{\mathcal{W}}$ (or $\widehat{\boldsymbol{\mathcal{W}}}$). The remaining task is to ascertain whether there exists a unique u that satisfies the original system. To accomplish this, we rewrite the first equation of Equation (2.12), with the value of $\boldsymbol{\sigma}$ set to the solution $\boldsymbol{\sigma}^*$ obtained from Equation (2.14), yielding

$$b(\boldsymbol{w}, u) = g(\boldsymbol{w}) - a(\boldsymbol{\sigma}^*, \boldsymbol{w}) \quad \text{for all } \boldsymbol{w} \in \boldsymbol{\mathcal{W}}.$$

Assuming that g is a bounded linear functional and identifying that $a(\boldsymbol{\sigma}^*, \cdot)$ is also a bounded linear functional on $\boldsymbol{\mathcal{W}}$ for a fixed $\boldsymbol{\sigma}^*$, $g(\cdot) - a(\boldsymbol{\sigma}^*, \cdot)$ qualifies as a bounded linear functional as well. By leveraging this insight along with the continuity of $b(\cdot, \cdot)$ on $\boldsymbol{\mathcal{W}} \times \mathcal{V}$, we establish the existence and uniqueness of u through a demonstration of weak coercivity.

Definition 10 ***Generalized Weak Coercivity.*** *A bilinear form, $b(\boldsymbol{w}, v)$, is weakly coercive on $\boldsymbol{\mathcal{W}} \times \mathcal{V}$ if there exists a constant $c_0 > 0$ such that*

$$\inf_{v \in \mathcal{V}} \sup_{\boldsymbol{w} \in \boldsymbol{\mathcal{W}}} \frac{b(\boldsymbol{w}, v)}{\|\boldsymbol{w}\|_{\boldsymbol{\mathcal{W}}} \|v\|_{\mathcal{V}}} \geq c_0.$$

After discussing generalized weak coercivity, it is important to highlight the significance of the inf-sup condition, which is crucial for determining the uniqueness of the solution u. The inf-sup condition, also known as the Ladyzhenskaya-Babuska-Brezzi (LBB) condition or the stability condition, is a crucial requirement in the context of mixed finite-element methods. It plays a pivotal role in ensuring stability, avoiding numerical issues, and providing accurate and reliable solutions. The inf-sup condition is essential for proving the well-posedness of the mixed Poisson problem. Without the inf-sup condition, the problem might be ill-posed, leading to numerical solutions that lack accuracy and reliability. The inf-sup condition can be understood by representing the bilinear form b as an operator $B : \boldsymbol{\mathcal{W}} \to \mathcal{V}^*$ and its adjoint $B^t : \mathcal{V} \to \boldsymbol{\mathcal{W}}^*$. The kernel of B is denoted as $\widehat{\boldsymbol{\mathcal{W}}}$, and the inf-sup condition imposes a coercivity assumption on this kernel. Essentially, the inf-sup condition ensures that for any $v \in \mathcal{V}$, there

exists a nonzero $\boldsymbol{w} \in \boldsymbol{\mathcal{W}}$ such that $b(\boldsymbol{w}, v) \geq c_0 \|\boldsymbol{w}\|_{\boldsymbol{\mathcal{W}}} \|v\|_{\mathcal{V}}$. Notably, such $\boldsymbol{w}$ cannot belong to $\widehat{\boldsymbol{\mathcal{W}}}$, as $b(\boldsymbol{w}, v) = 0$ for $\boldsymbol{w} \in \boldsymbol{\mathcal{W}}$, violating the inequality. Furthermore, optimal choices of $\boldsymbol{w}$ must be orthogonal to $\widehat{\boldsymbol{\mathcal{W}}}$ to maximize $b(\boldsymbol{w}, v)$ without increasing $\|\boldsymbol{w}\|_{\boldsymbol{\mathcal{W}}}$. This property of B implies that if $b(\boldsymbol{w}, v)$ satisfies the inf-sup condition, then B is an isomorphism from the orthogonal complement of $\widehat{\boldsymbol{\mathcal{W}}}$ onto $\mathcal{V}^*$. The inf-sup condition can be equivalently expressed as $\|B\boldsymbol{w}\|_{\mathcal{V}^*} \geq c_0 \|\boldsymbol{w}\|_{\boldsymbol{\mathcal{W}}}$ for $\boldsymbol{w} \in \widehat{\boldsymbol{\mathcal{W}}}^{\perp}$ and $\|B^t v\|_{\boldsymbol{\mathcal{W}}^*} \geq c_0 \|v\|_{\mathcal{V}}$ for $v \in \mathcal{V}$. This coercivity relation ensures the uniqueness of u as the solution to $b(\boldsymbol{w}, u) = g(\boldsymbol{w}) - a(\boldsymbol{\sigma}^*, \boldsymbol{w})$ for $\boldsymbol{w} \in \widehat{\boldsymbol{\mathcal{W}}}^{\perp}$. Furthermore, when $f \neq 0$, the inf-sup condition guarantees the existence of a unique $\boldsymbol{\sigma}_0 \in \widehat{\boldsymbol{\mathcal{W}}}^{\perp}$ such that $B\boldsymbol{\sigma}_0 = f$, leading to the final result given in Theorem 4.

Theorem 4 ***Well-Posedness of Mixed Formulation.*** *Let $\mathcal{W}$ and $\mathcal{V}$ be Hilbert spaces. Suppose that $a(\cdot,\cdot) : \boldsymbol{\mathcal{W}} \times \boldsymbol{\mathcal{W}} \to \mathbb{R}$ and $b(\cdot,\cdot) : \boldsymbol{\mathcal{W}} \times \mathcal{V} \to \mathbb{R}$ are bounded bilinear functionals, and that g and f are bounded linear functionals on $\boldsymbol{\mathcal{W}}$ and $\mathcal{V}$, respectively. If a is coercive on $\widehat{\boldsymbol{\mathcal{W}}} := \{\boldsymbol{w} \in \boldsymbol{\mathcal{W}} \mid b(\boldsymbol{w}, v) = 0$ for all $v \in \mathcal{V}\}$ and b satisfies the inf-sup condition in Definition* 15*, then there exists a unique solution, $(\boldsymbol{\sigma}, u) \in \boldsymbol{\mathcal{W}} \times \mathcal{V}$, that solves Equation* (2.12)*.*

After discussing the well-posedness of the continuum mixed formulation for the mixed Poisson problem, we now turn our attention to the discrete inf-sup condition, a crucial aspect in ensuring the stability and reliability of numerical solutions. The discrete inf-sup condition, also known as the discrete Ladyzhenskaya-Babuska-Brezzi (LBB) condition, for the mixed Poisson problem ensures the stability of the numerical discretization. In the context of finite element methods, the inf-sup condition is often expressed in terms of discrete spaces.

Let $\boldsymbol{\mathcal{W}}_h \subset \boldsymbol{\mathcal{W}}$ and $\mathcal{V}_h \subset \mathcal{V}$ be finite-dimensional subspaces (finite-element spaces) defined on some triangulation of the domain with mesh parameter h. Then, we discretize Equation (2.12) as finding $\boldsymbol{\sigma}_h \in \boldsymbol{\mathcal{W}}_h, u_h \in \mathcal{V}_h$ such that

$$a_h(\boldsymbol{\sigma}_h, \boldsymbol{w}_h) + b_h(\boldsymbol{w}_h, u_h) = g(\boldsymbol{w}_h) \quad \forall \boldsymbol{w} \in \boldsymbol{\mathcal{W}}_h, \tag{2.16a}$$

$$b_h(\boldsymbol{\sigma}_h, v_h) = f(v_h) \quad \forall v \in \mathcal{V}_h. \tag{2.16b}$$

Definition 11 ***Discrete inf-sup condition.*** *A family of finite-element spaces $(\boldsymbol{\mathcal{W}}_h, \mathcal{V}_h)$ satisfies the discrete inf-sup condition if the bilinear form $b(\boldsymbol{w}_h, v_h)$ is weakly*

coercive on $\boldsymbol{\mathcal{W}}_h \times \mathcal{V}_h$*, i.e., if there exists a constant* $\widehat{c_0} > 0$*, independent of* h*, such that*

$$\inf_{v_h \in \mathcal{V}_h} \sup_{\boldsymbol{w}_h \in \boldsymbol{\mathcal{W}}_h} \frac{b(\boldsymbol{w}_h, v_h)}{\|\boldsymbol{w}_h\|_{\boldsymbol{\mathcal{W}}_h} \|v_h\|_{\mathcal{V}_h}} \geq \widehat{c_0}.$$

Coercivity of the bilinear form $a(\cdot, \cdot)$ at the continuous level implies that it remains coercive at the discrete level when $\mathcal{V}_h \subset \mathcal{V}$. This preservation of coercivity is crucial for ensuring the stability and well-posedness of the discrete problem. Proving the discrete inf-sup condition becomes the additional requirement for establishing the well-posedness of the discrete mixed problem. This condition ensures stability and convergence of the numerical solution.

In the following, we provide an example of finite-element spaces for the Mixed-Poisson problem: $RT_0 - P_0$. Let $\{\Omega^h\}_{h>0}$ be a regular family of triangulations of Ω by triangles T in $\mathbb{R}^2$ or tetrahedra in $\mathbb{R}^3$ with diameter h_T. For each $T \in \Omega^h$, let $RT_0(T)$ denote the local Raviart-Thomas space of lowest order,

$$RT_0(T) := [P_0(T)]^n \oplus P_0(T)\boldsymbol{x},$$

where $\boldsymbol{x} := (x_1, ..., x_n)^T$ is a generic vector in $\mathbb{R}^n$, and $P_0(T)$ is the space of constant functions on T. We define $RT_0(\Omega^h)$ by requiring that restriction of $\boldsymbol{w}_h \in RT_0(\Omega^h)$ to any triangle, T, be in $RT_0(T)$, and that functions in $RT_0(\Omega^h)$ have continuous normal components between cells. In the context of infinite-dimensional spaces $\boldsymbol{\mathcal{W}} = H(\mathrm{div}, \Omega)$ and $\mathcal{V} = L^2(\Omega)$, finite element spaces on mesh Ω^h are naturally chosen as low-order spaces. These are represented as $\boldsymbol{\mathcal{W}}_h = RT_0(\Omega^h) \subset H(\mathrm{div}, \Omega)$ and $P_0(\Omega^h) \subset L^2(\Omega)$. The $RT_0 - P_0$ finite element pair, utilized for solving the mixed Poisson problem, plays a pivotal role in ensuring the stability and convergence of the numerical solution. To demonstrate the satisfaction of the discrete inf-sup condition, it is essential to verify the existence of a constant $\beta > 0$ such that:

$$\sup_{\boldsymbol{w}_h \in \boldsymbol{\mathcal{W}}_h} \frac{\langle \nabla \cdot \boldsymbol{w}_h, v_h \rangle}{\|\boldsymbol{w}_h\|_{\mathrm{div}}} \geq \beta \|v_h\|_{\mathcal{V}_h} \quad \forall v_h \in \mathcal{V}_h,$$

The proof of well-posedness is omitted here; for complete details, see [2]. Therefore, the $RT_0(\Omega^h) - P_0(\Omega^h)$ mixed finite element pair satisfies the discrete inf-sup condition, ensuring stability and convergence of the numerical solution for the mixed Poisson problem. Moreover, it can be observed that $b(\boldsymbol{w}_h, v_h)$ exhibits weak coercivity over $RT_0(\Omega^h) \times P_0(\Omega^h)$. Consequently, the mixed formulation remains well-posed

within these function spaces. Furthermore, these arguments extend seamlessly to higher-order variations of the Raviart-Thomas spaces. The Raviart-Thomas element is defined in [2] by $RT_k(T) = [P_k(T)]^n \otimes P_k(T)\boldsymbol{x}$. This generalization relies on the broader principle that $\operatorname{div} RT_k(\Omega^h) \subseteq P_k^-(\Omega^h)$, coupled with a consistently defined projection operator. In this context, $P_k^-(\Omega^h)$ denotes the space of piecewise polynomials on h, where continuity between elements is not necessarily preserved, a concept often referred to as the discontinuous Lagrange finite-element space.

2.1.3 Finite-element method for Stokes.

Let us consider the Stokes equations with Dirichlet boundary conditions for the velocity variable [9, 11, 22, 2]. The problem aims to find the velocity, $\boldsymbol{u}$, and the pressure, p, of a viscous fluid within specific function spaces satisfying

$$-\Delta \boldsymbol{u} + \nabla p = \boldsymbol{f} \quad \text{in } \Omega, \tag{2.17a}$$

$$-\nabla \cdot \boldsymbol{u} = 0 \quad \text{in} \quad \Omega, \tag{2.17b}$$

$$\boldsymbol{u} = 0 \quad \text{on } \partial\Omega, \tag{2.17c}$$

where $\boldsymbol{f}$ represents a known forcing term. Under these boundary conditions, the pressure only enters the Stokes equation inside a gradient.

Remark 2 *Let $\mathcal{V} = L_0^2(\Omega) = \{p \in L^2(\Omega) \mid \int_\Omega p\,dx = 0\}$. A suitable choice for the pressure space is $L^2(\Omega)$. It's important to note that $\int_\Omega div\,\boldsymbol{v}\,dx = \int_{\partial\Omega} \boldsymbol{v}\cdot\boldsymbol{n}\,dS = 0$ due to the boundary conditions. Consequently, the divergence operator maps $\boldsymbol{H}_0^1(\Omega)$ to the subspace $L_0^2(\Omega)$, where the pressure solving the Stokes equations is unique. However, in $L^2(\Omega)$, it is only unique up to a constant.*

Consequently, solving these equations for p may result in an indeterminate solution up to an additive constant. To address this, under these boundary conditions, we impose the additional condition that

$$\int_\Omega p\,dx = 0.$$

As above, we use the mathematical framework of multiplying by a test function $\boldsymbol{v} \in \boldsymbol{\mathcal{W}} = \boldsymbol{H}_0^1(\Omega)$ in Equation (2.17a) and $q \in L_0^2(\Omega) = \mathcal{V}$ in Equation (2.17b).

Subsequently, we integrate by parts in Equation (2.17a) to obtain the following weak variational formulation: Find $\boldsymbol{u} \in \boldsymbol{\mathcal{W}}$ and $p \in \mathcal{V}$ such that

$$\langle \nabla \boldsymbol{u}, \nabla \boldsymbol{v} \rangle - \int_{\partial\Omega} (\nabla \boldsymbol{u}) \boldsymbol{n} \cdot \boldsymbol{v} \, ds - \langle p, \nabla \cdot \boldsymbol{v} \rangle + \int_{\partial\Omega} p \, \boldsymbol{v} \cdot \boldsymbol{n} \, ds = \langle \boldsymbol{f}, \boldsymbol{v} \rangle \quad \forall \boldsymbol{v} \in \boldsymbol{\mathcal{W}},$$
$$\langle -\nabla \cdot \boldsymbol{u}, q \rangle = 0 \quad \forall q \in \mathcal{V}.$$

This leads to the following discrete linear system:

$$F\boldsymbol{u} + B\boldsymbol{p} = \boldsymbol{f}, \quad B^T \boldsymbol{u} = \boldsymbol{0} \tag{2.18}$$

where F is the discrete Laplacian (stiffness matrix) arising from the term $\langle \nabla \boldsymbol{u}, \nabla \boldsymbol{v} \rangle - \int_{\partial\Omega} (\nabla \boldsymbol{u}) \boldsymbol{n} \cdot \boldsymbol{v} \, ds$, and B is the discrete divergence operator matrix arising from the term $\langle -p, \nabla \cdot \boldsymbol{v} \rangle + \int_{\partial\Omega} p \, \boldsymbol{v} \cdot \boldsymbol{n} \, ds$.
By choosing $\boldsymbol{u} \in \boldsymbol{\mathcal{W}} = \boldsymbol{H}_0^1(\Omega)$ we derive the mixed formulation of the Stokes equations: to find $\boldsymbol{u} \in \boldsymbol{H}_0^1(\Omega)$ and $p \in L_0^2(\Omega)$ such that

$$\langle \nabla \boldsymbol{u}, \nabla \boldsymbol{v} \rangle - \langle p, \nabla \cdot \boldsymbol{v} \rangle = \langle \boldsymbol{f}, \boldsymbol{v} \rangle \quad \forall \boldsymbol{v} \in \boldsymbol{\mathcal{W}}, \tag{2.19a}$$
$$\langle -\nabla \cdot \boldsymbol{u}, q \rangle = 0 \quad \forall q \in \mathcal{V}. \tag{2.19b}$$

We can express this in the generalized saddle-point form by defining the bilinear and linear forms as follows:

$$\begin{aligned} a(\boldsymbol{u}, \boldsymbol{v}) &= \langle \nabla \boldsymbol{u}, \nabla \boldsymbol{v} \rangle \quad \text{for } \boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{\mathcal{W}}, \\ b(\boldsymbol{u}, q) &= \langle -\nabla \cdot \boldsymbol{u}, q \rangle \quad \text{for } \boldsymbol{u} \in \boldsymbol{\mathcal{W}}, \, q \in \mathcal{V}, \\ g(\boldsymbol{v}) &= \langle \boldsymbol{f}, \boldsymbol{v} \rangle \quad \text{for } \boldsymbol{v} \in \mathcal{V}, \\ f(q) &= 0 \quad \text{for } q \in \mathcal{V}. \end{aligned}$$

Remark 3 *In a similar approach, for Stokes equations with non-homogeneous Dirichlet boundary conditions $\boldsymbol{u}|_{\partial\Omega} = \boldsymbol{g}$, the data $\boldsymbol{g}$ must follow satisfy the compatibility condition $\int_{\partial\Omega} \boldsymbol{g} \cdot \boldsymbol{n} \, dS = \int_{\partial\Omega} div \, \boldsymbol{u} \, dx = 0$.*

The authors in [2] establish the well-posedness of the weak variational formulation of the mixed formulation of the Stokes equations in $\boldsymbol{H}_0^1(\Omega) \times L_0^2(\Omega)$. Once more, given the nature of $a(\cdot, \cdot)$, coercivity on the discrete level is inherent. Therefore, our focus shifts solely to verifying the discrete inf-sup conditions for Stokes within each space

pairing.
The pair $P_k(\Omega_h) \times P_{k-1}(\Omega_h)$ is stable for Stokes problems when $k \geq 2$. The simplest and most popular case is $k = 2$, where the pair (P_2, P_1) is used. This configuration is preferred over (P_1, P_0), which is unstable, as (P_2, P_1) provides a stable solution with one order higher approximation. While (P_2, P_1) uses fewer degrees of freedom compared to the stable pair (P_2, P_0), it still delivers a higher-order approximation, making it a favored choice for many applications.

For the (P_2, P_1) space, P_2 elements are used for velocity, placing nodes at both vertices and midpoints of each element, resulting in $(2n + 1) \times (2n + 1)$ nodes per velocity component. Pressure is represented with P_1 elements, which have nodes only at the vertices, leading to $(n + 1) \times (n + 1)$ pressure nodes. In contrast, the (P_2, P_0) space also employs P_2 elements for velocity but uses P_0 elements for pressure, with a single node at the center of each element, resulting in $2n \times n$ pressure nodes. Thus the (P_2, P_0) space has more pressure degrees of freedom when $n > 2$ resulting in a greater number of variables to solve and potentially increased computational effort.

2.1.4 Finite-element method for Poroelasticity

Poroelastic models serve as crucial tools for understanding mechanical deformation and fluid flow in porous media, with applications spanning various fields including medicine, biophysics, and geosciences. These models find utility in computations related to intracranial pressure, trabecular bone stiffness, reservoir simulation, and waste repository performance, among others [5, 6, 17, 19, 23, 25, 29, 31, 35, 30]. Constructing stable finite-element schemes for poroelastic models often involves selecting discrete spaces that adhere to appropriate inf-sup (or LBB) conditions or employing stabilization techniques to mitigate instabilities in finite-element pairs. In the realm of two-field formulations like Biot's problem, classical Taylor-Hood elements represent one approach, while recent work has explored stabilized discretizations using linear finite elements for both displacements and pressure [6, 23, 29, 31]. For three-field formulations incorporating the Darcy velocity, various conforming and nonconforming discretizations leveraging Stokes-stable finite-element spaces have been proposed. Notably, recent studies have introduced stable finite-element methods utilizing piecewise constants for pressure and parameter-robust three-field finite-element schemes, accompanied by a general theory for error analysis [6, 23, 31].

A classical and widely used model, introduced by Biot [8], is based on the following assumptions [23]:

1. The porous medium is saturated with fluid and maintains a constant temperature.
2. The fluid within the porous medium exhibits near-incompressibility.
3. The solid skeleton or matrix is comprised of an elastic material, with deformations and strains being relatively minor.
4. Fluid flow adheres to Darcy's law, suggesting laminar flow behavior.

Consider the quasi-static Biot model for soil consolidation. For a porous medium characterized by linear elasticity, homogeneity, and isotropy, and saturated with an incompressible Newtonian fluid, the consolidation process is described by the following system of partial differential equations in a domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$ with a sufficiently smooth boundary $\Gamma = \partial\Omega$ [5, 6, 31]:

$$\text{Equilibrium equation:} \quad -\operatorname{div}\boldsymbol{\sigma'} + \alpha\nabla p = \rho\boldsymbol{g} \quad \text{in } \Omega, \tag{2.20}$$

$$\text{Constitutive equation:} \quad \boldsymbol{\sigma'} = 2\mu\varepsilon(\boldsymbol{u}) + \lambda\operatorname{div}(\boldsymbol{u})I \quad \text{in } \Omega, \tag{2.21}$$

$$\text{Compatibility condition:} \quad \varepsilon(\boldsymbol{u}) = \frac{1}{2}(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^t) \quad \text{in } \Omega, \tag{2.22}$$

$$\text{Darcy's law:} \quad \boldsymbol{w} = -\frac{1}{\mu_f}\boldsymbol{K}(\nabla p - \rho_f\boldsymbol{g}) \quad \text{in } \Omega, \tag{2.23}$$

$$\text{Continuity equation:} \quad \frac{\partial}{\partial t}\left(\frac{1}{M}p + \alpha\operatorname{div}\boldsymbol{u}\right) + \operatorname{div}\boldsymbol{w} = f \quad \text{in } \Omega. \tag{2.24}$$

Here, μ_f is the viscosity of the fluid, I is the identity tensor, M is the Biot modulus, ρ and ρ_f are the bulk density and fluid density, respectively, and $\alpha = 1 - \dfrac{K_b}{K_s}$ is the Biot-Willis constant, with K_b and K_s denoting the drained and the solid-phase bulk moduli, respectively. The absolute permeability tensor is given by $\boldsymbol{K}$ which is symmetric and positive definite. The unknown functions are the displacement vector $\boldsymbol{u}$, the pore pressure p, and the percolation velocity of the fluid, or Darcy velocity, relative to the soil, $\boldsymbol{w}$. The vector-valued function $\boldsymbol{g}$ represents the gravitational force. Finally, $\mu = \dfrac{E}{2+2\nu}$ and $\lambda = \dfrac{E\nu}{(1-2\nu)(1+\nu)}$are the Lamé coefficients where ν is the Poisson ratio and E is the Young's modulus. As $\nu \to 0.5$, we have $\lambda \to \infty$, the

incompressible limit that causes difficulties in numerical simulations. The implications of $\lambda \to \infty$ are significant. As the material approaches incompressibility, $\mathrm{div}(\boldsymbol{u}) \approx 0$ becomes a constraint, leading to numerical challenges such as locking in standard finite element methods. To address these, specialized techniques like mixed formulations or stabilization methods are employed. These simulations demonstrate the necessity for advanced methods to handle the incompressible limit. The source term f represents a forced fluid extraction or injection process. Finally, this system is often subject to the following set of boundary conditions [5, 6, 31]:

$$\begin{aligned} p = 0, \quad \text{for } x \in \bar{\Gamma}_t, \quad \boldsymbol{\sigma'} \boldsymbol{n} = 0, \quad \text{for } x \in \Gamma_t, \\ \boldsymbol{u} = 0, \quad \text{for } x \in \bar{\Gamma}_c, \quad \frac{\partial p}{\partial \boldsymbol{n}} = 0, \quad \text{for } x \in \Gamma_c, \end{aligned}$$

where $\boldsymbol{n}$ is the outward unit normal to the boundary, $\bar{\Gamma} = \bar{\Gamma}_t \cup \bar{\Gamma}_c$, with Γ_t and Γ_c being open (with respect to Γ) subsets of Γ with nonzero measure. Appropriate initial conditions for the pressure and displacement (more precisely, for $\mathrm{div}(\boldsymbol{u})$) are also needed.

We consider a semi-discretized variational problem such that for each $t \in (0, T]$, $(\boldsymbol{u}(t), p(t), \boldsymbol{w}(t)) \in \boldsymbol{\mathcal{V}} \times Q \times \boldsymbol{\mathcal{W}}$ with

$$\begin{aligned} \boldsymbol{\mathcal{V}} &= \{\boldsymbol{u} \in \mathbf{H}^1(\Omega) \,|\, \boldsymbol{u}|_{\bar{\Gamma}_c} = 0\}, \;\; Q = L^2(\Omega), \\ \boldsymbol{\mathcal{W}} &= \{\boldsymbol{w} \in \mathbf{H}(\mathrm{div}, \Omega) \,|\, (\boldsymbol{w} \cdot \boldsymbol{n})|_{\Gamma_c} = 0\}. \end{aligned}$$

Using backward Euler as a time discretization on a time interval $(0, T]$ with constant time-step size τ , the fully discrete variational form for Biot's three-field consolidation model, $(2.19) - (2.23)$, is written as: Find $(\boldsymbol{u}_h^m, p_h^m, \boldsymbol{w}_h^m) \in \boldsymbol{\mathcal{V}}_h \times Q_h \times \boldsymbol{\mathcal{W}}_h$ such that

$$a(\boldsymbol{u}_h^m, \boldsymbol{v}_h) - \langle \alpha p_h^m, \mathrm{div} \boldsymbol{v}_h \rangle = \langle \rho \boldsymbol{g}, \boldsymbol{v}_h \rangle, \quad \forall \boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h, \tag{2.25}$$

$$\tau \langle \boldsymbol{K}^{-1} \mu_f \boldsymbol{w}_h^m, \boldsymbol{r}_h \rangle - \tau \langle p_h^m, \mathrm{div} \boldsymbol{r}_h \rangle = \tau \langle \rho_f \boldsymbol{g}, \boldsymbol{r}_h \rangle, \quad \forall \boldsymbol{r}_h \in \boldsymbol{\mathcal{W}}_h, \tag{2.26}$$

$$-\langle \frac{1}{M} p_h^m, q_h \rangle - \langle \alpha \mathrm{div} \boldsymbol{u}_h^m, q_h \rangle - \tau \langle \mathrm{div} \boldsymbol{w}_h^m, q_h \rangle = \langle \hat{f}, q_h \rangle, \quad \forall q_h \in Q_h, \tag{2.27}$$

where $\langle \cdot, \cdot \rangle$ denotes the standard $L^2(\Omega)$ inner product. Here, $(\boldsymbol{u}_h^m, p_h^m, \boldsymbol{w}_m^h)$ is an approximation to $(\boldsymbol{u}(\cdot, t_m), p(\cdot, t_m), \boldsymbol{w}(\cdot, t_m))$, at time $t_m = m\tau, m = 1, 2, ..., \langle \hat{f}, q_h \rangle =$

$\tau\langle f, q_h\rangle + \langle \frac{1}{M} p_h^{m-1}, q_h\rangle + \langle \alpha \mathrm{div} \boldsymbol{u}_h^{m-1}, q_h\rangle$, and $a(\boldsymbol{u}, \boldsymbol{v}) = 2\mu(\epsilon(\boldsymbol{u}), \epsilon(\boldsymbol{v})) + \lambda(\mathrm{div}\boldsymbol{u}, \mathrm{div}\boldsymbol{v})$ is the usual weak form for linear elasticity. Note that (2.25) has been scaled by τ and (2.26) has been scaled by -1 to make the system symmetric.

Definition 12 *[31] The triple of spaces $(\boldsymbol{\mathcal{V}}_h, \boldsymbol{\mathcal{W}}_h, Q_h)$ is Stokes-Biot stable if and only if the following conditions are satisfied:*

- $a(\boldsymbol{u}_h, \boldsymbol{v}_h) \leq C_{\boldsymbol{V}} \|\boldsymbol{u}_h\|_1 \|\boldsymbol{v}_h\|_1$, *for all* $\boldsymbol{u}_h \in \boldsymbol{\mathcal{V}}_h$, $\boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h$;
- $a(\boldsymbol{u}_h, \boldsymbol{u}_h) \geq \theta_{\boldsymbol{V}} \|\boldsymbol{u}_h\|_1^2$, *for all* $\boldsymbol{u}_h \in \boldsymbol{\mathcal{V}}_h$;
- *The pair of spaces $(\boldsymbol{\mathcal{W}}_h, Q_h)$ is Poisson stable, i.e., it satisfies stability and continuity conditions required by the mixed discretization of the Poisson equation;*
- *The pair of spaces $(\boldsymbol{\mathcal{V}}_h, Q_h)$ is Stokes stable, i.e., it satisfies the inf-sup stability condition for the Stokes equations.*

The authors in [31] suggest a parameter-robust stable scheme for Biot's system, building upon the conditions mentioned earlier. Inspired by this approach, we now define a norm on $(\boldsymbol{\mathcal{V}}_h, \boldsymbol{\mathcal{W}}_h, Q_h)$:

$$|||(\boldsymbol{u}_h, \boldsymbol{w}_h, p_h)|| := \left[\|\boldsymbol{u}_h\|_A + \tau \|\boldsymbol{w}_h\|^2_{K^{-1}\mu_f} + \tau^2 \xi^{-1} \|\mathrm{div}\, \boldsymbol{w}_h\|^2 + \xi \|p_h\|^2 \right]^{1/2},$$

where $\zeta = \sqrt{\lambda + \frac{2\mu}{d}}, \xi = \frac{\alpha^2}{\zeta^2} + \frac{1}{M}, \|\boldsymbol{r}\|_{K^{-1}\mu_f} = \left(K^{-1}\mu_f \boldsymbol{r}, \boldsymbol{r}\right)^{1/2}$.
Further, we associate a composite bilinear form on the space, $(\boldsymbol{\mathcal{V}}_h, \boldsymbol{\mathcal{W}}_h, Q_h)$,
$B(\boldsymbol{u}_h, \boldsymbol{w}_h, p_h; \boldsymbol{v}_h, \boldsymbol{r}_h, q_h) = a(\boldsymbol{u}_h^m, \boldsymbol{v}_h) - \langle \alpha p_h^m, \mathrm{div}\boldsymbol{v}_h\rangle + \tau\langle \boldsymbol{K}^{-1}\mu_f \boldsymbol{w}_h^m, \boldsymbol{r}_h\rangle - \tau\langle p_h^m, \mathrm{div}\boldsymbol{r}_h\rangle - \langle \frac{1}{M} p_h^m, q_h\rangle - \langle \alpha \mathrm{div}\boldsymbol{u}_h^m, q_h\rangle - \tau\langle \mathrm{div}\boldsymbol{w}_h^m, q_h\rangle$.

To ensure stability and convergence of the discretisation, the discrete subspace (mixed element) has to be chosen such that the following theorem is fullfilled:

Theorem 5 *[31]. If the triple $(\boldsymbol{\mathcal{V}}_h, \boldsymbol{\mathcal{W}}_h, Q_h)$ is Stokes-Biot stable, then: $B(\cdot,\cdot,\cdot\ ;\ \cdot,\cdot,\cdot)$ is continuous with respect to $|||\cdot,\cdot,\cdot|||$; and the following inf-sup condition holds:*

$$\sup_{(\boldsymbol{v}_h, \boldsymbol{r}_h, q_h) \in \boldsymbol{\mathcal{V}}_h \times \boldsymbol{\mathcal{W}}_h \times Q_h} \frac{B(\boldsymbol{u}_h, \boldsymbol{w}_h, p_h; \boldsymbol{v}_h, \boldsymbol{r}_h, q_h)}{|||(\boldsymbol{u}_h, \boldsymbol{w}_h, p_h)|||} \geq \beta |||(\boldsymbol{v}_h, \boldsymbol{r}_h, q_h)|||$$

with a constant $\beta > 0$ independent of mesh size h, time step size δ, and the physical parameters.

The next section will cover multigrid methods, including their motivation, application to the Poisson equation, and usage in solving systems of equations.

2.2 Multigrid methods

Consider the linear system $A\boldsymbol{u} = \boldsymbol{f}$, where A is a linear operator and $\boldsymbol{u}$ and $\boldsymbol{f}$ are vectors. This equation represents a fundamental problem in linear algebra and numerical analysis. The matrix A encodes the relationships between the elements of the vector $\boldsymbol{u}$ and the vector $\boldsymbol{f}$. Solving this system typically involves finding a solution $\boldsymbol{u}$ that satisfies the equation for a given right-hand side vector $\boldsymbol{f}$. Depending on the properties of A and $\boldsymbol{f}$, this problem may have unique solutions, infinitely many solutions, or no solutions at all. Solutions to $A\boldsymbol{u} = \boldsymbol{f}$ are crucial in various scientific and engineering applications, including solving differential equations, image processing, optimization problems, and data analysis.

In numerical analysis, the solution of $A\boldsymbol{u} = \boldsymbol{f}$ often involves the use of efficient algorithms tailored to exploit the structure of A. Sparse direct solvers are specialized algorithms designed to efficiently solve linear systems where A is sparse, meaning it contains mostly zero entries. These solvers exploit the sparsity of A to reduce computational complexity and memory requirements, making them particularly suitable for large-scale problems arising in scientific computing and engineering. Additionally, multigrid methods provide another approach for solving $A\boldsymbol{u} = \boldsymbol{f}$, especially for problems arising from discretizations of partial differential equations. Multigrid methods leverage a hierarchy of grids to accelerate convergence by effectively smoothing out error components at different spatial scales. This hierarchical approach makes multigrid methods highly efficient for solving large linear systems arising from discretizations of elliptic and parabolic partial differential equations.

Sparse direct solvers offer the capability to tackle exceptionally large problems that conventional "dense" solvers cannot handle efficiently. Sparse matrices can be broadly categorized into structured and unstructured types. Structured matrices exhibit a regular pattern in their nonzero entries, often along a few diagonals or in blocks of the same size forming a regular pattern, typically along a few (block) diagonals. Conversely, matrices with irregularly located entries are termed irregularly structured.

In practice, many finite-element or finite-volume methods applied to intricate geometries result in irregularly structured matrices [2, 32]. Many sparse direct methods aim to reduce computational cost by minimizing "fill-ins", which are non-zero elements introduced during the matrix's LU factorization process from initially zero positions. A common approach for solving sparse matrices involves four main steps. Firstly, preordering techniques such as minimum degree (MD) or nested dissection (ND) ordering are applied to minimize fill-in. Secondly, a symbolic factorization is conducted, where the factorization is computed without numerical values. Thirdly, the numerical factorization takes place, resulting in the formation of the actual factors L and U. Lastly, forward and backward triangular sweeps are performed for each individual right-hand side. The MD algorithm is widely recognized as the go-to approach for minimizing fill-in during sparse Gaussian elimination, especially for symmetric positive definite (SPD) matrices. In each step of the Gaussian elimination process, this algorithm chooses the node with the lowest degree as the next pivot row. This systematic selection helps in decreasing the amount of fill-in that occurs.

Definition 13 *[2, 32] Let π be a permutation of $\{1, 2, \ldots, n\}$ and define the permutation matrix, P, such that*

$$p_{i,j} = \begin{cases} 1 & \text{if } j = \pi(i), \\ 0 & \text{otherwise.} \end{cases}$$

Since P is an orthogonal matrix, with $P^{-1} = P^T$, solving $A\boldsymbol{u} = \boldsymbol{f}$ is equivalent to solving $(P^T AP)P^T\boldsymbol{u} = P^T\boldsymbol{f}$. The reordered matrix $P^T AP$ is a transformed version of the original system.

The nested dissection (ND) algorithm operates by identifying a separator in a graph, which is a set of nodes that, when removed, divides the graph into two or more disconnected parts. The ordering of nodes in the permutation is based on a sequence of separators selected in the graph. To achieve a target cost of $O(N^3)$ on an $N \times N$ mesh, it is aimed to utilize $O(N)$ separator nodes. This allows dense Gaussian elimination on a matrix involving this set to have an $O(N^3)$ cost for factorization. For instance, the "central cross" in the graph serves as a separator for a regular 7×7 mesh, comprising the nodes adjacent to the red edges in Figure 2.1. These nodes are ordered last, in lexicographic order, in the permutation. The algorithm is then

recursively applied to the four smaller 3×3 meshes, each with its central cross defined (nodes adjacent to the green edges), resulting in four subproblems on 1×1 meshes, or single nodes. In this process, each 3×3 separator is ordered immediately after the nodes in its four subproblems. To analyze the factorization complexity based on this reordering, the partial problem after the first reordering is examined. Assuming N is odd, as in the given example, the central cross consists of $2N - 1$ nodes, and each of the four subproblems has a size of $\frac{N-1}{2} \times \frac{N-1}{2}$. As these subproblems are entirely disjoint, the reordered system matrix was represented in [2] as:

$$P^T AP = \begin{pmatrix} A_{1,1} & 0 & 0 & 0 & A_{1,s} \\ 0 & A_{2,2} & 0 & 0 & A_{2,s} \\ 0 & 0 & A_{3,3} & 0 & A_{3,s} \\ 0 & 0 & 0 & A_{4,4} & A_{4,s} \\ A_{s,1} & A_{s,2} & A_{s,3} & A_{s,4} & A_{s,s} \end{pmatrix}$$

where $A_{i,i}$, $1 \leq i \leq 4$, correspond to the matrix restricted to each of the disjoint subdomains, while $A_{i,s}$ and $A_{s,i}$ for $1 \leq i \leq 4$ contain the connections from each subdomain to/from the separator (the central cross). Similarly, $A_{s,s}$ contains the connections in the original matrix between nodes in the separator. The cost of factoring the matrix in this reordering can easily be accounted for by summing the costs of factoring each of the $A_{i,i}$ plus those associated with the separator. As above, we assume the cost of accounting for the separator is $O(N^3)$, equal to that of dense Gaussian Elimination on $A_{s,s}$. The proof considers scenarios where $N = 2^k - 1$ so that recursively-defined subproblems are also one less than a power of two. Let $\theta(N)$ represent the cost of factoring the reordered matrix $P^T AP$ for an $N \times N$ mesh. The basic recursion is $\theta(N) = 4\theta(\frac{N-1}{2}) + O(N^3)$. Writing $\theta_k = \theta(2^k - 1)$, this recurrence relation becomes $\theta_k = 4\theta_{k-1} + c8^k$ for some constant θ. Further analysis yields the relation $\theta_{k+1} - 12\theta_k + 32\theta_{k-1} = 0$, which, using the ansatz that $\theta_k = s^k$, leads to the quadratic form $s^2 - 12s + 32 = 0$. Solving this gives $s = 4$ and $s = 8$, leading to the general solution $\theta_k = c_1 4^k + c_2 8^k = O((2^k - 1)^3)$. A detailed calculation confirms $\theta(N) = O(N^3)$, with a constant approximately 10. Unfortunately, the method described above does not easily extend to general cases. Additionally, separators can be identified in linear time, leading to similar bounds on factorization costs for certain types of two-dimensional discretizations. For regular $N \times N \times N$ grids in three dimensions, using geometric separators similar to "central crosses" in two dimensions

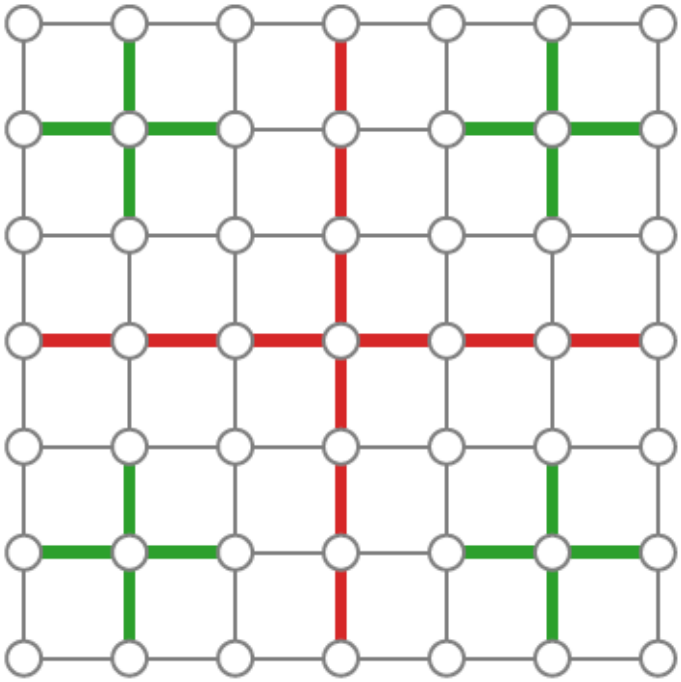

Figure 2.1: ND ordering

yields an $O(N^5)$ factorization cost. While this is a significant improvement over the $O(N^7)$ cost of factorization using banded Gaussian elimination in lexicographical order, it remains impractical for even moderate values of N. Generalizing this result to unstructured meshes in three-dimensional geometries is challenging, prompting exploration of alternative approaches.

Consider again a sparse linear system, $A\boldsymbol{u} = \boldsymbol{f}$, and let $\boldsymbol{e}_k = \boldsymbol{u} - \boldsymbol{u}_k$ represent the error in an approximation, $\boldsymbol{u}_k$, This error is generally unknown and not computable, as it requires knowledge of the exact solution, $\boldsymbol{u}$. We focus on iterative methods of the general form as in [2, 32]

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \alpha_k \delta \boldsymbol{u}_k,$$

for a scalar α_k and a vector $\delta \boldsymbol{u}_k$. The ideal choice for the update would satisfy $\alpha_k \delta \boldsymbol{u}_k = \boldsymbol{e}_k$. Therefore, we typically seek to approximate this ideal relation. One common approach is to employ the residual, $\boldsymbol{r}_k = \boldsymbol{f} - A\boldsymbol{u}_k$, as an approximation for $\boldsymbol{e}_k$. This proves effective when A is close to the identity matrix in some sense, since $\boldsymbol{r}_k = \boldsymbol{f} - A\boldsymbol{u}_k = A\boldsymbol{u} - A\boldsymbol{u}_k = A\boldsymbol{e}_k$; thus, if $A \approx I$, then $\boldsymbol{r}_k \approx \boldsymbol{e}_k$. Considering an

initial approximation or guess, $\boldsymbol{u}_0$, we iterate to arrive at

$$\boldsymbol{u}_k = \boldsymbol{u}_{k-1} + \alpha_k(\boldsymbol{f} - A\boldsymbol{u}_{k-1}).$$

To examine the convergence behavior of this iteration, we study the evolution of the error. By subtracting both sides of the preceding expression that defines $\boldsymbol{u}_k$ from the true solution, $\boldsymbol{u}$, we obtain:

$$\begin{aligned}
\boldsymbol{u} - \boldsymbol{u}_k &= \boldsymbol{u} - \boldsymbol{u}_{k-1} - \alpha_k(\boldsymbol{f} - A\boldsymbol{u}_{k-1}) \\
\boldsymbol{e}_k &= \boldsymbol{e}_{k-1} - \alpha_k A\boldsymbol{e}_{k-1} \\
\boldsymbol{e}_k &= (I - \alpha_k A)\boldsymbol{e}_{k-1} \\
\boldsymbol{e}_k &= \left(\prod_{i=1}^{k}(I - \alpha_i A)\right)\boldsymbol{e}_0.
\end{aligned}$$

We frequently denote the matrix in the final expression as $p_k(A) = \prod_{i=1}^{k}(I - \alpha_i A)$, recognizing this as a degree-k polynomial in the matrix A. It possesses the additional property that $p_k(0) = I$, where I denotes the identity matrix and the all-zero matrix is represented by 0. We can combine polynomial methods with the concept of preconditioning, as illustrated by the equation

$$MA\boldsymbol{u} = M\boldsymbol{f} \quad \text{(left preconditioning)}$$

or

$$AM(M^{-1}\boldsymbol{u}) = \boldsymbol{f} \quad \text{(right preconditioning)},$$

where the preconditioning matrix (or preconditioner), M, is an invertible matrix chosen to accelerate convergence. If we (left) precondition the system and iterate as above, then the iteration becomes

$$\begin{aligned}
\boldsymbol{u}_k &= \boldsymbol{u}_{k-1} + \alpha_k M(f - A\boldsymbol{u}_{k-1}) \\
\boldsymbol{e}_k &= (I - \alpha_k MA)\boldsymbol{e}_{k-1} \\
&= p_k(MA)\boldsymbol{e}_0.
\end{aligned}$$

Combining polynomial methods with preconditioning further improves solver performance by leveraging the benefits of both approaches. Preconditioning improves the

conditioning of the system matrix, which accelerates convergence rates. The choice of preconditioner is crucial and depends on its ability to modify the spectral properties of the matrix. Common strategies include diagonal scaling, incomplete factorizations, and multigrid methods. By selecting and designing preconditioners tailored to specific problem characteristics, significant improvements in solver efficiency and robustness can be achieved. Overall, integrating polynomial methods with preconditioning represents an effective strategy for solving large-scale linear systems efficiently in various scientific and engineering applications [32].

One approach to developing iterative methods is through a "matrix splitting", writing $A = M^{-1} - N$. Then, $A\boldsymbol{u} = \boldsymbol{f}$ is equivalent to $M^{-1}\boldsymbol{u} = N\boldsymbol{u} + \boldsymbol{f}$, which extends to an iteration of the form

$$M^{-1}\boldsymbol{u}_k = N\boldsymbol{u}_{k-1} + \boldsymbol{f}$$

which further leads to

$$\begin{aligned}\boldsymbol{u}_k &= MN\boldsymbol{u}_{k-1} + M\boldsymbol{f} \\ &= \boldsymbol{u}_{k-1} + M(\boldsymbol{f} - A\boldsymbol{u}_{k-1})\end{aligned}$$

We equate this iteration with a preconditioned polynomial method with all weights equal, $\alpha_i = 1$, and with preconditioner M — this is termed a stationary iterative method.

The Jacobi method is an iterative technique used to solve linear systems of equations. In this method, the system matrix is split into diagonal and off-diagonal components. Iterative updates are then applied using only the diagonal elements of the system matrix. Each iteration involves solving a set of one-dimensional equations. The Gauss-Seidel method is a similar iterative technique used to solve systems of linear equations, particularly when the coefficient matrix is diagonally dominant or symmetric and positive definite. Numerous classical methods prioritize achieving invertibility without necessarily emphasizing convergence. In this context, we align with this tradition and explore classical iterations where computing $\boldsymbol{u}_k$ is computationally feasible. Subsequently, we investigate the conditions required on matrix A to ensure rapid convergence of the iteration. To facilitate our analysis, we utilize the notation $A = D - L - U$, where D represents a diagonal matrix, and L and U denote

strictly lower and upper-triangular matrices, respectively. Within this framework, two standards options for matrix splittings emerge:

- Jacobi: $M^{-1} = D$
- Gauss-Seidel: $M^{-1} = D - L$

In iteration forms, these yield

- Jacobi: $\boldsymbol{u}_k = \boldsymbol{u}_{k-1} + D^{-1}(\boldsymbol{f} - A\boldsymbol{u}_{k-1}) = D^{-1}((L+U)\boldsymbol{u}_{k-1} + \boldsymbol{f})$
- Gauss-Seidel: $\boldsymbol{u}_k = \boldsymbol{u}_{k-1} + (D-L)^{-1}(\boldsymbol{f} - A\boldsymbol{u}_{k-1}) = (D-L)^{-1}(U\boldsymbol{u}_{k-1} + \boldsymbol{f})$

Iterative methods offer an attractive alternative to direct methods due to their optimal cost per iteration, operating by refining an initial guess until convergence is reached. They typically exhibit linear or near-linear computational complexity per iteration, making them more scalable for large problems compared to sparse direct solvers. However, iterative methods may converge slowly for certain PDEs or discretizations, particularly for problems with highly oscillatory or rapidly varying solutions. This slow convergence can offset their lower per-iteration cost.

Multigrid methods address these limitations by combining the scalability of iterative methods with the effectiveness of direct methods for smoother components of the solution. They exploit the multi-resolution nature of the problem to rapidly converge to an accurate approximate solution by efficiently handling both low-frequency and high-frequency components of the error. This makes them particularly well-suited for problems like the Poisson equation or other elliptic PDEs. The multigrid methods can serve as standalone iterative solvers or as effective preconditioners. By combining coarse-grid correction with relaxation techniques, multigrid methods accelerate convergence by damping high-frequency errors on fine grids while preserving low-frequency errors. Additionally, they exploit grid hierarchies, allowing for efficient information transfer between grids and systematic error correction across scales [11, 2].

2.2.1 Multigrid methods for Poisson

In the section about the multigrid method for solving the Poisson equation, we'll cover a few important topics. First, we'll talk about how the weighted-Jacobi iteration helps

to make our solutions better over time. Then, we'll explain why it's useful to correct mistakes on a simpler version of the problem. After that, we'll introduce two different methods, called the two-grid and multigrid algorithms, and talk about different ways to use them. Finally, we'll look at how much it costs to use these methods and how well they work, and we'll also talk about a specific technique called the V-cycle.

Let's consider the performance of the weighted Jacobi iteration for the one-dimensional Poisson problem, discretized using finite differences on a uniform mesh, $x_0, x_1, \ldots, x_n$, with spacing $h = 1/n$. The discretization is then

$$(A\boldsymbol{u})_i = \frac{1}{h^2}(-\boldsymbol{u}_{i-1} + 2\boldsymbol{u}_i - \boldsymbol{u}_{i+1}) \quad \text{for} \quad 1 \leq i \leq n-1,$$

where we implicitly take $\boldsymbol{u}_0 = \boldsymbol{u}_n = 0$. A direct calculation shows that the eigenvectors $\boldsymbol{v}^{(k)}$ of A can be written as $\boldsymbol{v}_i^{(k)} = \sin\left(\frac{k\pi i}{n}\right)$ for $\quad 1 \leq k \leq n-1$, as

$$\begin{aligned}
(A\boldsymbol{v}^{(k)})_i &= \frac{1}{h^2}\left(-\sin\left(\frac{k\pi(i-1)}{n}\right) + 2\sin\left(\frac{k\pi i}{n}\right) - \sin\left(\frac{k\pi(i+1)}{n}\right)\right) \\
&= \frac{1}{h^2}\left(2\sin\left(\frac{k\pi i}{n}\right) - 2\cos\left(\frac{k\pi}{n}\right)\sin\left(\frac{k\pi i}{n}\right)\right) \\
&= \frac{4}{h^2}\sin^2\left(\frac{k\pi}{2n}\right)\sin\left(\frac{k\pi i}{n}\right) \\
&= \frac{4}{h^2}\sin^2\left(\frac{k\pi}{2n}\right)\boldsymbol{v}_i^{(k)}.
\end{aligned}$$

In [2], it is shown that $\boldsymbol{v}^{(k)}$ is an eigenvector of matrix A with eigenvalue approximately $k^2\pi^2$ for $k \leq n$. This corresponds to the convergence of discrete operator eigenvalues to those of the continuous operator as $n \to \infty$. Furthermore, the error-propagation operator for the weighted-Jacobi iteration is expressed as $\boldsymbol{e}_{k+1} = (I - \omega D^{-1}A)\boldsymbol{e}_k$, where $D = \frac{2}{h^2}I$. The convergence of the weighted-Jacobi iteration can be analyzed for $0 < \omega \leq 1$, showing that while there's no single ω value leading to significant error reduction for small k while maintaining convergence, optimization strategies can be employed. Specifically, selecting $\omega = 2/3$ ensures consistent performance across a range of values of k not too close to zero, with the resulting bound established as $|\alpha_k(I - \omega D^{-1}A)| \leq \frac{1}{3}$ for $\frac{n}{2} \leq k < n$. After thorough examination, it becomes evident that the weighted Jacobi method serves as a highly efficient stationary iteration technique for reducing errors within the upper spectrum of matrix A, regardless of the problem size. This effectiveness extends to methods like Gauss-Seidel

[2, 32]. Similar favorable outcomes are observed across various problem types, including two- and three-dimensional Poisson problems employing both finite-difference and finite-element discretizations. When initiating with a typical error pattern comprising numerous frequencies (eigenfunctions), the application of such iterative methods necessarily results in the dominance of "smooth" modes (those with small k) in the remaining error. After several iterations of the Jacobi method, the error becomes notably smoother. Consequently, these iterative techniques are often referred to as "smoothers" or "relaxation methods".

Multigrid methods operate on two key insights: high-frequency errors are effectively reduced by smoothing techniques, while low-frequency errors can be accurately approximated on a coarser grid. It's important to note that some low-frequency errors on a fine grid translate into high-frequency errors on a coarser grid. By applying smoothing and leveraging the scale differences recursively, the classical multigrid formulation is achieved. In one-dimensional grids, linear interpolation serves as a natural operator to transfer corrections between grids. This interpolation assigns values from coarse-grid nodes to corresponding points on the fine grid. For points on the fine grid located between coarse-grid nodes, a linear interpolation of coarse-grid values determines the fine-grid values. This process defines a linear operator, denoted as P, mapping coarse-grid vectors to fine-grid vectors. Considering a uniform fine grid with mesh points $x_0^h, x_1^h, \ldots, x_n^h$ (where n is even) and mesh spacing $h = \frac{1}{n}$, and a corresponding coarse grid with mesh points $x_0^{2h}, \ldots, x_{n/2}^{2h}$ with spacing $2h$, the action of P on a given coarse-level vector $\boldsymbol{v}_{2h}$ can be described away from the boundaries as follows:

$$
\begin{aligned}
(P\boldsymbol{v}^{2h})_{2i} &= \boldsymbol{v}_i^{2h}, \\
(P\boldsymbol{v}^{2h})_{2i+1} &= (\boldsymbol{v}_i^{2h} + \boldsymbol{v}_{i+1}^{2h})/2,
\end{aligned}
$$

where i is a coarse level index. This is depicted in Figure 2.2. Careful consideration is required for boundary conditions. For example, with Neumann or Robin boundary conditions on both endpoints and n being even, the described method is applicable to all fine grid points. However, with Dirichlet boundary conditions, degrees of freedom such as $\boldsymbol{v}_0$ on both grids are typically eliminated. In such cases, adjustments are made to ensure corrections assume a zero error at the boundary, often yielding

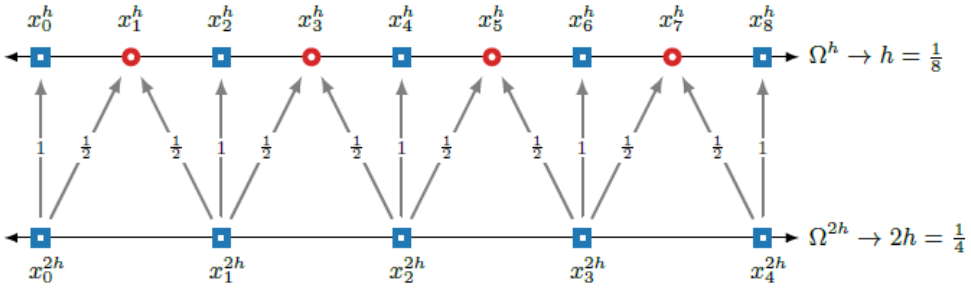

Figure 2.2: Interpolation pattern for one-dimensional grids with Neumann or Robin boundary conditions (from [2]).

$(P\boldsymbol{v}^{2h})_1 = \boldsymbol{v}_1^{2h}/2$. On tensor-product grids in two dimensions, tensor-product interpolation operators are commonly employed, particularly near boundaries such as pictured in Figure 2.3.

With an interpolation operator P mapping vectors from grid $2h$ to grid h, a specific correction to an approximation $\boldsymbol{u}_0^h$ is proposed for the solution of the grid h problem $\boldsymbol{A}\boldsymbol{u}^h = \boldsymbol{f}^h$, taking the form $P\boldsymbol{u}^{2h}$. Assuming $\boldsymbol{u}_0^h$ is derived from applying a few smoothing iteration steps to another vector, the corrected approximation is given by $\boldsymbol{u}_0^h + P\boldsymbol{u}^{2h}$, aiming for the best possible approximation of this form. As is typical, the use of the term "best" implies a metric that can be utilized to determine one corrected approximation as superior to another. In our context, we interpret this to be some matrix norm, $\|\boldsymbol{y}\|_M^2 = \boldsymbol{y}^T M \boldsymbol{y}$, where M is a symmetric and positive-definite matrix, resulting in the optimization problem in [2] as:

$$\min_{\boldsymbol{u}^{2h}} \|\boldsymbol{u}^h - \boldsymbol{u}_0^h + P\boldsymbol{u}^{2h}\|_M.$$

In the specific case where $M = 1$, the optimization focuses on finding the best approximation in the Euclidean l_2-norm. Conversely, when $M = A^T A$, the objective is to minimize the residual after correction. To find the minimum, the derivative is computed and set equal to zero, a process facilitated by first squaring the quantity to be minimized for convenience. Then,

$$\begin{aligned}\|\boldsymbol{u}^h - (\boldsymbol{u}_0^h + P\boldsymbol{u}^{2h})\|_M^2 &= (\boldsymbol{e}_0^h - P\boldsymbol{u}^{2h})^T M(\boldsymbol{e}_0^h - P\boldsymbol{u}^{2h}) \\ &= (\boldsymbol{e}_0^h)^T M \boldsymbol{e}_0^h - 2(\boldsymbol{u}^{2h})^T P^T M \boldsymbol{e}_0^h + (\boldsymbol{u}^{2h})^T P^T M P \boldsymbol{u}^{2h}.\end{aligned}$$

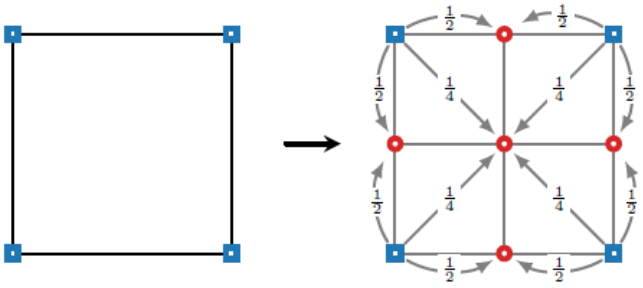

Figure 2.3: Interpolation pattern for two-dimensional tensor product grids with a coarse cell at left and its uniform refinement at right (from [2]).

Differentiating this with respect to $\boldsymbol{u}^{2h}$ and equating the resulting derivative to zero reveals that the optimal correction is defined by the solution, denoted as $\boldsymbol{u}^{2h}$, of

$$P^T M P \boldsymbol{u}^{2h} = P^T M \boldsymbol{e}_0^h = P^T M(\boldsymbol{u}^h - \boldsymbol{u}_0^h). \tag{2.28}$$

Although this method looks good mathematically, it presents a practical problem because the right-hand side might be hard to calculate since it depends on the unknown solution, $\boldsymbol{u}^h$. However, when A is symmetric and positive definite, choosing $M = A$ gives us a particularly nice coarse-grid problem. This choice results in the minimizer satisfying the equation:

$$P^T A P \boldsymbol{u}^{2h} = P^T(\boldsymbol{f} - A\boldsymbol{u}_0^h). \tag{2.29}$$

We term the update $P\boldsymbol{u}^{2h}$ as a Galerkin coarse-grid correction, while the coarse grid operator $A^c = P^T AP$ is denoted as a Galerkin coarse-grid operator. The two-grid iteration arises from combining the aforementioned components: a smoothing iteration and the coarse-grid correction process. Given A and $\boldsymbol{f}$ resulting from a discretization process on grid h, along with an initial guess $\boldsymbol{u}_0$ on grid h, a typical two-grid iteration is expressed as outlined in Algorithm 1. In this context, we define ν_1 pre-smoothing and ν_2 post-smoothing iterations specified by matrices M_1 and M_2; frequently, $M_1 = M_2$ or $M_1 = (M_2)^T$, although this choice depends on the specific problem. Typically, very small values of ν_1 and ν_2 are chosen, with optimal performance often observed for $\nu_1 + \nu_2 = 2$ or 3. However, the two-grid cycle proves inefficient as the grid $2h$ problem is not substantially smaller than the fine grid problem on grid h. Hence, we must explore methods to enhance the algorithm's efficiency without compromising its

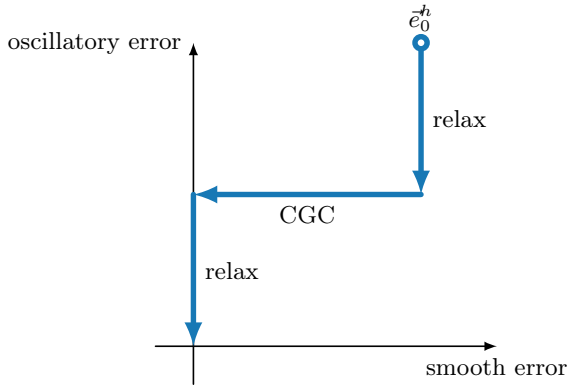

Figure 2.4: Ideal two-grid error-reduction when relaxation and coarse-grid correction are perfectly orthogonal (from [2]).

effectiveness.

Input : A, linear system matrix
$\boldsymbol{f}$, linear system right-hand side
$\boldsymbol{u}_0$, initial guess
R, restriction matrix
P, interpolation matrix
$M_{1,2}$, pre, post-relaxation matrix
$\nu_{1,2}$, number of pre, post-relaxation sweeps

Output: $\boldsymbol{u}$, approximation after one cycle

$\boldsymbol{u} \leftarrow \boldsymbol{u}_0$;
for $i = 1$ **to** ν_1 **do**
  $\boldsymbol{u} \leftarrow \boldsymbol{u} + M_1(\boldsymbol{f} - A\boldsymbol{u})$; `// Pre-relaxation`
end
$\boldsymbol{f}^{2h} \leftarrow R(\boldsymbol{f} - A\boldsymbol{u})$; `// Restrict residual of current approximation` $\boldsymbol{u}$
Solve $RAP\boldsymbol{u}^{2h} = \boldsymbol{f}^{2h}$ (or $A^{2h}\boldsymbol{u}^{2h} = \boldsymbol{f}^{2h}$) ; `// Solve coarse-level problem for` $\boldsymbol{u}^{2h}$
$\boldsymbol{u} \leftarrow \boldsymbol{u} + P\boldsymbol{u}^{2h}$; `// Coarse-grid correction`
for $i = 1$ **to** ν_2 **do**
  $\boldsymbol{u} \leftarrow \boldsymbol{u} + M_2(\boldsymbol{f} - A\boldsymbol{u})$; `// Post-relaxation`
end

Algorithm 1: Two-grid algorithm

The main computational challenge in Algorithm 1 arises from the solution of the linear system on grid $2h$. In the case of a one-dimensional problem on a uniform mesh,

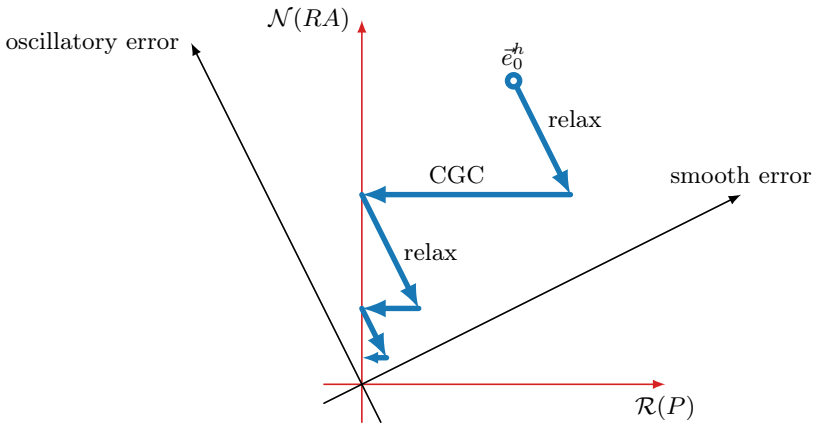

Figure 2.5: Typical two-grid error-reduction when relaxation and coarse-grid correction are not perfectly orthogonal (from [2]).

with a factor of two coarsening, A^{2h} is reduced to half the size of the original problem on grid h. Similarly, in two dimensions, it decreases to one-fourth the size, and in three dimensions, it decreases to one-eighth the size. To alleviate this computational cost, two options can be considered:

- Grid $2h$ is not revisited at every iteration.
- The grid $2h$ problem is not solved exactly.

While feasible in certain scenarios, implementing the first option generally proves challenging. This difficulty stems from the presence of a small discrepancy between the error spaces reduced by fine-scale relaxation and coarse-grid correction. If these error spaces were orthogonal, as sketched in Figure 2.4, handling error on the coarse grid once and complementing it with relaxation sweeps would be feasible. However, in most cases, these error spaces are not perfectly orthogonal, as illustrated in Figure 2.5. Consequently, subsequent relaxation reintroduces errors in spaces that can only be rectified by further coarse-grid correction. This lack of orthogonality necessitates revisiting the coarse-grid problem at each iteration step to eliminate errors within the interpolation range reintroduced by relaxation. Therefore, an efficient approach to approximately solve the grid $2h$ problem is required. The fundamental concept is to recognize the similarity between the grid $2h$ problem and the $2h$ version of the grid h problem, enabling a recursive approach. Specifically, the grid $2h$ version of Algorithm 1 is utilized to address the grid $2h$ problem. This recursion involves

addressing a grid $4h$ problem, which is tackled using the grid $4h$ version of the two-grid algorithm outlined in Algorithm 1. This recursive process continues until a grid size is reached that can be efficiently solved directly. Notationally, A is employed for the grid h, and A^c or A^{2h} for the grid $2h$ in defining the two-grid algorithm. In a multilevel algorithm, an index is employed to denote the level. For this purpose, A^ℓ is utilized, where $\ell = 0$ represents the fine-level grid-h matrix A^h, and ℓ_{max} designates the coarsest level. With this notation, A^ℓ represents the operator on level ℓ with a grid size of $2^\ell h$ (assuming a factor-of-two coarsening in the grids). Typical choices in Algorithm 2 are $\mu = 1$, referred to as the multigrid V-cycle, and $\mu = 2$, known as the W-cycle. The parameter μ specifies the number of times the coarse-grid correction step is applied at each level of the multigrid hierarchy. In the V-cycle ($\mu = 1$), the algorithm progresses down the grid hierarchy, performing smoothing operations and coarse-grid corrections, and then returns up the hierarchy, smoothing at each level again. In the W-cycle ($\mu = 2$), the algorithm makes two recursive visits to the coarser grids, allowing for additional corrections and potentially improved convergence rates, albeit with increased computational cost [2].

The Multigrid V-cycle is an iterative method for efficiently solving discretized partial differential equations. It combines smoothing operations with coarse-grid corrections to converge to the solution rapidly. Starting at the finest grid level, multiple relaxation sweeps smooth out high-frequency errors. The solution is then restricted to coarser grids, where smoothing is applied iteratively until reaching a tractable level. A direct solver or other relaxation scheme is used at this coarse level. After solving on the coarsest grid, the solution is interpolated back to finer grids, and correction updates refine the solution iteratively. This recursive smoothing and correcting on different grid levels resemble the shape of the letter "V," hence the name V-cycle.

Input: A, level-ℓ operator
$\boldsymbol{u}$, $\boldsymbol{f}$, initial approximation, level-ℓ right-hand side
R, P, level-ℓ restriction and interpolation matrices
$M_{1,2}, \nu_{1;2}$, level-ℓ pre, post-relaxation matrix and number of sweeps
ℓ, current grid level
Output: $\boldsymbol{u}$, approximation after one μ-cycle on level ℓ
for $i = 1$ *to* ν_1 **do**
  $\boldsymbol{u} \leftarrow \boldsymbol{u} + M_1(\boldsymbol{f} - A\boldsymbol{u})$; `// Pre-relaxation`
end
$\boldsymbol{f}^{\ell+1} \leftarrow R(\boldsymbol{f} - A\boldsymbol{u})$; `// Restrict level-`ℓ `residual`
if $\ell = \ell_{max} - 1$ **then**
  $\boldsymbol{u}^{\ell_{\max}} \leftarrow (A^{\ell_{\max}})^{-1}\boldsymbol{f}^{\ell_{\max}}$; `// Exact solve on coarsest grid`
end
else
  $\boldsymbol{u}^{\ell+1} \leftarrow 0$; `// Initialize coarse level correction as zero`
  for $i = 1$ *to* μ **do**
    $\boldsymbol{u}^{\ell+1} \leftarrow MU(A^{\ell+1}, \boldsymbol{u}^{\ell+1}, \boldsymbol{f}^{\ell+1}, \ldots, \ell+1)$; `// Recursive call to a` μ`-cycle`
  end
  $\boldsymbol{u} \leftarrow \boldsymbol{u} + P\boldsymbol{u}^{\ell+1}$; `// Coarse-grid correction`
end
for $i = 1$ *to* ν_2 **do**
  $\boldsymbol{u} \leftarrow \boldsymbol{u} + M_2(\boldsymbol{f} - A\boldsymbol{u})$; `// Post-relaxation`
end

Algorithm 2: The multigrid μ-cycle: $\boldsymbol{u} = MU(\cdot)$

2.2.2 Multigrid for systems

Considering the Stokes equations, we typically discretize Equation (2.17), leading to linear systems (2.18) described in [3, 11, 2] as follows:

$$\begin{pmatrix} F & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{pmatrix} \tag{2.30}$$

The matrix F represents discretization of the bilinear form $a(\boldsymbol{u}, \boldsymbol{v})$, or the discrete representation of the divergence of the strain rate tensor $\epsilon(\boldsymbol{u})$. The matrix B^T represents the discrete divergence operator, while its adjoint B corresponds to the discrete gradient operator. To tackle this saddle-point problem, we opt for a preconditioned Krylov subspace method. We explore two types of preconditioners: block-factorization methods and monolithic multigrid methods.

A common type of preconditioner relies on the block factorization of the system matrix,

$$\begin{pmatrix} F & B \\ B^T & 0 \end{pmatrix} = \begin{pmatrix} F & 0 \\ B^T & -B^T F^{-1} B \end{pmatrix} \begin{pmatrix} I & F^{-1}B \\ 0 & I \end{pmatrix}, \tag{2.31}$$

$$= \begin{pmatrix} I & 0 \\ B^T F^{-1} & I \end{pmatrix} \begin{pmatrix} F & 0 \\ 0 & -B^T F^{-1} B \end{pmatrix} \begin{pmatrix} I & F^{-1}B \\ 0 & I \end{pmatrix}. \tag{2.32}$$

From this, we consider two preconditioners, block diagonal,

$$\mathcal{M}_d = \begin{bmatrix} F & 0 \\ 0 & -B^T F^{-1} B \end{bmatrix} \tag{2.33}$$

and block triangular,

$$\mathcal{M}_t = \begin{bmatrix} F & 0 \\ B^T & -B^T F^{-1} B \end{bmatrix}. \tag{2.34}$$

One effective approach to enhance convergence rates in solving coupled systems like Equation (2.30) is to approximate the Schur complement by a mass matrix and construct a provably good preconditioner [12]. This involves utilizing block-factorization preconditioners derived from the block LU factorization of the system matrix. Two common preconditioners are the block-diagonal ($\mathcal{M}_d$) and block-triangular ($\mathcal{M}_t$) preconditioners. These exploit the sparsity and structure of the system matrix to efficiently approximate its inverse. To further improve preconditioning, multigrid methods can be integrated. By applying a multigrid cycle to approximate the inverse of the system matrix F, along with a simple preconditioner for the mass matrix approximation of the Schur complement, the overall preconditioner's performance can be enhanced. Multigrid methods provide an effective means to approximate the inverse of F.

An alternative to block-factorization preconditioners is to employ monolithic multigrid methods for the coupled system in Equation (2.30). In contrast to block preconditioners, where multigrid may be applied only to the subsystem involving F or smaller sub-blocks of F, monolithic approaches maintain coupling between $\boldsymbol{u}$ and $\boldsymbol{p}$ at all hierarchy levels. A common technique for monolithic geometric multigrid is to refine the underlying mesh similarly to an uncoupled problem, resulting in a grid $2h$ problem with the same degrees of freedom structure as the original grid h problem. Interpolation from grid $2h$ to grid h involves independently interpolating each system component, resulting in a composite interpolation operator of the form in [11] as,

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{P}_u & 0 \\ 0 & \boldsymbol{P}_P \end{bmatrix}$$

where $\boldsymbol{P}_u$ denotes the velocity interpolation operator, and $\boldsymbol{P}_p$ represents the pressure interpolation operator. While the Galerkin and rediscretization coarse-grid operators align when utilizing the canonical finite-element operators for all fields, in accordance with the geometric multigrid structure, we opt for the rediscretization operators over Galerkin. This choice is primarily made for the seamless extension from efficient two-level solvers to the multilevel scenario.

It is widely acknowledged that conventional relaxation methods, like Jacobi or Gauss-Seidel, are often ineffective for many saddle-point problems. Instead, researchers have proposed and investigated several families of relaxation methods specifically tailored to this context. In this discussion, we concentrate on four categories of such techniques: Braess-Sarazin, Uzawa, Vanka, and distributed relaxation. See [2] for more details.

First we discuss Braess-Sarazin relaxation schemes. Braess-Sarazin-type algorithms were originally proposed as relaxation schemes for the Stokes' equations, employing an approximate block factorization as an approximation to the original system. Braess-Sarazin approaches make use of the block factorization described in Equation (2.31), understanding that affordable approximations to solving with F and $B^T F^{-1} B$ can lead to effective relaxation methods. In a typical Braess-Sarazin relaxation, we employ the stationary iteration

$$\begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} + \begin{pmatrix} \omega C & B \\ B^T & 0 \end{pmatrix}^{-1} \left(\begin{pmatrix} \boldsymbol{f} \\ 0 \end{pmatrix} - \begin{pmatrix} F & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} \right),$$

as a relaxation scheme. Instead of using F, we employ the approximation C, where C is often chosen as a diagonal matrix, such as the identity matrix or the diagonal of F, and ω is a suitably selected relaxation parameter. It is important to note that, in this scenario, we need to both assemble and solve a linear system involving the approximate Schur complement, $-\omega^{-1}B^TC^{-1}B$. While assembling this approximate Schur complement is efficient when C is diagonal, its inversion remains computationally intensive and can be suitably approximated [28]. Although the proper selection of relaxation parameter(s) is crucial for achieving optimal multigrid performance, Braess-Sarazin methods have demonstrated successful application across various contexts. Uzawa methods only approximate the inverse of the block L factor. This leads to a stationary iteration of the form

$$\begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} + \begin{pmatrix} \omega C & 0 \\ B^T & -\widehat{S} \end{pmatrix}^{-1} \left(\begin{pmatrix} \boldsymbol{f} \\ 0 \end{pmatrix} - \begin{pmatrix} F & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} \right).$$

In this context, C is a simple version of F that's easy to invert. Now, $\widehat{S}$ is another simple-to-invert approximation, either to $B^TF^{-1}B$ or $B^TC^{-1}B$. Uzawa iterations are less computationally demanding than Braess-Sarazin iterations, but they're usually not as effective. Whether Uzawa relaxation is a good choice depends on the specific problem, its discretization, and how well the relaxation parameters are set. However, it's often found that inexact Braess-Sarazin methods, since they are a bit more flexible, provide better performance relative to their cost compared to Uzawa methods [2].

Distributed relaxation methods are founded upon approximating the continuum relationship $\Delta\nabla = \nabla\Delta$ at the discrete level, distinguishing between the vector Laplacian on the left and the scalar Laplacian on the right. This discrete representation is often symbolized as $FB \approx BF_p$, where F_p represents the discretization of the Laplacian operator on the pressure space (pertaining to Stokes equations). If this approximation remains valid, it offers an alternative block LU factorization (distinct from the unit U factor) compared to the one depicted in Equation (2.31),

$$\begin{pmatrix} F & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} I & B \\ 0 & -F_p \end{pmatrix} = \begin{pmatrix} F & 0 \\ B^T & B^TB \end{pmatrix}.$$

Distributive relaxation approaches thus consist of a stationary iteration given by

$$\begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} + \begin{pmatrix} I & B \\ & -F_p \end{pmatrix} \times \begin{pmatrix} \omega C & 0 \\ B^T & \widehat{B^T B} \end{pmatrix}^{-1} \left(\begin{pmatrix} \boldsymbol{f} \\ 0 \end{pmatrix} - \begin{pmatrix} F & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} \right).$$

In the context of finite-difference discretizations of Stokes equations, where the discrete relationship $FB \approx BF_p$ holds away from boundaries, distributed relaxation was initially proposed. However, it has not seen significant success in finite-element discretizations, except within the domain of Maxwell's equations. The effectiveness of distributed relaxation in the context of Maxwell's equations is attributed to the use of discrete exact sequences [2].

Vanka relaxation methods, in contrast to the aforementioned approaches, are grounded in domain decomposition concepts rather than block factorizations. In this scheme, the decomposition must align with the structure of the discretization. An algebraic Vanka approach can be naturally established by describing subdomains, denoted in [2] as Ω_i, where

$$\Omega_i = \{ j \,|\, b_{j,i} \neq 0 \} :$$

This signifies that the i^{th} subdomain consists of the sets of degrees of freedom associated with nonzero entries in the i^{th} row of B^T. Notably, this naturally introduces overlap in standard discretizations of the Stokes equations, where a single velocity degree of freedom is linked to multiple pressure degrees of freedom. Alternatively, a geometric Vanka methodology can be adopted, wherein the subdomains are constructed based on mesh connectivity. For the Taylor-Hood discretization of the Stokes equations, this results in subdomains comprising a single (nodal) pressure degree of freedom and all velocity degrees of freedom on elements adjacent to the node. Depending on how coincidental zeros in the discrete gradient operator are handled, this definition may align with the algebraic definition mentioned earlier. In recent years, Vanka relaxation techniques have found successful application in various scenarios, making them widely employed in numerous applications [2].

2.3 Time integration

In the context of numerical methods for solving partial differential equations (PDEs), ensuring the accuracy and stability of the solution is paramount. Previously, we discussed multigrid methods, which are effective for efficiently solving large-scale linear systems arising from the spatial discretization of PDEs. However, to fully resolve the temporal dynamics of the system, we must also focus on the integration of the resulting ordinary differential equations (ODEs) that arise from this discretization.

Time integration methods are crucial for advancing the solution in time while maintaining the desired properties imposed by the spatial discretization. Effective time integration techniques are necessary to handle the stiff and non-stiff ODEs resulting from the semi-discretization of PDEs. These methods ensure that the numerical solution remains stable and accurate over time, allowing for the realistic simulation of dynamic systems. By combining multigrid methods for spatial discretization with robust time integration schemes, we can achieve a comprehensive and reliable numerical approach for solving PDEs. This synergy allows for efficient and stable simulations, making it possible to accurately capture both the spatial and temporal aspects of complex physical phenomena.

We first consider a system of ordinary differential equations (ODEs) that governs the evolution of a dynamical system. These ODEs describe the rate of change of the system's state variables with respect to time. We are primarily interested in the case where the ODEs arise through semi-discretization, discretizing the spatial domain of a PDE while leaving the temporal domain continuous, resulting in a set of coupled ordinary differential equations in time, which can be solved numerically using time-stepping methods such as the explicit or implicit Euler methods, Runge-Kutta methods, or other advanced techniques. The semi-discretization allows for efficient numerical simulations while capturing the essential dynamics of the continuous system. Let's consider a system of ODEs represented as:

$$\boldsymbol{y}'(t) = \frac{d\boldsymbol{y}(t)}{dt} = \boldsymbol{f}(t, \boldsymbol{y}(t)), \tag{2.35}$$

where $\boldsymbol{y}$ is the state vector and $\boldsymbol{f}$ is a vector function describing the rate of change of $\boldsymbol{y}$ with respect to time t. We seek a numerical method to approximate the solution

to

$$\boldsymbol{y}'(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)) \quad a < t < b, \tag{2.36a}$$

$$\boldsymbol{y}(a) = \boldsymbol{y}_0. \tag{2.36b}$$

Consider trying to approximate $\boldsymbol{y}(t)$ at points $t_i = a + ih_t$ for $0 \leq i \leq N$, where $h_t = \dfrac{b-a}{N}$ is the step size. Denote the approximate solution for $\boldsymbol{y}(t_i)$ by $\boldsymbol{y}_i$. If we knew $\boldsymbol{y}(t_i)$ at these points, we could approximate $\boldsymbol{y}'(t_i) = \dfrac{\boldsymbol{y}(t_{i+1}) - \boldsymbol{y}(t_i)}{h_t} - \dfrac{h_t}{2}\boldsymbol{y}''(\xi_i)$. From the ODE, we know that $\boldsymbol{y}'(t_i) = \boldsymbol{f}(t_i, \boldsymbol{y}(t_i))$, so $\boldsymbol{y}(t_{i+1}) = \boldsymbol{y}(t_i) + h_t\boldsymbol{y}'(t_i) + \dfrac{h_t^2}{2}\boldsymbol{y}''(\xi_i)$. Dropping truncation error terms and approximating $\boldsymbol{y}_i \approx \boldsymbol{y}(t_i)$ we obtain the forward Euler method (explicit Euler formula) $\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h_t\boldsymbol{f}(t_i, \boldsymbol{y}_i)$.

We could also use backward differences, $\boldsymbol{y}'(t_{i+1}) = \dfrac{\boldsymbol{y}(t_{i+1}) - \boldsymbol{y}(t_i)}{h_t} - \dfrac{h_t}{2}\boldsymbol{y}''(\zeta_i)$ with $\boldsymbol{y}'(t_{i+1}) = \boldsymbol{f}(t_{i+1}, \boldsymbol{y}(t_{i+1}))$ to get $\boldsymbol{y}(t_{i+1}) = h_t\boldsymbol{y}'(t_{i+1}) + \boldsymbol{y}(t_i) - \dfrac{h_t^2}{2}\boldsymbol{y}''(\zeta_i)$. Again dropping truncation error terms and approximating $\boldsymbol{y}_i \approx \boldsymbol{y}(t_i)$, we obtain the backward Euler method (implicit Euler formula) $\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h_t\boldsymbol{f}(t_{i+1}, \boldsymbol{y}_{i+1})$.

Definition 14 ***Local truncation error.*** *The local truncation error, $\boldsymbol{d}_i$, is the residual of the approximation to $\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y})$ from a numerical method when $\boldsymbol{y}_i$ and $\boldsymbol{y}_{i+1}$ are replaced by $\boldsymbol{y}(t_i)$ and $\boldsymbol{y}(t_{i+1})$ for a sufficiently smooth solution, $\boldsymbol{y}(t)$, of the boundary value problem.*

Definition 15 ***Order of accuracy.*** *The order of accuracy, q, is the largest positive integer such that $\max_i |\boldsymbol{d}_i| = O(h_t^q)$ as $h_t \to 0$ for a sufficiently smooth solution, $\boldsymbol{y}(t)$, to the boundary value problem.*

For Explicit Euler: $\boldsymbol{d}_i = \dfrac{h}{2}\boldsymbol{y}''(\xi)$ for some $\xi \in (t_i, t_{i+1})$
For Implicit Euler: $\boldsymbol{d}_i = -\dfrac{h}{2}\boldsymbol{y}''(\eta)$ for some $\eta \in (t_i, t_{i+1})$
These estimates are valid so long as $\boldsymbol{y}''(t)$ is a continuous function, $\boldsymbol{y} \in C^2([a, b])$. Both implicit and explicit Euler have order of accuracy equal to 1, with $\max_i |\boldsymbol{d}_i| \leq \dfrac{Mh}{2}$ where $|\boldsymbol{y}''(t)| \leq M, a \leq t \leq b$.

Definition 16 ***Global error.*** *The global error of the method is $\boldsymbol{e}_i = \boldsymbol{y}(t_i) - \boldsymbol{y}_i$, for $i = 0, 1, 2, \ldots, N$*

Theorem 6 *Let $\boldsymbol{f}(t, \boldsymbol{y})$ be Lipschitz in $\boldsymbol{y}$ and let the solution to the BVP (boundary value problem) satisfy $\boldsymbol{y} \in C^2([a, b])$ with $|\boldsymbol{y}''(t)| \leq M$ for $a \leq t \leq b$. Let $\boldsymbol{y}_i$ be the explicit Euler approximation to $\boldsymbol{y}(t_i)$, then $|\boldsymbol{e}_i| = \frac{Mh}{2L}(e^{L(t_i-a)}-1)$ for $i = 0, 1, 2, \ldots, N$.*

The general approach for computing global error bounds is similar for both implicit and explicit Euler methods when considering a fixed time window $a \leq t \leq b$ and letting $h_t \to 0$. As h approaches zero, both methods converge to the solution of the boundary value problem (BVP). In this scenario, the global error for each method, denoted by $|\boldsymbol{e}_i|$, is typically of the order $O(h_t)$.

Definition 17 ***Convergence*** *[21]. The method is said to converge if the maximum global error tends to 0 as h tends to 0, provided the exact solution exists and is reasonably smooth.*

Now consider the test equation:

$$\boldsymbol{y}' = \lambda \boldsymbol{y} \quad t > 0, \tag{2.37a}$$

$$\boldsymbol{y}(0) = \boldsymbol{y}_0. \tag{2.37b}$$

We know the solution of (2.37) is $\boldsymbol{y}(t) = \boldsymbol{y}_0 e^{\lambda t}$, and that if $\lambda \leq 0$, $\boldsymbol{y}(t_{i+1}) \leq \boldsymbol{y}(t_i)$ for $0 \leq i \leq N-1$.

For this problem, the explicit Euler discretization becomes

$$\begin{aligned} \boldsymbol{y}_{i+1} &= \boldsymbol{y}_i + h_t \boldsymbol{f}(t_i, \boldsymbol{y}_i) \\ &= (1 + h_t\lambda)\boldsymbol{y}_i \\ &= (1 + h_t\lambda)^2 \boldsymbol{y}_{i-1} = (1 + h_t\lambda)^{i+1} \boldsymbol{y}_0 \end{aligned}$$

When $\lambda \leq 0$, $|1 + \lambda h_t| \leq 1$ if and only if $h_t \leq \dfrac{2}{|\lambda|}$.

The implicit Euler discretization for this problem becomes

$$\begin{aligned}
\boldsymbol{y}_{i+1} &= \boldsymbol{y}_i + h_t \boldsymbol{f}(t_{i+1}, \boldsymbol{y}_{i+1}) \\
\boldsymbol{y}_i &= (1 - h_t\lambda)\boldsymbol{y}_{i+1} \\
\boldsymbol{y}_{i+1} &= \left(\frac{1}{1 - h_t\lambda}\right) \boldsymbol{y}_i \\
&= \left(\frac{1}{1 - h_t\lambda}\right)^2 \boldsymbol{y}_{i-1} = \left(\frac{1}{1 - h_t\lambda}\right)^{i+1} \boldsymbol{y}_0
\end{aligned}$$

Note that $\left|\frac{1}{1 - h_t\lambda}\right| \leq 1$ for any $h_t \geq 0$ and all $\lambda \leq 0$. We refer to a method as conditionally stable if solutions to Equation (2.37) do not grow for $\lambda < 0$ only under some condition on h_t. Consequently, the explicit Euler method is conditionally stable, whereas the implicit Euler method is unconditionally stable. This is the main advantage provided by implicit schemes: they are more expensive per iteration than explicit schemes but are often unconditionally stable. When using implicit schemes, the primary consideration is determining the value of h_t based on accuracy constraints. In contrast, for explicit schemes, both accuracy and stability conditions need to be satisfied. We can improve upon first-order accuracy using the following approaches:

1. Introduce an approximation at the midpoint of $[t_i, t_{i+1}]$.

   (a) **Explicit midpoint method:** Use explicit Euler to predict $\boldsymbol{y}_{i+1/2} = \boldsymbol{y}_i + \frac{h_t}{2}\boldsymbol{f}(t_i, \boldsymbol{y}_i)$, then compute

   $$\begin{aligned}
   \boldsymbol{y}_{i+1} &= \boldsymbol{y}_i + h_t \boldsymbol{f}\left(t_{i+1/2}, \boldsymbol{y}_{i+1/2}\right) \\
   &= \boldsymbol{y}_i + h_t \boldsymbol{f}\left(t_{i+1/2}, \boldsymbol{y}_i + \frac{h_t}{2}\boldsymbol{f}(t_i, \boldsymbol{y}_i)\right) \quad \text{for} \quad t_{i+1/2} = \frac{(t_i + t_{i+1})}{2}.
   \end{aligned}$$

   (b) **Implicit midpoint method:** Approximate $\boldsymbol{y}_{i+1/2} = \frac{(\boldsymbol{y}_i + \boldsymbol{y}_{i+1})}{2}$, then solve $\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h_t\boldsymbol{f}(t_{i+1/2}, \boldsymbol{y}_{i+1/2})$. So the implicit midpoint method is given by $\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h_t\boldsymbol{f}\left(t_{i+1/2}, \frac{(\boldsymbol{y}_i + \boldsymbol{y}_{i+1})}{2}\right)$.

2. Trapezoidal rule

   (a) Use explicit Euler to predict $\boldsymbol{Y} = \boldsymbol{y}_i + h_t\boldsymbol{f}(t_i, \boldsymbol{y}_i)$, then compute

$\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \dfrac{h_t}{2}\left(\boldsymbol{f}(t_i, \boldsymbol{y}_i) + \boldsymbol{f}(t_{i+1}, \boldsymbol{Y})\right)$. The explicit trapezoidal rule is $\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \dfrac{h_t}{2}\left(\boldsymbol{f}(t_i, \boldsymbol{y}_i) + \boldsymbol{f}(t_{i+1}, \boldsymbol{y}_i + h_t\boldsymbol{f}(t_i, \boldsymbol{y}_i))\right)$.

(b) Explicit Euler linearizes at t_i with slope $\boldsymbol{y}'(t_i)$. Instead, take the average of $\boldsymbol{y}'(t_i)$ and $\boldsymbol{y}'(t_{i+1})$ and approximate $\boldsymbol{y}'(t_i) \approx \frac{1}{2}[\boldsymbol{f}(t_i, \boldsymbol{y}(t_i)) + \boldsymbol{f}(t_{i+1}, \boldsymbol{y}(t_{i+1}))]$. This gives the implicit trapezoidal rule:

$$\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{h_t}{2}[\boldsymbol{f}(t_i, \boldsymbol{y}_i) + \boldsymbol{f}(t_{i+1}, \boldsymbol{y}_{i+1})]$$

These four schemes are all instances of Runge-Kutta methods, which necessitate the evaluation of $\boldsymbol{f}(t, \boldsymbol{y})$ at points beyond $(t_i, \boldsymbol{y}_i)$. The concept of "stages" introduces additional evaluation points in $(t, \boldsymbol{y})$ space to compute $\boldsymbol{f}(t, \boldsymbol{y})$. Specifically, the implicit midpoint method constitutes a single-stage scheme, whereas the remaining three methods involve two-stage computations. We consider s-stage implicit RK methods in standard form,

$$\begin{aligned} \boldsymbol{y}_{i+1} &= \boldsymbol{y}_i + h_t \sum_{j=1}^{s} b_j \boldsymbol{f}(t_i + c_j h_t, \boldsymbol{Y}_j), \\ \text{with} \quad \boldsymbol{Y}_j &= \boldsymbol{y}_i + h_t \sum_{k=1}^{s} a_{jk} \boldsymbol{f}(t_i + c_k h_t, \boldsymbol{Y}_k), \quad j = 1, 2, \ldots, s. \end{aligned} \tag{2.38}$$

We need $\sum_{k=1}^{s} b_k = 1$ in order for the method to possibly be convergent. Typically, we take $\sum_{k=1}^{s} a_{jk} = c_j$, but this is not required. The coefficients of these methods are often represented in a Butcher tableau,

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}.$$

When $a_{jk} = 0$ for all $k \geq j$, the Runge-Kutta method is explicit, allowing each stage value, $\boldsymbol{Y}_j$, to be computed directly from previous stage values, requiring only evaluations of the function $\boldsymbol{f}(t, \boldsymbol{y})$. In contrast, when some $a_{jk} \neq 0$ for $k \geq j$, the method becomes implicit, necessitating the solution of a coupled system to determine the stage values. While RK methods are versatile and widely used, they can be computationally

expensive. Specifically, an s-stage explicit function demands s function evaluations. To achieve an error of $O(h_t^p)$ with an explicit RK method, it is imperative that $s \geq p$. For higher orders ($p \geq 5$), $s \geq p + 1$ is required. Implicit methods involve solving a coupled system of s equations for all stages simultaneously [21].

When $\boldsymbol{y} \in \mathbb{R}^m$, as is often the case in spatially discretized partial differential equations, Equation (2.38) represents a coupled nonlinear system of size $sm \times sm$. Solving this nonlinear system can be prohibitively costly, particularly when m is large. This cost can be mitigated with smaller h_t values or by using effective preconditioners for some linearization of the system. Consequently, diagonally implicit RK (DIRK) methods, where $a_{jk} = 0$ if $k > j$, are often employed to reduce computational expenses.

Concerning stability for IRK methods, define the $s \times s$ matrices $A = [a_{jk}]$ and I as the identity matrix, and column vectors of s entries $\mathbf{b} = [b_1, \ldots, b_s]^T$ and $\mathbf{e} = [1, 1, \ldots, 1]^T$ where the superscript T stands for transpose. Then, for any IRK method, its stability function ϕ can be calculated as:

$$\phi(z) = 1 + zb^T(I - zA)^{-1}\mathbf{e}$$

Definition 18 ***A-Stable*** *[21]. Runge–Kutta methods applied to the test equation $\boldsymbol{y}' = \lambda \boldsymbol{y}$ take the form $\boldsymbol{y}_{i+1} = \phi(h_t\lambda)\boldsymbol{y}_i$, and, by induction, $\boldsymbol{y}_i = (\phi(h_t\lambda))^i \cdot \boldsymbol{y}_0$. The function ϕ is called the stability function. Thus, the condition that $\boldsymbol{y}_i \to 0$ as $i \to \infty$ is equivalent to $|\phi(h_t\lambda)| < 1$. This motivates the definition of the region of absolute stability (sometimes referred to simply as stability region), which is the set $\{z \in \mathbb{C} : |\phi(z)| < 1\}$. The method is A-stable if the region of absolute stability contains the set $\{z \in \mathbb{C} : \mathrm{Re}(z) \leq 0\}$ that is, the left half plane.*

Practically, A-stability means that the numerical method remains stable for all sizes of time steps when applied to stiff problems. Stiffness is characterized by equations where certain components of the solution decay much faster than others, and in such cases, using a method that is not A-stable can lead to numerical solutions that grow without bound, even when the true solution does not. A-stable methods ensure that the numerical solution accurately follows the behavior of the true solution without requiring excessively small time steps. This property is crucial in simulations where stability and efficiency are both desired, such as in the modeling of poroelastic materials, where rapid fluid flow and slow solid deformation must be accurately captured.

L-stability is an even stronger form of stability that extends the concept of A-stability. L-stable methods not only remain stable for all sizes of time steps in stiff problems but also ensure that the numerical approximation to rapidly decaying modes diminishes as quickly as possible, preventing any lingering numerical artifacts. This is particularly beneficial in poroelastic simulations where the rapid dissipation of pressure or stress waves is essential.

Definition 19 *[21]. A method is L-stable if it is A-stable and* $\phi(z) \to 0$ *as* $z \to \infty$*, where* ϕ *is the stability function of the method. L-stable methods are, in general, very good at integrating stiff equations.*

Seeking higher-order accuracy to improve efficiency does not guarantee convergence. In fact, it can sometimes hinder convergence by compromising stability. This holds true even for implicit methods, meaning they do not always possess stability properties superior to those of explicit methods. For $s > 6$, the backward difference methods, BDF(s), are implicit multistep methods with arbitrarily high formal accuracy, yet they are not even stable. Multistep strategies are sometimes attractive because they have lower cost than high-order RK methods. For many problems in PDEs, however, employing an A-stable time stepper is strongly favored. Linear multistep methods with order more than 2 generally lack A-stability. Conversely, many implicit RK methods exhibit such stability. There are many families of implicit Runge-Kutta methods. Radau methods are fully implicit methods, that achieve an order of $2s - 1$ for s stages, and are A-stable. The first-order Radau method bears similarity to the backward Euler method [21]. Consider the Butcher tableaux of two stage RadauIIA and three stage RadauIIA,
Two stage RadauIIA:

$$\begin{array}{c|cc} 1/3 & 5/12 & -1/12 \\ 1 & 3/4 & 1/4 \\ \hline & 3/4 & 1/4 \end{array}$$

Three stage RadauIIA:

$$\begin{array}{c|ccc} \frac{2}{5} - \frac{\sqrt{6}}{10} & \frac{11}{45} - \frac{7\sqrt{6}}{360} & \frac{37}{225} - \frac{169\sqrt{6}}{1800} & -\frac{2}{225} + \frac{\sqrt{6}}{75} \\ \frac{2}{5} + \frac{\sqrt{6}}{10} & \frac{37}{225} + \frac{169\sqrt{6}}{1800} & \frac{11}{45} + \frac{7\sqrt{6}}{360} & -\frac{2}{225} - \frac{\sqrt{6}}{75} \\ 1 & \frac{4}{9} - \frac{\sqrt{6}}{36} & \frac{4}{9} + \frac{\sqrt{6}}{36} & \frac{1}{9} \\ \hline & \frac{4}{9} - \frac{\sqrt{6}}{36} & \frac{4}{9} + \frac{\sqrt{6}}{36} & \frac{1}{9} \end{array}$$

There are also three main families of Lobatto methods, denoted as IIIA, IIIB, and IIIC. All of them are implicit RK schemes with an order of $2s - 2$, where s is the number of stages. Notably, they all have $c_1 = 0$ and $c_s = 1$. Unlike explicit methods, it's possible for these methods to achieve an order greater than the number of stages.

The second-order Lobatto IIIC method is given by [21]:

$$\begin{array}{c|cc} 0 & 1/2 & -1/2 \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array}$$

The fourth-order Lobatto IIIC method is given by [21]:

$$\begin{array}{c|ccc} 0 & 1/6 & -1/3 & 1/6 \\ 1/2 & 1/6 & 5/12 & -1/12 \\ 1 & 1/6 & 2/3 & 1/6 \\ \hline & 1/6 & 2/3 & 1/6 \end{array}$$

They are L-stable. In this work we will use the RadauIIA schemes.

Now consider the system of differentiation algebraic equations:

$$\boldsymbol{y}' = \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{z}), \quad \boldsymbol{y}(t_0) = \boldsymbol{y}_0, \tag{2.39a}$$

$$\boldsymbol{g}(\boldsymbol{y}, \boldsymbol{z}) = 0, \quad \boldsymbol{z}(t_0) = \boldsymbol{z}_0. \tag{2.39b}$$

Suppose the initial values $\boldsymbol{y}_0$ and $\boldsymbol{z}_0$ adhere to the condition of consistency, indicated by their satisfaction of algebraic equation (2.39b). The differential index of system (2.39) signifies the minimal number of analytical differentiations necessary to convert this system into an explicit set of ordinary differential equations (ODEs) through algebraic manipulations. In systems with higher indices, such as two or above, the matrix $\dfrac{\partial \boldsymbol{g}}{\partial \boldsymbol{z}}$ becomes singular. As a consequence, the algebraic subsystem (2.39b) cannot be directly solved for the vector $\boldsymbol{z}$. Therefore, a simultaneous solution of algebraic and differential equations is facilitated through the application of an implicit method.

Implicit stiffly accurate methods, where $a_{si} = b_i$ for $i = 1, 2, \ldots, s$, are highly advantageous for solving stiff and differential algebraic equations. While diagonally implicit RK schemes can be applied to DAEs, they are generally found to produce

low-order accuracy. Despite classical theory suggesting that the global error of the numerical solution of an ODE behaves as $O(h_t^p)$ for sufficiently small step size h_t (where p is the order of the method), order reduction phenomena are observed, particularly for DIRK schemes when solving DAEs, especially for systems of higher indices.

We distinguish between the order p_d of the differential components and the order p_a of the algebraic ones. Suppose that system (2.39) is integrated on a given interval with a constant step size. Then, the above orders imply that the global errors of the corresponding components admit the estimates:

$$\begin{aligned} \boldsymbol{y}(t_n) - \boldsymbol{y}_n &= O(h_t^{p_d}) \\ \boldsymbol{z}(t_n) - \boldsymbol{z}_n &= O(h_t^{p_a}), \end{aligned}$$

where $\boldsymbol{y}(t_n), \boldsymbol{z}(t_n)$ is the exact solution at the right endpoint, while $\boldsymbol{y}_n, \boldsymbol{z}_n$ is the corresponding numerical solution.

While higher-order global error is appealing, especially for both stiff Differential Equations (DEs) and systems of Differential-Algebraic Equations, the so-called stage order of a Runge-Kutta method holds greater significance. Here, in addition to considering truncation error, the accuracy of a scheme is determined by bounding the approximation of stage j to $\boldsymbol{y}(t_i + c_j h_t)$ by some constant (dependent on $\boldsymbol{f}(t, \boldsymbol{y})$ and $\boldsymbol{y}(t)$) times h_t^{q+1}, thereby defining the stage order as $\min(q, p)$. For index-2 DAEs, the accuracy of a scheme is constrained by its stage order due to perturbation bounds on the solution of the constrained system. This limitation significantly narrows down the selection of schemes that allow higher-order accuracy. Although Diagonally Implicit Runge-Kutta (DIRK) methods can achieve reasonable global order, their stage order is typically restricted to 1. In contrast, the stage order of fully Implicit Runge-Kutta (IRK) schemes can be as large as the number of stages, making them the preferred schemes for integrating DAEs. Higher-order optimal Runge–Kutta methods often exhibit a significant disparity between their classical and stage order, potentially leading to a reduction in the actual order achieved. Radau IIA methods, characterized by their classical order equal to $2s - 1$, and stage order equal to s, stand out as optimal among stiffly accurate and L-stable methods [21, 33].

In this thesis, we propose a monolithic multigrid framework for solving the linear systems of equations resulting from employing higher-order Implicit Runge-Kutta (IRK) discretizations for poroelasticity.

Chapter 3

Mathematical Methods

In this section, we present the main contributions of this thesis. In Section 3.1, we will present Biot's three field formulation and its spatial discretization. Following that, Section 3.2 will focus on the spatiotemporal IRK discretization. Finally, we will discuss the monolithic multigrid framework in Section 3.3. Addressing the multifaceted challenges inherent in poroelasticity modeling involves several key considerations. Firstly, accurately capturing the intricate interplay between solid and fluid phases within heterogeneous porous media is paramount. This necessitates models capable of representing complex interactions and fluid-solid coupling phenomena effectively. Moreover, poroelastic materials exhibit nonlinear behavior under diverse loading conditions, requiring sophisticated numerical techniques to accurately simulate their response. Efficiently managing higher-order discretizations is another crucial aspect, ensuring that computational resources are utilized optimally while maintaining accuracy. Additionally, developing robust numerical methods capable of handling large-scale simulations with evolving geometries and boundary conditions is essential for practical applications, enabling reliable predictions of complex real-world phenomena. Addressing these challenges collectively contributes to advancing the state-of-the-art in poroelasticity modeling and simulation.

3.1 Biot's Three-Field Formulation and its Discretization

Poroelasticity theory, initially proposed by Maurice A. Biot in 1941, describes porous media as composed of two interdependent phases. This framework allows simultaneous modeling of medium deformation and fluid flow, integrating them through continuity and momentum conservation equations for each phase. Within this model, quantities such as stresses are considered "partial", indicating their dependence on phase fraction. In our work, we delve into Biot's linear poroelastic model [3], a coupled multiphysics system of partial differential equations (PDEs). The formulation involves three primary variables: displacement, pore pressure, and Darcy velocity (fluid flow). It accounts for the interaction between these variables and their impact on the overall mechanical response of the material. Discretizing Biot's Three-Field Formulation involves approximating these variables using finite element methods, where the domain is discretized into smaller elements, and numerical techniques are applied to solve the resulting system of equations. The discretization aims to accurately represent the physical phenomena while ensuring computational efficiency and stability.

3.1.1 Biot's Three-Field Formulation

In our work, we chose appropriate Dirichlet boundary conditions for the mathematical model of the three-field formulation of the consolidation process which is given in Section 2.1.4, as follows:

$$\boldsymbol{u}(0) = \boldsymbol{u}_0, \quad \text{for } x \in \bar{\Gamma}_c,$$

where Γ_c is an open (with respect to Γ) subset of Γ with nonzero measure. These boundary conditions are crucial for ensuring the accuracy and stability of numerical simulations in the context of the three-field formulation of the consolidation process. Additionally, specifying appropriate initial conditions for $\boldsymbol{u}$ is essential for maintaining mass conservation, ensuring numerical stability and convergence, and enhancing the physical realism of poroelasticity simulations. Now we consider the weak formulation of Biot's three field consolidation model in [31]:

For each $t \in (0, T]$, find $(\boldsymbol{u}(t), \boldsymbol{w}(t), p(t)) \in \boldsymbol{\mathcal{V}} \times \boldsymbol{\mathcal{W}} \times Q$ such that

$$a(\boldsymbol{u}, \boldsymbol{v}) - \langle \alpha p, \mathrm{div}\boldsymbol{v} \rangle = \langle \rho \boldsymbol{g}, \boldsymbol{v} \rangle, \quad \forall \boldsymbol{v} \in \boldsymbol{\mathcal{V}} \tag{3.1}$$

$$\langle \boldsymbol{K}^{-1} \mu_f \boldsymbol{w}, \boldsymbol{r} \rangle - \langle p, \mathrm{div}\boldsymbol{r} \rangle = \langle \rho_f \boldsymbol{g}, \boldsymbol{r} \rangle, \quad \forall \boldsymbol{r} \in \boldsymbol{\mathcal{W}} \tag{3.2}$$

$$\langle \frac{1}{M} \frac{\partial p}{\partial t}, q \rangle + \langle \alpha \mathrm{div} \frac{\partial \boldsymbol{u}}{\partial t}, q \rangle + \langle \mathrm{div}\boldsymbol{w}, q \rangle = \langle f, q \rangle, \quad \forall q \in Q \tag{3.3}$$

where $a(\boldsymbol{u}, \boldsymbol{v}) = 2\mu \langle \epsilon(\boldsymbol{u}), \epsilon(\boldsymbol{v}) \rangle + \lambda \langle \mathrm{div}\boldsymbol{u}, \mathrm{div}\boldsymbol{v} \rangle$ is the weak of the linear elasticity. The variational formulation employs specific function spaces, which are

$$\begin{aligned} \boldsymbol{\mathcal{V}} &= \{ \boldsymbol{u} \in \mathbf{H}^1(\Omega) \,|\, \boldsymbol{u}|_{\bar{\Gamma}_c} = 0 \}, \\ \boldsymbol{\mathcal{W}} &= \{ \boldsymbol{w} \in \mathbf{H}(\mathrm{div}, \Omega) \}, \\ Q &= L^2(\Omega), \end{aligned}$$

where $\mathbf{H}^1(\Omega)$ is the space of square integrable vector-valued functions whose first derivatives are also square integrable, and $\mathbf{H}(\mathrm{div}, \Omega)$ contains the square integrable vector-valued functions with square integrable divergences. Next, we turn our attention to finite-element discretizations of Biot's model.

3.1.2 Finite-Element Discretization.

Various discretization methods are available for different formulations of Biot's model. For instance, in the three-dimensional Biot poroelastic system, a finite-volume method on a staggered grid is outlined in [30]. Stable Taylor-Hood elements are used in the two-field formulation, as discussed in [25]. Another approach introduced in [5] utilizes a MINI element and a stabilized P1-P1 finite-element discretization, incorporating a stabilization term to eliminate nonphysical oscillations. A weak Galerkin finite-element method proposed in [24] demonstrates robustness on general polytopal meshes.

In the context of three-field formulations, [18] explores a nonconforming finite-element approach, utilizing Crouzeix-Raviart finite elements for displacements and lowest-order Raviart-Thomas-Nedelec elements for the Darcy velocity. Extensions to general cases are discussed in [36], introducing a mass-lumping technique to reduce computational costs. Hybridization schemes are developed in [27], while [36] outlines stable discretizations for a four-field formulation. The effect of the incompressibility

constraint associated with the elasticity block of the coupled system on the convergence of the proposed multigrid algorithm is addressed in [6]. Specifically, it is demonstrated that the concepts of reduced-quadrature discretization and divergence-free interpolation, initially proposed and analyzed for the incompressible elasticity subproblem, can be extended to the fully-coupled Biot model. The study shows that the modified discretization remains well-posed, and a robust monolithic multigrid approach for the resulting three-field formulation is developed.

Initially, we divide the domain Ω into n-dimensional simplices, creating a partition denoted as Ω^h. Each element of this partition, Ω^h, is associated with a triple of piecewise polynomial, finite-dimensional spaces: $\boldsymbol{\mathcal{V}}_h \subset \boldsymbol{\mathcal{V}}$, $\boldsymbol{\mathcal{W}}_h \subset \boldsymbol{\mathcal{W}}$, and $Q_h \subset Q$. Continuous function space $\boldsymbol{\mathcal{V}}_h$ ($\mathbf{H}^1$ space) comprises continuous functions and is primarily employed to approximate fields like displacement ($\boldsymbol{u}_h$). In practical implementations within the finite element method, $\boldsymbol{\mathcal{V}}_h$ is often realized using continuous Lagrange elements. These elements ensure that the basis functions are continuous within each element, facilitating the approximation of functions with continuous derivatives. $\boldsymbol{\mathcal{W}}_h$ ($\mathbf{H}$(div) space) is utilized for vector fields that possess divergence-free properties, such as fluid velocity fields. In finite-element discretizations, this space can be realized using Raviart-Thomas (RT) elements. These elements are designed to accurately represent vector fields while ensuring divergence compatibility across element boundaries. Discontinuous function space Q_h (L^2 space) represents scalar fields and is typically used to discretize quantities such as pressure (p). In finite-element simulations, Q_h is often implemented using discontinuous Lagrange elements. These elements allow for discontinuities in the basis functions across element boundaries, facilitating the approximation of scalar fields with square-integrable properties.

The spatial finite-element discretization is formulated in the weak form with rescaled variables. We introduce a semi-discretized variational problem. In Equation (3.3), we rescale p to $M^{1/2}p$ and multiply the third equation by $M^{1/2}$ to improve the error bound that follows. This results in the variational form for Biot's three-field consolidation model, encompassing Equations (3.1), (3.2) and (3.3). It is expressed as follows:

Find $(\boldsymbol{u}_h, \boldsymbol{w}_h, p_h) \in \boldsymbol{\mathcal{V}}_h \times \boldsymbol{\mathcal{W}}_h \times Q_h$ such that

$$a(\boldsymbol{u}_h, \boldsymbol{v}_h) - \langle \alpha M^{1/2} p_h, \mathrm{div}\boldsymbol{v}_h \rangle = \langle \rho \boldsymbol{g}, \boldsymbol{v}_h \rangle, \quad \forall \boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h \tag{3.4}$$

$$\langle \boldsymbol{K}^{-1} \mu_f \boldsymbol{w}_h, \boldsymbol{r}_h \rangle - \langle M^{1/2} p_h, \mathrm{div}\boldsymbol{r}_h \rangle = \langle \rho_f \boldsymbol{g}, \boldsymbol{r}_h \rangle, \quad \forall \boldsymbol{r}_h \in \boldsymbol{\mathcal{W}}_h \tag{3.5}$$

$$-\langle \frac{\partial p_h}{\partial t}, q_h \rangle - M^{1/2} \langle \alpha \mathrm{div} \frac{\partial \boldsymbol{u_h}}{\partial t}, q_h \rangle - M^{1/2} \langle \mathrm{div}\boldsymbol{w}_h, q_h \rangle = -M^{1/2} \langle f, q_h \rangle, \quad \forall q_h \in Q_h \tag{3.6}$$

where $\langle \cdot, \cdot \rangle$ represents the standard $L^2(\Omega)$ inner product. Here, $(\boldsymbol{u}_h, p_h, \boldsymbol{w}_h)$ is an approximation to $(\boldsymbol{u}(\cdot, t), p(\cdot, t), \boldsymbol{w}(\cdot, t))$. Note that (3.6) is scaled by -1 to retain some symmetry of the system. The vector function space $\boldsymbol{\mathcal{V}}_h$ serves as the domain for the trial functions representing the displacement field $\boldsymbol{u}_h$. It is constructed using continuous Lagrange elements of degree $k+1$ for each component of the displacement. $\boldsymbol{\mathcal{W}}_h$ represents the function space for the Darcy velocity $\boldsymbol{w}_h$. Constructed with Raviart-Thomas elements of degree k (where the lowest-order RT space has degree 1), $\boldsymbol{\mathcal{W}}_h$ approximates vector fields in the $H(\mathrm{div})$ space. The pressure field p_h resides in the function space Q_h, constructed with discontinuous Lagrange elements of degree $k-1$. Together, these function spaces provide the necessary discretization framework to accurately represent the displacement, Darcy velocity, and pressure fields, essential for solving the problem under consideration.

Using implicit Euler approximations for the time derivatives and rescaling Equation (3.5) byτ for symmetry, the discrete variational problem, (3.4), (3.5) and (3.6) can be defined in block matrix form as:

$$\mathcal{A} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{w} \\ p \end{pmatrix} = \boldsymbol{b}$$

with

$$\mathcal{A} = \begin{pmatrix} A_{\boldsymbol{u}} & 0 & \alpha M^{1/2} B_{\boldsymbol{u}}^T \\ 0 & \tau M_{\boldsymbol{w}} & \tau M^{1/2} B_{\boldsymbol{w}}^T \\ \alpha M^{1/2} B_{\boldsymbol{u}} & \tau M^{1/2} B_{\boldsymbol{w}} & -M_p \end{pmatrix}$$

where the blocks in the matrix A correspond to the following bilinear forms:

$$\begin{aligned} a(\boldsymbol{u}_h, \boldsymbol{v}_h) \to A_{\boldsymbol{u}}, -\langle \mathrm{div}\boldsymbol{u}_h, q_h\rangle \to B_{\boldsymbol{u}}, \\ -\langle \mathrm{div}\boldsymbol{w}_h, q_h\rangle \to B_{\boldsymbol{w}}, \\ \langle \boldsymbol{K}^{-1}\mu_f \boldsymbol{w}_h, \boldsymbol{r}_h\rangle \to M_{\boldsymbol{w}}, \quad \langle p_h, q_h\rangle \to M_p. \end{aligned}$$

The bilinear form for the reduced-quadrature discretization is defined in [6] as

$$a^{RQ}(\boldsymbol{u}, \boldsymbol{v}) := 2\mu\langle \epsilon(\boldsymbol{u}), \epsilon(\boldsymbol{v})\rangle + \lambda\langle P_{Q_h}\mathrm{div}\boldsymbol{u}, P_{Q_h}\mathrm{div}\boldsymbol{v}\rangle. \tag{3.7}$$

As discussed in [6], the non-local nature of the basis for divergence-free spaces stems from the direct evaluation of the term (div $\boldsymbol{u}$, div $\boldsymbol{v}$) in the weak form. This arises because the discrete divergence of the displacement space does not inherently align with the piecewise constant pressure space used there. To address this issue, a reduced integration approach is implemented. Instead of evaluating (div $\boldsymbol{u}$, div $\boldsymbol{v}$) directly, it is replaced with (P_{Q_h} div $\boldsymbol{u}$, P_{Q_h} div $\boldsymbol{v}$), where P_{Q_h} represents the L^2-projection from Q onto Q_h, which is the space of piecewise constant functions. By employing this reduced integration approach, a basis for the space of divergence-free functions can be constructed with local support. This allows for the effective use of local relaxation schemes for divergence-free components. The above variational problem possesses a unique solution and establishes an invertible operator whose inverse remains bounded, regardless of the mesh size h. We also adopt a reduced quadrature approach in this work, using L^2 projection into the degree k discontinuous Lagrange pressure space, Q_h.

Using the Equation (3.7), the poroelastic system with Euler time discretization is then written as

$$\mathcal{A}^{\mathcal{RQ}} = \begin{pmatrix} A_{\boldsymbol{u}}^{RQ} & 0 & \alpha M^{1/2} B_{\boldsymbol{u}}^T \\ 0 & \tau M_{\boldsymbol{w}} & \tau M^{1/2} B_{\boldsymbol{w}}^T \\ \alpha M^{1/2} B_{\boldsymbol{u}} & \tau M^{1/2} B_{\boldsymbol{w}} & -M_p \end{pmatrix},$$

where $a^{RQ}(\boldsymbol{u}_h, \boldsymbol{v}_h) \to A_{\boldsymbol{u}}^{RQ}$. Next we will prove the coercivity and continuity for the rescaled variational formulation using the following lemma.

Lemma 2 *[6] : Let the pair of finite-element spaces $\boldsymbol{\mathcal{V}}_h \times Q_h$ be Stokes-stable, i.e.,satisfy*

the inf-sup condition,

$$\sup_{\boldsymbol{v}\in\boldsymbol{\mathcal{V}}_h} \frac{\langle \mathrm{div}(\boldsymbol{v}), p\rangle}{\|\boldsymbol{v}\|_1} \leq \gamma_B^0 \|p\|, \quad \forall p \in Q_h,$$

where $\gamma_B^0 > 0$ *is a constant that does not depend on the mesh size. Then, for any* $p \in Q_h$*:*

$$\sup_{\boldsymbol{v}\in V_h} \frac{\langle \mathrm{div}(\boldsymbol{v}), p\rangle}{\|\boldsymbol{v}\|_{A_{\boldsymbol{u}}^{RQ}}} \leq \frac{\gamma_B^0}{\sqrt{d}\zeta}\|p\| =: \frac{\gamma_B}{\zeta}\|p\|, \tag{3.8}$$

where $\|\boldsymbol{v}\|^2_{A_{\boldsymbol{u}}^{RQ}} := a^{RQ}(\boldsymbol{v},\boldsymbol{v}), d$ *is the dimension, and* $\zeta := \sqrt{\lambda + 2\mu/d}$.

Now we will prove the following theorem, building on Definition 12 in Chapter 2 on the finite-element method for poroelasticity.

Theorem 7 *Let* $\boldsymbol{\mathcal{X}}_h = (\boldsymbol{\mathcal{V}}_h, \boldsymbol{\mathcal{W}}_h, Q_h)$ *be Stokes-Biot stable, that is,*

- $\exists C_{\boldsymbol{V}} > 0$ *such that* $a(\boldsymbol{u},\boldsymbol{v}) \leq C_{\boldsymbol{V}}\|\boldsymbol{u}\|_1\|\boldsymbol{v}\|_1$ *for all* $\boldsymbol{u},\boldsymbol{v} \in \boldsymbol{\mathcal{V}}_h$*;*
- $\exists \alpha_{\boldsymbol{V}} > 0$ *such that* $a(\boldsymbol{u},\boldsymbol{u}) \geq \alpha_{\boldsymbol{V}}\|\boldsymbol{u}\|_1^2$ *for all* $\boldsymbol{u} \in \boldsymbol{\mathcal{V}}_h$*;*
- $(\boldsymbol{\mathcal{W}}_h, Q_h)$ *is Poisson-stable, satisfying the necessary stability and continuity conditions for the mixed formulation of Poisson's equation; and*
- *The pair of spaces* $(\boldsymbol{\mathcal{V}}_h, Q_h)$ *is Stokes-stable,i.e., it satisfies stability for the Stokes equations.*

For $\boldsymbol{x} = (\boldsymbol{u},\boldsymbol{w},p) \in \boldsymbol{\mathcal{X}}_h$ *and* $\boldsymbol{y} = (\boldsymbol{v},\boldsymbol{w},p) \in \boldsymbol{\mathcal{X}}_h$*, define:*

$$\begin{aligned} B(\boldsymbol{x},\boldsymbol{y}) = a^{RQ}(\boldsymbol{u},\boldsymbol{v}) &- \langle \alpha M^{1/2} p, \mathrm{div}\boldsymbol{v}\rangle + \tau\langle \boldsymbol{K}^{-1}\mu_f \boldsymbol{w}, \boldsymbol{r}\rangle - \tau\langle M^{1/2}p, \mathrm{div}\boldsymbol{r}\rangle \\ &- \langle p, q\rangle - \langle M^{1/2}\alpha \mathrm{div}\boldsymbol{u}, q\rangle - \tau M^{1/2}\langle \mathrm{div}\boldsymbol{w}, q\rangle, \end{aligned} \tag{3.9}$$

$$\|\boldsymbol{x}\|^2_{D^{RQ}} = \|\boldsymbol{u}\|^2_{A_{\boldsymbol{u}}^{RQ}} + \|p\|^2 + \tau\|\boldsymbol{w}\|^2_{M_{\boldsymbol{w}}} + c_p\|\mathrm{div}\boldsymbol{w}\|^2, \tag{3.10}$$

where $\|\boldsymbol{w}\|_{M_{\boldsymbol{w}}}^2 = \langle \boldsymbol{K}^{-1}\mu_f \boldsymbol{w}, \boldsymbol{w}\rangle$ *and* $c_p = \left(1 + \frac{\alpha^2}{\zeta^2\tau^2}\right)^{-1}$. *Then,*

$$\sup_{0\neq \boldsymbol{x}\in\boldsymbol{\mathcal{X}}_h} \sup_{0\neq \boldsymbol{y}\in\boldsymbol{\mathcal{X}}_h} \frac{B(\boldsymbol{x},\boldsymbol{y})}{\|\boldsymbol{x}\|_{D^{RQ}} \|\boldsymbol{y}\|_{D^{RQ}}} \leq \bar{\zeta}, \tag{3.11}$$

$$\inf_{0\neq \boldsymbol{y}\in\boldsymbol{\mathcal{X}}_h} \sup_{0\neq \boldsymbol{x}\in\boldsymbol{\mathcal{X}}_h} \frac{B(\boldsymbol{x},\boldsymbol{y})}{\|\boldsymbol{x}\|_{D^{RQ}} \|\boldsymbol{y}\|_{D^{RQ}}} \geq \bar{\gamma}, \tag{3.12}$$

where the constants $\bar{\zeta}$ *and* $\bar{\gamma}$ *are independent of the physical and discretization parameters.*

Proof: Using Lemma 3, we know that for a given $p \in Q_h$, there exists $\boldsymbol{z} \in \boldsymbol{\mathcal{V}}_h$, such that $\|z\|_{A_{\boldsymbol{u}}^{RQ}} = \|p\|$. Let $\boldsymbol{v} = \boldsymbol{u}, \boldsymbol{r} = \boldsymbol{w}$, and $q = -p - \psi_1\tau M^{1/2}\mathrm{div}\boldsymbol{w}$ for constant ψ_1 that will be specified later. Then, by the Cauchy-Schwarz and Young's inequality,

$$\begin{aligned}
B(\boldsymbol{x},\boldsymbol{y}) &= a^{RQ}(\boldsymbol{u},\boldsymbol{u}) - \langle \alpha M^{1/2} p, \mathrm{div}\boldsymbol{u}\rangle + \tau\langle \boldsymbol{K}^{-1}\mu_f\boldsymbol{w},\boldsymbol{w}\rangle - \tau\langle M^{1/2}p, \mathrm{div}\boldsymbol{w}\rangle \\
&\quad - \langle p, -p - \psi_1\tau M^{1/2}\mathrm{div}\boldsymbol{w}\rangle - \langle M^{1/2}\alpha\mathrm{div}\boldsymbol{u}, -p - \psi_1\tau M^{1/2}\mathrm{div}\boldsymbol{w}\rangle \\
&\quad - \tau M^{1/2}\langle \mathrm{div}\boldsymbol{w}, -p - \psi_1\tau M^{1/2}\mathrm{div}\boldsymbol{w}\rangle, \\
B(\boldsymbol{x},\boldsymbol{y}) &= a^{RQ}(\boldsymbol{u},\boldsymbol{u}) + \tau\|\boldsymbol{w}\|_{M_{\boldsymbol{w}}}^2 + \tau^2 M\psi_1\|\mathrm{div}\boldsymbol{w}\|^2 + M^{1/2}\tau\psi_1\langle p, \mathrm{div}\boldsymbol{w}\rangle + \|p\|^2 \\
&\quad + M\alpha\tau\psi_1\langle P_{Q_h}\mathrm{div}\boldsymbol{u}, \mathrm{div}\boldsymbol{w}\rangle \\
&\geq \|\boldsymbol{u}\|_{A_{\boldsymbol{u}}^{RQ}}^2 - \frac{1}{2}\|\boldsymbol{u}\|_{A_{\boldsymbol{u}}^{RQ}}^2 + \tau\|\boldsymbol{w}\|_{M_{\boldsymbol{w}}}^2 \\
&\quad + \tau^2 M\psi_1\|\mathrm{div}\boldsymbol{w}\|^2 + \|p\|^2 - \frac{3}{4}\psi_1\|p\|^2 - \frac{1}{3}\psi_1\tau^2 M\|\mathrm{div}\boldsymbol{w}\|^2 \\
&\quad - \frac{1}{2}\psi_1\tau^2 M\|\mathrm{div}\boldsymbol{w}\|^2 - \frac{1}{2}\psi_1\alpha^2 M\|P_{Q_h}\mathrm{div}\boldsymbol{u}\|^2.
\end{aligned}$$

Using the properties of projection operators, we have that $\|P_{Q_h}\mathrm{div}\boldsymbol{v}\| \leq \|\mathrm{div}\boldsymbol{v}\|$, and by the Young's inequality we have that $\langle \mathrm{div}\boldsymbol{u}, \mathrm{div}\boldsymbol{u}\rangle \leq d\langle \epsilon(\boldsymbol{u}), \epsilon(\boldsymbol{u})\rangle$. This implies that,

$$\frac{1}{d}\langle P_{Q_h}\mathrm{div}\boldsymbol{u}, P_{Q_h}\mathrm{div}\boldsymbol{u}\rangle \leq \frac{1}{d}\langle \mathrm{div}\boldsymbol{u}, \mathrm{div}\boldsymbol{u}\rangle \leq \langle \epsilon(\boldsymbol{u}), \epsilon(\boldsymbol{u})\rangle.$$

Then, by direct calculation and the definition of $A_{\boldsymbol{u}}^{RQ}$, we have

$$\|P_{Q_h}\mathrm{div}\boldsymbol{u}\| \leq \frac{1}{\zeta}\|\boldsymbol{u}\|_{A_{\boldsymbol{u}}^{RQ}}. \tag{3.13}$$

Combining terms and applying (3.13) gives

$$B(\boldsymbol{x},\boldsymbol{y}) \geq \left(\frac{1}{2} - \frac{M\psi_1\alpha^2}{2\zeta^2}\right)\|\boldsymbol{u}\|^2_{A^{RQ}_{\boldsymbol{u}}} + \tau\|\boldsymbol{w}\|^2_{M_{\boldsymbol{w}}} + \left(1 - \frac{3}{4}\psi_1\right)\|p\|^2 + \frac{1}{6}\psi_1\tau^2 M\|\mathrm{div}\boldsymbol{w}\|^2.$$

Choosing $\psi_1 = \dfrac{1}{2M}\left(\tau^2 + \dfrac{\alpha^2}{\zeta^2}\right)^{-1}$, then gives

$$\begin{aligned} B(\boldsymbol{x},\boldsymbol{y}) &\geq \left(\frac{1}{2} - \frac{1}{4}\right)\|\boldsymbol{u}\|^2_{A^{RQ}_{\boldsymbol{u}}} + \tau\|\boldsymbol{w}\|^2_{M_{\boldsymbol{w}}} + \left[1 - \frac{3}{8M\tau^2}c_p\right]\|p\|^2 + \frac{1}{12}c_p\|\mathrm{div}\boldsymbol{w}\|^2, \\ &\geq \frac{1}{4}\|\boldsymbol{u}\|^2_{A^{RQ}_{\boldsymbol{u}}} + \tau\|\boldsymbol{w}\|^2_{M_{\boldsymbol{w}}} + \left[1 - \frac{3}{8M\tau^2}c_p\right]\|p\|^2 + \frac{1}{12}c_p\|\mathrm{div}\boldsymbol{w}\|^2, \\ &\geq \bar{\gamma}\|(\boldsymbol{u},\boldsymbol{w},p)\|^2_{D^{RQ}}, \end{aligned}$$

where $\bar{\gamma} = \min\{\frac{1}{4}, \frac{1}{12}\}$. Then, by the triangle inequality,
$\|\boldsymbol{y}\|^2_{D^{RQ}} = \|\boldsymbol{v}\|^2_{A^{RQ}_{\boldsymbol{u}}} + \|q\|^2 + \tau\|\boldsymbol{r}\|^2_{M_{\boldsymbol{w}}} + c_p\|\mathrm{div}\boldsymbol{r}\|^2 \leq \gamma^*\|\boldsymbol{x}\|^2_{D^{RQ}}$ where $\gamma^* = \max\{2, \frac{1}{4}\}$. Thus, the bilinear form $B(\boldsymbol{x},\boldsymbol{y})$ defined in (3.9) satisfies (3.12). For the continuty, using Cauchy-Schwarz and (3.13), we have,

$$\begin{aligned} B(\boldsymbol{x},\boldsymbol{y}) &= a^{RQ}(\boldsymbol{u},\boldsymbol{v}) - \langle\alpha M^{1/2}p, \mathrm{div}\boldsymbol{v}\rangle + \tau\langle\boldsymbol{K}^{-1}\mu_f\boldsymbol{w},\boldsymbol{r}\rangle - \tau\langle M^{1/2}p, \mathrm{div}\boldsymbol{r}\rangle \\ &\quad - \langle p,q\rangle - \langle M^{1/2}\alpha\mathrm{div}\boldsymbol{u}, q\rangle - \tau M^{1/2}\langle\mathrm{div}\boldsymbol{w}, q\rangle, \\ &\leq \|\boldsymbol{u}\|_{A^{RQ}_{\boldsymbol{u}}}\|\boldsymbol{v}\|_{A^{RQ}_{\boldsymbol{u}}} + \tau M^{1/2}c_p^{-1/2}\|p\| \ \ (\zeta\|\mathrm{div}\boldsymbol{u}\|) \\ &\quad + \tau\langle\boldsymbol{K}^{-1/2}\mu_f^{-1/2}\boldsymbol{w}\rangle\langle\boldsymbol{K}^{-1/2}\mu_f^{-1/2}\boldsymbol{r}\rangle + \tau M^{1/2}c_p^{-1/2}\|p\| \ \ (c_p^{1/2}\|\mathrm{div}\boldsymbol{r}\|) \\ &\quad + \|p\| \ \ \|q\| + \tau M^{1/2}c_p^{-1/2}(\zeta\|\mathrm{div}\boldsymbol{u}\|) \ \ \|q\| + \tau M^{1/2}c_p^{-1/2}(c_p^{1/2}\|\mathrm{div}\boldsymbol{w}\|) \ \ \|q\|, \\ &\leq (2 + \tau + 4\tau M^{1/2}c_p^{-1/2})\|\boldsymbol{x}\|_{D^{RQ}}\|\boldsymbol{y}\|_{D^{RQ}}. \end{aligned}$$

which completes the proof.

In [6], a similar result is shown, but with parameter c_p defined as $c_p = \left(\frac{1}{M} + \frac{\alpha^2}{\zeta^2}\right)^{-1}$. When M is large, the term $\frac{1}{M}$ becomes very small. Consequently, c_p is approximately $c_p \approx \left(\frac{\alpha^2}{\zeta^2}\right)^{-1}$, meaning that $(c_p)^{-1}$ is approximately $\frac{\alpha^2}{\zeta^2}$, which can be quite small if ζ is also large, as it is in the incompressible limit. This results in the pressure term $(c_p)^{-1}\|p\|^2$ being underrepresented the overall norm $\|\boldsymbol{x}\|^2_{D^{RQ}}$, which can potentially lead to numerical instability and less accurate pressure approximation. In contrast, the new approach defines c_p as $\left(1 + \frac{\alpha^2}{\zeta^2\tau^2}\right)^{-1}$. This formulation ties c_p to the discretization

parameter τ rather than M. This avoids the problem of $(c_p)^{-1}$ becoming excessively small and provides a more stable and balanced treatment of the pressure term in the norm. By reducing the dependency on large values of M and controlling c_p through τ, the new approach gives better balance in the product norm.

To gain deeper insights into the selection of the weighted norm (3.10), let's examine two extreme scenarios. As λ approaches infinity, $B(\boldsymbol{x}, \boldsymbol{y})$ becomes dominated by $\zeta\langle P_{Q_h} \operatorname{div} \boldsymbol{u}, P_{Q_h} \operatorname{div} \boldsymbol{v}\rangle$, highlighting the significance of the term $\lambda\|P_{Q_h} \operatorname{div} \mathbf{u}\|^2$ in the weighted norm. Conversely, as τ tends to zero, $B(\boldsymbol{x}, \boldsymbol{y})$ simplifies to $a^{RQ}(\boldsymbol{u}; \boldsymbol{v}) - M^{1/2}\alpha\langle p, \operatorname{div} \boldsymbol{v}\rangle - M^{1/2}\alpha\langle \operatorname{div} \boldsymbol{u}, q\rangle - \langle p, q\rangle$, resembling a problem relative to Stokes flow. In this scenario, the weighted norm (3.10) reduces to $\|\boldsymbol{u}\|^2_{A^{RQ}_{\boldsymbol{u}}} + \|p\|^2$, making it an appropriate choice for Stokes-type problems. Thus, the weighted norm (3.10) proves to be well-suited for handling these limiting cases.

3.2 Spatiotemporal IRK discretization

In this section, we present the application of Implicit Runge-Kutta (IRK) methods to Biot's model. Here, we rewrite the time-dependent problem:

$$-\operatorname{div}(2\mu\epsilon(\boldsymbol{u})) - \lambda\nabla(\operatorname{div}\boldsymbol{u}) + \alpha\nabla p = \rho\boldsymbol{g} \qquad \text{in } \Omega \times (0, T_{final}), \tag{3.14a}$$

$$\boldsymbol{K}^{-1}\mu_f\boldsymbol{w} + \nabla p = \rho_f\boldsymbol{g} \qquad \text{in } \Omega \times (0, T_{final}), \tag{3.14b}$$

$$\frac{\partial}{\partial t}\left(\frac{1}{M}p + \alpha\operatorname{div}\boldsymbol{u}\right) + \operatorname{div}\boldsymbol{w} = f \qquad \text{in } \Omega \times (0, T_{final}). \tag{3.14c}$$

which we endow with appropriate initial data:

$$\boldsymbol{u}(0) = \boldsymbol{u}_0, \quad \text{in } \Omega \times \{t = 0\}.$$

The presence of algebraic constraints, such as the divergence-free condition in Biot's model, poses challenges for time integration. IRK methods are particularly well-suited for handling such constraints, as they can handle both differential and algebraic equations simultaneously. The choice of IRK method and its order affect the accuracy and stability of the numerical solution. Higher-order IRK methods allow for more accurate approximations but may require more degrees of freedom to maintain stability, especially in the presence of stiff constraints.

We rewrite a rescaled semi-discretized weak form for Biot's three-field consolidation model of (3.14a) – (3.14c). For each $t \in (0, T]$, find $(\boldsymbol{u}(t), \boldsymbol{w}(t), p(t)) \in \boldsymbol{\mathcal{V}}_h \times \boldsymbol{\mathcal{W}}_h \times Q_h$ such that

$$a(\boldsymbol{u}, \boldsymbol{v}) - \alpha M^{1/2} \langle p, \mathrm{div} \boldsymbol{v} \rangle = \langle \rho \boldsymbol{g}, \boldsymbol{v} \rangle, \quad \forall \boldsymbol{v} \in \boldsymbol{\mathcal{V}} \tag{3.15a}$$

$$\langle \boldsymbol{K}^{-1} \mu_f \boldsymbol{w}, \boldsymbol{r} \rangle - M^{1/2} \langle p, \mathrm{div} \boldsymbol{r} \rangle = \langle \rho_f \boldsymbol{g}, \boldsymbol{r} \rangle, \quad \forall \boldsymbol{r} \in \boldsymbol{\mathcal{W}} \tag{3.15b}$$

$$-\langle \frac{\partial p}{\partial t}, q \rangle - \alpha M^{1/2} \langle \mathrm{div} \frac{\partial \boldsymbol{u}}{\partial t}, q \rangle - M^{1/2} \langle \mathrm{div} \boldsymbol{w}, q \rangle = -M^{1/2} \langle f, q \rangle, \quad \forall q \in Q. \tag{3.15c}$$

Now denoting $\tilde{\boldsymbol{u}}(t), \tilde{\boldsymbol{w}}(t)$ and $\tilde{p}(t)$ as the time-dependent coefficients of $\boldsymbol{u}(x,t), \boldsymbol{w}(x,t)$ and $p(x,t)$ in the finite-element basis, we can write this as a linear system of DAEs as

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \alpha M^{1/2} B_{\boldsymbol{u}} & 0 & -M_p \end{pmatrix} \begin{pmatrix} \frac{\partial \tilde{\boldsymbol{u}}}{\partial t} \\ \frac{\partial \tilde{\boldsymbol{w}}}{\partial t} \\ \frac{\partial \tilde{p}}{\partial t} \end{pmatrix} + \begin{pmatrix} A_{\boldsymbol{u}}^{RQ} & 0 & \alpha M^{1/2} B_{\boldsymbol{u}}^T \\ 0 & M_{\boldsymbol{w}} & M^{1/2} B_{\boldsymbol{w}}^T \\ 0 & M^{1/2} B_{\boldsymbol{w}} & 0 \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{u}} \\ \tilde{\boldsymbol{w}} \\ \tilde{p} \end{pmatrix} = \begin{pmatrix} \tilde{\boldsymbol{f}}_1 \\ \tilde{\boldsymbol{f}}_2 \\ \tilde{f}_3 \end{pmatrix}, \tag{3.16}$$

where $\tilde{\boldsymbol{f}}_1 = \rho \tilde{\boldsymbol{g}}, \tilde{\boldsymbol{f}}_2 = \rho_f \tilde{\boldsymbol{g}}$ and $\tilde{f}_3 = -M^{1/2} \tilde{f}$. Next we consider using an s-stage implicit Runge-Kutta method applied to a system of ordinary differential equations $\boldsymbol{u}'(t) = \boldsymbol{f}(\boldsymbol{u}(t), t)$, given by

$$\begin{aligned} \boldsymbol{k}_i &= \boldsymbol{f}\left(\boldsymbol{u}^n + \Delta t \sum_{j=1}^{s} a_{ij} \boldsymbol{k}_j, t^n + c_i \Delta t \right), \quad \text{for } i = 1, 2, \ldots, s, \\ \boldsymbol{u}^{n+1} &= \boldsymbol{u}^n + \Delta t \sum_{j=1}^{s} b_j \boldsymbol{k}_j. \end{aligned} \tag{3.17}$$

The coefficients in the scheme are the stage times (or nodes) c_i, the weights b_j, and the Runge-Kutta matrix $A = [a_{ij}]$. The s stage values are represented by the set $\{\boldsymbol{k}_i\}_{i=1}^s$, and the approximation at time $t^n = t^0 + n\Delta t$ is denoted by $\boldsymbol{u}^n$. Since the system in (3.16) comprises differential-algebraic equations rather than ordinary differential equations, we employ the DAE analogue of the standard Runge-Kutta scheme replacing time derivative by stage derivative approximations and function

values by their stage value approximations,

$$\begin{aligned} A_{\boldsymbol{u}}^{RQ}\tilde{\boldsymbol{u}}_i^n + \alpha M^{1/2} B_{\boldsymbol{u}}^T \tilde{p}_i^n &= \tilde{\boldsymbol{f}}_{\mathbf{1}_i}^{\ n}, \\ \tilde{\boldsymbol{w}}_i^n + M^{1/2} B_{\boldsymbol{w}}^T \tilde{p}_i^n &= \tilde{\boldsymbol{f}}_{\mathbf{2}_i}^{\ n}, \\ \alpha M^{1/2} B_{\boldsymbol{u}} \tilde{k}_i^{(\boldsymbol{u})} - M_p \tilde{k}_i^{(p)} + M^{1/2} B_{\boldsymbol{w}} \tilde{\boldsymbol{w}}_i{}^n &= \tilde{f}_{3_i}^{\ n}. \end{aligned}$$

This involves updating the solution vectors $\tilde{\boldsymbol{u}}_i^n, \tilde{\boldsymbol{w}}_i^n$ and $\tilde{p}_i^n$ using the scheme's coefficients and the terms in (3.16),

$$\begin{aligned} \tilde{\boldsymbol{u}}_i^n &= \tilde{\boldsymbol{u}}^n + \Delta t \sum_{j=1}^{s} a_{ij} \tilde{k}_j^{(\boldsymbol{u})}, \\ \tilde{\boldsymbol{w}}_i^n &= \tilde{\boldsymbol{w}}^n + \Delta t \sum_{j=1}^{s} a_{ij} \tilde{k}_j^{(\boldsymbol{w})} \\ \tilde{p}_i^n &= \tilde{p}^n + \Delta t \sum_{j=1}^{s} a_{ij} \tilde{k}_j^{(p)}. \end{aligned}$$

The next time step is computed using the updated values $\tilde{\boldsymbol{u}}^{n+1}, \tilde{\boldsymbol{w}}^{n+1}$ and $\tilde{p}^{n+1}$,

$$\begin{aligned} \tilde{\boldsymbol{u}}^{n+1} &= \tilde{\boldsymbol{u}}^n + \Delta t \sum_{j=1}^{s} b_j \tilde{k}_j^{(\boldsymbol{u})}, \\ \tilde{\boldsymbol{w}}^{n+1} &= \tilde{\boldsymbol{w}}^n + \Delta t \sum_{j=1}^{s} b_j \tilde{k}_j^{(\boldsymbol{w})}, \\ \tilde{p}^{n+1} &= \tilde{p}^n + \Delta t \sum_{j=1}^{s} b_j \tilde{k}_j^{(p)}. \end{aligned}$$

Where $\tilde{\boldsymbol{f}}_i^n$ represents the basis coefficient representation of $\boldsymbol{f}$ in $\boldsymbol{\mathcal{V}}_h$ at time $t_n + c_i \Delta t$, $\tilde{\boldsymbol{u}}_i^n, \tilde{\boldsymbol{w}}_i^n$ and $\tilde{p}_i^n$ denote the approximations of $\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{w}}$ and $\tilde{p}$ at time $t_n + c_i \Delta t$, and $\tilde{k}_i^{(\boldsymbol{u})}, \tilde{k}_i^{(\boldsymbol{w})}$ and $\tilde{k}_i^{(p)}$ denote the RK stages for which we solve. The equations for $\tilde{k}_i^{(\boldsymbol{u})}, \tilde{k}_i^{(\boldsymbol{w})}$ and

$\tilde{k}_i^{(p)}$ can be rewritten as follows:

$$A_{\boldsymbol{u}}^{RQ}\left(\tilde{\boldsymbol{u}}^n+\Delta t\sum_{j=1}^{s}a_{ij}\tilde{k}_j^{(\boldsymbol{u})}\right)+\alpha M^{1/2}B_{\boldsymbol{u}}^T\left(\tilde{p}^n+\Delta t\sum_{j=1}^{s}a_{ij}\tilde{k}_j^{(p)}\right)=\tilde{\boldsymbol{f}}_{\mathbf{1}i}{}^n \tag{3.18a}$$

$$M_{\boldsymbol{w}}\left(\tilde{\boldsymbol{w}}^n+\Delta t\sum_{j=1}^{s}a_{ij}\tilde{k}_j^{(\boldsymbol{w})}\right)+M^{1/2}B_{\boldsymbol{w}}^T\left(\tilde{p}^n+\Delta t\sum_{j=1}^{s}a_{ij}\tilde{k}_j^{(p)}\right)=\tilde{\boldsymbol{f}}_{\mathbf{2}i}{}^n \tag{3.18b}$$

$$\alpha M^{1/2}B_{\boldsymbol{u}}\tilde{k}_i^{(\boldsymbol{u})}+M^{1/2}B_{\boldsymbol{w}}\left(\tilde{\boldsymbol{w}}^n+\Delta t\sum_{j=1}^{s}a_{ij}\tilde{k}_j^{(\boldsymbol{w})}\right)-M_p\tilde{k}_i^{(p)}n=\tilde{f}_{3i}{}^n \tag{3.18c}$$

for $1\leq i\leq s$.

3.3 Monolithic Multigrid

Monolithic Multigrid for higher-order IRK discretizations of poroelasticity is an efficient numerical technique. It combines multigrid methods with implicit Runge-Kutta schemes, optimizing computational performance for solving poroelasticity problems. This approach ensures rapid convergence and scalability, making it suitable for handling complex, large-scale simulations in poroelasticity with higher-order accuracy.

3.3.1 Divergence-Preserving Interpolation

An essential aspect of developing robust solvers for elasticity, particularly in scenarios where the material approaches incompressibility (i.e., for large values of λ), is the interpolation of divergence-free functions from the coarse mesh to the fine mesh. This process ensures that if $\boldsymbol{u}_H$ represents a divergence-free function on the coarse grid, then, according to the divergence theorem

$$\int_{\partial T}\boldsymbol{n}^T\boldsymbol{u}_H\,ds=0\quad\text{for all } T\in\Omega^H.$$

In the equation above, the subscript H denotes the coarse grid, with its elements forming the set Ω^H. Ensuring that the prolongation of $\boldsymbol{u}_H$ to the fine grid remains

divergence-free leads to the requirement that:

$$\int_{\partial T} \boldsymbol{n}^T (P_H \boldsymbol{u}_H)\, ds = 0 \quad \text{for all } T \in \Omega^h \tag{3.19}$$

where P represents the prolongation operator from the coarse to the fine grid. The standard finite-element interpolation operator used on the displacement space fails to meet this requirement. In the realm of developing robust solvers for poroelasticity using monolithic multigrid with higher-order implicit Runge-Kutta discretization, a pivotal concern lies in conserving the divergence-free properties during the interpolation phase from coarse to fine grids, especially as the material approaches incompressibility. To achieve this, during the interpolation process to the fine grid using the prolongation operator P_H, it is imperative to preserve the divergence-free nature of $\boldsymbol{u}_H$. This entails modifying fine-grid functions within coarse-grid macro cells to eliminate any divergence introduced during interpolation. We define a subspace $\widetilde{\boldsymbol{\mathcal{V}}_h}$ comprising functions that vanish on macro cell boundaries, addressing divergence introduced by interpolation within cell boundaries. Then, by solving for a modified function $\widetilde{\boldsymbol{u}}_h$ within $\widetilde{\boldsymbol{\mathcal{V}}_h}$, the divergence is properly accounted for and eliminated within coarse-grid macro cells. The modified prolongation operator $\widetilde{P}_H$ is finally defined as the difference between the interpolated function $P_H \boldsymbol{u}_H$ and the corrected function $\widetilde{\boldsymbol{u}}_h$, ensuring that the fine-grid solution maintains divergence-free properties critical for precise poroelasticity simulations [16]. This comprehensive approach enhances the robustness and accuracy of the solver by preserving the divergence-free nature of the velocity field throughout the interpolation process.

3.3.2 Vanka relaxation

Vanka relaxation is a technique commonly employed within the realm of multigrid methods for efficiently solving saddle-point problems. It has been successfully adapted for various discretization schemes and applied to a wide range of saddle-point problems [26, 27].

Again consider the bilinear form for the reduced-quadrature discretization is

$$a^{RQ}(\boldsymbol{u}, \boldsymbol{v}) := 2\mu(\epsilon(\boldsymbol{u}), \epsilon(\boldsymbol{v})) + \lambda(P_{Qh}\mathrm{div}\boldsymbol{u}, P_{Qh}\mathrm{div}\boldsymbol{v}).$$

The implicit Euler discretization of the poroelastic system is written in [6] as

$$\mathcal{A}^{RQ} = \begin{pmatrix} A_{\boldsymbol{u}}^{RQ} & 0 & \alpha M^{1/2} B_{\boldsymbol{u}}^T \\ 0 & M_{\boldsymbol{w}} & M^{1/2} B_{\boldsymbol{w}}^T \\ \alpha M^{1/2} B_{\boldsymbol{u}} & M^{1/2} B_{\boldsymbol{w}} & -M_p \end{pmatrix}$$

Given a decomposition of the set of DoFs into L (overlapping) blocks, a standard Schwarz method is most easily defined by defining the restriction operator, V_ℓ, from global vectors to local vectors on block ℓ. Then, given a current residual, $r^{(j)} = b - \mathcal{A}^{RQ} x^{(j)}$, we can solve the projected system

$$V_\ell \mathcal{A}^{RQ} V_\ell^T \hat{\boldsymbol{x}}_\ell = V_\ell \boldsymbol{r}^{(j)}$$

on each block. The weighted additive form of the relaxation is then

$$\boldsymbol{x}^{(j+1)} = \boldsymbol{x}^{(j)} + \omega \sum_\ell V_\ell^T D_\ell^{-1} \hat{\boldsymbol{x}}_\ell$$

where ω is a damping parameter and D_ℓ is a diagonal weight matrix chosen to account for the fact that different (global) degrees of freedom (DoFs) appear in different numbers of patches. In this context, D_ℓ is defined based on the "natural weights" derived from the overlapping block decomposition. Each diagonal entry of D_ℓ is determined by the inverse of the number of patches containing the respective Degree of Freedom (DoF).

To apply monolithic multigrid methods to the higher-order implicit Runge-Kutta (IRK) discretization, the selection of blocks in Equation (3.22) is critical. Following the approach outlined in [1], we form blocks using the vertex star construction detailed in [15]. This construction couples the degrees of freedom from all IRK stages at all gridpoints in the patch. Reference [15] provides essential details on implementing these methods, ensuring robust and efficient multigrid relaxation in poroelasticity simulations.

Chapter 4

Numerical Results

In this chapter, we present the numerical results obtained from simulations aimed at analyzing the performance of implicit Euler discretization, implicit Runge-Kutta (IRK) discretization, and monolithic multigrid preconditioners coupled with a Vanka relaxation scheme for achieving higher-order spatial discretization. All numerical experiments were conducted on a workstation with two 8-core 1.7GHz Intel Xeon Bronze 3106 CPUs and 384 GB of RAM, and the results are presented in terms of computational efficiency and accuracy.

4.1 Time steady Problem

We first consider the following example: in this case, the right-hand side functions are chosen so that the exact solution is given by:

$$\boldsymbol{u}(x,y,t) = \text{curl}\phi = \begin{pmatrix} \dfrac{\partial \phi}{\partial y} \\ -\dfrac{\partial \phi}{\partial x} \end{pmatrix}, \quad \phi(x,y) = [xy(1-x)(1-y)]^2,$$
$$p(x,y,t) = 1, \quad \boldsymbol{w}(x,y,t) = \mathbf{0}.$$

To verify the accuracy of the reduced quadrature formulation in approximating the problem, we conduct a convergence analysis of the finite-element discretization concerning the mesh size, denoted by $h = \frac{1}{N}$, where N represents the number of vertices in each dimension. For this analysis, we choose $\tau = 1.0$ and $K = 10^{-6}$ as illustrative

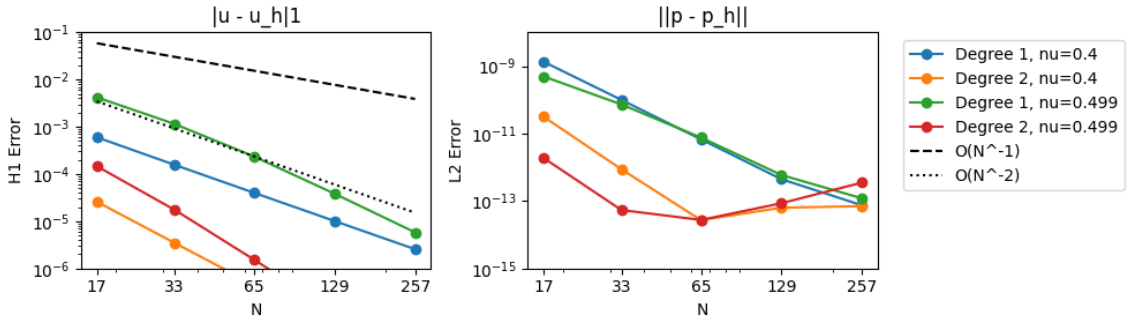

Figure 4.1: Convergence study for steady state problem, using implicit Euler scheme with Vanka relaxation scheme. Left: H^1-seminorm error for displacement vs. mesh size. Right: L^2-error for pressure vs. mesh size.

parameters. The results are depicted in Figure 4.1 for $\nu = 0.4$ and $\nu = 0.499$, with degrees 1 and 2. The displacement with degree $k = 1$ exhibits second-order convergence with respect to the H^1 seminorm, while the pressure demonstrates superconvergence. In the left-hand figure, differences in errors between $\nu = 0.4$ and $\nu = 0.499$ are observed. Comparing this with the right-hand figure, major variations in error values are noticed. Notably, at degree 2, there is rapid convergence, followed by a rise in error after $N = 65$. Here, enhanced convergence occurs due to the inherent smoothness of the solution (constant pressure) and the use of a uniform mesh, achieving machine precision with degree 2.

4.2 Time-Dependent Model Problem

We next consider a slightly more realistic test problem, now with a time-dependent smooth solution taken from [6, 19], The manufactured solution is defined on $\Omega = [0, 1]^2$, as

$$\begin{aligned}
\boldsymbol{u}(x, y, t) &= e^{-t} \begin{pmatrix} \sin(\pi y)(-\cos(\pi x) + \dfrac{1}{\mu + \lambda} \sin(\pi x)) \\ \sin(\pi x)(\cos(\pi y) + \dfrac{1}{\mu + \lambda} \sin(\pi y)) \end{pmatrix}, \\
p(x, y, t) &= e^{-t} \sin(\pi x) \sin(\pi y), \\
\boldsymbol{w}(x, y, t) &= -K \nabla p.
\end{aligned}$$

with right-hand sides chosen appropriately. We consider Dirichlet boundary conditions on all sides for displacement and pressure. The physical parameters are $\mu_f = 1$, $M = 10^6$, $K = 10^{-6}$ and $E = 3 \times 10^4$. Given our primary focus on the incompressible limit, we restrict our analysis to Poisson ratios above 0.4. Figure 4.2 illustrates the convergence study for a steady-state smooth test problem employing an implicit Euler scheme with monolithic multigrid solver using Poisson ratios of 0.4 and 0.499. Notably, we observe a decrease in error for both Poisson ratios in degree 1, with a particularly significant reduction noted for $\nu = 0.4$. However, there is minimal discrepancy in error values across different ν values, consistent with observations in degrees 3 and 4. On the right-hand side, no significant changes are observed in error values from degree 1 to degree 4 with varying ν. Nonetheless, both displacement and pressure exhibit convergence. This indicates that accuracy is limited by the time-stepper, so, we will use higher order IRK method for more accuracy.

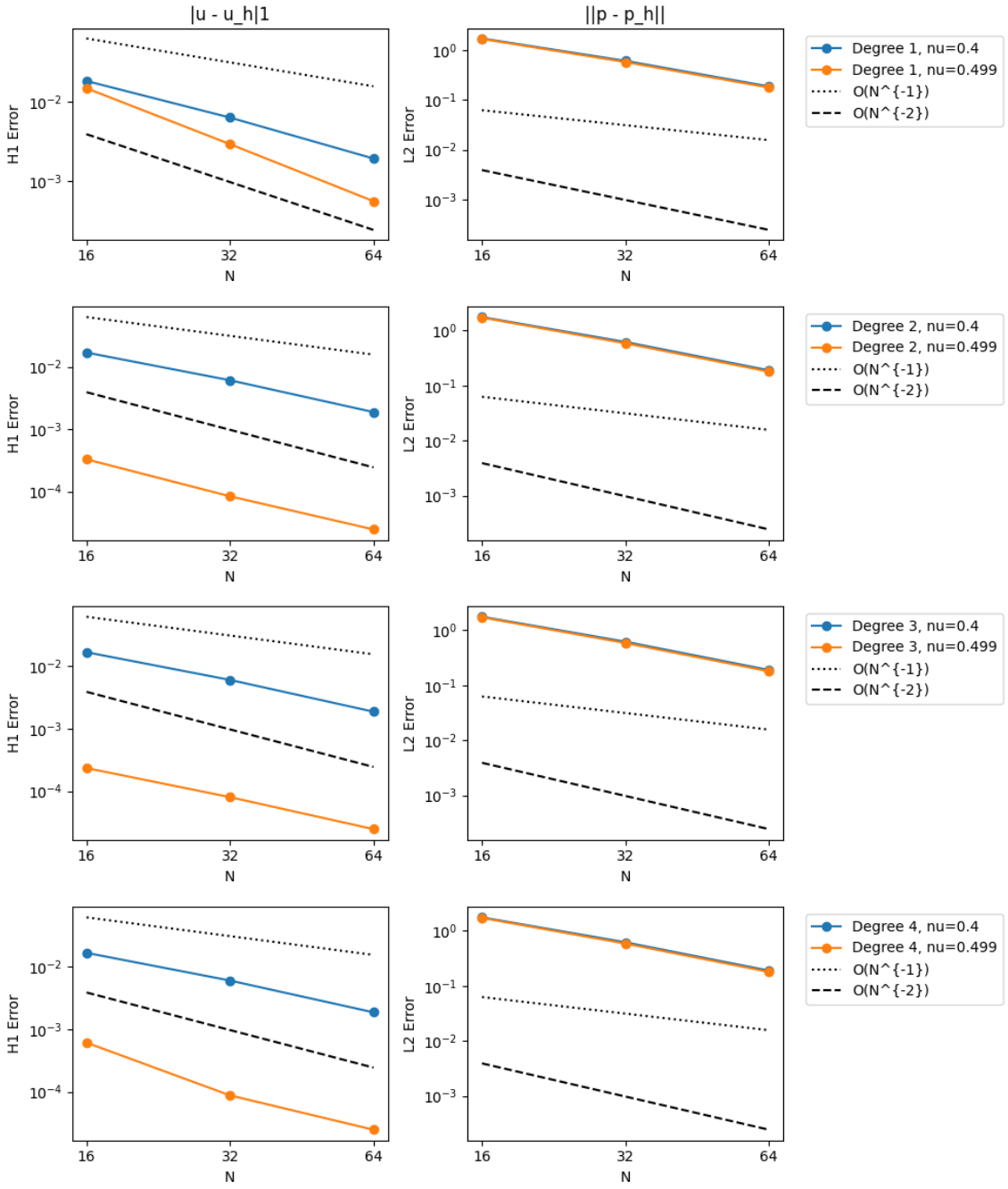

Figure 4.2: Convergence study for steady state problem, using implicit Euler scheme with monolithic multigrid. Left: H^1-seminorm error for displacement vs. mesh size. Right: L^2-error for pressure vs. mesh size.

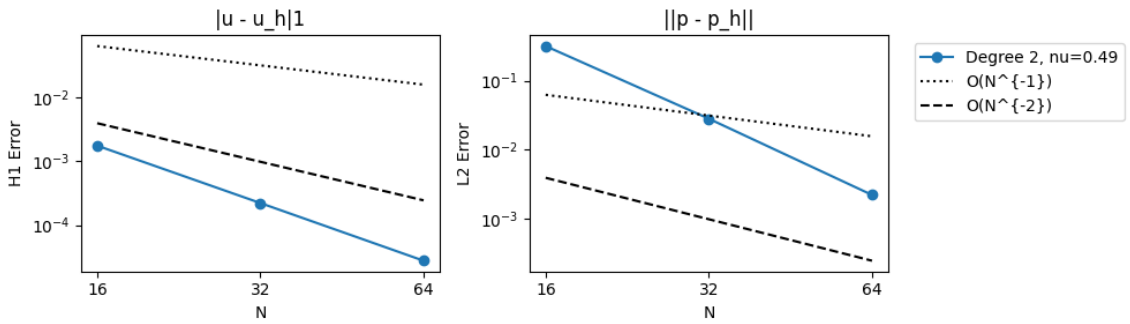

Figure 4.3: Convergence study for time varying problem, using implicit RK scheme in time with monolithic multigrid solver using $\nu = 0.49$.

4.3 Implicit higher order RK scheme

We again consider the time-dependent test problem to analyse the performance of the IRK discretization. In Figure 4.3, a convergence study for a transient problem is presented. The study employs the implicit Runge-Kutta (IRK) scheme for time discretization, coupled with the monolithic multigrid scheme using $\nu = 0.49$. The displacement exhibits second-order convergence with respect to the H^1-seminorm, consistent with expectations. Moreover, the pressure demonstrates second-order convergence, even as ν approaches 0.5. Comparison of this data, using degree $k = 2$ for the spatial discretization and a second-order, three-stage IRK scheme, exhibits enhanced accuracy and second-order convergence faster than the implicit Euler scheme, as seen in the results in Figure 4.2.

4.4 Monolithic multigrid for higher order IRK scheme

In this section, we explore the application of a monolithic multigrid method combined with Vanka relaxation to solve systems using the higher-order implicit Runge-Kutta (IRK) discretization. Through numerical experiments and analysis, we evaluate the performance and scalability of the monolithic multigrid solver with Vanka relaxation for different degrees of IRK schemes and mesh sizes. The results provide insights into the effectiveness of the proposed approach in handling stiff systems efficiently and accurately.

Analyzing the results from Table 4.1, we focus on the case where $\nu = 0.45$. In

Stage 1	Mesh size	Velocity Error	Pressure Error	Iterations
Degree 1	16	2.827×10^{-2}	1.219×10^{4}	4
	32	8.561×10^{-3}	6.089×10^{3}	5
	64	2.912×10^{-3}	3.040×10^{3}	4
Degree 2	16	6.797×10^{-3}	1.224×10^{4}	4
	32	3.379×10^{-3}	6.095×10^{3}	4
	64	1.686×10^{-3}	3.041×10^{3}	5
Degree 3	16	6.773×10^{-3}	1.224×10^{4}	4
	32	3.379×10^{-3}	6.095×10^{3}	4
	64	1.686×10^{-3}	3.041×10^{3}	5
Stage 2	Mesh size	Velocity Error	Pressure Error	Iterations
Degree 1	16	2.312×10^{-2}	9.326×10^{1}	5
	32	5.828×10^{-3}	2.381×10^{1}	5
	64	1.461×10^{-3}	5.999×10^{0}	5
Degree 2	16	3.317×10^{-4}	1.173×10^{1}	4
	32	4.190×10^{-5}	1.744×10^{0}	5
	64	5.259×10^{-6}	2.498×10^{-1}	5
Degree 3	16	8.805×10^{-6}	1.240×10^{1}	4
	32	6.505×10^{-7}	1.790×10^{0}	4
	64	5.755×10^{-8}	2.522×10^{-1}	4
Stage 3	Mesh size	Velocity Error	Pressure Error	Iterations
Degree 1	16	2.312×10^{-2}	9.337×10^{1}	5
	32	5.282×10^{-3}	2.38×10^{1}	5
	64	1.461×10^{-3}	5.996×10^{0}	5
Degree 2	16	3.318×10^{-4}	7.574×10^{-1}	5
	32	4.190×10^{-5}	4.603×10^{-2}	5
	64	5.260×10^{-6}	3.037×10^{-3}	5

Table 4.1: Velocity error, pressure error, and iterations for $\nu = 0.45$ at different stages s with different mesh sizes.

comparing the velocity and pressure errors as well as the number of iterations across different stages and degrees of the IRK scheme, several observations emerge. Firstly, regarding the convergence behavior, we note distinct trends across the various stages. For Stage 1, both the velocity and pressure errors decrease as the mesh size increases for all degrees of the IRK scheme. Interestingly, the number of iterations remains relatively stable even as the mesh size increases, suggesting consistent convergence behavior. Comparing different degrees of the IRK scheme, we find that Degree 1 generally exhibits higher errors compared to higher degrees (2 and 3). This trend holds across all stages and mesh sizes, indicating the superior convergence properties of higher degrees. Specifically, higher degrees tend to converge to lower errors with fewer iterations, particularly evident with smaller mesh sizes. Considering the best performing stage, it appears that Stage 3 offers the most promising results for $\nu = 0.45$. This conclusion is drawn from its consistent achievement of lower errors compared to Stage 1 and Stage 2 across all degrees of the IRK scheme. Additionally, the number of iterations required for convergence in Stage 3 is comparable to other stages, suggesting efficient convergence behavior. For $\nu = 0.45$, Stage 3 coupled with higher degrees of the IRK scheme provides the optimal combination of low errors and efficient convergence. In Table 4.2, focusing on $\nu = 0.49$, we evaluate velocity and pressure errors alongside iteration counts across different mesh sizes and degrees. In Stage 1, for Degree 2, at mesh size 16, the velocity error is 1.742×10^{-3} with a pressure error of 1.758×10^{1} and 7 iterations, improving to 2.214×10^{-4} velocity error, 2.437×10^{0} pressure error, and 6 iterations at mesh size 32. In contrast, for Degree 3, mesh size 16 yields a higher velocity error of 1.754×10^{-2} with a substantially greater pressure error of 1.076×10^{4} and 37 iterations, decreasing to 2.237×10^{-3} velocity error, 5.318×10^{3} pressure error, and 55 iterations at mesh size 32. Comparing Degree 2 and Degree 3 at mesh size 32, Degree 2 exhibits lower errors and iteration counts, indicating superior convergence. Therefore, for Stage 1 with $\nu = 0.49$, the Degree 2 IRK scheme is preferable. Moving on to Stage 2 and Stage 3, similar comparisons can be made between different degrees and mesh sizes to identify the optimal configuration based on error convergence and iteration counts. After comprehensive analysis, the optimal combination of stage and degree for the higher-order IRK scheme with monolithic multigrid, under $\nu = 0.49$, reveals that Stage 3 with Degree 2 consistently exhibits the lowest errors and iteration counts across diverse mesh sizes. This configuration yields the most promising convergence behavior, ensuring both accuracy and computational

Stage 1	Mesh size	Velocity Error	Pressure Error	Iterations
	16	1.742×10^{-3}	1.758×10^{1}	7
Degree 2	32	2.214×10^{-4}	2.437×10^{0}	6
	16	1.754×10^{-2}	1.076×10^{4}	37
Degree 3	32	2.237×10^{-3}	5.318×10^{3}	55
Stage 2	Mesh size	Velocity Error	Pressure Error	Iterations
	16	1.289×10^{-1}	1.187×10^{2}	5
Degree 1	32	3.254×10^{-2}	3.036×10^{1}	5
	64	8.162×10^{-3}	7.656×10^{0}	6
	16	1.742×10^{-3}	1.758×10^{1}	7
Degree 2	32	2.214×10^{-3}	2.437×10^{0}	6
	64	2.786×10^{-4}	3.269×10^{-1}	7
	16	4.457×10^{-5}	1.790×10^{1}	8
Degree 3	32	2.799×10^{-6}	2.463×10^{0}	7
	64	1.765×10^{-7}	3.289×10^{-1}	9
Stage 3	Mesh size	Velocity Error	Pressure Error	Iterations
	16	1.289×10^{-1}	1.188×10^{2}	6
Degree 1	32	3.254×10^{-2}	3.034×10^{1}	6
	64	8.162×10^{-3}	7.653×10^{0}	6
	16	1.743×10^{-3}	3.668×10^{-1}	7
Degree 2	32	2.214×10^{-4}	3.17×10^{-2}	7
	64	2.786×10^{-5}	2.437×10^{-3}	6

Table 4.2: Velocity error, pressure error, and iterations for $\nu = 0.49$ at different stages s with different mesh sizes.

efficiency. Moreover, the term $M^{1/2}$ appears in multiple equations and significantly impacts the pressure term p. When $M^{1/2}$ is large, it scales the pressure contributions in the equations, which can amplify any numerical inaccuracies or instabilities, leading to higher pressure errors. We can see this effect in the above tables.

Chapter 5

Conclusion

Our study successfully integrates higher-order discretization techniques with monolithic multigrid methods, demonstrating their efficacy in solving poroelasticity problems. By employing higher-order finite elements and implicit Runge-Kutta methods, we achieve both accuracy and stability in simulating complex poroelastic phenomena. We extend Vanka-style relaxation techniques to poroelastic equations, enhancing the efficiency and robustness of our numerical approach. Our results demonstrate the effectiveness of our approach in accurately capturing poroelastic behavior while maintaining computational efficiency, paving the way for further advancements in poroelasticity modeling and simulation. Moving forward, future work will focus on further investigation into the scalability of our method for large-scale poroelastic simulations particularly in 3D, exploration of adaptive mesh refinement strategies, development of parallel and distributed computing techniques, enhancing the efficiency and robustness of our numerical approach.

Bibliography

[1] R. Abu-Labdeh, S. MacLachlan, and P. E. Farrell. Monolithic multigrid for implicit Runge-Kutta discretizations of incompressible fluid flow. *J. Comput. Phys.*, 478:Paper No. 111961, 18, 2023.

[2] J. Adler, S. MacLachlan, H. Sterck, and L. Olson. *Numerical Partial Differential Equations*. SIAM, 2024. To appear.

[3] J. H. Adler, T. R. Benson, and S. P. MacLachlan. Preconditioning a mass-conserving discontinuous Galerkin discretization of the Stokes equations. *Numer. Linear Algebra Appl.*, 24(3):e2047, 23, 2017.

[4] J. H. Adler, F. J. Gaspar, X. Hu, P. Ohm, C. Rodrigo, and L. T. Zikatanov. Robust preconditioners for a new stabilized discretization of the poroelastic equations. *SIAM Journal on Scientific Computing*, 42(3):B761–B791, 2020.

[5] J. H. Adler, F. J. Gaspar, X. Hu, C. Rodrigo, and L. T. Zikatanov. Robust block preconditioners for Biot's model. In *Domain Decomposition Methods in Science and Engineering XXIV*, volume 125 of *Lect. Notes Comput. Sci. Eng.*, pages 3–16. Springer, Cham, 2018.

[6] J. H. Adler, Y. He, X. Hu, S. MacLachlan, and P. Ohm. Monolithic multigrid for a reduced-quadrature discretization of poroelasticity. *SIAM J. Sci. Comput.*, 45(3):S54–S81, 2023.

[7] M. A. Biot. General theory of three-dimensional consolidation. *Journal of Applied Physics*, 12(2):155–164, 1941.

[8] M. A. Biot. Theory of elasticity and consolidation for a porous anisotropic solid. *J. Appl. Phys.*, 26:182–185, 1955.

[9] D. Boffi, F. Brezzi, and M. Fortin. *Mixed Finite Element Methods and Applications*, volume 44 of *Springer Series in Computational Mathematics*. Springer, Heidelberg, 2013.

[10] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer, New York, third edition, 2008.

[11] J. Brown, Y. He, S. MacLachlan, M. Menickelly, and S. M. Wild. Tuning multigrid methods with robust optimization and local Fourier analysis. *SIAM Journal on Scientific Computing*, 43(1):A109–A138, 2021.

[12] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, second edition, 2014.

[13] A. Ern and J.-L. Guermond. *Finite elements II—Galerkin Approximation, Elliptic and Mixed PDEs*, volume 73 of *Texts in Applied Mathematics*. Springer, Cham, [2021].

[14] L. C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, second edition, 2010.

[15] P. E. Farrell, M. G. Knepley, L. Mitchell, and F. Wechsung. PCPATCH: Software for the topological construction of multigrid relaxation methods. *ACM Trans. Math. Softw.*, 47(3), 2021.

[16] P. E. Farrell, L. Mitchell, L. R. Scott, and F. Wechsung. A Reynolds-robust preconditioner for the Scott-Vogelius discretization of the stationary incompressible Navier-Stokes equations. *The SMAI Journal of Computational Mathematics*, 7:75–96, 2021.

[17] P. E. Farrell, L. Mitchell, L. R. Scott, and F. Wechsung. Robust multigrid methods for nearly incompressible elasticity using macro elements. *IMA J. Numer. Anal.*, 42(4):3306–3329, 2022.

[18] B. Flemisch, A. Fumagalli, and A. Scotti. A Review of the XFEM-Based Approximation of Flow in Fractured Porous Media. In G. Ventura and E. Benvenuti, editors, *Advances in Discretization Methods: Discontinuities, Virtual Elements, Fictitious Domain Methods*, SEMA SIMAI Springer Series, pages 47–76. Springer International Publishing, Cham, 2016.

[19] G. Fu. A high-order HDG method for the Biot's consolidation model. *Comput. Math. Appl.*, 77(1):237–252, 2019.

[20] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier-Stokes Equations*, volume 5 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1986.

[21] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, volume 14 of *Springer Verlag Series in Comput. Math.* Springer, 01 1996.

[22] Y. He and S. P. MacLachlan. Local Fourier analysis for mixed finite-element methods for the Stokes equations. *Journal of Computational and Applied Mathematics*, 357:161–183, 2019.

[23] Q. Hong and J. Kraus. Parameter-robust stability of classical three-field formulation of Biot's consolidation model. *Electron. Trans. Numer. Anal.*, 48:202–226, 2018.

[24] X. Hu, L. Mu, and X. Ye. Weak Galerkin method for the Biot's consolidation model. *Computers & Mathematics with Applications*, 75(6):2017–2030, 2018.

[25] X. Hu, C. Rodrigo, F. J. Gaspar, and L. T. Zikatanov. A nonconforming finite element method for the Biot's consolidation model in poroelasticity. *J. Comput. Appl. Math.*, 310:143–154, 2017.

[26] V. John and L. Tobiska. Numerical performance of smoothers in coupled multigrid methods for the parallel solution of the incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 33(4):453–473, 2000.

[27] M. Larin and A. Reusken. A comparative study of efficient iterative solvers for generalized Stokes equations. *Numerical Linear Algebra with Applications*, 15(1):13–34, 2008.

[28] S. P. MacLachlan and C. W. Oosterlee. Local Fourier analysis for multigrid with overlapping smoothers applied to systems of PDEs. *Num. Lin. Alg. Appl.*, 18(4):751–774, 2011.

[29] M. A. Murad and A. F. D. Loula. Improved accuracy in finite element analysis of Biot's consolidation problem. *Comput. Methods Appl. Mech. Engrg.*, 95(3):359–382, 1992.

[30] A. Naumovich and F. J. Gaspar. On a multigrid solver for the three-dimensional Biot poroelasticity system in multilayered domains. *Comp. Vis. Sci.*, 11(2):77–87, 2008.

[31] C. Rodrigo, X. Hu, P. Ohm, J. H. Adler, F. J. Gaspar, and L. T. Zikatanov. New stabilized discretizations for poroelasticity and the Stokes' equations. *Computer Methods in Applied Mechanics and Engineering*, 341:467–484, 2018.

[32] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, second edition, 2003.

[33] L. M. Skvortsov. A fifth order implicit method for the numerical solution of differential-algebraic equations. *Comput. Math. Math. Phys.*, 55(6):962–968, 2015.

[34] K. Terzaghi. *Theoretical Soil Mechanics*. Wiley: New York, 1943.

[35] R. Wienands, F. J. Gaspar, F. J. Lisbona, and C. W. Oosterlee. An efficient multigrid solver based on distributive smoothing for poroelasticity equations. *Computing*, 73(2):99–119, 2004.

[36] S.-Y. Yi. Convergence analysis of a new mixed finite element method for Biot's consolidation model. *Num. Meth. Partial Diff. Eqns.*, 30(4):1189–1210, 2014.