Local Fourier Analysis for Saddle-Point Problems

by

© Yunhui He

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

Department of Mathematics and Statistics
Memorial University

August, 2018

St. John's, Newfoundland and Labrador, Canada

Abstract

The numerical solution of saddle-point problems has attracted considerable interest in recent years, due to their indefiniteness and often poor spectral properties that make efficient solution difficult. While much research already exists, developing efficient algorithms remains challenging. Researchers have applied finite-difference, finite-element, and finite-volume approaches successfully to discretize saddle-point problems, and block preconditioners and monolithic multigrid methods have been proposed for the resulting systems. However, there is still much to understand.

Magnetohydrodynamics (MHD) models the flow of a charged fluid, or plasma, in the presence of electromagnetic fields. Often, the discretization and linearization of MHD leads to a saddle-point system. We present vector-potential formulations of MHD and a theoretical analysis of the existence and uniqueness of solutions of both the continuum two-dimensional resistive MHD model and its discretization.

Local Fourier analysis (LFA) is a commonly used tool for the analysis of multigrid and other multilevel algorithms. We first adapt LFA to analyse the properties of multigrid methods for both finite-difference and finite-element discretizations of the Stokes equations, leading to saddle-point systems. Monolithic multigrid methods, based on distributive, Braess-Sarazin, and Uzawa relaxation are discussed. From this LFA, optimal parameters are proposed for these multigrid solvers. Numerical experiments are presented to validate our theoretical results. A modified two-level LFA is proposed for high-order finite-element methods for the Lapalce problem, curing the failure of classical LFA smoothing analysis in this setting and providing a reliable way to estimate actual multigrid performance. Finally, we extend LFA to analyze the balancing domain decomposition by constraints (BDDC) algorithm, using a new choice of basis for the space of Fourier harmonics that greatly simplifies the application of LFA. Improved performance is obtained for some two- and three-level variants.

To my grandmother

Lay summary

The study of numerical simulation is important in our world, since we cannot always obtain exact solutions to accurate mathematical models of many real-world phenomena. Numerical simulation has, thus, penetrated into many fields, including meteorology, fluid mechanics, the biomedical sciences, and so on. For a good numerical algorithm, we should consider the choice of relevant parameters, comparison of the cost and effectiveness among different algorithms, and parallelism. Our goal in this work is to better understand algorithmic performance for simulation of fluid models and to design efficient algorithms.

Much research on simulation needs mathematical theories and tools to help study the algorithms before applying them to general problems. With the aid of mathematical analysis, we can know properties of models and develop efficient algorithms. The focus of this thesis is on the validity and applicability of such an analysis tool. Recent work has reported failure of the existing tool in some cases, and we aim to make up for this failure.

Our work addresses several issues. We develop a theoretical analysis for a model of charged fluids to answer an open question about the existence and uniqueness of solutions of this model. For the design of good algorithms, we employ a mathematical tool to analyze and predict the actual performance of algorithms. In some cases, this tool gives a good prediction and, based on this tool, we optimize the parameters in the algorithms and obtain efficient performance. To address the failure of the analysis for some models, we propose a modified analysis, which obtains a reliable prediction and efficient performance. Furthermore, we build a framework of analysis to study a parallel algorithm. In this case, we better understand the existing algorithm and develop improved variants. The results and tools presented here may help us design efficient numerical algorithms for many other models.

Acknowledgements

Firstly, I am deeply grateful to my supervisor, Professor Dr. Scott MacLachlan, for his professional advice, financial support, patience, and passion, throughout my Ph.D. study. What he has done enriches my mind and broadens my view. He guides me in writing code step by step, corrects my research writing over and over again, and shares advanced work and books with me. He offers me a comfortable research environment: encourages me to try new directions, allows me to hold my ideas, and provides equal-debating discussions, creating a nutritious growth environment. He supports me in attending many conferences and workshops. These research communications excite and consolidate my passion for math, and I have more chances to talk with other professors and make friends. Out of research, he gives me many suggestions on my personal life. I must say that he sets a good example to me, showing me what a good supervisor is and what a good human is. He has a very positive effect on my life, so that I have a more pleasant time in St. John's that I have ever had anywhere before.

Special thanks go to my friend, Mengliu, who taught me English for one year for my English exams, so that I can come to Canada to study. The first time I came to Canada, I received her warm hospitality. She offered me many opportunities to adapt to life abroad, knowing Canadian culture, and overcoming the language barrier. Beyond that, she is always a good listener and advisor for me, sharing my happiness and pulling me through obstacles. She lifts me up in spirits and makes the dark and empty world suddenly seem bright and full. She is my eternal mentor.

I would like to single out for special acknowledgment Professor Dr. Hermann Brunner, who introduced me to my supervisor. I could not have imagined having a better supervisor for my Ph.D study.

I am grateful to my collaborators, James, Xiaozhe, and Jed. It is fantastic to have opportunities to work with them. James and Xiaozhe lead me to a new direction,

keeping me on my theoretical toes with their questions, which have substantially improved my analysis abilities. Jed gives me much great advice and insights on coding and graphic design to get results of better quality. Under their guidance, I can make progress in my research.

I am also indebted to Professors Qun Lin, Hehu Xie, and Hongwei Li, who give me much encouragement and suggestions in my research by emails and calls.

I would like to thank Professor Ronald Haynes, who is my mentor for the *Teaching Skills Enhancement Program.* He spent much time guiding me in how to design curriculum, mark assignments, and shared his teaching experience.

In the department, I would like to express my gratitude to Ros and Colleen, who have assistanted me in attending conferences, Leonce,Tara, Jennifer, and Kimberly, who arrange my teaching assistant work and study, Lawrence who offers computer technical support, J Concepción who gives advice on the thesis Latex, and the other staff. I have been blessed with cheerful colleagues, Hisham, Razan, and other nice people. I miss the interesting and long-lasting chats, and enjoyable life in this beautiful island, Newfoundland.

My sincere thanks also goes to the members of my committee for devoting some of their precious time to scrutinize my thesis, and their insightful comments and encouragement.

I would not have been able to complete this work without the financial support of the School of Graduate Studies, the Department of Mathematics and Statistics, and the Natural Sciences and Engineering Research Council.

Last but not the least, I am deeply grateful to my family members who have supported me throughout all my studies for more than twenty years. Without their support and understanding, I cannot move here.

I appreciate all that makes me happy!

Statement of contributions

The work represented in Chapter 3 is the result of collaborative research between James H. Adler, Yunhui He, Xiaozhe Hu and Scott MacLachlan, with its intellectual property equally co-owned by all. It was written by Yunhui He with the guidance of Scott MacLachlan.

The work represented in Chapters 4, 5, and 6, is the result of collaborative research between Yunhui He and Scott MacLachlan, with its intellectual property equally co-owned by both. It was written by Yunhui He with the guidance of Scott MacLachlan.

The work represented in Chapter 7 is the result of collaborative research between Jed Brown, Yunhui He and Scott MacLachlan, with its intellectual property equally co-owned by all. It was written by Yunhui He with the guidance of Scott MacLachlan.

Table of contents

List of tables

List of figures

Chapter 1

Introduction

Saddle-point problems naturally arise in fluid and solid mechanics. There is great current interest in developing fast and efficient linear solvers for saddle-point systems, since their indefiniteness and often poor spectral properties pose some difficulties in numerical computing. While numerical experiments and theoretical analysis of saddle-point problems have been well-studied in the literature, there is a need to understand different problems more precisely, such as magnetohydrodynamics [1], mixed finite element approximations of elliptic PDEs [11], constrained optimization [9, 14, 15, 24, 27], and optimal control [25, 28]. The main goal of this thesis is the development of local Fourier analysis (LFA) [29, 30] tools to understand the performance of multigrid methods for saddle-point systems [5, 11] and higher-order finite-element discretizations.

Magnetohydrodynamics (MHD) models the flow of a charged fluid, or plasma, in the presence of electromagnetic fields. There are many formulations of MHD, depending on the domain and physical parameters considered. Often, the discretization and linearization of MHD leads to a saddle-point system. The set of equations that describe MHD are a combination of the Navier-Stokes equations of fluid dynamics and Maxwell's equations of electro-magnetism. These differential equations must be solved simultaneously. The equations of stationary, incompressible single fluid MHD posed in three dimensions are considered in (for example) [13, 26]. Under some conditions on the data, the existence and uniqueness of solutions to weak formulations of the equations is known both in the continuum and for certain discretizations. When writing the magnetic field variable using a vector potential form, the divergence-free condition is automatically satisfied. Consequently, vector potential formulations for MHD substantially reduce the complexity of the resulting equations and allow flexibility in the finite-element approximation. Numerical results using the vector potential formulation already exist in the literature [1, 6]. However, these papers focus mainly on linear-algebraic aspects of the solution of the resulting linearized systems of equations. Until now, unfortunately, rigorous study of the existence and uniqueness of solutions are still lacking. In our work presented here, we demonstrate that standard analysis techniques can be extended from three-dimensional MHD [13, 26] to the two-dimensional discretizations considered in [1, 6].

Discretization of the Stokes equations naturally leads to a saddle-point system. Finite-difference, finite-volume, and finite-element discretization approaches have been studied in the literature. Considerable attention must be paid to avoid instability when choosing appropriate discretizations of the Stokes equations. For finite differences, the Marker-and-Cell (MAC) scheme is known to be suitable for the Stokes equations [29]. Thus, we consider this approach as one of our discretizations. Fast and efficient solvers for the resulting systems are needed, and it is necessary to employ some mathematical theories and tools to help us analyze the properties of the systems and design efficient algorithms. In the past several decades, local Fourier analysis (LFA) has attracted much attention as an analysis tool to quantitatively predict convergence properties of multigrid methods and multilevel algorithms. There is a large volume of published studies concentrating on LFA of different relaxation schemes for many problems. We extend this work to multigrid schemes of current interest for the solution of saddle-point systems.

Recent developments in LFA have investigated the validity of LFA, and several studies have found that the smoothing analysis of LFA fails to be a good predictor of true performance, especially for overlapping multiplicative relaxation for the Q_2-Q_1 (Taylor-Hood) approximation of the Stokes equations [19]. One natural question that needs to be asked is whether this failure is due to the relaxation scheme or the discretization itself. We investigate this here.

For the Stokes equations, block preconditioners and monolithic multigrid methods have been designed for the resulting saddle-point systems. Recently, several families of relaxation schemes, including distributive Gauss-Seidel, Braess-Sarazin, and Uzawa relaxation, have been further developed for monolithic multigrid methods for the Stokes equations and more complicated saddle-point systems. These methods have been shown to outperform block preconditioners in some cases (see, e.g., [2]). Thus, monolithic multigrid methods are attractive. However, we note that most existing research using LFA is based on (symmetric) Gauss-Seidel approaches, even for simple scalar problems. Distributive Gauss-Seidel is well-known for its high efficiency [22, 23], but Jacobi relaxation is simpler and cheaper. However, to our knowledge, there is no research on distributive Jacobi relaxation for other problems. Braess-Sarazin relaxation has been shown to generally outperform other relaxation schemes [1, 2, 3]; however, there is still much more to understand about this approach, and no LFA has been performed for Braess-Sarazin relaxation. A simple version of Braess-Sarazin relaxation is Uzawa, which is popular for its simplicity and easy implementation [10, 12, 20]. Thus, we investigate monolithic multigrid methods based on common block-structured relaxations, including distributive Jacobi, Braess-Sarazin, and Uzawa relaxation, using LFA. Considering modern parallelism, variants based on weighted Jacobi are examined.

Besides multigrid methods, domain decomposition methods are also very popular for large-scale problems, offering high efficiency and natural parallelism. Balancing domain decomposition by constraints (BDDC), a nonoverlapping domain decomposition method, has been successfully applied to many problems [4, 7, 8, 16, 17, 18, 21]. Although there exists some convergence analysis of BDDC based on finite-element theorems in the literature, no study of BDDC using LFA has been carried out. We build a framework suitable for the analysis of BDDC and analyze some two- and three-level variants of BDDC here.

This thesis makes several noteworthy contributions to our knowledge by addressing four important issues. Firstly, we offer a more rigorous understanding of the existence and uniqueness of solutions of the vector potential formulations of two-dimensional magnetohydrodynamics. Secondly, we apply LFA to analyze block-structured relaxations for the Stokes equations discretizated with the MAC scheme and finite-element methods, and obtain efficient multigrid methods. Thirdly, the study of higher-order finite-element discretizations adds substantially to our understanding of the failure of classical LFA smoothing analysis for some types of problems. Lastly, the study of BDDC contributes a fundamentally new approach to the LFA literature suitable for analysis of domain decomposition methods.

This thesis is organized as follows.

In Chapter 2, we review the the existing work on linear solvers for saddle-point problems, and the history of local Fourier analysis and its applications.

In Chapter 3, a vector-potential formulation is presented for electromagnetic problems in two dimensions. Existence and uniqueness are considered separately for the continuum nonlinear equations of magnetohydrodynamics. At the same time, the discretized and linearized form that arises from Newton's method applied to a modified system is discussed.

In Chapter 4, we discuss the performance of multigrid for the Stokes equations discretized by the MAC scheme. Distributive weighted Jacobi, Uzawa, and Braess-Sarazin relaxations are investigated. Local Fourier analysis is applied to these relaxation schemes to analyse the convergence behavior, and we compare the efficiency of multigrid methods based on these schemes.

Chapter 5 begins by examining higher-order finite-element approximations to the Laplace problem. A modified Fourier analysis is presented to evaluate the performance of weighted Jacobi relaxation and the related two-grid method. Two-grid and multigrid performance is presented to validate our theoretical results.

In Chapter 6, two stabilized $Q_1 - Q_1$ and the stable $Q_2 - Q_1$ discretizations are considered for the Stokes equations. Optimal smoothing factors for distributive and Braess-Sarazin relaxation for the stabilized discretizations are determined by LFA. Just as LFA fails to predict the convergence factor of multigrid for the Q_2 discretization of the Laplace problem, the same is true for the $Q_2 - Q_1$ discretization of the Stokes equations. Thus, we numerically optimize the two-grid convergence factor. Inexact variants, using a few steps of Jacobi or multigrid iterations on the Schur complement system for Braess-Sarazin relaxation, are investigated as well.

In Chapter 7, we extend local Fourier analysis to the balancing domain decomposition by constraints family of algorithms, one of the classes of nonoverlapping domain decomposition methods. In this LFA, we use a new basis for the Fourier space allowing us to simplify the analysis. Two- and three-level variants of BDDC methods are proposed. Quantitative estimates of condition numbers of the resulting preconditioned operators are given by local Fourier analysis.

In Chapter 8, some conclusions are drawn and some potential projects for future work are discussed.

Bibliography

[1] J. Adler, T. R. Benson, E. Cyr, S. P. MacLachlan, and R. S. Tuminaro. Monolithic multigrid methods for two-dimensional resistive magnetohydrodynamics. *SIAM J. Sci. Comput.*, 38(1):B1–B24, 2016.

[2] 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.

[3] J. H. Adler, D. B. Emerson, S. P. MacLachlan, and T. A. Manteuffel. Constrained optimization for liquid crystal equilibria. *SIAM J. Sci. Comput.*, 38(1):B50–B76, 2016.

[4] L. Beirão da Veiga, D. Cho, L. F. Pavarino, and S. Scacchi. BDDC preconditioners for isogeometric analysis. *Math. Models Methods Appl. Sci.*, 23(6):1099–1142, 2013.

[5] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numer.*, 14:1–137, 2005.

[6] E. C. Cyr, J. N. Shadid, R. S. Tuminaro, R. P. Pawlowski, and L. Chacón. A new approximate block factorization preconditioner for two-dimensional incompressible (reduced) resistive MHD. *SIAM Journal on Scientific Computing*, 35(3):B701–B730, 2013.

[7] C. R. Dohrmann. An approximate BDDC preconditioner. *Numerical Linear Algebra with Applications*, 14(2):149–168, 2007.

[8] C. R. Dohrmann and O. B. Widlund. Some recent tools and a BDDC algorithm for 3D problems in $H(\mathrm{curl})$. In *Domain Decomposition Methods in Science and Engineering XX*, volume 91 of *Lect. Notes Comput. Sci. Eng.*, pages 15–25. Springer, Heidelberg, 2013.

[9] H. S. Dollar. *Iterative linear algebra for constrained optimization.* PhD thesis, University of Oxford, 2005.

[10] H. C. Elman and G. H. Golub. Inexact and preconditioned Uzawa algorithms for saddle point problems. *SIAM J. Numer. Anal.*, 31(6):1645–1661, 1994.

[11] 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.

[12] F. J. Gaspar, Y. Notay, C. W. Oosterlee, and C. Rodrigo. A simple and efficient segregated smoother for the discrete Stokes equations. *SIAM J. Sci. Comput.*, 36(3):A1187–A1206, 2014.

[13] M. D. Gunzburger, A. J. Meir, and J. S. Peterson. On the existence, uniqueness, and finite element approximation of solutions of the equations of stationary, incompressible magnetohydrodynamics. *Mathematics of Computation*, 56(194):523–563, 1991.

[14] M. Kollmann. *Efficient iterative solvers for saddle point systems arising in PDE-constrained optimization problems with inequality constraints*. PhD thesis, Johannes Kepler University Linz, 2013.

[15] M. Kollmann and W. Zulehner. A robust preconditioner for distributed optimal control for Stokes flow with control constraints. In *Numerical Mathematics and Advanced Applications 2011*, pages 771–779. Springer, Heidelberg, 2013.

[16] J. Li and O. Widlund. BDDC algorithms for incompressible Stokes equations. *SIAM J. Numer. Anal.*, 44(6):2432–2455, 2006.

[17] J. Li and O. Widlund. A BDDC preconditioner for saddle point problems. In *Domain Decomposition Methods in Science and Engineering XVI*, volume 55 of *Lect. Notes Comput. Sci. Eng.*, pages 413–420. Springer, Berlin, 2007.

[18] J. Li and O. Widlund. On the use of inexact subdomain solvers for BDDC algorithms. *Computer Methods in Applied Mechanics and Engineering*, 196(8):1415–1428, 2007.

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

[20] J.-F. Maitre, F. Musy, and P. Nigon. A fast solver for the Stokes equations using multigrid with a Uzawa smoother. In *Advances in multigrid methods (Oberwolfach, 1984)*, volume 11 of *Notes Numer. Fluid Mech.*, pages 77–83. Vieweg, Braunschweig, 1985.

[21] J. Mandel and C. R. Dohrmann. Convergence of a balancing domain decomposition by constraints and energy minimization. *Numerical Linear Algebra with Applications*, 10(7):639–659, 2003.

[22] A. Niestegge and K. Witsch. Analysis of a multigrid Stokes solver. *Appl. Math. Comput.*, 35(3):291–303, 1990.

[23] C. W. Oosterlee and F. J. Gaspar. Multigrid methods for the Stokes system. *Computing in Science & Engineering*, 8(6):34–43, 2006.

[24] T. Rees, H. S. Dollar, and A. J. Wathen. Optimal solvers for PDE-constrained optimization. *SIAM Journal on Scientific Computing*, 32(1):271–298, 2010.

[25] J. Schöberl, R. Simon, and W. Zulehner. A robust multigrid method for elliptic optimal control problems. *SIAM Journal on Numerical Analysis*, 49(4):1482–1503, 2011.

[26] D. Schötzau. Mixed finite element methods for stationary incompressible magneto–hydrodynamics. *Numerische Mathematik*, 96(4):771–800, 2004.

[27] R. Simon. *Multigrid solvers for saddle point problems in PDE-constrained optimization.* PhD thesis, Johannes Kepler University Linz, 2008.

[28] S. Takacs. *All-at-once multigrid methods for optimality systems arising from optimal control problems.* PhD thesis, Johannes Kepler University Linz, 2012.

[29] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid.* Academic Press, Inc., San Diego, CA, 2001. With contributions by A. Brandt, P. Oswald and K. Stüben.

[30] R. Wienands and W. Joppich. *Practical Fourier analysis for multigrid methods.* CRC press, 2004.

Chapter 2

Background

In recent decades, saddle-point problems have arisen in a wide variety of applications throughout computational science and engineering, and have formed one of the most significant topics of research in scientific computing. The substantial challenges for solving these problems arise due to their indefiniteness and often poor spectral properties. A considerable amount of research has been carried out on the numerical analysis of the saddle-point problems, see [6, 8, 31] and the references therein.

2.1 Saddle-point systems

Here, we first introduce some common problems in fluid dynamics. The Navier-Stokes (Stokes) equations are a common saddle-point problem. In the literature, both analytical and numerical aspects of the solution of the Navier-Stokes or Stokes equations for viscous incompressible fluids have been considered [4, 31, 41, 73, 80]. However, the analysis of the existence and uniqueness of the solution in linear and nonlinear, steady and time-dependent cases is still difficult and receives much attention. We introduce the stationary incompressible Navier-Stokes equations here. Let Ω be a Lipschitz, bounded open set in $\mathbb{R}^n$ with boundary Γ, and $\vec{f} \in (L^2(\Omega))^n$ be a given vector function. The stationary incompressible Navier-Stokes equations [4, 73, 75] are to find a fluid velocity $\vec{u} = (u_1, u_2, \cdots, u_n)$ and pressure p such that

$$\begin{cases} -\Delta\vec{u} + Re\left(\displaystyle\sum_{i=1}^{n} u_i D_i \vec{u} + \nabla p\right) = \vec{f}, \text{ in } \Omega \\ \nabla \cdot \vec{u} = 0, \text{ in } \Omega \\ \vec{u} = 0, \text{ on } \Gamma \end{cases} \tag{2.1}$$

where Re is the Reynolds number, which is proportional to a characteristic velocity, a characteristic length, and the reciprocal of fluid viscosity, and D_i is the differentiation operator defined as $D_i = \frac{\partial}{\partial x_i}$. We call the first equation in (2.1) the momentum equation, since it arises from the physical principle of conservation of momentum. The second equation in (2.1) is called the continuity equation for steady flow, which states that the rate at which mass enters a system is equal to the rate at which mass leaves the system.

Remark 2.1.1. *Here, we consider the vector Laplacian for the diffusion term in the first equation of (2.1), consistent with the Dirichlet boundary condition given. For more general models and/or boundary conditions, this could be replaced with the divergence of the symmetric part of the gradient, as will be considered in Chapter 3.*

Challenges in analyzing the existence and uniqueness of solutions of the Navier-Stokes equations arise in many aspects. For example, some technical difficulties arise in applying Sobolev inequalities and dealing with the nonlinear terms. The treatment of the equations heavily depends on the dimension of the problems considered, and non-uniqueness of solutions also happens. Uniqueness results are known only when the data, for example, Re, are small enough, or the viscosity is large enough. For more details, we refer to [22, 73].

A special case of fluid dynamics is the Stokes equations, which describe highly viscous incompressible flows characterized by the diffusion term in the momentum equation. The Stokes equations are linear, and simpler than the incompressible Navier-Stokes equations. We consider the Stokes equations as follows

$$\begin{cases} -\varepsilon\Delta\vec{u} + \nabla p = \vec{f}, \text{ in } \Omega \\ \nabla \cdot \vec{u} = 0, \text{ in } \Omega \\ \vec{u} = 0, \text{ on } \Gamma \end{cases}$$

where ε is the fluid viscosity.

Remark 2.1.2. *The Stokes equations considered here are a limiting(Re $\rightarrow 0$) case of the Navier-Stokes equations in (2.1) with a proper scaling of the pressure term.*

The discretizations of the above models by finite-element, finite-difference, and finite-volume methods for the unknown variables lead to some difficulty. For high Reynolds number in the Navier-Stokes equations, the momentum equation is singularly perturbed and the h-ellipticity measure [75] of the standard (central) discretization schemes decreases. For the Stokes equations, equal-order finite element methods cannot be used, since the required stability condition is not satisfied in this situation. Instability also arises when using central differencing of the first-order derivatives in the pressure term and continuity equation, if all variables are located at the grid points.

To overcome this instability, the Marker-and-Cell (MAC) scheme was first established by Harlow and Welch [78]. In the MAC scheme, the velocity unknowns are located at the midpoints of the x- and y-edges of the mesh, and the pressure is located at the cell centres. The MAC scheme has been successfully adapted and applied to many PDEs. Another option is to add additional “artificial viscosity” or “artificial pressure” terms to keep the discrete equations stable [75]. A semi-implicit method was designed by Chorin [20, 21], where an artificial compressibility was introduced to the continuity equation. One of our interests in this work is to analyze solution of discretizations using the MAC scheme.

Using higher-order finite-element methods is another option to avoid instability. Taylor-Hood elements ($Q_2 - Q_1$, $P_2 - P_1$) have been applied to the Stokes equations. We consider the $Q_2 - Q_1$ approximation and solvers for the resulting systems. Other stabilized finite-element methods are also well-studied, for example, in [31].

The momentum equation is associated with $\vec{u}$ and p, but p is not present in the continuity equation. This leads to complications in the discretization and in the numerical treatment. For well-posedness of the discrete system, the discretization of the velocity and pressure unknowns should satisfy an inf-sup stability [31]

$$\inf_{q_h \neq 0} \sup_{\vec{v}_h \neq \vec{0}} \frac{|(\nabla q_h, \vec{v}_h)|}{\|\vec{v}_h\|_1 \|q_h\|_0} \geq \gamma > 0,$$

where γ is a constant.

A more complicated example of fluids is magnetohydrodynamics (MHD). MHD is the study of the properties of electrically conducting fluids, and has many valuable applications, including in modelling plasma confinement, in astrophysics, aerospace engineering, and so on. Based on the MHD equations, for example, scientists have made a supercomputer model of the Earth's interior. Often, MHD is modelled using a combination of the Navier-Stokes equations of fluid dynamics and Maxwell's equations of electro-magnetism. We consider the one-fluid visco-resistive MHD model, where the dependent variables are the fluid velocity, $\vec{u}$, the hydrodynamic pressure, p, and the magnetic field $\vec{B}$. The equations are

$$\begin{aligned} \frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} - \nabla \cdot (T + T_M) + \nabla p &= \vec{F}, \qquad (2.2) \\ \frac{\partial \vec{B}}{\partial t} - \nabla \times (\vec{u} \times \vec{B}) + \nabla \times (\frac{1}{Re_m} \nabla \times \vec{B}) &= \vec{G}, \qquad (2.3) \\ \nabla \cdot \vec{u} &= 0, \qquad (2.4) \\ \nabla \cdot \vec{B} &= 0, \qquad (2.5) \end{aligned}$$

where $\vec{G} = -\nabla \times \vec{E}_{\text{stat}}$, and $\vec{E}_{\text{stat}}$ is the static component of the electric field. The Newtonian and magnetic stress tensors are given by

$$T = \frac{1}{2Re}\left[\nabla \vec{u} + \nabla \vec{u}^T\right], \text{ and } T_M = \vec{B} \otimes \vec{B} - \frac{1}{2}|\vec{B}|^2 I,$$

respectively.

The stationary, incompressible MHD model in three-dimensions has been discussed in [44], where the existence and uniqueness of solutions of the continuous and approximate problems are guaranteed under some conditions on the data. A new mixed variational formulation of MHD is presented in [70], where standard inf-sup stable velocity-pressure pairs are used for the hydrodynamic unknowns, and a mixed approach using Nédélec elements is used for the magnetic variables. In this model, there is another scalar variable, r, included in (2.3) as ∇r. In that case, if $\nabla \cdot \vec{G}$ is zero, it can be shown that $r = 0$, yielding equivalence to the standard MHD model or the linearized version in [40], where the author focuses on the stabilized finite-element method for MHD. In recent years, numerical experiments using vector potential formulations of (2.2)-(2.5) have been discussed in [1, 23]. However, there has been no attempt to explore the existence and uniqueness of solutions of the formulations. Noting that the discretizations of MHD lead to saddle-point systems, it is necessary to better understand the properties of either the continuum formulation or the discrete formulations of MHD before designing good algorithms. In Chapter 3, we extend the tools of [44, 70] to prove the existence and uniqueness of solutions of vector potential formulations of MHD in two dimensions.

2.2 Multigrid preliminaries

For a general nonsingular linear system, $Ku = b$, we can consider finding its solution exactly or inexactly. With increasing problem size, it is often a challenge to solve the linear-algebraic equation exactly. Thus, iterative methods are used to find approximate solutions. In the literature, many iterative methods have been well studied [31, 43, 59, 68, 72], including stationary iterative methods, Krylov subspace methods, and multigrid methods. The idea of a stationary iteration is to find an approximation, M, to K that can be inverted easily, then compute the approximate solution via the iteration

$$u^{j+1} = u^j + M^{-1}(b - Ku^j),$$

or

$$u^{j+1} = (I - M^{-1}K)u^j + M^{-1}b. \tag{2.6}$$

The matrix $\mathcal{S} := I - M^{-1}K$ is called the iteration matrix. If $\rho(\mathcal{S}) := \max|\lambda(\mathcal{S})| < 1$, then (2.6) is said to be convergent. Often, we choose M to be the diagonal part of K (Jacobi iteration), the lower triangle part of K (Gauss-Seidel iteration), or a scalar multiple of the identity (Richardson iteration).

Many classical iterative methods (for example, Jacobi) appropriately applied to discrete problems have poor convergence but a strong “smoothing” effect on the error in any approximation. That is, the schemes can reduce high frequency error components quickly, but are slow to reduce low-frequency errors. Because of this, we call such schemes “relaxation” methods. Based on this smoothing property, we can construct a coarse grid, where the low frequencies on the fine grid can be treated as relatively high frequencies, so the smooth error can be approximated on the coarse grid, which is simple, compared with the fine-grid problem. This leads to two-grid and multigrid methods.

Multigrid methods [10, 16, 71, 75, 79, 81] have been successfully applied to saddle-point problems either as standalone iterative solvers or as preconditioners, due to their high efficiency. Precisely, multigrid offers the possibility of solving problems with N unknowns with $O(N)$ work and storage for large classes of problems. In the literature, there are two families of multigrid methods, geometric multigrid [71, 79] and algebraic multigrid [16, 67, 68, 75]. In this thesis, we focus on geometric multigrid methods. Assume we have two meshes, with fine-grid meshsize h and coarse-grid meshsize H (often, $H = 2h$, by doubling the meshsize in each spatial direction). A two-grid algorithm is as follows,

Algorithm 2.2.1. *Two-grid method:* $u_h^{j+1} = \mathbf{TGAlg}(K_h, b_h, u_h^j, \nu_1, \nu_2)$

1. *Pre-smoothing: Applying* ν_1 *sweeps of relaxation to* u_h^j*:*

$$\bar{u}_h^j = \mathbf{Smoothing}^{\nu_1}(u_h^j, K_h, b_h). \tag{2.7}$$

2. *Coarse grid correction (CGC):*

   - *Compute the residual:* $r_h = b_h - K_h \bar{u}_h^j$*;*
   - *Restrict the residual:* $r_H = R_h r_h$*;*
   - *Solve the coarse-grid problem:* $K_H^* u_H = r_H$*;*
   - *Interpolate the correction:* $\delta u_h = P_h u_H$*;*
   - *Update the corrected approximation:* $\hat{u}_h^j = \bar{u}_h^j + \delta u_h$*;*

3. *Post-smoothing: Applying* ν_2 *sweeps of relaxation to* $\hat{u}_h^j$*,*

$$u_h^{j+1} = \mathbf{Smoothing}^{\nu_2}(\hat{u}_h^j, K_h, b_h) \tag{2.8}$$

Applying this two-grid method, the two-grid error propagation operator is

$$M^{\text{TGM}} = \mathcal{S}_h^{\nu_2}(I - P_h(K_H^*)^{-1} R_h K_h)\mathcal{S}_h^{\nu_1}, \tag{2.9}$$

where $\mathcal{S}_h = I - M_h^{-1} K_h$ is the error propagation operator for relaxation.

From the above discussion, the important components in a two-grid algorithm are:

- The smoothing procedure: $\bar{u}_h = \mathbf{Smoothing}^{\nu}(*, K_h, b_h)$;

- The fine-to-coarse restriction operator: R_h;
- The coarse-grid operator: K_H^*;
- The coarse-to-fine interpolation operator: P_h;

For the pre- and post-smoothing relaxation, Jacobi, Gauss-Seidel, and Richardson relaxation can all be used. Usually, we use the same relaxation for both pre- and post-smoothing, but it can be different. For the restriction operator, R_h, there are many choices, which depend on the problems. Here, we focus on choices of R_h tied to the mesh and the particular discretization scheme used to generate K_h. K_H^* can be the Galerkin operator, $K_H^* = R_h K_h P_h$, or the natural rediscretization operator, K_H, and $I - P_h(K_H^*)^{-1} R_h K_h$ is called the coarse-grid correction operator. The interpolation operator, P_h, is usually taken to be the conjugate transpose of R_h, with scaling depending on the discretization scheme and the dimension of the considered problem. For more details on the choice of multigrid components, we refer to [71, 75, 79].

If we solve the coarse-grid problem recursively by the two-grid method, then we obtain a multigrid method. Over the past decades, a variety of types of multigrid methods have been developed, including W, V, and F-cycles [71].

The choice of the components of multigrid methods, such as coarse-grid correction, prolongation, restriction, and relaxation schemes, is very crucial to design efficient algorithms. A well-developed tool, local Fourier analysis (LFA), can aid proper choice of these multigrid components in many cases. Thus, it is worth investigating and understanding how to apply LFA to different problems and when it can be an effective tool.

2.3 LFA preliminaries

Local Fourier analysis was first introduced by Brandt in [13], where the smoothing factor is presented as a good predictor for multigrid performance. The principal advantage of LFA is that it provides realistic quantitative estimates of the asymptotic multigrid convergence factor of some model problems and classes of relaxation schemes and multigrid algorithms. Two-grid local Fourier analysis [75] contains analysis of two parts: the relaxation scheme and the coarse-grid correction. Here, we present a brief introduction to LFA.

In order to describe LFA, we consider d-dimensional infinite uniform grids, $\mathbf{G}_h$, as follows,

$$\mathbf{G}_h = \Big\{ \boldsymbol{x} := (x_1, x_2, \ldots, x_d) = \boldsymbol{k}h = (k_1, k_2, \ldots, k_d)h, k_i \in \mathbb{Z} \Big\}, \tag{2.10}$$

and Fourier modes $\varphi(\boldsymbol{\theta}, \boldsymbol{x}) = e^{\imath \boldsymbol{\theta} \cdot \boldsymbol{x}/h}$ on $\mathbf{G}_h$, where $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_d)$ and $\imath^2 = -1$. Because $\varphi(\boldsymbol{\theta}, \boldsymbol{x})$ is periodic in $\boldsymbol{\theta}$ with period 2π, we consider θ_i to vary continuously in the interval $\big(-\frac{\pi}{2}, \frac{3\pi}{2}\big]$ (or any interval with length 2π). The coarse grid $\mathbf{G}_{2h}$ is defined similarly.

Remark 2.3.1. *In practical use, the grids might be more complicated than (2.10). However, LFA can be modified to adapt to the corresponding discretizations, as it will be later in this thesis.*

Let L_h be a scalar Toeplitz operator acting on $l^2(\mathbf{G}_h)$ as follows,

$$\begin{aligned} L_h &\overset{\wedge}{=} [s_{\boldsymbol{\kappa}}]_h \ (\boldsymbol{\kappa} = (\kappa_1, \kappa_2, \cdots, d) \in \mathbb{Z}^d); \\ L_h w_h(\boldsymbol{x}) &= \sum_{\boldsymbol{\kappa} \in \boldsymbol{V}} s_{\boldsymbol{\kappa}} w_h(\boldsymbol{x} + \boldsymbol{\kappa} h), \end{aligned}$$

with constant coefficients $s_{\boldsymbol{\kappa}} \in \mathbb{R}$ (or $\mathbb{C}$), where $w_h(\boldsymbol{x})$ is a function in $l^2(\mathbf{G}_h)$. Here, $\boldsymbol{V}$ is taken to be a finite index set. Note that because L_h is Toeplitz, it is diagonalized by the Fourier modes $\varphi(\boldsymbol{\theta}, \boldsymbol{x})$.

A general 2D stencil can be written as

$$[s_{\boldsymbol{\kappa}}]_h = \begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & s_{-1,1} & s_{0,1} & s_{1,1} & \cdots \\ \cdots & s_{-1,0} & s_{0,0} & s_{1,0} & \cdots \\ \cdots & s_{-1,-1} & s_{0,-1} & s_{1,-1} & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix}.$$

If we consider the 2D Laplace problem using the 5-point finite-difference approximation, then the stencil of $L_h = -\Delta_h$ is

$$\frac{1}{h^2} \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}, \text{ denoted as } \begin{bmatrix} & s_{0,1} & \\ s_{-1,0} & s_{0,0} & s_{1,0} \\ & s_{0,-1} & \end{bmatrix}. \tag{2.11}$$

Definition 2.3.1. *We call* $\widetilde{L}_h(\boldsymbol{\theta}) = \sum\limits_{\boldsymbol{\kappa}\in\boldsymbol{V}} s_{\boldsymbol{\kappa}} e^{i\boldsymbol{\theta}\cdot\boldsymbol{\kappa}}$ *the symbol of* L_h*.*

Note that for all grid functions $\varphi(\boldsymbol{\theta}, \boldsymbol{x})$,

$$L_h\varphi(\boldsymbol{\theta}, \boldsymbol{x}) = \widetilde{L}_h(\boldsymbol{\theta})\varphi(\boldsymbol{\theta}, \boldsymbol{x}).$$

We note that the symbol of the stencil L_h is closely related to the standard definition of the symbol of a differential operator. By standard calculation, the symbol of stencil $-\Delta_h$, defined in (2.11), is given by $\widetilde{L}_h(\theta_1, \theta_2) = \frac{4-2\cos\theta_1 - 2\cos\theta_2}{h^2}$.

For multigrid methods, we construct a sequence of coarse grids by doubling the mesh size in each spatial direction. High and low frequencies for standard coarsening are given by

$$\boldsymbol{\theta} \in T^{\text{low}} = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^d, \ \boldsymbol{\theta} \in T^{\text{high}} = \left[-\frac{\pi}{2}, \frac{3\pi}{2}\right)^d \bigg\backslash \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^d.$$

It is easy to check that

$$\varphi(\boldsymbol{\theta}, \boldsymbol{x}) \equiv \varphi(\boldsymbol{\theta}', \boldsymbol{x}) \ \text{ for } \boldsymbol{x} \in \mathbf{G}_{2h}, \ \text{ iff } \ \boldsymbol{\theta} = \boldsymbol{\theta}' (\text{mod } \pi).$$

We define 2^d-dimensional spaces of harmonics over $\boldsymbol{\theta} \in (-\frac{\pi}{2}, \frac{\pi}{2}]^d$ as

$$\mathbf{E}_h(\boldsymbol{\theta}) = \text{span}\big\{\varphi_h(\boldsymbol{\theta}^{\boldsymbol{\xi}}, *) : \boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_d), \xi_j \in \{0, 1\}\big\}, \tag{2.12}$$

with

$$\boldsymbol{\theta}^{\boldsymbol{\xi}} = \boldsymbol{\theta} + \boldsymbol{\xi}\pi.$$

Definition 2.3.2. *The error-propagation symbol,* $\widetilde{\mathcal{S}}_h(\boldsymbol{\theta})$*, for smoother* $\mathcal{S}_h$ *on the infinite grid* $\mathbf{G}_h$ *satisfies*

$$\mathcal{S}_h\varphi(\boldsymbol{\theta}, \boldsymbol{x}) = \widetilde{\mathcal{S}}_h\varphi(\boldsymbol{\theta}, \boldsymbol{x}), \ \boldsymbol{\theta} \in \big[-\frac{\pi}{2}, \frac{3\pi}{2}\big)^d,$$

for all $\varphi(\boldsymbol{\theta}, \boldsymbol{x})$*, and the corresponding smoothing factor for* $\mathcal{S}_h$ *is given by*

$$\mu_{\text{loc}} = \mu_{\text{loc}}(\mathcal{S}_h) = \max_{\boldsymbol{\theta}\in T^{\text{high}}} \big\{\big|\widetilde{\mathcal{S}}_h(\boldsymbol{\theta})\big| \ \big\}. \tag{2.13}$$

Remark 2.3.2. *If* L_h *is not a scalar operator, then* $\big|\widetilde{\mathcal{S}}_h(\boldsymbol{\theta})\big|$ *in (2.13) can be modified* *to be* $\left|\lambda\big(\widetilde{\mathcal{S}}_h(\boldsymbol{\theta})\big)\right|$, *taking the absolute value of the eigenvalues of* $\widetilde{\mathcal{S}}_h(\boldsymbol{\theta})$.

The error-propagation symbol for a relaxation scheme, represented by matrix M_h, applied to L_h is

$$\widetilde{\mathcal{S}}_h(\boldsymbol{p},\omega,\boldsymbol{\theta}) = I - \omega \widetilde{M}_h^{-1}(\boldsymbol{\theta})\widetilde{L}_h(\boldsymbol{\theta}),$$

where $\boldsymbol{p}$ represents parameters within M_h, the approximation to L_h, ω is an overall weighting factor, and $\widetilde{M}_h$ and $\widetilde{L}_h$ are the symbols for M_h and L_h, respectively. Denote $\bar{\boldsymbol{p}} = (\boldsymbol{p},\omega)$.

For the 2D Laplace problem, we consider the Jacobi relaxation, where M_h is given by

$$M_h = \frac{1}{h^2}\begin{bmatrix} & 0 & \\ 0 & 4 & 0 \\ & 0 & \end{bmatrix},$$

with its symbol $\widetilde{M}_h(\theta_1,\theta_2) = \frac{4}{h^2}$.

Then, the error propagation symbol of weighted Jacobi relaxation for the Laplace problem is

$$\widetilde{\mathcal{S}}_h(\omega,\boldsymbol{\theta}) = 1 - \omega\frac{4 - 2\cos\theta_1 - 2\cos\theta_2}{4}.$$

According to (2.13), we have

$$\mu_{\text{loc}} = \max\left\{|1-2\omega|, |1-\frac{\omega}{2}|\right\}, \tag{2.14}$$

since the maximum and minimum values of $4 - 2\cos\theta_1 - 2\cos\theta_2$ are achieved at $(\theta_1,\theta_2) = (\pi,\pi)$, and $(\theta_1,\theta_2) = (0,\frac{\pi}{2})$ or $(\frac{\pi}{2},0)$, respectively.

Because μ_{loc}, defined in (2.13), is a function of $\bar{\boldsymbol{p}}$, the following is a natural question: how can we optimize the parameters in (2.13) to obtain the most efficient performance? This is one of the central topics in our work presented here.

Definition 2.3.3. *The optimal smoothing factor over* $\mathcal{D}$ *is defined as*

$$\mu_{\text{opt}} = \min_{\mathcal{D}} \mu_{\text{loc}}, \tag{2.15}$$

where $\mathcal{D}$ *is a bounded and closed set of allowable parameters.*

Now, optimizing (2.14), we obtain

$$\mu_{\text{opt}} = \frac{3}{5}, \text{ if and only if } \omega = \frac{4}{5}. \tag{2.16}$$

2.3.1 Two-grid LFA

Here, we introduce the two-grid LFA. For simplicity, we consider $d = 2$ (other cases are similar). We use the ordering of $\boldsymbol{\alpha} = (0,0), (1,0), (0,1), (1,1)$ for the four harmonics. Given any $\boldsymbol{\theta}^{00}$, (2.12) can be written as the following 4-dimensional subspaces

$$\mathbf{E}_h(\boldsymbol{\theta}) = \text{span}\{\varphi(\boldsymbol{\theta}^{00}, *), \varphi(\boldsymbol{\theta}^{10}, *), \varphi(\boldsymbol{\theta}^{10}, *), \varphi(\boldsymbol{\theta}^{11}, *)\}.$$

We consider applying LFA to the two-grid operator,

$$\boldsymbol{M}_h^{\text{TGM}} = \mathcal{S}_h^{\nu_2} \mathcal{M}_h^{\text{CGC}} \mathcal{S}_h^{\nu_1}, \tag{2.17}$$

with CGC operator,

$$\mathcal{M}_h^{\text{CGC}} = I - P_h (L_H^*)^{-1} R_h L_h,$$

where L_H^* is the coarse-grid operator. Assume that $L_h, R_h, P_h, \mathcal{S}_h$, and L_H^* are represented by stencils on $\mathbf{G}_h$ and $\mathbf{G}_{2h}$. Then, $\mathbf{E}_h(\boldsymbol{\theta})$ is invariant under the two-grid operator, $\boldsymbol{M}_h^{\text{TGM}}$.

To derive symbols for the grid-transfer operators, we first consider an arbitrary restriction operator characterized by a constant coefficient stencil $R_h \overset{\wedge}{=} [r_{\boldsymbol{\kappa}}]$. Then, an infinite grid function $w_h : \mathbf{G}_h \to \mathbb{R}$ (or $\mathbb{C}$) is transferred to the coarse grid, $\mathbf{G}_{2h}$, in the following way

$$(R_h w_h)(\boldsymbol{x}) \;\; = \;\; \sum_{\boldsymbol{\kappa} \in V} r_{\boldsymbol{\kappa}} w_h(\boldsymbol{x} + \boldsymbol{\kappa} h) \;\; (\boldsymbol{x} \in \mathbf{G}_{2h}).$$

Definition 2.3.4. *We call* $\widetilde{R}_h(\boldsymbol{\theta}^{\boldsymbol{\alpha}}) = \sum_{\boldsymbol{\kappa} \in V} r_{\boldsymbol{\kappa}} e^{\iota \boldsymbol{\kappa} \cdot \boldsymbol{\theta}^{\boldsymbol{\alpha}}}$ *the restriction symbol of* R_h.

Inserting the representations of $\mathcal{S}_h, L_h, L_H^*, P_h, R_h$ into (2.17), we obtain the Fourier representation of two-grid error-propagation operator as

$$\widetilde{\boldsymbol{M}}_h^{\text{TGM}}(\boldsymbol{\theta}) = \widetilde{\boldsymbol{S}}_h^{\nu_2}(\boldsymbol{\theta}) \big(I - \widetilde{\boldsymbol{P}}_h(\boldsymbol{\theta}) (\widetilde{L}_H^*(2\boldsymbol{\theta}))^{-1} \widetilde{\boldsymbol{R}}_h(\boldsymbol{\theta}) \widetilde{\boldsymbol{L}}_h(\boldsymbol{\theta}) \big) \widetilde{\boldsymbol{S}}_h^{\nu_1}(\boldsymbol{\theta}),$$

where

$$\begin{aligned}
\widetilde{\boldsymbol{L}}_h(\boldsymbol{\theta}) &= \operatorname{diag}\left\{\widetilde{L}_h(\boldsymbol{\theta}^{00}), \widetilde{L}_h(\boldsymbol{\theta}^{10}), \widetilde{L}_h(\boldsymbol{\theta}^{01}), \widetilde{L}_h(\boldsymbol{\theta}^{11})\right\},\\
\widetilde{\boldsymbol{S}}_h(\boldsymbol{\theta}) &= \operatorname{diag}\left\{\widetilde{\mathcal{S}}_h(\boldsymbol{\theta}^{00}), \widetilde{\mathcal{S}}_h(\boldsymbol{\theta}^{10}), \widetilde{\mathcal{S}}_h(\boldsymbol{\theta}^{01}), \widetilde{\mathcal{S}}_h(\boldsymbol{\theta}^{11})\right\},\\
\widetilde{\boldsymbol{P}}_h(\boldsymbol{\theta}) &= \left(\widetilde{P}_h(\boldsymbol{\theta}^{00}); \widetilde{P}_h(\boldsymbol{\theta}^{10}); \widetilde{P}_h(\boldsymbol{\theta}^{01}); \widetilde{P}_h(\boldsymbol{\theta}^{11})\right),\\
\widetilde{\boldsymbol{R}}_h(\boldsymbol{\theta}) &= \left(\widetilde{R}_h(\boldsymbol{\theta}^{00}), \widetilde{R}_h(\boldsymbol{\theta}^{10}), \widetilde{R}_h(\boldsymbol{\theta}^{01}), \widetilde{R}_h(\boldsymbol{\theta}^{11})\right),
\end{aligned}$$

in which $\operatorname{diag}\{T_1, T_2, T_3, T_4\}$ stands for the block diagonal matrix with diagonal blocks, T_1, T_2, T_3, and T_4.

Remark 2.3.3. *Considering general dimensions d, the above block matrices will have 2^d blocks.*

Definition 2.3.5. *The asymptotic two-grid convergence factor, ρ_{asp}, is defined as*

$$\rho_{\text{asp}} = \sup\{\rho(\widetilde{\boldsymbol{M}_h(\boldsymbol{\theta})}^{\text{TGM}}) : \boldsymbol{\theta} \in T^{\text{low}}\}. \tag{2.18}$$

For practical use, we usual consider a discrete form of ρ_{asp}, denoted by ρ_h, resulting from sampling ρ_{asp} over only finite set of frequencies.

Now, consider the two-grid LFA for the 2D Laplace problem. Here, we use $L_H^* = R_h L_h P_h$, where R_h is the full weighted (FW) restriction operator given by

$$R_h = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix},$$

with its symbol $\widetilde{R}_h(\theta_1, \theta_2) = \frac{1}{4}(1 + \cos\theta_1)(1 + \cos\theta_2)$, and P_h is taken to be the bilinear interpolation as follows,

$$P_h = \frac{1}{4}\left]\begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix}\right[,$$

with its symbol $\widetilde{P}_h(\theta_1,\theta_2)=\frac{1}{4}(1+\cos\theta_1)(1+\cos\theta_2)$. Thus,

$$\widetilde{\boldsymbol{P}}_h(\boldsymbol{\theta})=\frac{1}{4}\begin{pmatrix}(1+\cos\theta_1)(1+\cos\theta_2)\\(1-\cos\theta_1)(1+\cos\theta_2)\\(1+\cos\theta_1)(1-\cos\theta_2)\\(1-\cos\theta_1)(1-\cos\theta_2)\end{pmatrix},$$

and $\widetilde{\boldsymbol{R}}_h(\boldsymbol{\theta})=\widetilde{\boldsymbol{P}}_h^T(\boldsymbol{\theta})$. For more details on calculation of symbols of grid-transfer operators, we refer to [53, 75, 81].

Here, we show LFA predictions for the 2D Laplace problem with weighted Jacobi relaxation. ρ_h is computed with $h=\frac{1}{64}$. In the smoothing analysis above, we give the optimal parameter choice for weighted Jacobi relaxation scheme. Figure 2.1 presents the two-grid LFA convergence factor for weighted Jacobi, as a function of ω, to show the sensitivity of performance to parameter choice. From Figure 2.1, we see that the LFA smoothing factor and predicted two-grid convergence factors match well. Note that the optimal parameter is $\frac{4}{5}$, which is consistent with the smoothing analysis.

Now we take $\omega=\frac{4}{5}$ to show the LFA predictions. At the left of Figure 2.2, we present the spectral radius of the error-propagation symbol for the weighted Jacobi relaxation, as a function of $\boldsymbol{\theta}$, showing that weighted Jacobi relaxation reduces errors over the high frequencies quickly. The right of Figure 2.2 shows the spectra of the two-grid error-propagation operators for weighted Jacobi relaxation. We see the two-grid convergence factor $\rho_h=\frac{3}{5}$, which is equal to the optimal smoothing factor.

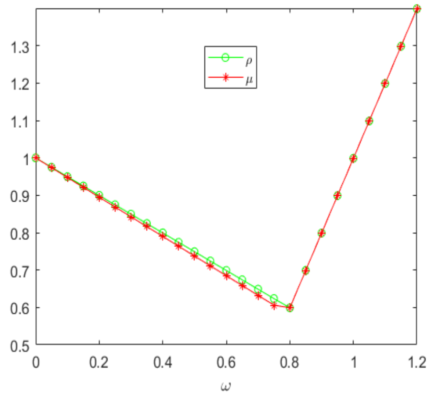

Figure 2.1: The two-grid local Fourier analysis convergence and smoothing factors for weighted Jacobi, as a function of ω.

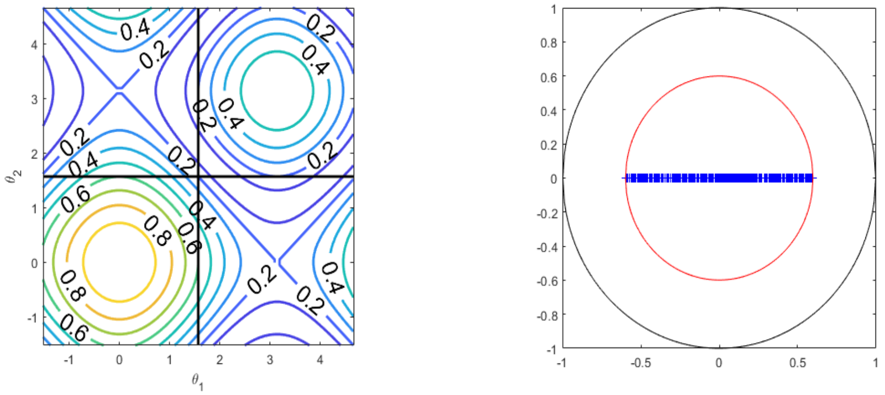

Figure 2.2: At left, the spectral radius of the error-propagation symbol for weighted Jacobi relaxation with $\omega = \frac{4}{5}$, as a function of the Fourier mode $\boldsymbol{\theta}$. At right, the spectrum of the two-grid error-propagation operator for weighted Jacobi relaxation with $\omega = \frac{4}{5}$. The radius of the red circle is the smoothing factor.

Remark 2.3.4. *If we use rediscretization operator, $L_H^* = L_H$, for the 2D Laplace problem with weighted Jacobi relaxation, we obtain the same LFA predictions as those with the Galerkin operator.*

The convergence factor of the two-grid method can be estimated directly from the two-grid LFA convergence factor in (2.18). If we assume that we have an “ideal” coarse-grid-correction operator that annihilates low-frequency error components and leaves high-frequency components unchanged, then the resulting LFA smoothing analysis usually gives a good prediction for the actual multigrid performance. For precise prediction by LFA, we usually consider an infinite-grid operator, that is, we ignore the boundary conditions. In practical computing, extra work, for example, pre-relaxation, might be needed to deal with boundary conditions and obtain better performance. Under these circumstances, the smoothing factor, (2.13), of LFA can be used to analyse the multigrid algorithm and easily optimize any parameters available. Two- and multi-level local Fourier analysis have been established to adapt to different multigrid cycling strategies. For more details, we refer the reader to [75, 81].

2.4 LFA applications

LFA can be applied to different types of discretization schemes, including discontinuous Galerkin finite-element [46], finite-difference [75], and finite-volume methods [52]. Both staggered and unstaggered grids, and even more complicated grids, can be considered. Both vertex-centred and cell-centred multigrid methods have been considered [57, 75]. There are also some recent extensions to more complicated meshes, for example, discretizations on Voronoi meshes [66].

Different relaxation schemes have also been investigated using LFA. For PDEs with a single (scalar) unknown, pointwise relaxation schemes, such as classical Jacobi, and Gauss-Seidel are widely used. Alternating line relaxation [75], a combination of x-line and y-line relaxation, is attractive, due to its robustness, yielding excellent properties for a large class of complicated problems, including anisotropic model problems. Collective relaxation schemes are also very efficient, updating the solution over subsets, whose union covers all of the unknowns. The advantage of these methods is that one can solve the resulting small-scale problems over subsets of the unknowns more accurately and efficiently. For example, Local Fourier analysis has been applied to the curl-curl equation with overlapping block relaxation [9, 53]. Collective (Vanka-type) relaxation has also been well-studied for scalar PDEs or systems of PDEs, see [53, 65], including theoretical analysis of the validity of LFA for multigrid methods with staggered grid transfers and multiplicative overlapping smoothers [53]. Arbitrary finite-element discretizations can also be analysed in that framework for LFA. Recently, LFA has also been presented for periodic stencils with collective relaxation [63], with a flexible computer implementation [64].

2.5 Block-structured solvers

Researchers have recently shown increased interest in numerical solution of the Stokes equations, whose discretization naturally leads to saddle-point systems. The design of fast solvers for the Stokes equations has been a major research subject in recent years, developing efficient algorithms for the Navier-Stokes equations [41, 73] and control problems governed by the Stokes equations [51, 62]. We mainly employ LFA to help us analyze and construct better algorithms for the solution of the Stokes equations with multigrid methods.

The discretization of saddle-point problems, considered here, has the following form

$$Kx = \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \mathcal{U} \\ p \end{pmatrix} = b, \tag{2.19}$$

where we focus on the case where A is symmetric and positive definite on the kernel of B, B has full rank, and C is symmetric and positive semidefinite, including the case where C is zero.

In the literature, researchers have developed two main families of preconditioners for the Stokes equations. Block preconditioners (cf. [31] and the references therein) are commonly used, since they can easily be constructed from standard multigrid algorithms for scalar elliptic PDEs, such as algebraic multigrid [67]. Monolithic multigrid methods, in contrast, are directly applied to the system in coupled form. However, the construction and analysis of these methods poses some difficulty, because standard pointwise relaxation schemes cannot be applied. Thus, several families of relaxation schemes have been developed for monolithic multigrid methods for the Stokes equations and more complicated saddle-point systems. These methods have been shown to outperform block preconditioners in some cases (see, e.g., [2]).

Distributive relaxation [14, 60, 82] was the first approach to be proposed, and can be regarded as a generalization of decoupled relaxation, that has been further developed [5, 77]. For a theoretical description and corresponding analysis, we refer to [82, 83]. The central idea is to use a distribution operator, $\mathcal{P}$, to allow use of pointwise relaxation schemes on transformed variables. For distributive Gauss-Seidel or weighted-Jacobi relaxation (with weights α_1, α_2), we solve a system of the form

$$M\delta\hat{x} = \begin{pmatrix} \alpha_1 D_1 & 0 \\ B & \alpha_2 D_2 \end{pmatrix} \begin{pmatrix} \delta\hat{\mathcal{U}} \\ \delta\hat{p} \end{pmatrix} = \begin{pmatrix} r_{\mathcal{U}} \\ r_p \end{pmatrix}, \tag{2.20}$$

where D_1 and D_2 are approximations to the corresponding blocks in $K\mathcal{P}$, respectively. Then, distribute the updates as $\delta x = \mathcal{P}\delta\hat{x}$. Equation (2.20) is equivalent to computing the updates as

$$\begin{aligned} \delta\hat{\mathcal{U}} &= (\alpha_1 D_1)^{-1} r_{\mathcal{U}}, \\ \delta\hat{p} &= (\alpha_2 D_2)^{-1}(r_p - B\delta\hat{\mathcal{U}}), \end{aligned}$$

followed by distribution to the original unknowns by computing

$$\begin{pmatrix} \delta\mathcal{U} \\ \delta p \end{pmatrix} = \mathcal{P} \begin{pmatrix} \delta\hat{\mathcal{U}} \\ \delta\hat{p} \end{pmatrix}.$$

Different choices of D_1 and D_2 lead to the distributive Gauss-Seidel or distributive Jacobi relaxation, or other schemes in this family.

A collective relaxation scheme was introduced by Vanka [76], based on solving a sequence of localized saddle-point problems in a block overlapping Gauss-Seidel iteration. Although collective relaxation is more robust for coupled systems, it is also more expensive in practice than decoupled relaxation. More detailed comparison between coupled and decoupled relaxations can be found, for example, in [75]. Others block relaxation schemes include the Braess-Sarazin [12] and Uzawa [54] approaches.

Using Braess-Sarazin relaxation (BSR) for system (2.19), one solves a system of the form

$$M\delta x = \begin{pmatrix} \alpha D & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \delta\mathcal{U} \\ \delta p \end{pmatrix} = \begin{pmatrix} r_{\mathcal{U}} \\ r_p \end{pmatrix}, \tag{2.21}$$

where D is an approximation to A, the inverse of which is easy to apply, for example $I,$ or diag(A). Solutions of (2.21) are computed in two stages as

$$\begin{aligned} S\delta p &= \frac{1}{\alpha} BD^{-1} r_{\mathcal{U}} - r_p, \\ \delta\mathcal{U} &= \frac{1}{\alpha} D^{-1}(r_{\mathcal{U}} - B^T\delta p), \end{aligned} \tag{2.22}$$

where $S = \frac{1}{\alpha}BD^{-1}B^T + C$, and $\alpha > 0$ is a chosen weight to obtain a better relaxation scheme. Iterative methods can be applied to the solution for δp in (2.22), which leads to inexact BSR methods [84].

The Uzawa-type relaxation schemes that we consider can be written in terms of a simpler block solve than that used in BSR,

$$M\delta x = \begin{pmatrix} \alpha D & 0 \\ B & -S \end{pmatrix} \begin{pmatrix} \delta\mathcal{U} \\ \delta p \end{pmatrix} = \begin{pmatrix} r_{\mathcal{U}} \\ r_p \end{pmatrix}, \tag{2.23}$$

where αD is an approximation of A and $-S$ is an approximation of the Schur complement, $-BA^{-1}B^T - C$. Equation (2.23) is equivalent to computing the updates as

$$\begin{aligned} \delta\mathcal{U} &= (\alpha D)^{-1} r_{\mathcal{U}}, \\ S\delta p &= B\delta\mathcal{U} - r_p. \end{aligned}$$

The choice of S leads to different types of Uzawa relaxation, which we will investigate.

Each of the above families has been further developed in recent years, including Braess-Sarazin-type relaxation schemes [1, 2, 3, 11, 12], Vanka-type relaxation schemes [1, 2, 3, 53, 56, 65, 69, 76], Uzawa-type relaxation schemes [39, 42, 47, 58], and other types of methods [19, 72].

Here, we first consider the MAC scheme for the Stokes equations, and address the natural question of how to solve the resulting saddle-point systems. Block relaxation schemes, such as Braess-Sarazin, Uzawa, and distributive approaches, have each been investigated in this setting. However, few studies have been carried out comparing these schemes. Thus, we concentrate on LFA for Braess-Sarazin, Uzawa, and distributive relaxation schemes, and focus on optimizing the parameters for each to provide a fair comparison of performance. Considering parallel implementation on modern architectures, we consider variants based on weighted-Jacobi relaxation.

For predicting performance, early studies mainly have focused on LFA smoothing analysis. However, recently, some studies have reported that smoothing analysis is unable to give a good prediction of multigrid behavior for some problems [36, 37, 53]. Specifically, in [37], local Fourier analysis failed to provide its usual predictivity of the convergence behavior of multigrid applied to the space-time diffusion equation and its generalizations. In [36], however, a semi-algebraic mode analysis (SAMA) was proposed to remedy standard LFA and provide insight into asymptotic convergence behaviour of multigrid methods. In [53], the smoothing factor of LFA overestimates the two-grid convergence factor for the $Q_2 - Q_1$ discretization of the Stokes equations with Vanka-type relaxation.

Our work is motivated by the failure of the classical smoothing analysis for the $Q_2 - Q_1$ approximation. Since this failure might be related to the Q_2 approximation for the velocity unknowns, we first investigate higher-order finite-element methods for the Laplace problem. Even for the simple weighted Jacobi relaxation for the Laplace problem, although the two-grid LFA convergence factor matches with realistic multigrid performance, we find that the LFA smoothing factor fails to predict this performance. A modified two-grid local Fourier analysis is presented, and the correct parameter choice is shown to yield a significant improvement in two-grid and multigrid convergence factors. This study further helps us understand the previous findings that the classical smoothing analysis of LFA loses its predictivity of multigrid performance for the $Q_2 - Q_1$ approximation to the Stokes equations.

Following this, we discuss LFA for multigrid methods applied to Taylor-Hood and two stabilized ($Q_1 - Q_1$) finite-element discretizations of the Stokes equations. Similarly to the case of the MAC discretization, block-structured relaxations are considered for these finite-element methods. As the exact application of these schemes is expensive, we also experiment with the inexact case, in which the subsystem solves are performed by a few steps of Jacobi or multigrid iteration. Rediscretization and Galerkin coarse-grid operators are discussed. Many interesting results are found.

2.6 Domain decomposition

With increasing problem sizes, there is an urgent need to design fast and efficient algorithms. Direct solvers usually are too costly, especially considering memory. Domain decomposition is well-suited for parallelism and can be applied to some challenging problems, for example indefinite Helmholtz equations [18, 32]. There are two common families: nonoverlapping and overlapping domain decomposition, and many approaches, including Neumann-Neumann [45, 61, 74], FETI [29, 33, 34], Schwarz [27, 28, 74], and Optimized Schwarz [27, 38] methods. Balancing domain decomposition by constraints (BDDC), one of the nonoverlapping domain decomposition methods, was first introduced by Dohrmann in [24]. Recently, BDDC has been extended to many problems either as a solver or preconditioner, including for the Stokes equations [48], elliptic problems [7, 50], 3D problems in $H(\text{curl})$ [26], and others [25, 49, 55]. However, existing research focuses mainly on either the linear-algebraic aspects of solutions or the analysis of error estimates based on the finite-element theory. In contrast to this work, we extend LFA to BDDC to examine the condition number of the preconditioned operators.

The idea of domain decomposition methods is very natural and simple. First, partition the domain, Ω, into N subdomains, $\Omega_i, i = 1, 2, \cdots, N$, such that $\bar{\Omega} =$

$\bigcup_{i=1}^N \bar{\Omega}_i$. Then, solve smaller-scale problems on each subdomain, Ω_i. Finally, "glue" the local solutions together to obtain a global approximation to the solution. There are many techniques for this "glue" to obtain the correct solution. How we choose the subdomain and glue the local solutions together determines the different classes of domain decomposition methods.

Here, we give a brief introduction to Schwarz domain decomposition methods, to shed light on the BDDC approach that we will investigate. For more details about domain decomposition methods, we refer to [27, 74]. Consider the Laplace problem on a bounded domain, Ω, with Lipschitz boundary, with homogenous Dirichlet boundary conditions as follows

$$\begin{cases} \Delta w = f, \text{ in } \Omega \\ w = 0, \text{ on } \partial\Omega \end{cases} \tag{2.24}$$

For simplification, suppose that Ω is partitioned into two nonoverlapping subdomains Ω_i (as shown at the left of Figure 2.3):

$$\bar{\Omega} = \bar{\Omega}_1 \cup \bar{\Omega}_2, \ \partial\Omega_1 \cap \partial\Omega_2 = \partial\Omega_s. \tag{2.25}$$

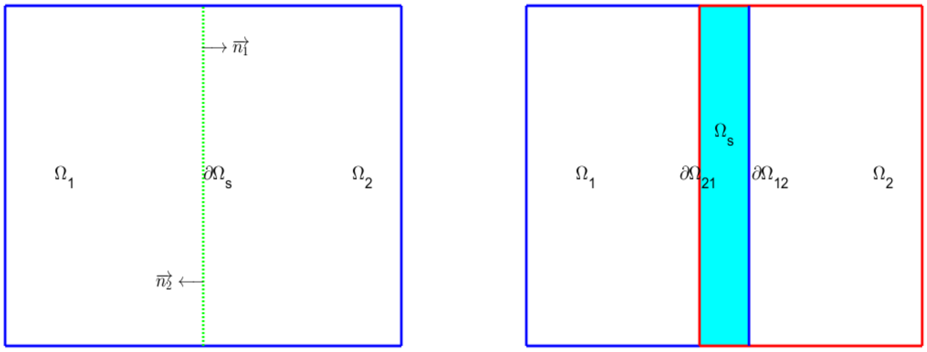

Figure 2.3: At left, partition of Ω into two nonoverlapping subdomains, and $\vec{n}_i$ $(i = 1, 2)$ denote the outward normals to the boundary $\partial\Omega$ corresponding to subdomain Ω_i. At right, partition of Ω into two overlapping subdomains.

The Laplace problem (2.24) is equivalent to the following coupled system [74],

$$\begin{cases} -\Delta w_1 = f, \text{ in } \Omega_1 \\ w_1 = 0, \text{ on } \partial\Omega_1 \cap \partial\Omega \\ w_2 = w_1, \text{ on } \partial\Omega_s \\ \frac{\partial w_2}{\partial \vec{n}_2} = -\frac{\partial w_1}{\partial \vec{n}_1}, \text{ on } \partial\Omega_s \\ -\Delta w_2 = f, \text{ in } \Omega_2 \\ w_2 = 0, \text{ on } \partial\Omega_2 \cap \partial\Omega \end{cases} \tag{2.26}$$

Several domain decomposition approaches arise from (2.26), alternately solving for w_1 and w_2 based on the conditions imposed on Ω_S.

The finite-element discretization of (2.24) leads to the linear algebraic system,

$$A\mathrm{w} = \mathrm{f}, \tag{2.27}$$

which, similarly to (2.26), can be ordered as

$$A = \begin{pmatrix} A_I^{(1)} & 0 & A_{I\Gamma}^{(1)} \\ 0 & A_I^{(2)} & A_{I\Gamma}^{(2)} \\ A_{\Gamma I}^{(2)} & A_{\Gamma I}^{(2)} & A_{\Gamma\Gamma} \end{pmatrix}, \ \mathrm{w} = \begin{pmatrix} \mathrm{w}_I^{(1)} \\ \mathrm{w}_I^{(2)} \\ \mathrm{w}_\Gamma \end{pmatrix}, \ \mathrm{f} = \begin{pmatrix} \mathrm{f}_I^{(1)} \\ \mathrm{f}_I^{(2)} \\ \mathrm{f}_\Gamma \end{pmatrix},$$

where the variables corresponding to Ω_1, Ω_2, and $\partial\Omega_s$ are labelled by $\mathrm{w}_I^{(1)}$, $\mathrm{w}_I^{(2)}$, and w_Γ, respectively. Note that A is written in block form, similar to the saddle-point structure. The challenge for (2.27) is how to construct fast solvers using this decomposition.

In contrast, suppose that Ω is partitioned into two overlapping subdomains Ω_i (shown at the right of Figure 2.3):

$$\bar{\Omega} = \bar{\Omega}_1 \cup \bar{\Omega}_2, \ \Omega_1 \cap \Omega_2 = \Omega_s.$$

Here, we discuss additive Schwarz methods, based on the overlapping partition.

To avoid complications from the overlap between subdomains, we introduce the partition of unity functions. Since $\Omega_1 \cap \Omega_2 = \Omega_s$, to obtain global solutions from the subdomains, we first define an extension operator E_i. For a function $w_i : \Omega_i \to \mathbb{R}$, $E_i(w_i) : \Omega_i \to \Omega$ is the extension of w_i by zero outside Ω_i. Another option to "glue" the solutions together is to use partition of unity functions, g_i, mapping Ω_i to $\mathbb{R}$, with $g_i(x) \geq 0, g_1(x) + g_2(x) = 1$, and $\text{supp}(g_i) \subset \Omega_i$ for each i, and such that for all functions $w : \Omega_i \to \mathbb{R}$,

$$w = E_1(g_1 w|_{\Omega_1}) + E_2(g_2 w|_{\Omega_2}).$$

Based on the above, we introduce the famous additive Schwarz (AS) and restricted additive Schwarz (RAS) approaches based on the overlapping partition (see the right of Figure 2.3), following [27].

Algorithm 2.6.1. *AS and RAS Algorithms*

Given w^0 that satisfies boundary conditions on Ω,

1. *Compute the residual: $r^n = f + \Delta w^n$;*
2. *For $i = 1, 2$, solve the following local subdomain problem:*

$$\begin{cases} -\Delta v_i^n = r^n, \text{ in } \Omega_i \\ v_i^n = 0, \text{ on } \partial\Omega_i \end{cases}$$

3. *Two choices to update solution w^n:*
   - *(a) AS choice*

$$w^{n+1} = w^n + E_1(v_1^n) + E_2(v_2^n). \tag{2.28}$$

   - *(b) RAS choice*

$$w^{n+1} = w^n + E_1(g_1 v_1^n) + E_2(g_2 v_2^n). \tag{2.29}$$

The advantage of AS is that it is suitable to parallelize, but its convergence is very slow. In practice, AS is always used as a preconditioner for a Krylov method such as GMRES, CG, or BiCGSTAB. For more details, including about the implementation of AS and RAS, we refer to [27]. Two-level AS methods have also been developed for scalar second-order symmetric positive-definite elliptic boundary value problems and for the biharmonic equation [15], using a nonconforming finite-element method. A finite-element-based additive Schwarz preconditioner has been developed for the Navier-Stokes equations [35]. Often, RAS shows a faster convergence than AS [30]. An extension of RAS preconditioning has been designed for symmetric positive-definite problems [17], where sharp condition number bounds on the preconditioned system and the combination with CG are discussed, and also for general sparse systems [18].

BDDC can be treated as a combination of nonoverlapping and overlapping decomposition methods, using a nonoverlapping partition, but the idea of (2.28) and (2.29) to give two values to represent the solution along the boundary $\partial\Omega_s$, and to construct a preconditioner for the global problem (2.24). Taking Figure 2.4 as example, where Ω has the same partition as (2.25). We duplicate $\partial\Omega_s$, and introduce independent degrees of freedom for subdomains Ω_1 and Ω_2, along this line. We call the union of the duplicated subdomains the "duplicated global" domain, corresponding to a duplicated global problem. Then, we glue the solutions of the two subdomain problems together to get the approximate solution of the global problem (2.24). To be specific, we have the matrix representation of the duplicated global problem,

$$\hat{A} = \sum_{i=1}^{2} (\bar{R}^{(i)})^T A^{(i)} \bar{R}^{(i)}, \tag{2.30}$$

where $\bar{R}_i$ is a restriction operator mapping from the "duplicated global" variables to the i-th subdomain variables, and $A^{(i)}$ corresponds to matrix representation of subdomain problem in Ω_i with Neumann boundary conditions on $\partial\Omega_s$. Based on (2.30), a preconditioner for A is given by

$$M^{-1} = \mathcal{R}^T \hat{A}^{-1} \mathcal{R}, \tag{2.31}$$

where $\mathcal{R}$ is a mapping from the standard global variables to the "duplicated global" variables. The role of $\mathcal{R}^T$ is the same as E_i in AS and RAS. In [50], "lumped" and Dirichlet operators are used to construct $\mathcal{R}$. Our BDDC work is based on the preconditioners introduced in [50].

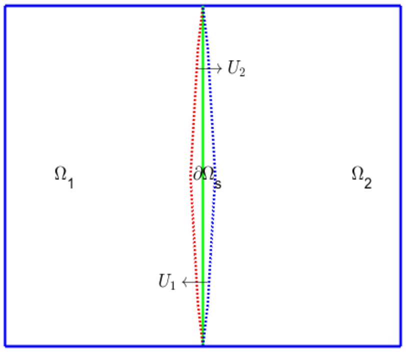

Figure 2.4: Nonoverlapping partition for BDDC method with two subdomains.

In (2.31), $\hat{A}$ has a block structured form with block LU decomposition,

$$\hat{A} = \begin{pmatrix} A_{rr} & \hat{A}_{\Pi r}^T \\ \hat{A}_{\Pi r} & A_{\Pi\Pi} \end{pmatrix} = \begin{pmatrix} A_{rr} & 0 \\ \hat{A}_{\Pi r} & \hat{S}_{\Pi\Pi} \end{pmatrix} \begin{pmatrix} I & A_{rr}^{-1} A_{\Pi r}^T \\ 0 & I \end{pmatrix},$$

where A_{rr} corresponds to the subdomain interior and interface degrees of freedom, $A_{\Pi\Pi}$ corresponds to the coarse-level degrees of freedom, which are located at the corners of the subdomains, $\hat{A}_{\Pi r}$ is the connections between the coarse-level and subdomain and interface degrees of freedom, and $\hat{S}_{\Pi\Pi} = A_{\Pi\Pi} - \hat{A}_{\Pi r} A_{rr}^{-1} \hat{A}_{\Pi r}^T$ is the Schur complement. In BDDC, the solution of the Schur complement equation ($\hat{S}_{\Pi\Pi}$) is needed, which is the main bottleneck of the BDDC approach. To mitigate this, we propose variants of BDDC algorithms based on multiplicative preconditioning ideas. From LFA, we can quantitatively estimate the conditioner numbers of BDDC-like algorithms, which gives us some insight into the design of efficient solvers.

Bibliography

[1] J. H. Adler, T. R. Benson, E. C. Cyr, S. P. MacLachlan, and R. S. Tuminaro. Monolithic multigrid methods for two-dimensional resistive magnetohydrodynamics. *SIAM J. Sci. Comput.*, 38(1):B1–B24, 2016.

[2] 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.

[3] J. H. Adler, D. B. Emerson, S. P. MacLachlan, and T. A. Manteuffel. Constrained optimization for liquid crystal equilibria. *SIAM J. Sci. Comput.*, 38(1):B50–B76, 2016.

[4] W. F. Ames. *Numerical methods for partial differential equations.* Academic Press [Harcourt Brace Jovanovich, Publishers], New York; Thomas Nelson & Sons, London-Lagos-Melbourne, second edition, 1977. Computer Science and Applied Mathematics, Applications of Mathematics Series.

[5] C. Bacuta, P. S. Vassilevski, and S. Zhang. A new approach for solving Stokes systems arising from a distributive relaxation method. *Numerical Methods for Partial Differential Equations*, 27(4):898–914, 2011.

[6] R. E. Bank, B. D. Welfert, and H. Yserentant. A class of iterative methods for solving saddle point problems. *Numer. Math.*, 56(7):645–666, 1990.

[7] L. Beirão da Veiga, D. Cho, L. F. Pavarino, and S. Scacchi. BDDC preconditioners for isogeometric analysis. *Math. Models Methods Appl. Sci.*, 23(6):1099–1142, 2013.

[8] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numer.*, 14:1–137, 2005.

[9] T. Boonen, J. Van lent, and S. Vandewalle. Local Fourier analysis of multigrid for the curl-curl equation. *SIAM J. Sci. Comput.*, 30(4):1730–1755, 2008.

[10] D. Braess. *Finite elements: Theory, fast solvers, and applications in elasticity theory.* Cambridge University Press, Cambridge, third edition, 2007. Translated from the German by Larry L. Schumaker.

[11] D. Braess and W. Dahmen. A cascadic multigrid algorithm for the Stokes equations. *Numer. Math.*, 82(2):179–191, 1999.

[12] D. Braess and R. Sarazin. An efficient smoother for the Stokes problem. *Appl. Numer. Math.*, 23(1):3–19, 1997.

[13] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comp.*, 31(138):333–390, 1977.

[14] A. Brandt and N. Dinar. Multigrid solutions to elliptic flow problems. In *Numerical Methods for Partial Differential Equations (Proc. Adv. Sem., Math. Res. Center, Univ. Wisconsin, Madison, Wis., 1978)*, volume 42 of *Publ. Math. Res. Center Univ. Wisconsin*, pages 53–147. Academic Press, New York-London, 1979.

[15] S. C. Brenner. Two-level additive Schwarz preconditioners for nonconforming finite element methods. *Math. Comp.*, 65(215):897–921, 1996.

[16] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A multigrid tutorial*. SIAM, 2000.

[17] X.-C. Cai, M. Dryja, and M. Sarkis. Restricted additive Schwarz preconditioners with harmonic overlap for symmetric positive definite linear systems. *SIAM J. Numer. Anal.*, 41(4):1209–1231, 2003.

[18] X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.*, 21(2):792–797, 1999.

[19] L. Chen. Multigrid methods for saddle point systems using constrained smoothers. *Comput. Math. Appl.*, 70(12):2854–2866, 2015.

[20] A. J. Chorin. The numerical solution of the Navier-Stokes equations for an incompressible fluid. *Bull. Amer. Math. Soc.*, 73:928–931, 1967.

[21] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Math. Comp.*, 22:745–762, 1968.

[22] P. Constantin and C. Foias. *Navier-Stokes equations*. Chicago Lectures in Mathematics. University of Chicago Press, Chicago, IL, 1988.

[23] E. C. Cyr, J. N. Shadid, R. S. Tuminaro, R. P. Pawlowski, and L. Chacón. A new approximate block factorization preconditioner for two-dimensional incompressible (reduced) resistive MHD. *SIAM Journal on Scientific Computing*, 35(3):B701–B730, 2013.

[24] C. R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. *SIAM Journal on Scientific Computing*, 25(1):246–258, 2003.

[25] C. R. Dohrmann. An approximate BDDC preconditioner. *Numerical Linear Algebra with Applications*, 14(2):149–168, 2007.

[26] C. R. Dohrmann and O. B. Widlund. Some recent tools and a BDDC algorithm for 3D problems in $H(\mathrm{curl})$. In *Domain Decomposition Methods in Science and Engineering XX*, volume 91 of *Lect. Notes Comput. Sci. Eng.*, pages 15–25. Springer, Heidelberg, 2013.

[27] V. Dolean, P. Jolivet, and F. Nataf. *An introduction to domain decomposition methods: Algorithms, theory, and parallel implementation.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2015.

[28] M. Dryja and O. B. Widlund. Towards a unified theory of domain decomposition algorithms for elliptic problems. In *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Houston, TX, 1989)*, pages 3–21. SIAM, Philadelphia, PA, 1990.

[29] M. Dryja and O. B. Widlund. A FETI-DP method for a mortar discretization of elliptic problems. *Lecture Notes in Computational Science and Engineering*, 23:41–52, 2002.

[30] E. Efstathiou and M. J. Gander. Why restricted additive Schwarz converges faster than additive Schwarz. *BIT*, 43(suppl.):945–959, 2003.

[31] 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.

[32] O. G. Ernst and M. J. Gander. Why it is difficult to solve Helmholtz problems with classical iterative methods. In *Numerical analysis of multiscale problems*, volume 83 of *Lect. Notes Comput. Sci. Eng.*, pages 325–363. Springer, Heidelberg, 2012.

[33] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. FETI-DP: a dual-primal unified FETI method. I. A faster alternative to the two-level FETI method. *Internat. J. Numer. Methods Engrg.*, 50(7):1523–1544, 2001.

[34] C. Farhat, J. Mandel, and F.-X. Roux. Optimal convergence properties of the FETI domain decomposition method. *Comput. Methods Appl. Mech. Engrg.*, 115(3-4):365–385, 1994.

[35] P. F. Fischer. An overlapping Schwarz method for spectral element solution of the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 133(1):84–101, 1997.

[36] S. Friedhoff and S. MacLachlan. A generalized predictive analysis tool for multigrid methods. *Numerical Linear Algebra with Applications*, 22(4):618–647, 2015.

[37] S. Friedhoff, S. MacLachlan, and C. Borgers. Local Fourier analysis of space-time relaxation and multigrid schemes. *SIAM Journal on Scientific Computing*, 35(5):S250–S276, 2013.

[38] M. J. Gander, F. Magoulès, and F. Nataf. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM J. Sci. Comput.*, 24(1):38–60, 2002.

[39] F. J. Gaspar, Y. Notay, C. W. Oosterlee, and C. Rodrigo. A simple and efficient segregated smoother for the discrete Stokes equations. *SIAM J. Sci. Comput.*, 36(3):A1187–A1206, 2014.

[40] J.-F. Gerbeau. A stabilized finite element method for the incompressible magnetohydrodynamic equations. *Numer. Math.*, 87(1):83–111, 2000.

[41] V. Girault and P.-A. Raviart. *Finite element methods for Navier-Stokes equations:Theory and algorithms*. Springer Science & Business Media, 2012.

[42] B. Gmeiner, M. Huber, L. John, U. Rüde, and B. Wohlmuth. A quantitative performance study for Stokes solvers at the extreme scale. *J. Comput. Sci.*, 17(part 3):509–521, 2016.

[43] A. Greenbaum. *Iterative methods for solving linear systems*, volume 17 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[44] M. D. Gunzburger, A. J. Meir, and J. S. Peterson. On the existence, uniqueness, and finite element approximation of solutions of the equations of stationary, incompressible magnetohydrodynamics. *Mathematics of Computation*, 56(194):523–563, 1991.

[45] M. Heinkenschloss and H. Nguyen. Neumann-Neumann domain decomposition preconditioners for linear-quadratic elliptic optimal control problems. *SIAM J. Sci. Comput.*, 28(3):1001–1028, 2006.

[46] P. W. Hemker, W. Hoffmann, and M. H. van Raalte. Fourier two-level analysis for discontinuous Galerkin discretization with linear elements. *Numer. Linear Algebra Appl.*, 11(5-6):473–491, 2004.

[47] L. John, U. Rüde, B. Wohlmuth, and W. Zulehner. On the analysis of block smoothers for saddle point problems. *arXiv preprint arXiv:1612.01333*, 2016.

[48] J. Li and O. Widlund. BDDC algorithms for incompressible Stokes equations. *SIAM J. Numer. Anal.*, 44(6):2432–2455, 2006.

[49] J. Li and O. Widlund. A BDDC preconditioner for saddle point problems. In *Domain Decomposition Methods in Science and Engineering XVI*, volume 55 of *Lect. Notes Comput. Sci. Eng.*, pages 413–420. Springer, Berlin, 2007.

[50] J. Li and O. Widlund. On the use of inexact subdomain solvers for BDDC algorithms. *Computer Methods in Applied Mechanics and Engineering*, 196(8):1415–1428, 2007.

[51] W. Liu and N. Yan. A posteriori error estimates for control problems governed by Stokes equations. *SIAM Journal on Numerical Analysis*, 40(5):1850–1869, 2002.

[52] P. Luo, C. Rodrigo, F. Gaspar, and C. Oosterlee. On an Uzawa smoother in multigrid for poroelasticity equations. *Numerical Linear Algebra with Applications*, 24(1), 2017.

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

[54] J.-F. Maitre, F. Musy, and P. Nigon. A fast solver for the Stokes equations using multigrid with a Uzawa smoother. In *Advances in Multigrid Methods (Oberwolfach, 1984)*, volume 11 of *Notes Numer. Fluid Mech.*, pages 77–83. Vieweg, Braunschweig, 1985.

[55] J. Mandel and C. R. Dohrmann. Convergence of a balancing domain decomposition by constraints and energy minimization. *Numerical Linear Algebra with Applications*, 10(7):639–659, 2003.

[56] S. Manservisi. Numerical analysis of Vanka-type solvers for steady Stokes and Navier-Stokes flows. *SIAM J. Numer. Anal.*, 44(5):2025–2056, 2006.

[57] M. Mohr and R. Wienands. Cell-centred multigrid revisited. *Comput. Vis. Sci.*, 7(3-4):129–140, 2004.

[58] M. A. Olshanskii. Multigrid analysis for the time dependent Stokes problem. *Math. Comp.*, 81(277):57–79, 2012.

[59] M. A. Olshanskii and E. E. Tyrtyshnikov. *Iterative methods for linear systems: theory and applications*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014.

[60] C. W. Oosterlee and F. J. Gaspar. Multigrid methods for the Stokes system. *Computing in Science & Engineering*, 8(6):34–43, 2006.

[61] L. F. Pavarino and O. B. Widlund. Balancing Neumann-Neumann methods for incompressible Stokes equations. *Comm. Pure Appl. Math.*, 55(3):302–335, 2002.

[62] T. Rees. *Preconditioning iterative methods for PDE constrained optimization*. PhD thesis, University of Oxford, 2010.

[63] H. Rittich. *Extending and Automating Fourier Analysis for Multigrid Methods*. PhD thesis, University of Wuppertal, June 2017.

[64] H. Rittich. https://hrittich.github.io/lfa-lab/. June 2017.

[65] C. Rodrigo, F. J. Gaspar, and F. J. Lisbona. On a local Fourier analysis for overlapping block smoothers on triangular grids. *Appl. Numer. Math.*, 105:96–111, 2016.

[66] C. Rodrigo, P. Salinas, F. J. Gaspar, and F. J. Lisbona. Local Fourier analysis for cell-centered multigrid methods on triangular grids. *J. Comput. Appl. Math.*, 259(part A):35–47, 2014.

[67] J. W. Ruge and K. Stüben. Algebraic multigrid. In *Multigrid methods*, volume 3 of *Frontiers Appl. Math.*, pages 73–130. SIAM, Philadelphia, PA, 1987.

[68] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2003.

[69] J. Schöberl and W. Zulehner. On Schwarz-type smoothers for saddle point problems. *Numer. Math.*, 95(2):377–399, 2003.

[70] D. Schötzau. Mixed finite element methods for stationary incompressible magneto–hydrodynamics. *Numerische Mathematik*, 96(4):771–800, 2004.

[71] K. Stüben and U. Trottenberg. Multigrid methods: Fundamental algorithms, model problem analysis and applications. *Multigrid Methods*, pages 1–176, 1982.

[72] S. Takacs. A robust multigrid method for the time-dependent Stokes problem. *SIAM J. Numer. Anal.*, 53(6):2634–2654, 2015.

[73] R. Temam. *Navier-Stokes equations: Theory and numerical analysis.* American Mathematical Soc., 2001.

[74] A. Toselli and O. Widlund. *Domain decomposition methods: Algorithms and theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2005.

[75] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid.* Academic Press, Inc., San Diego, CA, 2001. With contributions by A. Brandt, P. Oswald and K. Stüben.

[76] S. P. Vanka. Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. *J. Comput. Phys.*, 65(1):138–158, 1986.

[77] M. Wang and L. Chen. Multigrid methods for the Stokes equations using distributive Gauss-Seidel relaxations based on the least squares commutator. *J. Sci. Comput.*, 56(2):409–431, 2013.

[78] J. E. Welch, F. H. Harlow, J. P. Shannon, and B. J. Daly. The MAC method. A computing technique for solving viscous, incompressible, transient fluid-flow problems involving free surfaces. 1966.

[79] P. Wesseling. *An introduction to multigrid methods.* Pure and Applied Mathematics (New York). John Wiley & Sons, Ltd., Chichester, 1992.

[80] P. Wesseling. *Principles of computational fluid dynamics*, volume 29 of *Springer Series in Computational Mathematics.* Springer-Verlag, Berlin, 2001.

[81] R. Wienands and W. Joppich. *Practical Fourier analysis for multigrid methods.* CRC press, 2004.

[82] G. Wittum. Multi-grid methods for Stokes and Navier-Stokes equations. *Numerische Mathematik*, 54(5):543–563, 1989.

[83] G. Wittum. On the convergence of multi-grid methods with transforming smoothers. Theory with applications to the Navier-Stokes equations. *Numer. Math.*, 57(1):15–38, 1990.

[84] W. Zulehner. A class of smoothers for saddle point problems. *Computing*, 65(3):227–246, 2000.

Chapter 3

Vector-potential finite-element formulations for two-dimensional resistive magnetohydrodynamics

Abstract

[1] Vector-potential formulations are attractive for electromagnetic problems in two dimensions, since they reduce both the number and complexity of equations, particularly in coupled systems, such as magnetohydrodynamics (MHD). In this paper, we consider the finite-element formulation of a vector-potential model of two-dimensional resistive MHD. Existence and uniqueness are considered separately for the continuum nonlinear equations and the discretized and linearized form that arises from Newton's method applied to a modified system. Under some conditions, we prove that the solutions of the original and modified weak forms are the same, allowing us to prove convergence of both the discretization and the nonlinear iteration.

Keywords: Magnetohydrodynamics, mixed finite-element method, Newton's method

AMS subject classification: 76W05, 65N30

[1] Authors: J. H. Adler, Y. He, X. Hu and S. P. MacLachlan
This work is submitted as *Vector-potential finite-element formulations for two-dimensional resistive magnetohydrodynamics*, to Journal of Computers & Mathematics with Applications, 2017.

3.1 Introduction

Magnetohydrodynamics (MHD) models the flow of a charged fluid, or plasma, in the presence of electromagnetic fields. There are many formulations of MHD, depending on the domain and physical parameters considered. This includes assumptions associated with the coupling between the electric field, current density, and Ohm's law, leading to formulations such as ideal, resistive, and Hall MHD [16]. In this paper, we use a single incompressible fluid model, treating ions and electrons together, along with a resistive formulation. The resulting visco-resistive model couples the Navier-Stokes equations with Maxwell's equations, forming a nonlinear system of partial differential equations (PDEs). Moreover, we focus on time-independent solutions, with our primary focus on existence and uniqueness of solutions to the nonlinear and linearized systems of equations.

The equations of stationary, incompressible single fluid MHD posed in three dimensions are considered in (for example) [17, 18]. Under some conditions on the data, the existence and uniqueness of solutions to weak formulations of the equations is known both in the continuum and for certain discretizations. The focus of this paper is on MHD in two dimensions (2D). Here, a vector potential formulation was used in [2, 10]. Vector potential formulations are attractive for electromagnetic problems with two-dimensional dynamics, since they substantially reduce the complexity of the resulting equations, by trading vector for scalar unknowns, and the curl terms that arise in Maxwell's equations for standard gradient and diffusion operators. Despite this attractiveness, there is a scarcity of analysis for multiphysics systems using vector potential formulations, for both the continuum and discretized models. In this paper, we demonstrate that standard analysis techniques can be extended from three-dimensional MHD [17, 18] to the two-dimensional discretizations considered in [2, 10], although some complications arise that can only be addressed (to our knowledge) by making more restrictive assumptions.

Two-dimensional models of MHD arise when considering magnetically confined plasmas, such as in a large aspect-ratio tokamak reactor, as illustrated in Figure 3.1. In this setting, the magnetic field along the toroidal direction (denoted by z) is very large in order to contain the plasma. Consequently, the resulting dynamics decouple into a two-dimensional problem posed over the poloidal cross-section. While such a configuration can be accurately studied using full three-dimensional models, the

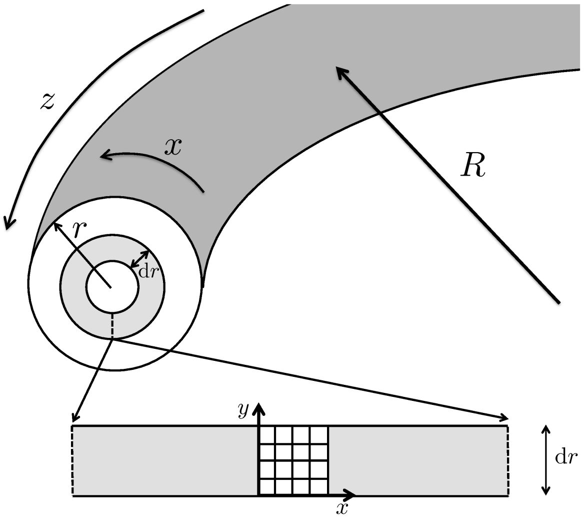

Figure 3.1: Cross-sectional view of large aspect-ratio tokamak geometry, with major radius, R, and minor radius, r, satisfying $R \gg r$. A cross-section of thickness dr can be unfolded to create a Cartesian grid as pictured.

computational cost of such models is substantially more than their two-dimensional counterparts, thus motivating the many numerical studies of MHD in two dimensions.

While numerical results using the vector potential formulation already exist in the literature, [2, 10] focus primarily on linear algebraic aspects of the solution of the resulting linearized systems of equations, leaving open the questions of existence and uniqueness of solutions. In this paper, we focus on the theoretical analysis of both the continuum model and its discretization, applying standard theoretical tools for the existence and uniqueness of solutions at both the continuum and discrete levels. For the discretization, this is complicated when considering a nonconforming discretization, as was used in [2, 10]. Nonetheless, under moderate conditions, we prove that Newton's method yields well-posed linearizations and converges to the solution of the weak formulation.

An outline of this paper is as follows. In Section 3.2, we detail the vector-potential formulation for the MHD problem in 2D and, under standard conditions, we prove the existence and uniqueness of the continuum solution. In Section 3.3, we introduce a modified, "uncurled", formulation for the MHD problem and present the analysis of the discretized problem using a mixed finite-element method. In Section 3.4, we consider Newton's method for solving the nonlinear system and analyze convergence.

Numerical results supporting the theory are presented in Section 3.5. Finally, some concluding remarks are given in Section 3.6.

In what follows, the letter C (with or without subscripts) denotes a generic positive constant which may be different depending on the context. For a Lipschitz domain $\Omega \subset \mathbb{R}^2$, denote by L^p, $1 \leq p \leq \infty$, the Lebesgue space of p-integrable functions, endowed with the norm $\|\cdot\|_{0,p}$. Denote the standard Euclidean norm as $|\cdot|$, the classical $L^2(\Omega)$ inner product and norm as $\langle\cdot,\cdot\rangle_0$ and $\|\cdot\|_0$, respectively, and $\langle f, g\rangle = \int_\Omega fg\mathrm{dX}$, where $fg \in L^1(\Omega)$. The standard L^2-based Sobolev space with integer or fractional exponent s is denoted by $H^s(\Omega)$. We write $\|\cdot\|_s$ for its norm.

For convenience, we introduce the spaces

$$\mathbf{J} := \left(H_0^1(\Omega)\right)^2 \cap H(\mathrm{div}^0;\Omega), \quad \mathbf{W} := \left(H_0^1(\Omega)\right)^2, \quad \mathbf{Q} := L_0^2(\Omega),$$

$$\mathbf{X} := H_\tau^2(\Omega) \cap L_0^2(\Omega), \quad \widetilde{\mathbf{X}} := H^1(\Omega) \cap L_0^2(\Omega), \quad \mathbf{X}_0 := H_\gamma^2(\Omega), \quad \widetilde{\mathbf{X}}_0 := H_0^1(\Omega),$$

endowed with natural Sobolev norms. Here, in addition to the standard (scalar and vector) spaces $H^1(\Omega)$ and $H_0^1(\Omega)$, we take

$$H(\mathrm{div}^0;\Omega) := \left\{\vec{v}\,\middle|\,\vec{v} \in \left(L^2(\Omega)\right)^2,\, \nabla\cdot\vec{v} = 0 \text{ in } \Omega\right\}, \quad L_0^2(\Omega) := \left\{q\,\middle|\,q \in L^2(\Omega), \int_\Omega q\,\mathrm{dX} = 0\right\},$$
$$H_\tau^2(\Omega) := \left\{\phi\,\middle|\,\phi \in H^2(\Omega), \frac{\partial\phi}{\partial\vec{n}}|_{\partial\Omega} = 0\right\}, \quad H_\gamma^2(\Omega) := \{\phi|\phi \in H^2(\Omega), \phi|_{\partial\Omega} = 0\}.$$

3.2 Steady-state visco-resistive MHD

In this paper, we consider cylindrical three-dimensional domains, $\hat{\Omega} = \Omega \times [z_0, z_1]$, where $\Omega \subset \mathbb{R}^2$ is Lipschitz, bounded and connected, which are coupled with a large incident magnetic field in the z-direction. To begin, we consider the one-fluid visco-resistive MHD model, where the dependent variables are the fluid velocity $\vec{u}$, the hydrodynamic pressure p, and the magnetic field $\vec{B}$. The equations are

$$\begin{aligned} \frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} - \nabla \cdot (T + T_M) + \nabla p &= \vec{F}, & (3.1)\\ \frac{\partial \vec{B}}{\partial t} - \nabla \times (\vec{u} \times \vec{B}) + \nabla \times (\frac{1}{Re_m} \nabla \times \vec{B}) &= \vec{G}, & (3.2)\\ \nabla \cdot \vec{u} &= 0, & (3.3)\\ \nabla \cdot \vec{B} &= 0, & (3.4) \end{aligned}$$

where $\vec{G} = -\nabla \times \vec{E}_{\text{stat}}$, and $\vec{E}_{\text{stat}}$ is the static component of the electric field. The Newtonian and magnetic stress tensors are

$$T = \frac{1}{2Re}\left[\nabla \vec{u} + \nabla \vec{u}^T\right], \text{ and } T_M = \vec{B} \otimes \vec{B} - \frac{1}{2}|\vec{B}|^2 I,$$

respectively. We define the tensor $\vec{B} \otimes \vec{B}$ component-wise as $(\vec{B} \otimes \vec{B})_{i,j} = B_i B_j$ and $\vec{F} = (\vec{f}, 0) \in \left(H^{-1}(\hat{\Omega})\right)^3$ for $\vec{f} \in \left(H^{-1}(\Omega)\right)^2$(where $H^{-1}(\Omega)$ is the dual space of $H^1(\Omega)$, and which is isomorphic to the dual space of $H_0^1(\Omega)$), $\vec{G} \in \left(L^2(\hat{\Omega})\right)^3$. Additionally, we define the standard nondimensional Reynolds number, Re, and magnetic Reynolds number, Re_m:

$$Re = \frac{\rho U L}{\nu}, \qquad Re_m = \frac{\mu_0 U L}{\eta},$$

for a characteristic velocity, U, and a characteristic length scale, L. The physical parameters, all assumed constant, are the fluid viscosity ν, the fluid density ρ, the magnetic permeability of free space μ_0, and the magnetic resistivity η.

Assuming that the domain is coupled with a large incident magnetic field in the z-direction, the resulting dynamics decouple into a two-dimensional problem over Ω with simple behaviour in the z-direction. For the tokamak pictured in Figure 3.1, this is equivalent to assuming both a large incident magnetic field in the toroidal direction as well as a large aspect-ratio, so that the curvature of the tokamak is negligible. Considering the resulting plasma behaviour over Ω (the poloidal cross-section of the tokamak), and assuming no variation in the z- (toroidal-)direction, we take $\vec{B} = (B_1(x, y), B_2(x, y), B_0)$ and $\vec{u} = (u_1(x, y), u_2(x, y), u_0)$. Then, we complete the above system with homogeneous boundary conditions on the velocity, $\vec{u} = \vec{0}$ on $\partial\Omega$, and either *perfect conductor* or *perfect insulator* boundary conditions on $\vec{B}$, $\vec{B} \cdot \vec{n} = 0$ or $\vec{B} \times \vec{n} = \vec{0}$ on $\partial\Omega$, respectively, where $\vec{n}$ denotes the outward normal vector on $\partial\Omega$.

Noting that $\nabla \cdot \vec{B} = 0$, we must have $\frac{\partial B_1}{\partial x} + \frac{\partial B_2}{\partial y} = 0$, which allows us to write $\vec{B} = \nabla \times \vec{A} + (0, 0, B_0)$, where $\vec{A} = (0, 0, A(x, y))$. A standard result (see, for example [15]), is that if $B \in \left(H^1(\hat{\Omega})\right)^3$, then $A \in H^2(\hat{\Omega})$. Consequently, we rewrite Equations (3.1)-(3.4) in terms of the vector potential, $\vec{A}$. Considering the continuum problem (3.1)-(3.4), direct calculation shows that B_0 and u_0 do not appear in the resulting equations for the other components of $\vec{B}$ and $\vec{u}$ and, so, we ignore them (by treating them as zero) in what follows.

3.2.1 $H^2(\Omega)$ weak formulation

We now introduce the weak formulation of (3.1)-(3.4) for the two-dimensional domain Ω. Writing $\vec{B} = \nabla \times \vec{A}$ for vector potential, $\vec{A}$, gives $\nabla \cdot \vec{B} = 0$ and Equation (3.4) is automatically satisfied. Thus, we no longer include it in the formulation.

A standard vector calculus identity is that if $\vec{B} \in \left(H^1(\hat{\Omega})\right)^3$,

$$\nabla \cdot (\vec{B} \otimes \vec{B} - \frac{1}{2}|\vec{B}|^2 I) = (\nabla \times \vec{B}) \times \vec{B} + (\nabla \cdot \vec{B}) \cdot \vec{B},$$

and if $\vec{B} \in \left(H^1(\hat{\Omega})\right)^3 \cap H(\text{div}^0; \hat{\Omega})$, then

$$\nabla \cdot (\vec{B} \otimes \vec{B} - \frac{1}{2}|\vec{B}|^2 I) = (\nabla \times \vec{B}) \times \vec{B}.$$

Taking $\vec{B} = (\frac{\partial A}{\partial y}, -\frac{\partial A}{\partial x}, 0)$ ensures that $\vec{B} \in \left(H^1(\hat{\Omega})\right)^3 \cap H(\text{div}^0; \hat{\Omega})$ when $A \in \mathbf{X}$, giving

$$\begin{aligned} \int_{\hat{\Omega}} \nabla \cdot (\vec{B} \otimes \vec{B} - \frac{1}{2}|\vec{B}|^2 I) \cdot \vec{V} \mathrm{d}\hat{\mathrm{X}} &= \int_{\hat{\Omega}} (\nabla \times \vec{B}) \times \vec{B} \cdot \vec{V} \mathrm{d}\hat{\mathrm{X}} \\ &= \int_{\hat{\Omega}} (-\triangle A \cdot \frac{\partial A}{\partial x}, -\triangle A \cdot \frac{\partial A}{\partial y}, 0) \cdot \vec{V} \mathrm{d}\hat{\mathrm{X}} \\ &= -(z_1 - z_0) \int_{\Omega} \triangle A \cdot (\nabla A \cdot \vec{v}) \, \mathrm{dX}, \qquad (3.5) \end{aligned}$$

for any $\vec{V} = (\vec{v}, v_3) \in \left(H^1(\hat{\Omega})\right)^3$, with $\vec{v} \in \left(H^1(\Omega)\right)^2$.

Taking $\vec{C} = \nabla \times (0, 0, \varphi)$ for $\varphi \in \mathbf{X}$, then we can rewrite the weak formulation of (3.2), discarding the time derivative,

$$\int_{\hat{\Omega}} \left[-\nabla \times (\vec{u} \times \vec{B}) \cdot \vec{C} + \nabla \times (Re_m^{-1} \nabla \times \vec{B}) \cdot \vec{C} \right] \mathrm{d}\hat{\mathrm{X}} = \int_{\hat{\Omega}} \vec{G} \cdot \vec{C} \mathrm{d}\hat{\mathrm{X}},$$

as

$$\int_\Omega -(u_1, u_2) \cdot \nabla A \cdot \triangle\varphi \, \mathrm{dX} + \int_\Omega Re_m^{-1} \triangle A \cdot \triangle\varphi \, \mathrm{dX} = \int_\Omega E^0 \cdot \triangle\varphi \mathrm{dX},$$

where E^0 is the z-component of the electrostatic part, $\vec{E}_{\text{stat}}$, and we choose E^0 so that $\int_\Omega E^0 \mathrm{dX} = 0$. We drop the common scaling of $(z_1 - z_0)$ when switching from integrals over $\hat{\Omega}$ to those over Ω. In the following, we denote $\vec{u} = (u_1(x,y), u_2(x,y))$.

Note that with $\vec{B} = (\partial A/\partial y, -\partial A/\partial x, 0)$, the perfect conductor boundary condition, $\vec{B} \cdot \vec{n} = 0$ is implied by a homogeneous Dirichlet boundary condition on A, as is included in the space $\mathbf{X}_0$, while the perfect insulator boundary condition, $\vec{B} \times \vec{n} = \vec{0}$, is implied by a homogeneous Neumann boundary condition on A, as is included in the space $\mathbf{X}$. In what follows, we state weak formulations and results for the latter case, $A \in \mathbf{X}$ (and, from Section 3.3 onwards, $A \in \widetilde{\mathbf{X}}$) as proofs for this case are slightly more technical than for $A \in \mathbf{X}_0$ (or $A \in \widetilde{\mathbf{X}}_0$). Where substantial differences occur between the two cases, we provide remarks to clarify. With homogeneous Dirichlet boundary conditions on $\vec{u}$ and perfect insulator boundary conditions on A, the weak form of (3.1)-(3.4) in two dimensions is : find $\vec{u} \in \mathbf{W}, A \in \mathbf{X}, p \in \mathbf{Q}$ such that

$$\begin{aligned} a_1(\vec{u}, \vec{v}) + c_0(\vec{u}; \vec{u}, \vec{v}) + c_1(A; \vec{v}, A) + b(p, \vec{v}) &= \langle \vec{f}, \vec{v} \rangle, & (3.6) \\ a_2(A, \varphi) - c_1(A; \vec{u}, \varphi) &= \langle E^0, \triangle\varphi \rangle, & (3.7) \\ b(q, \vec{u}) &= 0, & (3.8) \end{aligned}$$

for all $\vec{v} \in \mathbf{W}, \varphi \in \mathbf{X}, q \in \mathbf{Q}$, with $\mathcal{S}\vec{u} = \frac{1}{2}(\nabla\vec{u} + \nabla\vec{u}^T)$, where

$$\begin{aligned} a_1(\vec{u}, \vec{v}) &:= Re^{-1} \int_\Omega \mathcal{S}\vec{u} : \nabla\vec{v} \, \mathrm{dX} = Re^{-1} \int_\Omega \mathcal{S}\vec{u} : \mathcal{S}\vec{v} \, \mathrm{dX}, \\ a_2(\phi, \psi) &:= Re_m^{-1} \int_\Omega \triangle\phi \cdot \triangle\psi \, \mathrm{dX}, \\ b(q, \vec{v}) &:= -\int_\Omega q(\nabla \cdot \vec{v}) \, \mathrm{dX}, \\ c_0(\vec{w}; \vec{u}, \vec{v}) &:= \frac{1}{2} \int_\Omega (\vec{w} \cdot \nabla)\vec{u} \cdot \vec{v} \, \mathrm{dX} - \frac{1}{2} \int_\Omega (\vec{w} \cdot \nabla)\vec{v} \cdot \vec{u} \, \mathrm{dX}, \\ c_1(\psi; \vec{v}, \phi) &:= \int_\Omega \triangle\phi \cdot \nabla\psi \cdot \vec{v} \, \mathrm{dX}. \end{aligned}$$

3.2.2 Properties of the weak formulation

In this section, we briefly analyze the weak form in Equations (3.6)-(3.8), which we write as

Formulation 3.2.1. *Find* $(\vec{u}, p, A) \in \mathbf{W} \times \mathbf{Q} \times \mathbf{X}$ *such that*

$$\begin{aligned} \mathcal{A}(\vec{u}, A; \vec{v}, \varphi) + \mathcal{C}(\vec{u}, A; \vec{u}, A; \vec{v}, \varphi) + \mathcal{B}(p; \vec{v}, \varphi) &= \mathcal{L}(\vec{v}, \varphi), \qquad (3.9) \\ \mathcal{B}(q; \vec{u}, A) &= 0, \qquad (3.10) \end{aligned}$$

for all $(\vec{v}, q, \varphi) \in \mathbf{W} \times \mathbf{Q} \times \mathbf{X}$*,*

with

$$\begin{aligned} \mathcal{A}(\vec{u}, A; \vec{v}, \varphi) &:= a_1(\vec{u}, \vec{v}) + a_2(A, \varphi), \\ \mathcal{B}(q; \vec{v}, \varphi) &:= b(q, \vec{v}), \\ \mathcal{C}(\vec{w}, \psi; \vec{u}, \phi; \vec{v}, \varphi) &:= c_0(\vec{w}; \vec{u}, \vec{v}) + c_1(\psi; \vec{v}, \phi) - c_1(\psi; \vec{u}, \varphi), \\ \mathcal{L}(\vec{v}, \varphi) &:= \langle \vec{f}, \vec{v} \rangle + \langle E^0, \triangle\varphi \rangle. \end{aligned}$$

We define the product space $\mathbf{W} \times \mathbf{X}$ with the norm $|||(\vec{v}, \varphi)|||^2 := \|\vec{v}\|_1^2 + \|\varphi\|_2^2$ and define the operator norm, $|||\mathcal{L}|||_- := \sup_{(\vec{0},0) \neq (\vec{v},\varphi) \in \mathbf{J} \times \mathbf{X}} \dfrac{|\mathcal{L}(\vec{v}, \varphi)|}{|||(\vec{v}, \varphi)|||}$. Next, we consider the properties of the forms $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$.

Lemma 3.2.1. *For any* $(\vec{v}, \varphi), (\vec{w}, \psi) \in \mathbf{W} \times \mathbf{X}$*, we have*

$$\begin{aligned} \mathcal{A}(\vec{v}, \varphi; \vec{v}, \varphi) &\geq c_\alpha \min\{Re^{-1}, Re_m^{-1}\} |||(\vec{v}, \varphi)|||^2, \qquad (3.11) \\ \mathcal{A}(\vec{w}, \psi; \vec{v}, \varphi) &\leq \max\{2Re^{-1}, Re_m^{-1}\} |||(\vec{w}, \psi)||| \cdot |||(\vec{v}, \varphi)|||, \end{aligned}$$

where $c_\alpha \leq 1$ *is a constant depending only on* Ω*.*

Proof. Since $(\vec{v}, \varphi) \in \mathbf{W} \times \mathbf{X}$, we have

$$\begin{aligned} \mathcal{A}(\vec{v}, \varphi; \vec{v}, \varphi) &= Re^{-1} \int_\Omega \mathcal{S}\vec{v} : \mathcal{S}\vec{v} \, \mathrm{dX} + \int_\Omega Re_m^{-1} \triangle\varphi \cdot \triangle\varphi \, \mathrm{dX} \\ &= Re^{-1} \|\mathcal{S}\vec{v}\|_0^2 + Re_m^{-1} \|\triangle\varphi\|_0^2 \\ &\geq \beta_1 Re^{-1} \|\vec{v}\|_1^2 + \beta_2 Re_m^{-1} \|\varphi\|_2^2 \\ &\geq c_\alpha \min\{Re^{-1}, Re_m^{-1}\} |||(\vec{v}, \varphi)|||^2, \end{aligned}$$

where $c_\alpha = \min\{\beta_1, \beta_2\}$, β_1 comes from Korn's Inequality [7, Corollary 11.2.22], and β_2 comes from a regularity argument [15, Chapter I, Theorem 1.10]. This gives the coercivity of $\mathcal{A}$.

For continuity,

$$\begin{aligned}
\mathcal{A}(\vec{u},\psi;\vec{v},\varphi) &= Re^{-1}\int_\Omega \mathcal{S}\vec{u} : \mathcal{S}\vec{v}\,\mathrm{dX} + Re_m^{-1}\int_\Omega \triangle\psi\cdot\triangle\varphi\,\mathrm{dX} \\
&\le 2Re^{-1}\|\vec{u}\|_1\|\vec{v}\|_1 + Re_m^{-1}\|\psi\|_2\|\varphi\|_2 \\
&\le \max\{2Re^{-1}, Re_m^{-1}\}|||(\vec{u},\psi)|||\cdot|||(\vec{v},\varphi)|||,
\end{aligned}$$

via the Cauchy-Schwarz inequality. □

Remark 3.2.1. *If $\varphi \in \mathbf{X}_0$, then $\|\Delta\varphi\|_0^2 \ge \beta_2\|\varphi\|_2^2$ also holds (see [15,* Chapter *I,* Theorem *1.8]).*

We state two Lemmas that follow directly from the standard Compact Imbedding Theorem for Sobolev spaces (see, e.g., [15], Theorem I.1.2), showing the trilinear forms c_0 and c_1 are well defined.

Lemma 3.2.2. *If $\vec{u},\vec{v},\vec{w} \in \left(H^1(\Omega)\right)^2$, then*

$$|c_0(\vec{w};\vec{u},\vec{v})| \le C_0\|\vec{w}\|_{0,4}\cdot\|\nabla\vec{u}\|_0\cdot\|\vec{v}\|_{0,4} \le C_0\|\vec{w}\|_1\cdot\|\vec{u}\|_1\cdot\|\vec{v}\|_1, \tag{3.12}$$

where C_0 is a constant depending only on Ω.

Lemma 3.2.3. *If $\psi,\phi \in H^2(\Omega)$ and $\vec{v} \in \left(H^1(\Omega)\right)^2$, then*

$$|c_1(\psi;\vec{v},\phi)| \le C_1\|\nabla\psi\|_{0,4}\cdot\|\triangle\phi\|_0\cdot\|\vec{v}\|_{0,4} \le C_1\|\psi\|_2\cdot\|\phi\|_2\cdot\|\vec{v}\|_1, \tag{3.13}$$

where C_1 is a constant depending only on Ω.

Lemma 3.2.4. *For any $\vec{w},\vec{u},\vec{v} \in \mathbf{W}$ and $\psi,\phi,\varphi \in \mathbf{X}$, the trilinear form $\mathcal{C}$ has the following properties*

$$|\mathcal{C}(\vec{w},\psi;\vec{u},\phi;\vec{v},\varphi)| \le C_c|||(\vec{w},\psi)|||\cdot|||(\vec{u},\phi)|||\cdot|||(\vec{v},\varphi)|||, \tag{3.14}$$

where C_c is a constant only depending on Ω. Furthermore,

$$\mathcal{C}(\vec{w},\psi;\vec{v},\varphi;\vec{v},\varphi) = 0. \tag{3.15}$$

Proof. The continuity bound follows directly from inequalities (3.12) and (3.13). That $\mathcal{C}(\vec{w}, \psi; \vec{v}, \varphi; \vec{v}, \varphi) = 0$ follows directly from its definition, and those of c_0 and c_1. □

The form $b(q, \vec{v})$ is continuous and satisfies the following inf-sup condition

$$\inf_{0 \neq q \in \mathbf{Q}} \sup_{\vec{0} \neq \vec{v} \in \mathbf{W}} \frac{b(q, \vec{v})}{\|\vec{v}\|_1 \|q\|_0} \geq \Gamma > 0, \tag{3.16}$$

where Γ is a constant depending only on Ω [15, Chapter I.5.1].

The form $\mathcal{B}$ is obviously continuous:

$$|\mathcal{B}(q; \vec{v}, \varphi)| \leq C_b \|q\|_0 \|\vec{v}\|_1 \leq C_b \|q\|_0 |||(\vec{v}, \varphi)|||,$$

for all $(\vec{v}, q, \varphi) \in \mathbf{W} \times \mathbf{Q} \times \mathbf{X}$, with a constant $C_b > 0$. Furthermore, it inherits the inf-sup condition from b.

Lemma 3.2.5. *There exists a constant $\Gamma > 0$ depending only on Ω, such that*

$$\sup_{(\vec{0},0) \neq (\vec{v},\varphi) \in \mathbf{W} \times \mathbf{X}} \frac{\mathcal{B}(q; \vec{v}, \varphi)}{|||(\vec{v}, \varphi)|||} \geq \Gamma \|q\|_0,$$

for all $q \in \mathbf{Q}$.

Proof. Since

$$\mathcal{B}(q; \vec{v}, \varphi) = b(q, \vec{v}),$$

we have

$$\sup_{(\vec{0},0) \neq (\vec{v},\varphi) \in \mathbf{W} \times \mathbf{X}} \frac{\mathcal{B}(q; \vec{v}, \varphi)}{|||(\vec{v}, \varphi)|||} \geq \sup_{\vec{0} \neq \vec{v} \in \mathbf{W}} \frac{b(q, \vec{v})}{\|\vec{v}\|_1} \geq \|q\|_0 \cdot \Gamma,$$

where the last inequality follows directly from (3.16). □

3.2.3 Existence and uniqueness of solutions

From [15], we quote the main theorem that we will apply to this weak formulation.

Theorem 3.2.1 ([15], Theorem IV.1.3). *Let V be a separable Hilbert space with the norm $\|\cdot\|_V$, l be a linear functional in the dual space V' and, for $w \in V$, the mapping $(u, v) \to a(w; u, v)$ be a bilinear continuous form on $V \times V$. Assume that the following hold:*

- *the bilinear form $a(w;v,v)$ is uniformly V-coercive with respect to w, i.e., there exists a constant $\alpha > 0$ such that*

$$a(w;v,v) \geq \alpha \|v\|_V^2, \quad \forall v, w \in V.$$

- *there exists a continuous and monotonically increasing function $L : \mathbb{R}_+ \to \mathbb{R}_+$ such that for all $\mu > 0$*

$$|a(w_1;u,v) - a(w_2;u,v)| \leq L(\mu)\|u\|_V \|v\|_V \|w_1 - w_2\|_V,$$

$$\forall u, v \in V, \quad w_1, w_2 \in S_\mu = \{w \in V; \|w\|_V \leq \mu\}.$$

- *the linear function l and α satisfy*

$$\frac{\|l\|_{V'}}{\alpha^2} \cdot L(\|l\|_{V'}/\alpha) < 1.$$

Then the problem: find $u \in V$ such that

$$a(u;u,v) = l(v), \quad \forall v \in V,$$

has a unique solution that satisfies the stability bound $\|u\|_V \leq \alpha^{-1}\|l\|_{V'}$.

Theorem 3.2.2. *Let $\vec{f} \in \left(H^{-1}(\Omega)\right)^2$ and $E^0 \in L^2(\Omega)$, and assume that*

$$\frac{C_c|||\mathcal{L}|||_-}{c_\alpha^2 \min\{Re^{-2}, Re_m^{-2}\}} < 1, \tag{3.17}$$

where c_α comes from (3.11), and C_c comes from (3.14). Then, there exists a unique solution $(\vec{u}, p, A)$ in $\mathbf{W} \times \mathbf{Q} \times \mathbf{X}$ of Formulation 3.2.1. Furthermore, we have the stability bounds

$$|||(\vec{u}, A)||| \leq \frac{|||\mathcal{L}|||_-}{c_\alpha \min\{Re^{-1}, Re_m^{-1}\}}$$

and

$$\|p\|_0 \leq \Gamma^{-1}\left[||\vec{f}||_{-1} + 2Re^{-1}\|\vec{u}\|_1 + C_0\|\vec{u}\|_1^2 + C_1\|A\|_1^2\right],$$

where C_0 comes from (3.12), and C_1 comes from (3.13).

Proof. We first apply Theorem 3.2.1 to Formulation 3.2.1 restricted to $(\vec{u}, A) \in \mathbf{J} \times \mathbf{X}$, satisfying the constraint in Equation (3.10). We note that $\mathbf{J} \times \mathbf{X}$ is separable, since $\mathbf{J}$ and $\mathbf{X}$ are closed subsets of $\big(H^1(\Omega)\big)^2$ and $H^2(\Omega)$ respectively, and $\big(H^1(\Omega)\big)^2$ and $H^2(\Omega)$ are separable Hilbert Spaces.

For any $(\vec{w}, \psi)$, define the mapping $((\vec{u}, \phi), (\vec{v}, \varphi)) \to \mathcal{A}_1(\vec{w}, \psi; \vec{u}, \phi, \vec{v}, \varphi)$, where

$$\mathcal{A}_1(\vec{w}, \psi; \vec{u}, \phi, \vec{v}, \varphi) = \mathcal{A}(\vec{u}, \phi; \vec{v}, \varphi) + \mathcal{C}(\vec{w}, \psi; \vec{u}, \phi; \vec{v}, \varphi).$$

From inequalities (3.11) and (3.15), we have

$$\begin{aligned} |\mathcal{A}_1(\vec{w}, \psi; &\vec{v}, \varphi; \vec{v}, \varphi)| = |\mathcal{A}(\vec{v}, \varphi; \vec{v}, \varphi) + \mathcal{C}(\vec{w}, \psi; \vec{v}, \varphi; \vec{v}, \varphi)| = |\mathcal{A}(\vec{v}, \varphi; \vec{v}, \varphi)| \\ &\geq c_\alpha \text{min}\{Re^{-1}, Re_m^{-1}\}|||(\vec{v}, \varphi)|||^2 \quad \forall (\vec{w}, \psi), (\vec{v}, \varphi) \in \mathbf{J} \times \mathbf{X}. \end{aligned}$$

Finally, linearity in the first argument of $\mathcal{C}$ and inequality (3.14) give

$$\begin{aligned} |\mathcal{A}_1(\vec{w}_1, \psi_1; \vec{u}, \phi; \vec{v}, \varphi) &- \mathcal{A}_1(\vec{w}_2, \psi_2; \vec{u}, \phi; \vec{v}, \varphi)| \\ &= |\mathcal{C}((\vec{w}_1, \psi_1; \vec{u}, \phi; \vec{v}, \varphi) - \mathcal{C}(\vec{w}_2, \psi_2; \vec{u}, \phi; \vec{v}, \varphi)| \\ &= |\mathcal{C}(\vec{w}_1 - \vec{w}_2, \psi_1 - \psi_2; \vec{u}, \phi; \vec{v}, \varphi)| \\ &\leq C_c|||(\vec{w}_1 - \vec{w}_2, \psi_1 - \psi_2)||| \cdot |||(\vec{u}, \phi)||| \cdot |||(\vec{v}, \varphi)|||, \end{aligned}$$

$\forall (\vec{w}_1, \psi_1), (\vec{w}_2, \psi_2), (\vec{u}, \phi), (\vec{v}, \varphi) \in \mathbf{J} \times \mathbf{X}$. In the notation of Theorem 3.2.1, this gives $L(\mu) = C_c$, where C_c comes from (3.14).

Thus, by Theorem 3.2.1, assumption (3.17) proves existence of a unique solution to Formulation 3.2.1 restricted to $\mathbf{J} \times \mathbf{X}$. Let $(\vec{u}, A) \in \mathbf{J} \times \mathbf{X}$ be that unique solution, which satisfies the stability bound stated.

By the inf-sup condition in Equation (3.16), there also exists a unique solution of the following problem: find $p \in \mathbf{Q}$ such that

$$\begin{aligned} b(p, \vec{v}) = \mathcal{B}(p; \vec{v}, \varphi) &= \mathcal{L}(\vec{v}, \varphi) - \mathcal{A}(\vec{u}, A; \vec{v}, \varphi) - \mathcal{C}(\vec{u}, A; \vec{u}, A; \vec{v}, \varphi), \\ &= \langle \vec{f}, \vec{v} \rangle - a_1(\vec{u}, \vec{v}) - c_0(\vec{u}; \vec{u}, \vec{v}) - c_1(A; \vec{v}, A), \end{aligned}$$

for all $\vec{v} \in \mathbf{W} \setminus \mathbf{J}$ [15, Theorem IV.1.4].

From the inf-sup condition, we have

$$\begin{aligned}\Gamma\|p\|_0 &\leq \sup_{\vec{0}\neq\vec{v}\in\mathbf{W}} \frac{b(q,\vec{v})}{\|\vec{v}\|_1} \\ &= \sup_{\vec{0}\neq\vec{v}\in\mathbf{W}} \frac{\langle\vec{f},\vec{v}\rangle - a_1(\vec{u},\vec{v}) - c_0(\vec{u};\vec{u},\vec{v}) - c_1(A;\vec{v},A)}{\|\vec{v}\|_1}.\end{aligned}$$

Combining this with Equations (3.12) and (3.13), we obtain the bound on p. □

Any conforming mixed finite-element discretization of (3.9) and (3.10) necessarily requires the use of H^2-conforming elements for $A \in \mathbf{X}$, such as Argyris triangle elements, or Bogner-Fox-Schmit elements [9]. By using the antisymmetric form of c_0 in the weak formulation, existence and uniqueness of the solution to the discretized form of Formulation 3.2.1 follows immediately, so long as an appropriate inf-sup stable finite-element pair is used for the velocity and pressure unknowns. While these approximations have been thoroughly studied, particularly for fourth-order problems, their use also poses some additional difficulties for implementation and efficient solution of the resulting linearized systems. Thus, we next consider a modified approach using H^1-conforming elements, following [2, 10].

3.3 Uncurled formulation of MHD

Introducing the vector potential into Equation (3.2) leads to the bilinear form $a_2(\phi,\psi)$, which requires H^2-conforming elements for discretization. Notice, however, that, in the steady-state case, Equation (3.2) can be rewritten as $\nabla\times(-\vec{u}\times\vec{B} + Re_m^{-1}\nabla\times\vec{B}) = -\nabla\times\vec{E}_{\text{stat}}$, which can be simplified into a first-order equation in $\vec{B}$, resulting in a second-order equation in A. Using this in place of (3.2), we derive an "uncurled" weak formulation: find $(\vec{u},A)\in\mathbf{W}\times\widetilde{\mathbf{X}}, p\in\mathbf{Q}$ such that

$$a_1(\vec{u},\vec{v}) + c_0(\vec{u};\vec{u},\vec{v}) + \widetilde{c}_1(A;\vec{v},A) + b(p,\vec{v}) = \langle\vec{f},\vec{v}\rangle, \tag{3.18}$$

$$\widetilde{a}_2(A,\psi) + \widetilde{c}_2(A;\vec{u},\psi) = \langle -E^0,\psi\rangle, \tag{3.19}$$

$$b(q,\vec{u}) = 0, \tag{3.20}$$

for all $(\vec{v}, \psi) \in \mathbf{W} \times \widetilde{\mathbf{X}}, q \in \mathbf{Q}$, where

$$\begin{aligned}
\widetilde{a}_2(\phi, \psi) &:= Re_m^{-1} \int_\Omega \nabla \phi \cdot \nabla \psi \, \mathrm{dX}, \\
\widetilde{c}_1(\phi; \vec{v}, A) &:= \frac{1}{2}\left\langle \left(\frac{\partial A}{\partial y} \cdot \frac{\partial \phi}{\partial y} - \frac{\partial A}{\partial x} \cdot \frac{\partial \phi}{\partial x}, -\left[\frac{\partial A}{\partial x} \cdot \frac{\partial \phi}{\partial y} + \frac{\partial A}{\partial x} \cdot \frac{\partial \phi}{\partial y} \right] \right), \frac{\partial \vec{v}}{\partial x} \right\rangle_0 \\
&\quad + \frac{1}{2}\left\langle \left(-\left[\frac{\partial A}{\partial x} \cdot \frac{\partial \phi}{\partial y} + \frac{\partial A}{\partial x} \cdot \frac{\partial \phi}{\partial y} \right], \frac{\partial A}{\partial x} \cdot \frac{\partial \phi}{\partial x} - \frac{\partial A}{\partial y} \cdot \frac{\partial \phi}{\partial y} \right), \frac{\partial \vec{v}}{\partial y} \right\rangle_0, \\
\widetilde{c}_2(\phi; \vec{u}, \psi) &:= \int_\Omega \vec{u} \cdot \nabla \phi \cdot \psi \, \mathrm{dX}.
\end{aligned}$$

Note, we now integrate by parts on the stress tensor in (3.1) since $c_1(A, \vec{v}, A)$ is obviously ill-defined if $A \notin H^2(\Omega)$. The corresponding term in (3.7) becomes $\widetilde{c}_2(\phi; \vec{u}, \psi)$ due to the "uncurling" of (3.2). This is the formulation used in [2, 10]; in [2], an inf-sup stable finite-element method pair is used for discretization of $\vec{u}$ and p, while a stabilized pair was used in [10]. Neither of these papers considered theoretical analysis of this formulation, which we do here.

The analysis below shows that, in contrast to the formulation considered above, this formulation does not directly yield unique solutions under the classical theory. To address this, we augment analysis of the continuum weak form with that at the discrete level. We separately consider the well-posedness of the Newton linearizations in Section 4.

3.3.1 Mixed variational formulation

Extending the bilinear form $\mathcal{B}$ to act on $\widetilde{\mathbf{X}}$ gives

$$\widetilde{\mathcal{B}}(q; \vec{v}, \psi) := b(q, \vec{v}),$$

where the only difference between $\mathcal{B}$ and $\widetilde{\mathcal{B}}$ is that they act on $\mathbf{X}$ and $\widetilde{\mathbf{X}}$, respectively. The mixed variational formulation in (3.18)-(3.20) can then be rewritten as

Formulation 3.3.1. *Find $(\vec{u}, p, A) \in \mathbf{W} \times \mathbf{Q} \times \widetilde{\mathbf{X}}$ such that*

$$\begin{aligned}
\widetilde{\mathcal{A}}(\vec{u}, A; \vec{v}, \psi) + \widetilde{\mathcal{C}}(\vec{u}, A; \vec{u}, A; \vec{v}, \psi) + \widetilde{\mathcal{B}}(p; \vec{v}, \psi) &= \widetilde{\mathcal{L}}(\vec{v}, \psi), \qquad (3.21) \\
\widetilde{\mathcal{B}}(q; \vec{u}, A) &= 0,
\end{aligned}$$

for all $(\vec{v}, q, \psi) \in \mathbf{W} \times \mathbf{Q} \times \widetilde{\mathbf{X}}$, *where*

$$\begin{aligned} \widetilde{\mathcal{A}}(\vec{u}, A; \vec{v}, \psi) &:= a_1(\vec{u}, \vec{v}) + \widetilde{a}_2(A, \psi), \\ \widetilde{\mathcal{C}}(\vec{w}, \phi; \vec{u}, A; \vec{v}, \psi) &:= c_0(\vec{w}; \vec{u}, \vec{v}) + \widetilde{c}_1(\psi; \vec{v}, A) + \widetilde{c}_2(\psi; \vec{u}, \phi), \\ \widetilde{\mathcal{L}}(\vec{v}, \psi) &:= \langle \vec{f}, \vec{v} \rangle + \langle -E^0, \psi \rangle. \end{aligned}$$

For our later analysis, we note some properties of the terms in this formulation.

Lemma 3.3.1. *Let* $\psi, \phi \in H^1(\Omega)$ *and* $\vec{u} \in \left(H^1(\Omega)\right)^2$, *then*

$$|\widetilde{c}_2(\phi; \vec{u}, \psi)| \le C \|\vec{u}\|_{0,4} \cdot \|\nabla \phi\|_0 \cdot \|\psi\|_{0,4} \le C \|\vec{u}\|_1 \cdot \|\phi\|_1 \cdot \|\psi\|_1,$$

where C *is a constant depending only on* Ω.

We define the product space $\mathbf{W} \times \widetilde{\mathbf{X}}$ with the norm

$$\|(\vec{v}, \psi)\|_1^2 := \|\vec{v}\|_1^2 + \|\psi\|_1^2,$$

and consider ellipticity of $\widetilde{\mathcal{A}}$ on this product space.

Lemma 3.3.2. *For any* $(\vec{v}, \varphi) \in \mathbf{W} \times \widetilde{\mathbf{X}}$, *we have*

$$\begin{aligned} \widetilde{\mathcal{A}}(\vec{v}, \varphi; \vec{v}, \varphi) &\ge \widetilde{c}_\alpha \min\{Re^{-1}, Re_m^{-1}\} \|(\vec{v}, \varphi)\|_1^2, \\ \widetilde{\mathcal{A}}(\vec{w}, \psi; \vec{v}, \varphi) &\le \max\{2Re^{-1}, Re_m^{-1}\} \|(\vec{w}, \psi)\|_1 \|(\vec{v}, \varphi)\|_1, \end{aligned}$$

where $\widetilde{c}_\alpha \le 1$ *is a constant depending only on* Ω.

Proof. The proof follows that of Lemma 3.2.1, substituting Friedrichs' Inequality [7],

$$\|\nabla \varphi\|_0^2 \ge \xi \|\varphi\|_1^2, \quad \forall \varphi \in \widetilde{X},$$

for the regularity argument used in the coercivity bound. □

Remark 3.3.1. *For* $\varphi \in \widetilde{X}_0$, *the standard Friedrichs' Inequality also gives the coercivity result.*

The form $\widetilde{\mathcal{B}}$ is again continuous:

$$|\widetilde{\mathcal{B}}(q; \vec{v}, \psi)| \le C_b \|q\|_0 \|\vec{v}\|_1 \le \widetilde{C}_b \|q\|_0 \|(\vec{v}, \psi)\|_1, \tag{3.22}$$

for all $(\vec{v}, q, \psi) \in \mathbf{W} \times \mathbf{Q} \times \widetilde{\mathbf{X}}$, with a constant $\widetilde{C}_b > 0$, and inherits the inf-sup condition from b:

Lemma 3.3.3. *There exists a constant* $\Gamma > 0$ *depending only on* Ω *such that*

$$\sup_{(\vec{0},0)\neq(\vec{v},\psi)\in\mathbf{W}\times\widetilde{\mathbf{X}}} \frac{\widetilde{\mathcal{B}}(q;\vec{v},\psi)}{\|(\vec{v},\psi)\|_1} \geq \Gamma\|q\|_0, \tag{3.23}$$

for all $q \in \mathbf{Q}$.

The form $\widetilde{\mathcal{C}}$ no longer satisfies the desired zero property $\widetilde{\mathcal{C}}(\vec{w},\phi;\vec{v},\psi;\vec{v},\psi) = 0$. Also, $\widetilde{c}_1$ is not obviously continuous in $H^1(\Omega)$. Consequently, classical results, such as Theorem 3.2.1, cannot be directly applied to establish existence and uniqueness of solutions to Formulation 3.3.1. Instead, we tackle this question indirectly, leveraging the result given in Theorem 3.2.2 for Formulation 3.2.1.

3.3.2 Relationship between solutions of the two formulations

Formulations 3.2.1 and 3.3.1 offer two weak formulations of the steady-state visco-resistive MHD problem, (3.1)-(3.4). A natural question is whether the solutions of these two formulations are the same. Here, we provide conditions under which this is the case. These results follow naturally from the fact that $\mathbf{X} \subseteq \widetilde{\mathbf{X}}$.

Theorem 3.3.1. *Assume that* Ω *has* $C^{1,1}$ *boundary and* $(\vec{u}, p, A) \in \mathbf{W} \times \mathbf{Q} \times \mathbf{X}$ *is a solution of Formulation 3.2.1, then* $(\vec{u}, p, A)$ *is also a solution of Formulation 3.3.1.*

Proof. Let $(\vec{u}, p, A) \in \mathbf{W} \times \mathbf{Q} \times \mathbf{X}$ be a solution of Formulation 3.2.1. According to (3.5), the following equality holds

$$\int_\Omega \triangle A \cdot (\nabla A \cdot \vec{v})\,\mathrm{dX} = -\int_\Omega (\nabla \cdot T_M) \cdot \vec{v}\,\mathrm{dX} = \int_\Omega T_M : \nabla\vec{v}\,\mathrm{dX}, \quad \forall \vec{v} \in \mathbf{W}.$$

Then, (3.6) is the same as (3.18). For any $\psi \in \widetilde{\mathbf{X}} \subseteq L^2(\Omega)$, there exists $\varphi \in \mathbf{X}$ such that $\triangle\varphi = \psi$ (see [15, Chapter I, Theorem 1.10]). In (3.7),

$$\int_\Omega -\vec{u} \cdot \nabla A \cdot \triangle\varphi\,\mathrm{dX} + \int_\Omega Re_m^{-1} \triangle A \cdot \triangle\varphi\,\mathrm{dX} = \langle E^0, \triangle\varphi\rangle, \quad \forall\varphi \in \mathbf{X},$$

taking $\triangle\varphi = \psi$ implies (3.19). So $(\vec{u}, p, A)$ is also a solution of Formulation 3.3.1. □

Remark 3.3.2. *When $\psi \in \widetilde{\mathbf{X}}_0$, [15](*Chapter *I,* Theorem *1.8) gives the existence of $\varphi \in \mathbf{X}_0$ such that $\Delta\varphi = \psi$ in Ω.*

Theorem 3.3.2. *Assume that Ω has $C^{1,1}$ boundary and $(\vec{u}, p, A) \in \mathbf{W} \times \mathbf{Q} \times \widetilde{\mathbf{X}}$ is a solution of Formulation 3.3.1 and that this solution is smooth enough such that $A \in H^2(\Omega)$. Then, $(\vec{u}, p, A)$ is also a solution of Formulation 3.2.1.*

Proof. Let $(\vec{u}, p, A) \in \mathbf{W} \times \mathbf{Q} \times \widetilde{\mathbf{X}}$ be a solution of Formulation 3.3.1. Since $A \in H^2(\Omega)$ and $\vec{v} \in \left(H_0^1(\Omega)\right)^2$, the following equality holds

$$\int_\Omega T_M : \nabla\vec{v}\,\mathrm{dX} = -\int_\Omega (\nabla\cdot T_M)\cdot\vec{v}\,\mathrm{dX} = \int_\Omega \triangle A\cdot(\nabla A\cdot\vec{v})\,\mathrm{dX}, \quad \forall\vec{v}\in\mathbf{W}.$$

Then, (3.18) is the same as (3.6). Furthermore,

$$\int_\Omega \left[\vec{u}\cdot\nabla A\cdot\psi + Re_m^{-1}\nabla A\cdot\nabla\psi\right]\mathrm{dX} = -\int_\Omega E^0\cdot\psi\,\mathrm{dX}, \quad \forall\psi\in\widetilde{\mathbf{X}},$$

can be rewritten as

$$\int_\Omega \nabla A\cdot\nabla\psi\mathrm{dX} = -Re_m\int_\Omega \left(E^0 + \vec{u}\cdot\nabla A\right)\cdot\psi\,\mathrm{dX}, \quad \forall\psi\in\widetilde{\mathbf{X}}.$$

Since $\int_\Omega E^0\mathrm{dX} = 0$ and $\int_\Omega \vec{u}\cdot\nabla A\,\mathrm{dX} = -\int_\Omega(\nabla\cdot\vec{u})A\,\mathrm{dX} + \int_{\partial\Omega}(\vec{u}\cdot\vec{n})A\,\mathrm{dX} = 0$, we have $\int_\Omega(E^0 + \vec{u}\cdot\nabla A)\,\mathrm{dX} = 0$. Using the results of Proposition 1.2 of [15], the weak form of finding $w \in \widetilde{\mathbf{X}}$ such that

$$\int_\Omega \nabla w\cdot\nabla\psi\mathrm{dX} = \int_\Omega -Re_m\left(E^0 + \vec{u}\cdot\nabla A\right)\cdot\psi\,\mathrm{dX}, \quad \forall\psi\in\widetilde{\mathbf{X}}, \tag{3.24}$$

has a unique solution, and if $w \in H^2(\Omega)$, then it is the strong solution of the Neumann problem,

$$\left\{\begin{array}{rcl} -\Delta w & = & -Re_m(E^0 + \vec{u}\cdot\nabla A), \quad \text{in } \Omega, \\ \frac{\partial w}{\partial\vec{n}} & = & 0, \quad \text{on } \partial\Omega, \\ \int_\Omega w\,\mathrm{dX} & = & 0. \end{array}\right. \tag{3.25}$$

Thus, from [15, Chapter I, Theorem 1.10], we have that (3.25) has a unique solution, $w \in H_\tau^2(\Omega)$, which is given by $w = A$, implying that $-\vec{u}\cdot\nabla A + Re_m^{-1}\triangle A = E^0$. For $\varphi \in H_\tau^2(\Omega)$, multiplying both sides by $\triangle\varphi$ and integrating yields (3.7). So $(\vec{u}, p, A)$ is also a solution of Formulation 3.2.1. □

Remark 3.3.3. *Using the Lax-Milgram Lemma, problem (3.24) considered over $H_0^1(\Omega)$, has one and only one solution, $w \in H^1(\Omega)$. By Theorem 1.8 of [15], if $w \in H^2(\Omega)$, then it is the strong solution of the corresponding Dirichlet problem. Thus, Theorem 3.3.2 also applies in the case when $A \in \widetilde{\mathbf{X}}_0$.*

Theorem 3.3.3. *Assume that Ω has $C^{1,1}$ boundary and (3.17) holds. Then, Formulation 3.3.1 has at least one solution $(\vec{u}, p, A) \in \mathbf{W} \times \mathbf{Q} \times \widetilde{\mathbf{X}}$, which is the unique solution of Formulation 3.2.1. Furthermore, if all of the solutions of Formulation 3.3.1 satisfy $(\vec{u}, p, A) \in \mathbf{W} \times \mathbf{Q} \times \mathbf{X}$, then Formulation 3.2.1 and Formulation 3.3.1 have the same solution, and the solution is unique.*

Proof. Since (3.17) holds, Theorem 3.2.2 states that Formulation 3.2.1 has a unique solution $(\vec{u}, p, A)$. According to Theorem 3.3.1, $(\vec{u}, p, A)$ is also a solution of Formulation 3.3.1.

If $A \in \mathbf{X}$, Theorem 3.3.2 states that the solution $(\vec{u}, p, A)$ of Formulation 3.3.1 is also a solution of Formulation 3.2.1. However, since (3.17) holds, Formulation 3.2.1 has only one solution. This means that Formulation 3.3.1 has only one solution. □

3.3.3 Finite-element discretization

In this subsection, we introduce a mixed finite-element approximation of the uncurled formulation and discuss the convergence rates that are obtained under some standard smoothness assumptions.

Let $\mathcal{T}_h$ be a quasi-uniform family of subdivisions that partition Ω into triangles or quadrilaterals, $\mathcal{K}$, with diameters bounded by h [15, Chapter I, Definitions A.2]. Based on these meshes, we construct a series of finite-element spaces satisfying

$$\mathbf{W}_h \subset \mathbf{W}, \mathbf{X}_h \subset \widetilde{\mathbf{X}}, \mathbf{Q}_h \subset \mathbf{Q}.$$

The discretization of Formulation 3.3.1 can be written as

Formulation 3.3.2. *Find $(\vec{u}_h, p_h, A_h) \in \mathbf{W}_h \times \mathbf{Q}_h \times \mathbf{X}_h$ such that*

$$\begin{aligned}
\widetilde{\mathcal{A}}(\vec{u}_h, A_h; \vec{v}, \psi) + \widetilde{\mathcal{C}}(\vec{u}_h, A_h; \vec{u}_h, A_h; \vec{v}, \psi) + \widetilde{\mathcal{B}}(p_h; \vec{v}, \psi) &= \widetilde{\mathcal{L}}(\vec{v}, \psi), \\
\widetilde{\mathcal{B}}(q; \vec{u}_h, A_h) &= 0,
\end{aligned}$$

for all $(\vec{v}, q, \psi) \in \mathbf{W}_h \times \mathbf{Q}_h \times \mathbf{X}_h$.

In the following, we assume that Formulation 3.3.2 is well-posed. In this paper, we consider the 2D problem and assume that the solution $A \in H^{s+1}(\Omega)$, $s > 1$, then we have

$$|\nabla A|_\infty \le C_A \|\nabla A\|_s \le C_A \|A\|_{s+1}, \quad s > 1. \tag{3.26}$$

More details can be found in [1, Theorem IV4.12].

Theorem 3.3.4. *Assume that* (3.17) *holds and that* $(\vec{u}, A)$ *is the solution of Formulation 3.3.1 with* $\vec{u} \in \left(H^1(\Omega)\right)^2$ *and* $A \in H^{s+1}(\Omega)$ *for* $s > 1$*, and* $(\vec{u}_h, A_h)$ *is the solution of Formulation 3.3.2 satisfying* $\|\vec{u}_h\|_1 + |\nabla A_h|_\infty \le d$*, where* d *is a constant. Then,*

$$\|(\vec{u} - \vec{u}_h, A - A_h)\|_1 \le C \left(\inf_{(\vec{v},\psi) \in \mathbf{W}_h \times \mathbf{X}_h} \|(\vec{u} - \vec{v}, A - \psi)\|_1 + \inf_{q \in \mathbf{Q}_h} \|p - q\|_0 \right),$$

with a constant $C > 0$*, depending on* d*, for sufficiently small values of* Re *and* Re_m*.*

Proof. Subtracting Formulation 3.3.2 from Equality (3.21), we have

$$\begin{aligned} &\widetilde{\mathcal{A}}(\vec{u} - \vec{u}_h, A - A_h; \vec{v}, \psi) + \widetilde{\mathcal{C}}(\vec{u} - \vec{u}_h, A - A_h; \vec{u}, A; \vec{v}, \psi) + \widetilde{\mathcal{C}}(\vec{u}_h, A_h; \vec{u} - \vec{u}_h, A - A_h; \vec{v}, \psi) \\ &+ \widetilde{\mathcal{B}}(p - p_h; \vec{v}, \psi) = 0, \end{aligned} \tag{3.27}$$

for all $(\vec{v}, \psi) \in \mathbf{W}_h \times \mathbf{X}_h$.

From (3.27), for any $\vec{v}$ such that $b(q, \vec{v}) = 0$ for all $q \in \mathbf{Q}_h$, we have

$$\begin{aligned} & \widetilde{\mathcal{A}}(\vec{v} - \vec{u}_h, \psi - A_h; \vec{v} - \vec{u}_h, \psi - A_h) + \widetilde{\mathcal{C}}(\vec{v} - \vec{u}_h, \psi - A_h; \vec{u}, A; \vec{v} - \vec{u}_h, \psi - A_h) \\ & + \widetilde{\mathcal{C}}(\vec{u}_h, A_h; \vec{v} - \vec{u}_h, \psi - A_h; \vec{v} - \vec{u}_h, \psi - A_h) \\ = \quad & \widetilde{\mathcal{A}}(\vec{v} - \vec{u}, \psi - A; \vec{v} - \vec{u}_h, \psi - A_h) + \widetilde{\mathcal{C}}(\vec{v} - \vec{u}, \psi - A; \vec{u}, A; \vec{v} - \vec{u}_h, \psi - A_h) \\ & + \widetilde{\mathcal{C}}(\vec{u}_h, A_h; \vec{v} - \vec{u}, \psi - A; \vec{v} - \vec{u}_h, \psi - A_h) - \widetilde{\mathcal{B}}(p - p_h; \vec{v} - \vec{u}_h, \psi - A_h), \end{aligned} \tag{3.28}$$

For such a $\vec{v}$, we also have

$$\widetilde{\mathcal{B}}(p - p_h; \vec{v} - \vec{u}_h, \psi - A_h) = \widetilde{\mathcal{B}}(p - q; \vec{v} - \vec{u}_h, \psi - A_h), \tag{3.29}$$

for all $q \in \mathbf{Q}_h$.

From (3.28) and (3.29), we have the estimate

$$\begin{aligned}\text{r.h.s of (3.28)} &\leq \|(\vec{v}-\vec{u}_h,\psi-A_h)\|_1\big[\max\{2Re^{-1},Re_m^{-1}\}\|(\vec{v}-\vec{u},\psi-A)\|_1\\ &\quad +C\|(\vec{v}-\vec{u},\psi-A)\|_1\big(\|\vec{u}\|_1+C_A\|A\|_{s+1}\big)\\ &\quad +C\|(\vec{v}-\vec{u},\psi-A)\|_1\big(\|\vec{u}_h\|_1+|\nabla A_h|_\infty\big)+\widetilde{C}_b\|p-q\|_0\big]\\ &\leq C_r\|(\vec{v}-\vec{u}_h,\psi-A_h)\|_1\big(\|(\vec{u}-\vec{v},A-\psi)\|_1+\|p-q\|_0\big), \quad (3.30)\end{aligned}$$

where $C_r=\max\{2Re^{-1},Re_m^{-1}\}+2C\cdot\max\{\|\vec{u}\|_1+C_A\|A\|_{s+1,2},\|\vec{u}_h\|_1+|\nabla A_h|_\infty\}+\widetilde{C}_b$, C_A comes from (3.26), and $\widetilde{C}_b$ comes from (3.22). Since $(\vec{u},A)$ is the solution of the continuous problem and $\vec{u}\in H^1(\Omega)$ and $A\in H^{s+1}(\Omega)$, then $\|\vec{u}\|_1+C_A\|A\|_{s+1,2}$ can be bounded by some constant. By assumption, so can $\|\vec{u}_h\|_1+|\nabla A_h|_\infty$.

Similarly,

$$\begin{aligned}\text{l.h.s of (3.28)} &\geq \widetilde{c}_\alpha\min\{Re^{-1},Re_m^{-1}\}\cdot\|(\vec{v}-\vec{u}_h,\psi-A_h)\|_1^2\\ &\quad -C\|(\vec{v}-\vec{u}_h,\psi-A_h)\|_1^2\cdot\big(\|\vec{u}\|_1+\|A\|_{s+1,2}\big)\\ &\quad -C\|(\vec{v}-\vec{u}_h,\psi-A_h)\|_1^2\cdot\big(\|\vec{u}_h\|_1+|\nabla A_h|_\infty\big)\\ &\geq C_l\|(\vec{v}-\vec{u}_h,\psi-A_h)\|_1^2, \quad (3.31)\end{aligned}$$

where $C_l=\widetilde{c}_\alpha\min\{Re^{-1},Re_m^{-1}\}-2C\cdot\max\{\|\vec{u}\|_1+C_A\|A\|_{s+1,2},\|\vec{u}_h\|_1+|\nabla A_h|_\infty\}$ and $\widetilde{c}_\alpha$ comes from Lemma 3.3.2. Here, we assume that $\widetilde{c}_\alpha\min\{Re^{-1},Re_m^{-1}\}$ is large enough such that $C_l\geq\frac{\widetilde{c}_\alpha}{2}\min\{Re^{-1},Re_m^{-1}\}$.

According to (3.30) and (3.31), we have the following estimate

$$\|(\vec{v}-\vec{u}_h,\psi-A_h)\|_1\leq C\bigg(\|(\vec{u}-\vec{v},A-\psi)\|_1+\|p-q\|_0\bigg),$$

where $C=C_r/C_l$. Furthermore,

$$\begin{aligned}\|(\vec{u}-\vec{u}_h,A-A_h)\|_1 &\leq \sqrt{2}\big(\|(\vec{u}-\vec{v},A-\psi)\|_1+\|(\vec{v}-\vec{u}_h,\psi-A_h)\|_1\big)\\ &\leq C\|(\vec{u}-\vec{v},A-\psi)\|_1+C\|p-q\|_0.\end{aligned}$$

Now, let $\vec{v}\in W_h$ be arbitrary and take $\vec{w}\in W_h$ to be a solution of

$$b(q,\vec{w})=b(q,\vec{u}-\vec{v}),\quad \forall q\in\mathbf{Q}_h.$$

Since b satisfies an inf-sup condition and a continuity condition, then there exists a solution to this problem such that

$$\|\vec{w}\|_1 \leq C\|\vec{u} - \vec{v}\|_1,$$

and such that $b(q, \vec{w} + \vec{v}) = 0$ for all $q \in \mathbf{Q}_h$. By the triangle inequality and using the result above, we then have

$$\begin{aligned}
\|(\vec{u} - \vec{u}_h, A - A_h)\|_1 &\leq C\|(\vec{u} - (\vec{w} + \vec{v}), A - \psi)\|_1 + C\|p - q\|_0 \\
&\leq C\|(\vec{u} - \vec{v}, A - \psi)\|_1 + C\|\vec{w}\|_1 + C\|p - q\|_0 \\
&\leq C\|(\vec{u} - \vec{v}, A - \psi)\|_1 + C\|p - q\|_0.
\end{aligned}$$

□

To give a more precise definition of our finite-element approximations, define, on an element $\mathcal{K}$,

$$\mathcal{P}_k(\mathcal{K}) := \text{the space of polynomials of degree} \leq k,$$

and let $\mathcal{C}^0(\bar{\Omega})$ denote the standard space of continuous functions on $\bar{\Omega}$. The finite-element spaces are defined as

$$\begin{aligned}
\mathbf{W_h} &:= \{\vec{v}_h \in \mathcal{C}^0(\bar{\Omega}) : \vec{v}_h|_{\mathcal{K}} \in (\mathcal{P}_{k+1})^2, \quad \forall \mathcal{K} \in \mathcal{T}_h\}, \\
\mathbf{Q_h} &:= \{q_h \in \mathcal{C}^0(\bar{\Omega}) : q_h|_{\mathcal{K}} \in \mathcal{P}_k, \quad \forall \mathcal{K} \in \mathcal{T}_h\}, \\
\mathbf{X_h} &:= \{\psi_h \in \mathcal{C}^0(\bar{\Omega}) : \psi_h|_{\mathcal{K}} \in \mathcal{P}_{k+1}, \quad \forall \mathcal{K} \in \mathcal{T}_h\},
\end{aligned}$$

where $k \geq 1$. In what follows, we make standard approximation assumptions for generalized Taylor-Hood mixed finite-elements on either triangular or quadrilateral elements in 2D [6, Proposition 8.2.2] as well as for the scalar space $\mathbf{X_h}$.

Assumption 3.3.1. *Let $k \geq 1, s > 1$. Assume that*

$$\inf_{\vec{v}_h \in \mathbf{W}_h} \|\vec{u} - \vec{v}_h\|_1 + \inf_{q_h \in \mathbf{Q}_h} \|p - q_h\|_0 \leq C h^{\min\{s,k+1\}} \left[\|u\|_{s+1} + \|p\|_s\right],$$

for all $(\vec{u}, p) \in H^{s+1}(\Omega)^2 \times H^s(\Omega)$ *and that*

$$\inf_{\psi_h \in \mathbf{X}_h} \|A - \psi_h\|_1 \le Ch^{\min\{s,k+1\}} \|A\|_{s+1},$$

for all $A \in H^{s+1}(\Omega)$*.*

Corollary 3.3.1. *Let* $(\vec{u}_h, A_h) \in \mathbf{W}_h \times \mathbf{X}_h$ *be the finite-element approximation in Formulation 3.3.2. Under the assumptions of Theorem 3.3.4 and Assumption 3.3.1, we have the error bound*

$$\|(\vec{u} - \vec{u}_h, A - A_h)\|_1 \le Ch^{\min\{s,k+1\}} \left[\|\vec{u}\|_{s+1} + \|p\|_s + \|A\|_{s+1}\right].$$

3.4 Newton's method

Since the weak formulation in (3.18)-(3.20) is nonlinear, we use Newton's method to derive a linearized system. As expected, the discrete form leads to a saddle-point problem [5, 8]. Here, we focus on the linearization steps and show that the resulting systems are well-posed, and that the solutions converge to that of the original problem, under certain assumptions.

3.4.1 Newton linearizations

Let $\mathbf{S} = \mathbf{W} \times \widetilde{\mathbf{X}}$ with the norm $\|W\|_1^2 = \|\vec{v}\|_1^2 + \|\psi\|_1^2$ for all $W = (\vec{v}, \psi) \in \mathbf{S}$. For convenience, we denote the solutions of Formulations 3.3.1 and 3.3.2 as (U^*, p^*), (U_h^*, p_h^*), respectively.

For $U = (\vec{u}, A), W = (\vec{v}, \psi) \in \mathbf{S}$, define the following operators:

$$\begin{aligned}
\mathcal{L}_1(\vec{u}, A, p)[\vec{v}] &:= a_1(\vec{u}, \vec{v}) + b(p, \vec{v}) + c_0(\vec{u}; \vec{u}, \vec{v}) + \widetilde{c}_1(A; \vec{v}, A) - \langle \vec{f}, \vec{v} \rangle, \\
\mathcal{L}_2(\vec{u}, A, p)[\psi] &:= \widetilde{a}_2(A, \psi) + \widetilde{c}_2(A; \vec{u}, \psi) + \langle E^0, \psi \rangle, \\
\mathcal{L}_3(\vec{u}, A, p)[q] &:= -b(q, \vec{u}).
\end{aligned}$$

Problem (3.18)-(3.20) is equivalent to

$$\begin{aligned} \mathcal{L}_1(\vec{u}, A, p)[\vec{v}] &= 0, \qquad \forall \vec{v} \in \mathbf{W}, && (3.32) \\ \mathcal{L}_2(\vec{u}, A, p)[\psi] &= 0, \qquad \forall \psi \in \widetilde{\mathbf{X}}, && (3.33) \\ \mathcal{L}_3(\vec{u}, A, p)[q] &= 0, \qquad \forall q \in \mathbf{Q}. \end{aligned}$$

Since the variational system contains nonlinearities in both (3.32) and (3.33), we linearize the above forms. Let $\vec{u}_k, A_k, p_k$ be the current approximations for $\vec{u}, A, p$, respectively and $\delta\vec{u}_k = \vec{u}_{k+1} - \vec{u}_k, \delta A = A_{k+1} - A_k, \delta p = p_{k+1} - p_k$ be the update to the approximations, then the linear systems that arise within Newton's method are denoted

$$\begin{bmatrix} \mathcal{L}_{1,\vec{u}} & \mathcal{L}_{1,A} & \mathcal{L}_{1,p} \\ \mathcal{L}_{2,\vec{u}} & \mathcal{L}_{2,A} & 0 \\ \mathcal{L}_{3,\vec{u}} & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\vec{u} \\ \delta A \\ \delta p \end{bmatrix} = - \begin{bmatrix} \mathcal{L}_1 \\ \mathcal{L}_2 \\ \mathcal{L}_3 \end{bmatrix},$$

where each of the system components is evaluated at $\vec{u}_k, A_k, p_k$. That is

$$\begin{aligned} \mathcal{L}_{1,\vec{u}}[\vec{v}] \cdot \delta\vec{u} = \frac{\partial}{\partial\vec{u}}(\mathcal{L}_1(\vec{u}_k, A_k, p_k)[\vec{v}])[\delta\vec{u}] &= a_1(\delta\vec{u}, \vec{v}) + c_0(\vec{u}_k; \delta\vec{u}, \vec{v}) + c_0(\delta\vec{u}; \vec{u}_k, \vec{v}), \\ \mathcal{L}_{1,A}[\vec{v}] \cdot \delta A = \frac{\partial}{\partial A}(\mathcal{L}_1(\vec{u}_k, A_k, p_k)[\vec{v}])[\delta A] &= \hat{a}(A_k; \vec{v}, \delta A), \\ \mathcal{L}_{1,p}[\vec{v}] \cdot \delta p = \frac{\partial}{\partial p}(\mathcal{L}_1(\vec{u}_k, A_k, p_k)[\vec{v}])[\delta p] &= b(\delta p, \vec{v}), \\ \mathcal{L}_{2,\vec{u}}[\psi] \cdot \delta\vec{u} = \frac{\partial}{\partial\vec{u}}(\mathcal{L}_2(\vec{u}_k, A_k, p_k)[\psi])[\delta\vec{u}] &= \widetilde{c}_2(A_k; \delta\vec{u}, \psi), \\ \mathcal{L}_{2,A}[\psi] \cdot \delta A = \frac{\partial}{\partial A}(\mathcal{L}_2(\vec{u}_k, A_k, p_k)[\psi])[\delta A] &= \widetilde{a}_2(\delta A, \psi) + \widetilde{c}_2(\delta A; \vec{u}_k, \psi), \\ \mathcal{L}_{3,\vec{u}}[q] \cdot \delta\vec{u} = \frac{\partial}{\partial\vec{u}}(\mathcal{L}_3(\vec{u}_k, A_k, p_k)[q])[\delta\vec{u}] &= b(q, \delta\vec{u}), \end{aligned}$$

where

$$\begin{aligned} \hat{a}(A_k; \vec{v}, A) \;:=\; & \left\langle \left(\frac{\partial A_k}{\partial y} \cdot \frac{\partial A}{\partial y} - \frac{\partial A_k}{\partial x} \cdot \frac{\partial A}{\partial x}, -\left[\frac{\partial A_k}{\partial x} \cdot \frac{\partial A}{\partial y} + \frac{\partial A}{\partial x} \cdot \frac{\partial A_k}{\partial y} \right] \right), \frac{\partial \vec{v}}{\partial x} \right\rangle_0 \\ & + \left\langle \left(-\left[\frac{\partial A_k}{\partial x} \cdot \frac{\partial A}{\partial y} + \frac{\partial A}{\partial x} \cdot \frac{\partial A_k}{\partial y} \right], \frac{\partial A_k}{\partial x} \cdot \frac{\partial A}{\partial x} - \frac{\partial A_k}{\partial y} \cdot \frac{\partial A}{\partial y} \right), \frac{\partial \vec{v}}{\partial y} \right\rangle_0. \end{aligned}$$

Define the following forms:

$$\begin{aligned}
\mathfrak{A}(U_k; U, W) &:= \hat{a}(A_k; \vec{v}, A) + a_1(\vec{u}, \vec{v}) + \widetilde{a}_2(A, \psi) + c_0(\vec{u}_k; \vec{u}, \vec{v}) + c_0(\vec{u}; \vec{u}_k, \vec{v}) \\
&\quad + \widetilde{c}_2(A_k; \vec{u}, \psi) + \widetilde{c}_2(A; \vec{u}_k, \psi), \\
\mathfrak{B}(W, q) &:= b(q, \vec{v}), \\
F(U_k, p_k; W) &:= \widetilde{\mathcal{L}}(\vec{v}, \psi) - \widetilde{\mathcal{A}}(\vec{u}_k, A_k; \vec{v}, \psi) - \widetilde{\mathcal{C}}(\vec{u}_k, A_k; \vec{u}_k, A_k; \vec{v}, \psi) - \widetilde{\mathcal{B}}(p_k; \vec{v}, \psi), \\
G(U_k; q) &:= -\mathfrak{B}(U_k, q).
\end{aligned}$$

For Newton's method applied in a linearize-then-discretize formulation, we consider the finite-element spaces $\mathbf{S}_h = \mathbf{W}_h \times \mathbf{X}_h \subset \mathbf{S}$ and $\mathbf{Q}_h \subset \mathbf{Q}$. Given an approximation, $(U_{h,k}, p_{h,k}) \in \mathbf{S}_h \times \mathbf{Q}_h$, the discrete Newton update is given by

Formulation 3.4.1. *Find $(\delta U_h, \delta p_h) \in \mathbf{S}_h \times \mathbf{Q}_h$ such that*

$$\begin{aligned}
\mathfrak{A}(U_{h,k}; \delta U_h, W_h) + \mathfrak{B}(W_h, \delta p_h) &= F(U_{h,k}, p_{h,k}; W_h), && (3.34) \\
\mathfrak{B}(\delta U_h, q_h) &= G(U_{h,k}; q_h), && (3.35)
\end{aligned}$$

for all $(W_h, q_h) \in \mathbf{S}_h \times \mathbf{Q}_h$. Let $U_{h,k+1} = U_{h,k} + \delta U_h, p_{h,k+1} = p_{h,k} + \delta p_h$.

For simplicity, throughout the remainder of this section, we drop the subscript h. Since we consider finite-element approximations $\vec{u}_k$ and A_k, we denote $C_{sup} = \sup\limits_{(x,y)\in\Omega} |\nabla \vec{u}_k|, D_{sup} = \sup\limits_{(x,y)\in\Omega} |\nabla A_k|$, and $M_{sup} = \sup\limits_{(x,y)\in\Omega} |\vec{u}_k|$, and note that they are all finite quantities.

Lemma 3.4.1. *$\mathfrak{A}(U_k; U, W)$ and $\mathfrak{B}(W, q)$ are continuous on $\mathbf{S}_h$ and $\mathbf{Q}_h$ for the norms $\|\cdot\|_1$ and $\|\cdot\|_0$.*

Proof. For the continuity of $\mathfrak{A}(U_k; U, W)$, observe that

$$\begin{aligned}
|\mathfrak{A}(U_k; U, W)| &\leq |\hat{a}(A_k; \vec{v}, A) + a_1(\vec{u}, \vec{v}) + \widetilde{a}_2(A, \psi) + c_0(\vec{u}_k; \vec{u}, \vec{v}) + c_0(\vec{u}; \vec{u}_k, \vec{v}) \\
&\quad + \widetilde{c}_2(A_k; \vec{u}, \psi) + \widetilde{c}_2(A; \vec{u}_k, \psi)|.
\end{aligned}$$

Next, consider the above summands separately. First, note that

$$|\hat{a}(A_k; \vec{v}, A)| \leq 2D_{sup} \|\nabla A\|_0 \|\nabla \vec{v}\|_0.$$

Recalling the definitions of the rest of these terms, we obtain the following estimates

$$\begin{aligned}
|a_1(\vec{u},\vec{v})| &\leq CR_e^{-1}\|\vec{u}\|_1\|\vec{v}\|_1,\\
|\widetilde{a}_2(A,\psi)| &\leq Re_m^{-1}\|A\|_1\|\psi\|_1,\\
|c_0(\vec{u}_k;\vec{u},\vec{v})| &\leq \frac{M_{sup}}{2}\left(\||\nabla\vec{u}\|_0\|\vec{v}\|_0+\|\vec{u}\|_0\|\nabla\vec{v}\|_0\right),\\
|c_0(\vec{u};\vec{u}_k,\vec{v})| &\leq \frac{1}{2}\left(C_{sup}\|\vec{u}\|_0\|\vec{v}\|_0+M_{sup}\|\vec{u}\|_0\|\nabla\vec{v}\|_0\right),\\
|\widetilde{c}_2(A_k;\vec{u},\psi)| &\leq D_{sup}\|\vec{u}\|_0\|\psi\|_0,\\
|\widetilde{c}_2(A;\vec{u}_k,\psi)| &\leq M_{sup}\|\nabla A\|_0\|\psi\|_0.
\end{aligned}$$

An application of the Cauchy-Schwarz inequality shows that

$$|\mathfrak{A}(U_k;U,W)|\leq C\|U\|_1\|W\|_1,$$

where C is a constant depending on C_{sup}, D_{sup}, M_{sup}, Re and Re_m.

Continuity of $\mathfrak{B}(W,q)$ holds by standard arguments. □

Lemma 3.4.2. *$F(U_k,p_k;W)$ and $G(U_k;q)$ are bounded linear functionals on $\mathbf{S}_h$ and $\mathbf{Q}_h$, respectively.*

Proof. The components of $F(U_k,p_k;W)$ can be bounded as in the proof of Lemma 3.4.1. Since, additionally,

$$\begin{aligned}
|\langle E^0,\psi\rangle_0| &\leq \|E^0\|_0\|\psi\|_0,\\
|\langle\vec{f},\vec{v}\rangle| &\leq \|\vec{f}\|_{-1}\|\vec{v}\|_1,
\end{aligned}$$

and $b(q,\vec{v})$ is continuous, we have

$$|F(U_k,p_k;W)|\leq C\|W\|_1,$$

where C is a constant only depending on the norms of U_k and p_k.

By Hölder's inequality, we have

$$|G(U_k;q)|=|-\mathfrak{B}(U_k,q)|\leq\|U_k\|_1\|q\|_0,$$

implying that $G(U_k;q)$ is bounded. □

To illustrate the existence and uniqueness of solutions to the system given by (3.34) and (3.35), we now give conditions under which $\mathfrak{A}(U_k; U, W)$ is a coercive and continuous bilinear form. When $\mathfrak{B}(W, q)$ is continuous and weakly coercive in the chosen finite-element spaces, existence and uniqueness of solutions to the discretized Newton linearization is automatic.

Theorem 3.4.1. *Let Re and Re_m be small enough such that*

$$\min\{\alpha_1 Re^{-1}, \alpha_2 Re_m^{-1}\} - (C_{sup} + D_{sup} + \frac{M_{sup}}{2}) > 0,$$

where α_1, α_2 are constants defined below, and C_{sup}, D_{sup}, and M_{sup} are as given above. Then, there exists a constant $\gamma > 0$ depending on U_k and Ω such that

$$\mathfrak{A}(U_k; W, W) \geq \gamma \|W\|_1^2, \quad \forall W \in \mathbf{S}_h. \tag{3.36}$$

Proof. By standard arguments,

$$\langle \nabla\vec{v} + \nabla\vec{v}^T, \nabla\vec{v}\rangle_0 \geq \alpha_1 \|\vec{v}\|_1^2, \quad \forall \vec{v} \in \mathbf{W}_h,$$

where α_1 is a constant depending only on Ω (see [7], Corollary 11.2.22) and

$$\langle \nabla\psi, \nabla\psi\rangle_0 \geq \alpha_2 \|\psi\|_1^2, \quad \forall \psi \in \mathbf{X}_h,$$

where α_2 depends only on Ω (see Friedrichs' inequality [7]).

The remaining terms in $\mathfrak{A}(U_k; W, W)$ can be bounded as in the proof of Lemma 3.4.1, giving

$$\begin{aligned}
\mathfrak{A}(U_k; W, W) &\geq \alpha_1 Re^{-1}\|\vec{v}\|_1^2 + \alpha_2 Re_m^{-1}\|\psi\|_1^2 - 2D_{sup}\|\nabla\psi\|_0\|\nabla\vec{v}\|_0 \\
&\quad - M_{sup}\|\vec{v}\|_0\|\nabla\vec{v}\|_0 - \frac{C_{sup}}{2}\|\vec{v}\|_0^2 - \frac{M_{sup}}{2}\|\vec{v}\|_0\|\nabla\vec{v}\|_0 \\
&\quad - D_{sup}\|\vec{v}\|_0\|\psi\|_0 - M_{sup}\|\nabla\psi\|_0\|\psi\|_0 \\
&\geq \min\{\alpha_1 Re^{-1}, \alpha_2 Re_m^{-1}\}\|W\|_1^2 - \frac{2C_{sup} + 6D_{sup} + 5M_{sup}}{4}\|W\|_1^2 \\
&= (\gamma_1 - \gamma_2)\|W\|_1^2,
\end{aligned}$$

where $\gamma_1 = \min\{\alpha_1 Re^{-1}, \alpha_2 Re_m^{-1}\}, \gamma_2 = (2C_{sup}+6D_{sup}+5M_{sup})/4$. Let $\gamma = \gamma_1 - \gamma_2 > 0$. Thus, $\mathfrak{A}(U_k; W, W)$ is coercive. □

Remark 3.4.1. *Since the standard Friedrichs' inequality applies for $\psi \in \widetilde{\mathbf{X}}_0$, the coercivity bound will also hold for the appropriate finite-element space in the case of perfect conductor boundary conditions.*

Assumption 3.4.1. *There exists a constant $\Gamma_s > 0$ depending on Ω such that*

$$\inf_{0 \neq q \in \mathbf{Q}_h} \sup_{\vec{0} \neq \vec{v} \in \mathbf{W}_h} \frac{b(q, \vec{v})}{\|\vec{v}\|_1 \|q\|_0} \geq \Gamma_s > 0. \tag{3.37}$$

Remark 3.4.2. *The major difference between (3.23) and (3.37) is that the inf-sup condition must be satisfied on the discrete space. There is, however, no restriction on the discrete space chosen to approximate A. Choosing a pair of spaces for which the discrete inf-sup condition (3.37) holds is well-known to be a delicate matter, and seemingly natural choices of velocity and pressure approximation do not always work [13]. For example, the simplest globally continuous approximations, using linear or bilinear elements for both velocity and pressure on triangles or quadrilaterals, respectively (the so-called $P_1 - P_1$ and $Q_1 - Q_1$ approximations), are unstable. In general, care must be taken to make the velocity space rich enough compared to the pressure space, otherwise the discrete solution will be "over-constrained". Any stable element pair for the Navier-Stoke equations (e.g., $P_2 - P_1$ or $Q_2 - Q_1$ Taylor-Hood elements) can be used for $\vec{u}$ and p (see [6, 13, 14, 15]) to satisfy (3.37).*

Theorem 3.4.2. *Under the assumptions of Theorem 3.4.1 and Assumption 3.4.1, there is a unique solution to Formulation 3.4.1.*

Proof. Following Theorem 1.2 of [15, Chapter III], Lemmas 3.4.1, 3.4.2, and Theorem 3.4.1 prove the result. □

3.4.2 Solvability of stabilized discretizations

In this subsection, we give a solvability condition for stabilized finite-element methods, since our analysis is also suitable for this setting. From Formulation 3.4.1, the matrix equations that result from a stabilized finite-element discretization have the following block form:

$$\mathcal{M}x = \begin{bmatrix} K & Z & B \\ Y & D & 0 \\ B^T & 0 & -T \end{bmatrix} \begin{bmatrix} x_{\vec{u}} \\ x_A \\ x_p \end{bmatrix} = \begin{bmatrix} \mathfrak{f}_{\vec{u}} \\ \mathfrak{f}_A \\ \mathfrak{f}_p \end{bmatrix}, \tag{3.38}$$

where $x_{\vec{u}}, x_A$, and x_p are the discrete Newton corrections for $\vec{u}, A$, and p, respectively, and $\mathfrak{f}_{\vec{u}}, \mathfrak{f}_A$, and $\mathfrak{f}_p$ are the corresponding blocks of the residual, while T is the stabilization term.

Let

$$\hat{K} = \begin{bmatrix} K & Z \\ Y & D \end{bmatrix}, \hat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, x_{\hat{\vec{u}}} = \begin{bmatrix} x_{\vec{u}} \\ x_A \end{bmatrix}, \mathfrak{f}_{\hat{\vec{u}}} = \begin{bmatrix} \mathfrak{f}_{\vec{u}} \\ \mathfrak{f}_A \end{bmatrix}.$$

Then, Equation (3.38) can be rewritten as

$$\mathcal{M}x = \begin{bmatrix} \hat{K} & \hat{B} \\ \hat{B}^T & -T \end{bmatrix} \begin{bmatrix} x_{\hat{\vec{u}}} \\ x_p \end{bmatrix} = \begin{bmatrix} \mathfrak{f}_{\hat{\vec{u}}} \\ \mathfrak{f}_p \end{bmatrix}, \tag{3.39}$$

where $\hat{K} \in \mathbb{R}^{n\times n}$, $\hat{B} \in \mathbb{R}^{n\times m}$, $\mathfrak{f}_{\hat{\vec{u}}} \in \mathbb{R}^n$, $\mathfrak{f}_p \in \mathbb{R}^m$ and $m \leq n$.

Lemma 3.4.3. *Under the assumptions of Theorem 3.4.1, $\hat{K}$ is positive definite.*

Proof. This is a consequence of (3.36). □

With homogeneous Dirichlet boundary conditions on $\vec{v} \in \mathbf{W}$, $b(p, \vec{v}) = 0$ for all $\vec{v} \in \mathbf{W}$ implies that the pressure, p, is a constant. When using a nodal finite-element basis, $\text{Span}\{\vec{1}\} \subset \text{Ker}(B)$ is a natural consequence of this. If the two spaces are equal, the resulting pressure is unique up to constants. When a discrete inf-sup condition (as in (3.37)) does not hold, $\text{Ker}(B) \neq \text{Span}\{\vec{1}\}$. However, we have the following condition that guarantees the solvability of the stabilized method, and gives insight into the construction of T.

Theorem 3.4.3. *Under the assumptions of Theorem 3.4.1, let $S = -(T + \hat{B}^T\hat{K}^{-1}\hat{B})$ be the Schur complement of $\hat{K}$ in $\mathcal{M}$, with T symmetric and positive semidefinite. If* $\text{Ker}(T) \cap \text{Ker}(B) \subseteq \text{Span}\{\vec{1}\}$, *then* $\text{Ker}(S) \subseteq \text{Span}\{\vec{1}\}$.

Proof. Since $\hat{K}$ is positive definite, $\hat{K}^{-1}$ is also positive definite. This implies that $p^T\hat{B}^T\hat{K}^{-1}\hat{B}p \geq 0$ with equality if and only if $Bp = 0$. On the other hand, because T is symmetric positive semidefinite, $\text{Ker}(S) = \text{Ker}(T) \cap \text{Ker}(B)$. □

This theorem tells us that (3.39) is well-posed if the stabilized pressure Schur Complement, S, is a positive semi-definite matrix with the following stability condition:

$$\text{Ker}(S) \subseteq \text{Span}\{\vec{1}\}.$$

The important consequence of Theorem 3.4.3 is that any stabilization approach that is suitable for the Stokes equations is also suitable in this context, since $\hat{K}$ does not enter the intersecting kernels condition. In particular, standard approaches for equal-order $Q_1 - Q_1$ approximations of velocity and pressure can be used, including diffusion stabilization and pressure-projection [12, 13]. Thus, the analysis above can be applied to discretization approaches similar to those in [10], which uses diffusion-type stabilization of the pressure equation (although we note that [10] also makes use of additional stabilization for the case when the Reynolds numbers are not small, which is not considered here). Based on the above discussions, we give the natural result.

Theorem 3.4.4. *Under the assumptions of Theorem 3.4.3, the stabilized discrete Newton approximation of Formulation 3.3.2 yields a unique solution with a pressure that is unique up to constants..*

We note here that, for both the stable and stabilized cases, the assumptions of Theorem 3.4.1 could be relaxed with the use of appropriate stabilized finite-elements for the convection-diffusion parts of the weak form, as was done in [10]. The general conclusions of Theorems 3.4.2 and 3.4.4 would naturally still hold in this case, notably that any standard mixed finite-element space for Stokes or Navier-Stokes can be used for the velocity and pressures, and an independent choice can be made for the potential, A.

3.4.3 Convergence of Newton's method

Finally, under much more restrictive assumptions, we give a local convergence analysis of Newton's method at the discrete level. Define $\|U\|_{1,\infty} := \max\{\|\vec{u}\|_{1,\infty}, \|A\|_{1,\infty}\}$ and $\mathbf{D}(U;r) = \{W : \|W - U\|_1 < r\}$ and assume the following.

Assumption 3.4.2. *Assume the conditions of Corollary 3.3.1 hold; furthermore, assume the solution U_h^* of Formulation 3.3.2 satisfies*

$$\kappa_h^* = \|U_h^*\|_{1,\infty} < \gamma_1,$$

where $\gamma_1 = \min\{\alpha_1 Re^{-1}, \alpha_2 Re_m^{-1}\}$ is from Theorem 3.4.1.

Assumption 3.4.3. *Assume that there exists $r_1 > 0$ such that for any initial iterate $U_k \in \mathbf{D}(U_h^*; r_1)$ Newton's method converges to the unique solution of Formulation 3.3.2* *and converges quadratically.*

Recalling constants γ_1, γ_2 from the proof of Theorem 3.4.1,

$$\gamma_2 = (2C_{sup} + 6D_{sup} + 5M_{sup})/4 < 4 \cdot \max\{C_{sup}, D_{sup}, M_{sup}\} < 4\|U_k\|_{1,\infty},$$

gives

$$\mathfrak{A}(U_k; W, W) > (\gamma_1 - 4\|U_k\|_{1,\infty})\|W\|_1^2.$$

Thus, if $\|U_k\|_{1,\infty} < \dfrac{\gamma_1}{4}$, then $\mathfrak{A}(U_k; W, W)$ is coercive.

Lemma 3.4.4. *Assume that $U \in \mathbf{S}_h$ and $\|U\|_{1,\infty} = \kappa_h$. Then,*

$$\|W\|_{1,\infty} \leq \kappa_h + C_1 h^{-1} r, \quad \forall W \in \mathbf{D}(U; r) \cap \mathbf{S}_h,$$

where C_1 is a constant depending on Ω.

Proof. According to the standard inverse inequality [7, Theorem IV.5.11],

$$\|U\|_{1,\infty} \leq C_1 h^{-1} \|U\|_1, \quad \forall U \in \mathbf{S}_h,$$

where C_1 is a constant. By the triangle inequality, for $W \in \mathbf{D}(U; r) \cap \mathbf{S}_h$

$$\begin{aligned} \|W\|_{1,\infty} &\leq \|U\|_{1,\infty} + \|W - U\|_{1,\infty} \\ &\leq \kappa_h + C_1 h^{-1} \|W - U\|_1 \\ &\leq \kappa_h + C_1 h^{-1} r. \end{aligned}$$

□

Remark 3.4.3. *Lemma 3.4.4 indicates that if we take $U_k \in \mathbf{D}(U; r_2)$, for $r_2 = \frac{h(\gamma_1/4 - \kappa_h^*)}{C_1}$, then $\mathcal{A}(U_k; W, W)$ is always coercive.*

If for the stabilized case, we have the same approximation result as in Theorem 3.3.1, then the next convergence theorem is not only true for stable element approximations, but also for the stabilized case.

Theorem 3.4.5. *Under Assumptions of Theorem 3.4.2 or Theorem 3.4.4, and Assumptions 3.4.2 and 3.4.3, for any initial $U_0 \in \mathbf{D}(U_h^*; r^*)$, $r^* = \min\{r_1, r_2\}$, the* *sequence $\{U_k\}$ produced by Newton's method is both well-defined and converges to the solution of Formulation 3.3.2.*

Proof. Since $U_0 \in \mathbf{D}(U_h^*; r^*)$, then according to Lemma 3.4.4, Formulation 3.4.1 has a unique solution for every U_k. By the triangle inequality, we have

$$\|U_k - U^*\|_1 \leq \|U_k - U_h^*\|_1 + \|U_h^* - U^*\|_1. \tag{3.40}$$

According to Assumptions 3.4.2 and 3.4.3, (3.40) goes to zero. □

Remark 3.4.1. *Conditions that guarantee convergence of Newton's method for finite-element discretizations of MHD in 3D can be found, for example, in [17].*

3.5 Numerical results

To demonstrate both the finite-element convergence and performance of Newton's method for this formulation, we consider the Hartmann flow test problem on the domain $\left[-\frac{1}{2}, \frac{1}{2}\right]^2$. For this problem, we have an analytical solution, given by $\vec{u} = (u_1, 0)$ and $\vec{B} = (B_1, B_2)$ with

$$\begin{aligned} u_1(x,y) &= \frac{1}{2\tanh(Ha/2)}\sqrt{\frac{Re}{Re_m}}\left(1 - \frac{\cosh(yHa)}{\cosh(Ha/2)}\right), \\ B_1(x,y) &= \frac{\sinh(yHa)}{2\sinh(Ha/2)} - y, \\ B_2(x,y) &= 1, \\ p(x,y) &= -x - \frac{1}{2}\left(B_1(x,y)\right)^2, \end{aligned}$$

where the Hartmann number is given by $Ha = \sqrt{ReRe_m}$. Increasing Ha leads to increased coupling between the velocity and magnetic field components of the solution, which is seen in [2] to lead to difficulties with some preconditioners for the discretized and linearized equations. In the numerical results that follow, we fix $Re = Re_m = Ha$. From this expression, we compute $A(x,y)$ such that $B_1(x,y) = \frac{\partial A}{\partial y}$ and $B_2(x,y) = -\frac{\partial A}{\partial x}$. For this solution, we have non-homogeneous conductor boundary conditions on $\vec{B}$, which we implement with suitable non-homogeneous Dirichlet boundary conditions on $A(x,y)$.

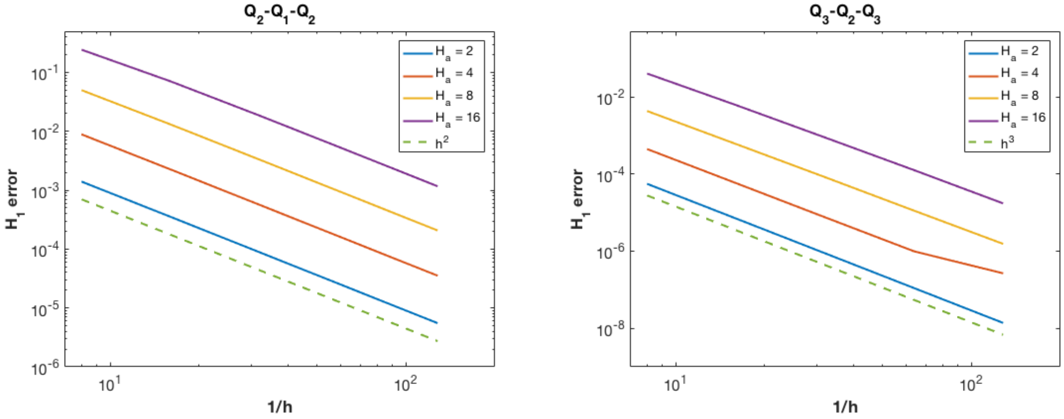

Figure 3.2: H^1 approximation error, $(\|\vec{u}-\vec{u}_h\|_1^2 + \|A - A_h\|_1^2)^{1/2}$, for finite-element solution of Hartmann test problem on uniform quadrilateral meshes with meshwidth h. At left, error for approximation with velocities and potential in Q_2 and pressure in Q_1, at right, error for approximation with velocities and potential in Q_3 and pressure in Q_2.

Figure 3.2 shows finite-element convergence for this problem with varying Ha and mesh-size h. We solve the problem using a linearize-then-discretize formulation, starting from an initial guess that matches the non-homogeneous Dirichlet boundary conditions, but is zero for all variables inside the domain. The discretization is done in deal.II [3, 4], with each linearization solved using a direct solver (UMFPACK [11]), and the nonlinear iteration stopped when the vector ℓ^2-norm, scaled by the mesh-size h, of the nonlinear residual or that of the Newton update is less than 10^{-8}. These results are presented in the setting of Corollary 3.3.1, using (generalized) Taylor-Hood elements for the velocity and pressure, and matching the degree of the velocity space for the potential. The numerical results presented here agree quite well with Corollary 3.3.1, with $\mathcal{O}(h^2)$ errors observed for approximation of velocities and potential in Q_2 and pressure in Q_1 and $\mathcal{O}(h^3)$ errors observed for approximation with velocities and potential in Q_3 and pressure in Q_2. For the range of Hartmann numbers considered in these figures, no difficulties are seen with convergence either of the nonlinear iteration or the finite-element approximations; convergence is seen within 4 to 7 Newton steps for all Hartmann numbers and all meshes. For larger Hartmann numbers, we did observe convergence issues with Newton's method.

3.6 Conclusions

In this paper, we present a theoretical analysis of the weak formulations of a steady-state visco-resistive vector-potential MHD formulation. Under certain conditions, we prove the uniqueness and existence of the solutions. Furthermore, we show that the solutions of the curled and uncurled formulations are the same, under some conditions. From this point of view, using the uncurled formulation to approximate the MHD problem is reasonable and meaningful. A mixed finite-element approximation of the uncurled formulation is discussed. The convergence rates obtained under some standard smoothness assumptions have been analysed and show that it is a suitable option. Thus, using Newton stepping and a stable Stokes finite-element method pair plus any space for A yields a convergent solution scheme for MHD.

Acknowledgements

This work was partially supported by NSF grant DMS-1216972. The work of S.M. and Y.H. was partially supported by an NSERC discovery grant.

Bibliography

[1] R. A. Adams and J. J. Fournier. *Sobolev spaces.* Academic press, 2003.

[2] J. Adler, T. R. Benson, E. Cyr, S. P. MacLachlan, and R. S. Tuminaro. Monolithic multigrid methods for two-dimensional resistive magnetohydrodynamics. *SIAM Journal on Scientific Computing*, 38(1):B1–B24, 2016.

[3] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II – a general purpose object oriented finite element library. *ACM Trans. Math. Softw.*, 33(4):24/1–24/27, 2007.

[4] W. Bangerth, T. Heister, G. Kanschat, et al. `deal.II` *Differential Equations Analysis Library, Technical Reference.* `http://www.dealii.org`.

[5] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, pages 1–137, 2005.

[6] D. Boffi, F. Brezzi, and M. Fortin. *Mixed finite element methods and applications.* Springer Series in Computational Mathematics. Springer, Heidelberg, 2013.

[7] S. Brenner and R. Scott. *The mathematical theory of finite element methods.* Springer Science & Business Media, 2007.

[8] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods.* Springer Science & Business Media, 2012.

[9] P. G. Ciarlet. *The finite element method for elliptic problems.* Society for Industrial and Applied Mathematics, Philadelphia, PA, 2002.

[10] E. C. Cyr, J. N. Shadid, R. S. Tuminaro, R. P. Pawlowski, and L. Chacón. A new approximate block factorization preconditioner for two-dimensional incompressible (reduced) resistive MHD. *SIAM Journal on Scientific Computing*, 35(3):B701–B730, 2013.

[11] T. A. Davis. Algorithm 832: Umfpack v4.3—an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Softw.*, (2):196–199, June.

[12] C. R. Dohrmann and P. B. Bochev. A stabilized finite element method for the Stokes problem based on polynomial pressure projections. *International Journal for Numerical Methods in Fluids*, 46(2):183–201, 2004.

[13] 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.

[14] M. Fortin. Old and new finite elements for incompressible flows. *International Journal for Numerical Methods in Fluids*, 1(4):347–364, 1981.

[15] V. Girault and P.-A. Raviart. *Finite element methods for Navier-Stokes equations: theory and algorithms.* Springer Science & Business Media, 2012.

[16] J. P. Goedbloed and S. Poedts. *Principles of magnetohydrodynamics: with applications to laboratory and astrophysical plasmas.* Cambridge university press, 2004.

[17] M. D. Gunzburger, A. J. Meir, and J. S. Peterson. On the existence, uniqueness, and finite element approximation of solutions of the equations of stationary, incompressible magnetohydrodynamics. *Mathematics of Computation*, 56(194):523–563, 1991.

[18] D. Schötzau. Mixed finite element methods for stationary incompressible magneto–hydrodynamics. *Numerische Mathematik*, 96(4):771–800, 2004.

Chapter 4

Local Fourier analysis of block-structured multigrid relaxation schemes for the Stokes equations

Abstract

[1] Multigrid methods that use block-structured relaxation schemes have been successfully applied to several saddle-point problems, including those that arise from the discretization of the Stokes equations. In this paper, we present a local Fourier analysis of block-structured relaxation schemes for the staggered finite-difference discretization of the Stokes equations to analyze their convergence behavior. Three block-structured relaxation schemes are considered: distributive relaxation, Braess-Sarazin relaxation, and Uzawa relaxation. In each case, we consider variants based on weighted Jacobi relaxation, as is most suitable for parallel implementation on modern architectures. From this analysis, optimal parameters are proposed, and we compare the efficiency of the presented algorithms with these parameters. Finally, some numerical experiments are presented to validate the two-grid and multigrid convergence factors.

[1] Authors: Y. He and S. P. MacLachlan
This work is published as *Local Fourier analysis of block-structured multigrid relaxation schemes for the Stokes equations*, Numerical Linear Algebra with Applications, 25(3):e2147, 2018.

Keywords: Braess-Sarazin relaxation, distributive relaxation, local Fourier analysis, multigrid, staggered finite-difference method (MAC scheme), Stokes equations, Uzawa relaxation

AMS subject classification: 65N22, 65N55

4.1 Introduction

Large linear systems of saddle-point type arise in a wide variety of applications throughout computational science and engineering. Such linear systems represent a significant challenge for computation owing to their indefiniteness and often poor spectral properties. Saddle-point problems are well known and well studied in numerical analysis [5, 6, 15]. Discretization of the Stokes equations naturally leads to saddle-point systems, and solvers for the Stokes equations are a natural first step in developing new algorithms for the Navier-Stokes equations and other saddle-point problems. Two main families of preconditioners are found in the literature for saddle-point systems, such as the Stokes equations. Block preconditioners (cf. [15] and the references therein) are commonly used, because they can easily be constructed from standard multigrid algorithms for scalar elliptic PDEs, such as algebraic multigrid [32]. Monolithic multigrid methods, which are applied directly to the system in coupled form, are potentially more difficult to construct and analyse, because standard pointwise relaxation schemes cannot be applied. Several families of relaxation schemes have, however, been developed for monolithic multigrid methods for the Stokes equations and more complicated saddle-point systems and have been shown to outperform block preconditioners in some cases (see, e.g., [2]). Distributive relaxation [11, 30, 41] was the first to be proposed, using a distributive operator to allow use of pointwise relaxation schemes on transformed variables. A strongly coupled relaxation scheme was introduced by Vanka [37], based on solving a sequence of localized saddle-point problems in a block overlapping Gauss-Seidel (GS) iteration. Two further families are based on using block preconditioning strategies as relaxation schemes, yielding the Braess-Sarazin [8] and Uzawa [25] approaches. Each of these families has been further developed in recent years, including Braess-Sarazin-type relaxation schemes [1, 2, 3, 7, 8], Vanka-type relaxation schemes [1, 2, 3, 24, 26, 31, 33, 37], Uzawa-type relaxation schemes [16, 17, 20, 28], distributive relaxation schemes [4, 38] and other types of methods [12, 35]. The aim of this paper is to analyse block-structured relaxation schemes, including distributive, Braess-Sarazin, and Uzawa relaxation.

Existing analysis of these relaxation schemes leaves several open questions. For finite-element discretizations, variational analysis techniques have been developed for both Braess-Sarazin [44] and Uzawa [20] relaxation. Local Fourier analysis (LFA) has been applied to all of the standard relaxation schemes, including distributive relaxation [27], Vanka relaxation [24, 31], and Braess-Sarazin and Uzawa-type schemes [16, 23]. However, the vast majority of the existing LFA has been for relaxation schemes using (symmetric) GS approaches. Here, in contrast, we focus on schemes that make use of weighted Jacobi relaxation. Considering modern multicore and accelerated parallel architectures, proper understanding of such schemes is critical to achieving excellent parallel and algorithmic scalability.

Supporting numerical results demonstrate some key conclusions of this analysis. First, distributive weighted-Jacobi (DWJ) relaxation retains the well-known advantages of distributive GS (DGS). This fact, coupled with the low cost per iteration and fine-scale parallelism, recommends this relaxation scheme, at least in the context of the finite-difference scheme considered herein. For Braess-Sarazin relaxation, we find that there is no degradation in predicated multigrid performance for the inexact variant of the algorithm introduced in [44] over the exact variant originally proposed in [7, 8]. The same is not true for Uzawa relaxation, where our results show a notable gap between the predicated performance with the exact inversion of the resulting approximate Schur complement and that with only the inexact inversion. Furthermore, we see that the assumptions made in [16] for algebraic analysis of Uzawa-type relaxation are sufficient but not necessary for convergence.

In this paper, we consider these three families of relaxation schemes in terms of the computational work and the optimal smoothing factors obtained. The results show that Braess-Sarazin relaxation provides better smoothing than Uzawa in the case of finite-difference discretization. This is in contrast to results in [20] for finite-element discretizations. The gap between finite-difference discretization and finite-element discretization using Braess-Sarazin relaxation is a question for our future work. However, we also see that distributive weighted Jacobi can match the performance of Braess-Sarazin, as has been seen for GS-based relaxation. Extending this analysis to the finite-element case is also a topic for future research.

The outline of the paper is as follows. In Section 4.2, we introduce the marker and cell (MAC) finite-difference discretization of the Stokes equations in two dimensions and some definitions of LFA. In Section 4.3, we present the DWJ relaxation schemes and the optimal smoothing factor is given by LFA. In Section 4.4, LFA is developed for Braess-Sarazin-type relaxation and optimal parameters are derived. In Section 4.5, we apply LFA to Uzawa-type relaxation to determine the optimal smoothing factor. Furthermore, a comparison of the relaxation schemes is given. Section 4.6 presents some experimentally measured two-grid and multigrid convergence factors to confirm the theoretical results. Conclusions are drawn in Section 4.7.

4.2 Discretization and local Fourier analysis

4.2.1 Staggered finite-difference discretization of the Stokes equations

We consider the Stokes equations,

$$
\begin{aligned}
-\triangle \boldsymbol{\mathcal{U}} + \nabla p &= \boldsymbol{\mathcal{F}}, && (4.1)\\
\nabla \cdot \boldsymbol{\mathcal{U}} &= 0, && (4.2)
\end{aligned}
$$

for velocity vector, $\boldsymbol{\mathcal{U}} = \begin{pmatrix} u \\ v \end{pmatrix}$, and scalar pressure, p, of a viscous fluid. Discretization of (4.1) and (4.2) typically leads to a linear system of the form

$$
Kx = \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\mathcal{U}}_h \\ p_h \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mathcal{F}}_h \\ 0 \end{pmatrix} = b, \tag{4.3}
$$

where A corresponds to the discretized vector Laplacian, B is the negative of the discrete divergence operator, and $\boldsymbol{\mathcal{U}}_h = \begin{pmatrix} u_h \\ v_h \end{pmatrix}$.

Remark 4.2.1. *Here, we consider the vector Laplacian of the velocity in the Stokes equations, as is standard. For more general models, the divergence of the symmetric part of the gradient could be considered, affecting only the symbol of A in what follows.*

In this paper, we consider the standard staggered finite-difference discretization in two-dimensions, known as the MAC scheme (see [19, 36]). The discrete pressure unknowns p_h are defined at cell centres ($\times$-points in Figure 1). The discrete values of u_h and v_h are located at the grid cell faces in the $\circ$- and $\bullet$-points, respectively, see Figure 1.

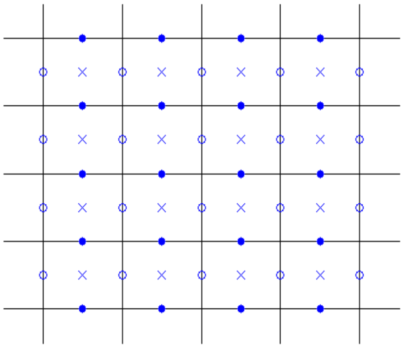

Figure 4.1: The staggered location of unknowns on mesh $\mathbf{G}_h$: $\times - p, \circ - u, \bullet - v$.

The discrete momentum equations read (see [36])

$$-\triangle_h u_h + (\partial_x)_{h/2}\, p_h = F_{1,h}, \qquad -\triangle_h v_h + (\partial_y)_{h/2}\, p_h = F_{2,h},$$

where $\boldsymbol{\mathcal{F}}_h = \begin{pmatrix} F_{1,h} \\ F_{2,h} \end{pmatrix}$. Here, we use the standard five-point discretization for $-\triangle_h$ (for u_h on the $\circ$ grid and for v_h on the $\bullet$ grid) and the approximations

$$\begin{aligned} (\partial_x)_{h/2}\, p_h(x,y) &= \frac{1}{h}\Big(p_h\big(x+h/2,y\big) - p_h\big(x-h/2,y\big)\Big), \\ (\partial_y)_{h/2}\, p_h(x,y) &= \frac{1}{h}\Big(p_h\big(x,y+h/2\big) - p_h\big(x,y-h/2\big)\Big). \end{aligned}$$

The discrete conservation of mass equation is given by

$$(\partial_x)_{h/2}\, u_h(x,y) + (\partial_y)_{h/2}\, v_h(x,y) = 0.$$

We consider uniform meshes with: $h_x = h_y = h$ in this paper.

4.2.2 Definitions and notations

Experience with multigrid methods and multigrid theory shows that the choice of multigrid components may have a strong influence on the efficiency of the resulting algorithm. Some rules are needed to choose the proper multigrid components. In general, the smoothing factor, μ, of LFA gives satisfactorily sharp predictions of actual multigrid convergence (ρ) and guarantees h-independent multigrid convergence [36]. In order to describe LFA for staggered grids, we first introduce some terminology. More details can be found in [36]. We consider two-dimensional infinite uniform grids $\mathbf{G}_h = \mathbf{G}_h^1 \bigcup \mathbf{G}_h^2 \bigcup \mathbf{G}_h^3$ with

$$\mathbf{G}_h^j = \left\{ \boldsymbol{x}_{k_1,k_2}^j := (k_1,k_2)h + \delta^j, (k_1,k_2) \in \mathbb{Z}^2 \right\}, \text{with } \delta^j = \begin{cases} (0, h/2) & \text{if} \quad j=1, \\ (h/2, 0) & \text{if} \quad j=2, \\ (h/2, h/2) & \text{if} \quad j=3, \end{cases}$$

and Fourier functions $\varphi(\boldsymbol{\theta}, \boldsymbol{x}_{k_1,k_2}) \in \operatorname{span}\left\{\varphi_1(\boldsymbol{\theta}, \boldsymbol{x}_{k_1,k_2}), \varphi_2(\boldsymbol{\theta}, \boldsymbol{x}_{k_1,k_2}), \varphi_3(\boldsymbol{\theta}, \boldsymbol{x}_{k_1,k_2})\right\}$ on $\mathbf{G}_h$, in which

$$\begin{aligned} \varphi_1(\boldsymbol{\theta}, \boldsymbol{x}_{k_1,k_2}) &= \begin{pmatrix} e^{i\boldsymbol{\theta}\cdot\boldsymbol{x}_{k_1,k_2}^1/h} & 0 & 0 \end{pmatrix}^T, \ \varphi_2(\boldsymbol{\theta}, \boldsymbol{x}_{k_1,k_2}) = \begin{pmatrix} 0 & e^{i\boldsymbol{\theta}\cdot\boldsymbol{x}_{k_1,k_2}^2/h} & 0 \end{pmatrix}^T, \\ \varphi_3(\boldsymbol{\theta}, \boldsymbol{x}_{k_1,k_2}) &= \begin{pmatrix} 0 & 0 & e^{i\boldsymbol{\theta}\cdot\boldsymbol{x}_{k_1,k_2}^3/h} \end{pmatrix}^T, \ \boldsymbol{\theta} = (\theta_1, \theta_2), \end{aligned}$$

where T denotes the (nonconjugate) transpose of the row vectors. Because $\varphi(\boldsymbol{\theta}, \boldsymbol{x}_{k_1,k_2})$ is periodic in $\boldsymbol{\theta}$ with period 2π, we consider the domain $\boldsymbol{\theta} \in \left[-\frac{\pi}{2}, \frac{3\pi}{2}\right)^2$ (or any interval with length 2π).

Let L_h be a Toeplitz operator acting on one of the components of $\mathbf{G}_h$,

$$\begin{aligned} L_h &\overset{\wedge}{=} [s_{\boldsymbol{\kappa}}]_h \ (\boldsymbol{\kappa} = (\kappa_1, \kappa_2) \in \mathbb{Z}^2); \\ L_h w_h(\boldsymbol{x}^j) &= \sum_{\boldsymbol{\kappa}\in\boldsymbol{V}} s_{\boldsymbol{\kappa}} w_h(\boldsymbol{x}^j + \boldsymbol{\kappa}h), \end{aligned}$$

with constant coefficients $s_{\boldsymbol{\kappa}} \in \mathbb{R}$ (or $\mathbb{C}$), where $w_h(\boldsymbol{x}^j)$ is a function in $l^2(\mathbf{G}_h^j)$. Here, $\boldsymbol{V}$ is a finite index set. Note that because L_h is Toeplitz, it is diagonalized by the Fourier modes $\varphi(\boldsymbol{\theta}, \boldsymbol{x}^j) = e^{i\boldsymbol{\theta}\cdot\boldsymbol{x}^j/\boldsymbol{h}} = e^{i\theta_1 x_1^j/h} e^{i\theta_2 x_2^j/h}$.

Definition 4.2.1. *We call* $\widetilde{L}_h(\boldsymbol{\theta}) = \sum_{\boldsymbol{\kappa}\in\boldsymbol{V}} s_{\boldsymbol{\kappa}} e^{i\boldsymbol{\theta}\boldsymbol{\kappa}}$ *the symbol of* L_h.

Note that for all grid functions $\varphi(\boldsymbol{\theta}, \boldsymbol{x}^j)$,

$$L_h \varphi(\boldsymbol{\theta}, \boldsymbol{x}^j) = \widetilde{L}_h(\boldsymbol{\theta}) \varphi(\boldsymbol{\theta}, \boldsymbol{x}^j).$$

The staggered discretization of the Stokes equations leads to the system

$$\mathcal{L}_h \mathbf{u}_h = \begin{pmatrix} -\triangle_h & 0 & (\partial_x)_{h/2} \\ 0 & -\triangle_h & (\partial_y)_{h/2} \\ -(\partial_x)_{h/2} & -(\partial_y)_{h/2} & 0 \end{pmatrix} \begin{pmatrix} u_h \\ v_h \\ p_h \end{pmatrix},$$

with stencils

$$-\triangle_h = \frac{1}{h^2} \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}, \quad (\partial_x)_h = \frac{1}{h} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}, \quad (\partial_y)_h = \frac{1}{h} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}.$$

The symbol of operator $\mathcal{L}_h$ is given by

$$\widetilde{\mathcal{L}}_h(\theta_1, \theta_2) = \frac{1}{h^2} \begin{pmatrix} 4m(\boldsymbol{\theta}) & 0 & i2h \sin\frac{\theta_1}{2} \\ 0 & 4m(\boldsymbol{\theta}) & i2h \sin\frac{\theta_2}{2} \\ -i2h \sin\frac{\theta_1}{2} & -i2h \sin\frac{\theta_2}{2} & 0 \end{pmatrix},$$

where $m(\boldsymbol{\theta}) = \frac{4 - 2\cos\theta_1 - 2\cos\theta_2}{4} = \sin^2(\frac{\theta_1}{2}) + \sin^2(\frac{\theta_2}{2})$. Each entry in $\widetilde{\mathcal{L}}_h$ is computed as the (scalar) symbol of the corresponding block of $\mathcal{L}_h$, following Definition 4.2.1. Because $\mathcal{L}_h$ is a 3×3 block operator, its symbol is naturally a 3×3 matrix. The error-propagation symbol for a relaxation scheme, represented by matrix M, applied to MAC scheme is

$$\widetilde{\mathcal{S}}_h(\boldsymbol{p}, \omega, \boldsymbol{\theta}) = I - \omega \widetilde{M}^{-1} \widetilde{\mathcal{L}}_h,$$

where $\boldsymbol{p}$ represents parameters within M, the block approximation to $\mathcal{L}_h$, ω is an overall weighting factor, and $\widetilde{M}$ and $\widetilde{\mathcal{L}}_h$ are the symbols for M and $\mathcal{L}_h$, respectively.

In this paper, we consider multigrid methods for staggered discretizations with standard geometric grid coarsening, that is, we construct a sequence of coarse grids by doubling the mesh size in each spatial direction. High and low frequencies for standard coarsening are given by

$$\boldsymbol{\theta} \in T^{\text{low}} = \Big[-\frac{\pi}{2}, \frac{\pi}{2}\Big)^2, \ \boldsymbol{\theta} \in T^{\text{high}} = \Big[-\frac{\pi}{2}, \frac{3\pi}{2}\Big)^2 \Big\backslash \Big[-\frac{\pi}{2}, \frac{\pi}{2}\Big)^2.$$

Definition 4.2.2. *The error-propagation symbol, $\widetilde{\mathcal{S}}_h(\boldsymbol{\theta})$, for a block smoother $\mathcal{S}_h$ on the infinite grid $\mathbf{G}_h$ satisfies*

$$\mathcal{S}_h\varphi(\boldsymbol{\theta}, \boldsymbol{x}_{k_1,k_2}) = \widetilde{\mathcal{S}}_h\varphi(\boldsymbol{\theta}, \boldsymbol{x}_{k_1,k_2}), \ \boldsymbol{\theta} \in \big[-\frac{\pi}{2}, \frac{3\pi}{2}\big)^2,$$

for all $\varphi(\boldsymbol{\theta}, \boldsymbol{x}_{k_1,k_2})$, and the corresponding smoothing factor for $\mathcal{S}_h$ is given by

$$\mu_{\text{loc}} = \mu_{\text{loc}}(\mathcal{S}_h) = \max_{\boldsymbol{\theta} \in T^{\text{high}}} \big\{ \big|\lambda(\widetilde{\mathcal{S}}_h(\boldsymbol{\theta}))\big| \ \big\},$$

where $\lambda\big(\widetilde{\mathcal{S}}_h(\boldsymbol{\theta})\big)$ is an eigenvalue of the 3×3 matrix-valued function $\widetilde{\mathcal{S}}_h(\boldsymbol{\theta})$. Throughout the rest of this paper, the developed theory applies to discrete spaces. Therefore, except when necessary for clarity, we drop the subscript h for simplicity.

Definition 4.2.3. *Because the smoothing factor is a function of some parameters, let $\mathcal{D}$ be a bounded and closed set of allowable parameters, and define the optimal smoothing factor over $\mathcal{D}$ as*

$$\mu_{\text{opt}} = \min_{\mathcal{D}} \mu_{\text{loc}}.$$

Set $\mathcal{D}$ may have many parameters depending on the selection of the relaxation scheme.

Remark 4.2.2. *Because the $\varphi(\boldsymbol{\theta}, \cdot)$ are defined on the infinite grid $\mathbf{G}_h$, the influence of boundaries and of boundary conditions is not taken into account here. The purpose of LFA is to determine the quantitative convergence behavior and efficiency an appropriate multigrid algorithm can attain if a proper boundary treatment is included [10, 34]. Experience with LFA shows that it is often exact for problems with periodic boundary conditions, but degradation in performance may be seen with Dirichlet boundary conditions [27], as will be seen here in the numerical results in Section 4.6.*

4.3 Distributive relaxation

DGS relaxation [11, 30] is well known to be highly efficient for the MAC discretization. The idea of distributive relaxation is as follows. To relax the equation $\mathcal{L}x = b$, we introduce a new variable $\hat{x}$ by $x = \mathcal{P}\hat{x}$ and consider the (transformed) system $\mathcal{L}^*\hat{x} = \mathcal{L}\mathcal{P}\hat{x} = b$. Here, $\mathcal{P}$ is chosen such that the resulting operator $\mathcal{L}\mathcal{P}$ is suitable for decoupled relaxation with a simple, efficient relaxation process, preferably for each of the equations (velocity and pressure) of the transformed system separately. After each sweep of relaxation, the correction $\delta\hat{x}$, is distributed to the original unknowns, $\delta x = \mathcal{P}\delta\hat{x}$. DGS-type relaxation has been widely used [12, 13]. One well-known drawback of DGS is a persistent "gap" between the smoothing factor predicted by LFA and the convergence factors observed in practice for problems with Dirichlet boundary conditions [27, 29, 42, 43]. In [27], it is noted that the LFA predictions are exact for periodic boundary conditions, but extra boundary relaxation is required for Dirichlet boundary conditions (consistent with later analysis of LFA in general in [10, 34]). Another possible solution, proposed in [43] is to replace GS with an incomplete LU factorization in this setting.

Motivated by potential parallelization, we consider DWJ relaxation here, although results in Section 4.6 will show that the above concerns also play a role in this setting. For the Stokes equations, the discretized distribution operator can be represented by the preconditioner

$$\mathcal{P} = \begin{pmatrix} I_h & 0 & (\partial_x)_{h/2} \\ 0 & I_h & (\partial_y)_{h/2} \\ 0 & 0 & \triangle_h \end{pmatrix}.$$

Then, we apply block weighted-Jacobi relaxation to the distributed operator,

$$\mathcal{L}^* = \mathcal{L}\mathcal{P} = \begin{pmatrix} -\triangle_h & 0 & 0 \\ 0 & -\triangle_h & 0 \\ -(\partial_x)_{h/2} & -(\partial_y)_{h/2} & -\triangle_h \end{pmatrix}. \tag{4.4}$$

Remark 4.3.1. *For the staggered MAC discretization, if the original problem has Dirichlet boundary conditions, then the last block operator, $-\triangle_h$, of $\mathcal{L}^*$ is the standard 5-point stencil of the Laplacian operator discretized at cell centers with Neumann boundary conditions [18]. If the original problem has periodic boundary conditions, then last block operator, $-\triangle_h$, should have periodic boundary conditions.*

The discrete matrix form of $\mathcal{P}$ is

$$\mathcal{P} = \begin{pmatrix} I & B^T \\ 0 & -A_p \end{pmatrix},$$

where A_p is the the standard five-point stencil of the Laplacian operator discretized at cell centers (see Remark 4.3.1). For DWJ (with weight α_D) relaxation, we need to solve a system of the form

$$M\delta\hat{x} = \begin{pmatrix} \alpha_D \text{diag}(A) & 0 \\ B & \alpha_D \text{diag}(A_p) \end{pmatrix} \begin{pmatrix} \delta\hat{\mathcal{U}} \\ \delta\hat{p} \end{pmatrix} = \begin{pmatrix} r_{\mathcal{U}} \\ r_p \end{pmatrix}, \tag{4.5}$$

then distribute the updates as $\delta x = \mathcal{P}\delta\hat{x}$. The error propagation operator for the scheme is then $I - \omega_D \mathcal{P} M^{-1} \mathcal{L}$.

4.3.1 DWJ relaxation

The symbol of operator $\mathcal{L}^*$ is given by

$$\widetilde{\mathcal{L}}^*(\theta_1, \theta_2) = \frac{1}{h^2} \begin{pmatrix} 4m(\boldsymbol{\theta}) & 0 & 0 \\ 0 & 4m(\boldsymbol{\theta}) & 0 \\ -i2h\sin\frac{\theta_1}{2} & -i2h\sin\frac{\theta_2}{2} & 4m(\boldsymbol{\theta}) \end{pmatrix},$$

and the symbol of the block weighted-Jacobi operator is

$$\widetilde{M}_D(\theta_1, \theta_2) = \frac{1}{h^2} \begin{pmatrix} 4\alpha_D & 0 & 0 \\ 0 & 4\alpha_D & 0 \\ -i2h\sin\frac{\theta_1}{2} & -i2h\sin\frac{\theta_2}{2} & 4\alpha_D \end{pmatrix}.$$

It is easy to see that all of the eigenvalues of the error-propagation symbol, $\widetilde{\mathcal{S}}_D(\alpha_D, \omega_D, \boldsymbol{\theta}) = I - \omega_D \widetilde{\mathcal{P}} \widetilde{M}_D^{-1} \widetilde{\mathcal{L}}$, are $1 - \omega_D \frac{m(\boldsymbol{\theta})}{\alpha_D}$.

Theorem 4.3.1. *The optimal smoothing factor for DWJ relaxation is*

$$\mu_{\text{opt},D} = \min_{(\alpha_D,\omega_D)} \max_{\boldsymbol{\theta}\in T^{\text{high}}} \left|\lambda(\widetilde{\mathcal{S}}_D(\alpha_D, \omega_D, \boldsymbol{\theta}))\right| = \frac{3}{5},$$

and is achieved if and only if $\alpha_D = \frac{5}{4}\omega_D$.

Proof. When $\boldsymbol{\theta} \in T^{\text{high}}$, $m(\boldsymbol{\theta}) = \sin^2(\frac{\theta_1}{2}) + \sin^2(\frac{\theta_2}{2})$ covers the interval $[\frac{1}{2}, 2]$. Because all of the eigenvalues of $\widetilde{\mathcal{S}}_D(\alpha_D, \omega_D, \boldsymbol{\theta}) = I - \omega_D \widetilde{\mathcal{P}} \widetilde{M}_D^{-1} \widetilde{\mathcal{L}}$ are $1 - \omega_D \frac{m(\boldsymbol{\theta})}{\alpha_D}$, $\max\limits_{\boldsymbol{\theta}\in T^{\text{high}}} \left|\lambda(\widetilde{\mathcal{S}}_D(\alpha_D, \omega_D, \boldsymbol{\theta}))\right| = \max\left\{|1 - \frac{\omega_D}{2\alpha_D}|, |1 - \frac{2\omega_D}{\alpha_D}|\right\}$. In order to minimize this, setting $|1 - \frac{\omega_D}{2\alpha_D}| = |1 - \frac{2\omega_D}{\alpha_D}|$ obtains $\frac{\omega_D}{\alpha_D} = \frac{4}{5}$ and $|1 - \frac{\omega_D}{2\alpha_D}| = \frac{3}{5}$. □

Remark 4.3.2. *The optimal smoothing factor for the ω-(damped) Jacobi relaxation for a five-point finite-difference discretization of the Laplacian is $\frac{3}{5}$ with $\omega = \frac{4}{5}$. Thus, it is not surprising that this serves as an intuitive lower bound on the possible performance of block relaxation schemes that include this as a piece of the overall relaxation.*

4.4 Braess-Sarazin-type relaxation schemes

Although the DWJ-type relaxation is efficient, proper construction of the preconditioner $\mathcal{P}$, is not always possible or straightforward, especially for other types of saddle-point problems. Considering this obstacle, we also analyse other block-structured relaxation schemes. Braess-Sarazin-type algorithms were originally developed as a relaxation scheme for the Stokes equations [8], requiring the solution of a greatly simplified but global saddle-point system. As a relaxation scheme for the system in (4.3), one solves a system of the form

$$Mx = \begin{pmatrix} \alpha C & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \delta\mathcal{U} \\ \delta p \end{pmatrix} = \begin{pmatrix} r_{\mathcal{U}} \\ r_p \end{pmatrix}, \tag{4.6}$$

where C is an approximation of A, the inverse of which is easy to apply, for example I, or $\text{diag}(A)$; $\alpha > 0$ is a chosen relaxation parameter. Solutions of (4.6) are computed in two stages as

$$\begin{aligned} (BC^{-1}B^T)\delta p &= BC^{-1}r_{\mathcal{U}} - \alpha r_p, \\ \delta\mathcal{U} &= \frac{1}{\alpha}C^{-1}(r_{\mathcal{U}} - B^T\delta p). \end{aligned} \tag{4.7}$$

In practice, (6.16) is not solved exactly; an approximate solve is sufficient [44], such as using a simple sweep of a GS or weighted Jacobi iteration. In the following, we consider two ways to solve (6.16): exact and inexact methods.

4.4.1 Exact Braess-Sarazin relaxation

We first take $C = \text{diag}(A)$ and analyze exact Braess-Sarazin relaxation (BSR), that is, solving (6.16) exactly. Denoting the corresponding M as M_E, the symbol of M_E is given by

$$\widetilde{M_E}(\theta_1,\theta_2) = \frac{1}{h^2}\begin{pmatrix} 4\alpha_E & 0 & i2h\sin\frac{\theta_1}{2} \\ 0 & 4\alpha_E & i2h\sin\frac{\theta_2}{2} \\ -i2h\sin\frac{\theta_1}{2} & -i2h\sin\frac{\theta_2}{2} & 0 \end{pmatrix}.$$

The symbol of the error-propagation matrix for weighted exact BSR is $\widetilde{\mathcal{S}}_E(\alpha_E,\omega_E,\boldsymbol{\theta}) = I - \omega_E\widetilde{M_E}^{-1}\widetilde{\mathcal{L}}$. A standard calculation shows that the determinant of $\widetilde{L} - \lambda\widetilde{M_E}$ is

$$\pi_E(\lambda;\alpha_E) \quad = \quad \frac{16m(\boldsymbol{\theta})\alpha_E}{h^4}(\lambda-1)^2(\lambda - \frac{m(\boldsymbol{\theta})}{\alpha_E}),$$

thus, the eigenvalues of $\widetilde{M_E}^{-1}\widetilde{\mathcal{L}}$ are $1, 1, \dfrac{m(\boldsymbol{\theta})}{\alpha_E}$.

Remark 4.4.1. *Note that 1 is an eigenvalue of $\widetilde{M_E}^{-1}\widetilde{\mathcal{L}}$ with multiplicity 2. This result matches with the general results for constraint preconditioners in [21], which considers the distribution of eigenvalues of the left preconditioned linear system, $G^{-1}Hx = G^{-1}b$.*

Theorem 4.4.1. *The optimal smoothing factor for (weighted) exact BSR is*

$$\mu_{\text{opt},E} = \min_{(\alpha_E,\omega_E)}\max_{\boldsymbol{\theta}\in T^{\text{high}}}\left|\lambda(\widetilde{\mathcal{S}}_E(\alpha_E,\omega_E,\boldsymbol{\theta}))\right| = \frac{3}{5},$$

and is achieved if and only if $\alpha_E = \frac{5}{4}\omega_E$, with $\omega_E \in [\frac{2}{5},\frac{8}{5}]$.

Proof. Since the symbol of the error-propagation operator, $\widetilde{\mathcal{S}}_E(\alpha_E,\omega_E,\boldsymbol{\theta}) = I - \omega_E\widetilde{M_E}^{-1}\widetilde{\mathcal{L}}$, has eigenvalues $1-\omega_E, 1-\omega_E, 1-\omega_E\frac{m(\boldsymbol{\theta})}{\alpha_E}$, the smoothing factor is given by $\max_{\boldsymbol{\theta}\in T^{\text{high}}}\left|\lambda(\widetilde{\mathcal{S}}_E(\alpha_E,\omega_E,\boldsymbol{\theta}))\right| = \max\left\{\left|1-\dfrac{\omega_E}{2\alpha_E}\right|, \left|1-\dfrac{2\omega_E}{\alpha_E}\right|, \left|1-\omega_E\right|\right\}$. As in Theorem 4.3.1, we know that $\min_{(\alpha_E,\omega_E)}\max_{\boldsymbol{\theta}\in T^{\text{high}}}\left\{\left|1-\dfrac{\omega_E}{2\alpha_E}\right|, \left|1-\dfrac{2\omega_E}{\alpha_E}\right|\right\} = \dfrac{3}{5}$. Because $|1-\omega_E|$ should be no larger than $\frac{3}{5}$ to achieve the overall bound, we have $\omega_E \in [\frac{2}{5},\frac{8}{5}]$. □

The natural choice is to take $\omega_E = 1$, with $\alpha_E = \frac{5}{4}\omega_E = \frac{5}{4}$. In this setting, the predicted rate of multigrid convergence is very fast, again matching the smoothing performance of weighted Jacobi on the finite-difference Poisson operator. Also note that for the analysis above, we considered $C = \text{diag}(A)$ rather than $C = I$; however, the same conclusion holds for the latter case because $\text{diag}(A) = 4I$ on the infinite grid. Taking $C = I$, we obtain the same smoothing factor $\mu_{\text{opt},E}(\boldsymbol{\theta}) = \frac{3}{5}$ with $\omega_E \in [\frac{2}{5},\frac{8}{5}]$ and $\alpha_E = 5\omega_E$.

Remark 4.4.2. *Choosing C to be (symmetric) GS relaxation ((S)GS) leads to an impractical exact BSR iteration that can, however, be easily analyzed following the above. For the Gauss-Seidel variant (GS-BSR), this leads to an optimal smoothing factor of 0.50 and an LFA-predicted convergence factor of 0.45 with optimal parameters. For the symmetric Gauss-Seidel variant (SGS-BSR), this leads to an optimal smoothing factor of 0.25 and an LFA-predicted convergence factor of 0.20 with optimal parameters. In Remarks 4.4.6 and 4.4.7, we revisit these results in comparison with inexact GS-BSR and SGS-BSR, respectively.*

4.4.2 Inexact Braess-Sarazin relaxation

The (exact) Braess-Sarazin approach was first introduced in [8], where it was shown that a multigrid convergence rate of $O(k^{-1})$ can be achieved, where k denotes the number of smoothing steps on each level. However, there is a significant difficulty in practical use of this method because it requires an exact inversion of the Schur complement, which is very expensive. A broader class of iterative methods for Stokes problem is discussed in [44], which demonstrated that the same $O(k^{-1})$ performance can be achieved as the exact Braess-Sarazin relaxation when the pressure correction equation is not solved exactly. In [44], this inexact BSR (IBSR) is seen to be slightly worse than exact BSR for a finite-element discretization of the Stokes Equations, even with a strong iteration used on the Schur complement system. This motivates us to explore inexact Braess-Sarazin relaxation for the MAC discretization, wondering whether it is possible to achieve the same smoothing factor of $\frac{3}{5}$. This will be answered in the following.

Considering parallel and graphics processing unit (GPU) computation, we focus on using a single sweep of weighted Jacobi iteration (with weight ω_J) to approximate the solution of Equation (6.16). In order to distinguish between the parameters α_E, ω_E used in the exact case, we use α_I, ω_I in the inexact case. Denote the resulting approximation matrix, M, as M_I. Considering the block factorization of M in Equation (4.6), we introduce the modified Schur complement that corresponds to applying only a single weighted Jacobi sweep of relaxation on the true Schur complement, $B(\alpha_I C)^{-1}B^T$, as $-S + B(\alpha_I C)^{-1}B^T$, where $C = \text{diag}(A)$ and $S = \omega_J^{-1}\text{diag}(B(\alpha_I C)^{-1}B^T)$. The stencil of $\alpha_I C$ is

$$\frac{1}{h^2}\begin{bmatrix} 4\alpha_I & 0 \\ 0 & 4\alpha_I \end{bmatrix},$$

and the stencils of $B(\alpha_I C)^{-1}B^T$ and the modified Schur complement for weighted Jacobi iteration are, respectively,

$$\frac{1}{\alpha_I}\begin{bmatrix} & -\frac{1}{4} & \\ -\frac{1}{4} & 1 & -\frac{1}{4} \\ & -\frac{1}{4} & \end{bmatrix}, \quad \frac{1}{\alpha_I}\begin{bmatrix} & -\frac{1}{4} & \\ -\frac{1}{4} & 1-\omega_J^{-1} & -\frac{1}{4} \\ & -\frac{1}{4} & \end{bmatrix}. \tag{4.8}$$

Therefore, according to the symbol Definition 4.2.1, the symbol of the weighted Jacobi iteration is

$$\beta = \frac{2-\cos\theta_1-\cos\theta_2-2\omega_J^{-1}}{2\alpha_I} = \frac{m(\boldsymbol{\theta})-\omega_J^{-1}}{\alpha_I}.$$

The symbol of matrix M_I is given by

$$\widetilde{M_I}(\theta_1,\theta_2) = \frac{1}{h^2}\begin{pmatrix} 4\alpha_I & 0 & i2h\sin\frac{\theta_1}{2} \\ 0 & 4\alpha_I & i2h\sin\frac{\theta_2}{2} \\ -i2h\sin\frac{\theta_1}{2} & -i2h\sin\frac{\theta_2}{2} & h^2\beta \end{pmatrix}.$$

Calculating the determinant of $\widetilde{\mathcal{L}} - \lambda\widetilde{M_I}$, we obtain the characteristic polynomial

$$\pi_I(\lambda;\alpha_I,\omega_J) = \frac{16\alpha_I(m(\boldsymbol{\theta})-\alpha_I\beta)}{h^4}\big(\lambda-\frac{m(\boldsymbol{\theta})}{\alpha_I}\big)\big(\lambda^2+\frac{\beta m(\boldsymbol{\theta})-2m(\boldsymbol{\theta})}{m(\boldsymbol{\theta})-\alpha_I\beta}\lambda+\frac{m(\boldsymbol{\theta})}{m(\boldsymbol{\theta})-\alpha_I\beta}\big) \tag{4.9}$$

Note that setting $\beta = 0$ (which would require $m(\boldsymbol{\theta})\omega_J = 1$) yields $\pi_I(\lambda;\alpha_I,\omega_J) = \pi_E(\lambda;\alpha_I)$, recovering the case of exact Braess-Sarazin. In the general case (when ω_J is a constant factor), we still recognize that $\lambda_* := \frac{m(\boldsymbol{\theta})}{\alpha_I}$ is an eigenvalue for both the exact and inexact Braess-Sarazin relaxation. Therefore, the optimal smoothing factor $\mu_{\text{opt},I}$ for the inexact case cannot be smaller than $\frac{3}{5}$, and will only achieve that value if $\frac{\omega_I}{\alpha_I} = \frac{4}{5}$. Thus, it is reasonable to try $\frac{\omega_I}{\alpha_I} = \frac{4}{5}$ in the analysis of the inexact case.

To analyze the other eigenvalues of the inexact Braess-Sarazin relaxation, substituting $\beta = \frac{m(\boldsymbol{\theta})-\omega_J^{-1}}{\alpha_I}$ into (4.9), these two eigenvalues, λ_1, λ_2, are the roots of

$$g_I(\lambda; \alpha_I, \omega_J) = \lambda^2 + \left(\frac{m(\boldsymbol{\theta})}{\alpha_I}\big(m(\boldsymbol{\theta})\omega_J - 1\big) - 2m(\boldsymbol{\theta})\omega_J\right)\lambda + m(\boldsymbol{\theta})\omega_J. \tag{4.10}$$

Consequently, we have

$$\begin{aligned} \lambda_1 + \lambda_2 &= \frac{m(\boldsymbol{\theta})}{\alpha_I}\big(1 - m(\boldsymbol{\theta})\omega_J\big) + 2m(\boldsymbol{\theta})\omega_J, & (4.11)\\ \lambda_1\lambda_2 &= m(\boldsymbol{\theta})\omega_J > 0. & (4.12) \end{aligned}$$

Denote the discriminant of the quadratic function g_I as

$$\Delta_I(\alpha_I, \omega_J) = \frac{\omega_J^2}{\alpha_I^2} m(\boldsymbol{\theta})\big(m(\boldsymbol{\theta}) - m_*\big)\big(m(\boldsymbol{\theta}) - m_+\big)\big(m(\boldsymbol{\theta}) - m_-\big), \tag{4.13}$$

where

$$m_* = \omega_J^{-1}, \quad m_\pm = \frac{4\alpha_I + \omega_J^{-1} \pm \sqrt{(4\alpha_I + \omega_J^{-1})^2 - (4\alpha_I)^2}}{2}.$$

For $m(\boldsymbol{\theta}) \in [0, 2]$, the sign of $\Delta_I(\alpha_I, \omega_J)$ is determined by the choices of α_I, ω_J. Hence, it is important to determine the relationship of m_*, m_+, m_-, for certain choices of α_I, ω_J. The next Lemma gives a useful characterization.

Lemma 4.4.1. *If $\alpha_I = \omega_J^{-1}$, then $m_- = m_*$. If, furthermore, $\frac{1}{2} \leq \alpha_I \leq 2$, then*

$$\Delta_I(\alpha_I, \omega_J) \leq 0, \quad \forall m(\boldsymbol{\theta}) \in [0, 2].$$

Proof. Since $\alpha_I = \omega_J^{-1}$, we have

$$m_- = \frac{4\alpha_I + \omega_J^{-1} - \sqrt{(4\alpha_I + \omega_J^{-1})^2 - (4\alpha_I)^2}}{2} = \alpha_I = m_*,$$

which is the first result.

If $\frac{1}{2} \le \alpha_I \le 2$, we have

$$\begin{aligned} m_+ &= \frac{4\alpha_I + \omega_J^{-1} + \sqrt{(4\alpha_I + \omega_J^{-1})^2 - (4\alpha_I)^2}}{2} = 4\alpha_I \ge 2, \\ m_- &= \alpha_I \le 2. \end{aligned}$$

According to the discriminant in (4.13) and the relationship that $\alpha_I = \omega_J^{-1}$, it follows that

$$\Delta_I(\alpha_I, \omega) = \frac{m(\boldsymbol{\theta})(m(\boldsymbol{\theta}) - 4\alpha_I)(m(\boldsymbol{\theta}) - \alpha_I)^2}{\alpha_I^4} \le 0,$$

for all $m(\boldsymbol{\theta}) \in [0, 2]$. □

Theorem 4.4.2. *If $\Delta_I(\alpha_I, \omega_J) \le 0$, then necessary and sufficient conditions for the convergence of inexact Braess-Sarazin iteration, $\widetilde{\mathcal{S}}_I(\boldsymbol{\theta}) = I - \omega_I(\widetilde{M}_I)^{-1}\widetilde{\mathcal{L}}$, for all frequencies $\boldsymbol{\theta} \neq 0$ are*

$$|1 - \omega_I\lambda_*| < 1, \tag{4.14}$$

$$(1 - \omega_I\lambda_1)(1 - \omega_I\lambda_2) < 1. \tag{4.15}$$

Proof. If $\Delta_I(\alpha_I, \omega_J) \le 0$, then $\lambda_1 = \overline{\lambda_2}$ and $|1 - \omega_I\lambda_1|^2 = |1 - \omega_I\lambda_2|^2 = (1 - \omega_I\lambda_1)(1 - \omega_I\lambda_2)$. Thus, the necessary and sufficient condition for convergence is $(1 - \omega_I\lambda_1)(1 - \omega_I\lambda_2) < 1$, along with $|1 - \omega_I\lambda_*| < 1$. □

Next, under the condition $\alpha_I = \omega_J^{-1}$, we optimize the smoothing factor $\mu_{\text{loc},I}(\boldsymbol{\theta})$. Considering the convergence conditions, using (4.11) and (4.12), (4.15) can be simplified as

$$m(\boldsymbol{\theta}) < \omega_J^{-1} + \alpha_I(2 - \omega_I),$$

which should hold for all $m(\boldsymbol{\theta}) \in [0, 2]$. This is clearly satisfied for all $m(\boldsymbol{\theta})$ if it is true for $m(\boldsymbol{\theta}) = 2$. From (4.14), becuase $\lambda_* = \frac{m(\boldsymbol{\theta})}{\alpha_I}$, we obtain $\omega_I < \alpha_I$. We thus define a set $\mathcal{D}^*$, of parameters that satisfy Theorem 4.4.2 (allowing for nonconvergence when $\boldsymbol{\theta} = 0$), as well as the assumption that $\alpha_I = \frac{5}{4}\omega_I$ needed to achieve the smoothing factor of $\frac{3}{5}$, as

$$\mathcal{D}^* = \left\{ \big(\alpha_I, \omega_J, \omega_I\big) : \frac{1}{2} \le \alpha_I = \omega_J^{-1} \le 2, 2 < \alpha_I(3 - \omega_I), \alpha_I = \frac{5}{4}\omega_I \right\}.$$

The next theorem demonstrates that IBSR can achieve the optimal smoothing factor of $\frac{3}{5}$.

Theorem 4.4.3. *For $(\alpha_I, \omega_J, \omega_I) \in \mathcal{D}^*$, the optimal smoothing factor for the IBSR is*

$$\mu_{\text{opt},I} = \min_{(\alpha_I,\omega_J,\omega_I)\in\mathcal{D}^*} \max_{\boldsymbol{\theta}\in T^{\text{high}}} \left\{ |1-\omega_I\lambda_*|,\, |1-\omega_I\lambda_1|,\, |1-\omega_I\lambda_2| \right\} = \frac{3}{5},$$

and is achieved if and only if $\alpha_I = \frac{5}{4}, \omega_I = 1$, and $\omega_J = \frac{4}{5}$.

Proof. Because $(\alpha_I, \omega_J, \omega_I) \in \mathcal{D}^*$, the convergence conditions are satisfied. For the high frequencies, the eigenvalues are either complex numbers or two equal real numbers, so we consider μ^2_{opt} in place of μ_{opt}. Let us set

$$\eta^2(m(\boldsymbol{\theta})) := (1-\omega_I\lambda_1)(1-\omega_I\lambda_2).$$

Following (4.11) and (4.12), and substituting $\omega_J^{-1} = \alpha_I, \omega_I = \frac{4}{5}\alpha_I$ into $\eta^2(m(\boldsymbol{\theta}))$, we have

$$\eta^2(m(\boldsymbol{\theta})) = \frac{4}{5\alpha_I} m(\boldsymbol{\theta})^2 + (\frac{16\alpha_I}{25} - \frac{12}{5}) m(\boldsymbol{\theta}) + 1.$$

Treating η^2 as a quadratic function of m, the symmetry axis is $m_0 = \dfrac{15\alpha_I - 4\alpha_I^2}{10}$. For $\alpha_I \in [\frac{1}{2}, 2], m_0 \in \left[\frac{13}{20}, \frac{45}{32}\right] \subseteq \left[\frac{1}{2}, 2\right]$, achieving its maximum value at $\alpha_I = \frac{15}{8}$. This tells us that $\eta^2(m(\boldsymbol{\theta}))$ obtains its maximum at either $m(\boldsymbol{\theta}) = \frac{1}{2}$ or $m(\boldsymbol{\theta}) = 2$, so our discussion is divided into two cases. Note also that $m_0 = \frac{5}{4}$, when $\alpha_I = \frac{5}{4}$.

Case 1: If $m_0 \geq \frac{5}{4}$, then

$$\max_{\boldsymbol{\theta}\in T^{\text{high}}} \eta^2(m(\boldsymbol{\theta})) = \eta^2(m(\boldsymbol{\theta}) = \frac{1}{2}) = \frac{1}{5\alpha_I} + \frac{8\alpha_I}{25} - \frac{1}{5}.$$

From $m_0 \geq \frac{5}{4}$ and $\alpha_I \in [\frac{1}{2}, 2]$, we have $\alpha_I \in \left[\frac{5}{4}, 2\right]$. The optimal smoothing factor is, then,

$$\min_{(\alpha_I,\omega_J,\omega_I)\in\mathcal{D}^*} \max_{\boldsymbol{\theta}\in T^{\text{high}}} \eta^2(m(\boldsymbol{\theta})) = \min_{\alpha_I\in[\frac{5}{4},2]} \left\{ \frac{1}{5\alpha_I} + \frac{8\alpha_I}{25} - \frac{1}{5} \right\} = \frac{9}{25}, \tag{4.16}$$

where $\alpha_I = \frac{5}{4}$ obtains the minimum.

Case 2: If $m_0 \leq \frac{5}{4}$, then

$$\max_{\boldsymbol{\theta}\in T^{\text{high}}} \eta^2(m(\boldsymbol{\theta})) = \eta^2(m(\boldsymbol{\theta}) = 2) = \frac{16}{5\alpha_I} + \frac{32\alpha_I}{25} - \frac{19}{5}.$$

From $m_0 \leq \frac{5}{4}$ and $\alpha_I \in \left[\frac{1}{2}, 2\right]$, we have $\alpha_I \in \left[\frac{1}{2}, \frac{5}{4}\right]$. The optimal smoothing factor is

$$\min_{(\alpha_I,\omega_J,\omega_I)\in\mathcal{D}^*} \max_{\boldsymbol{\theta}\in T^{\text{high}}} \eta^2(m(\boldsymbol{\theta})) = \min_{\alpha_I\in[\frac{1}{2},\frac{5}{4}]} \left\{ \frac{16}{5\alpha_I} + \frac{32\alpha_I}{25} - \frac{19}{5} \right\} = \frac{9}{25}, \tag{4.17}$$

where $\alpha_I = \frac{5}{4}$ obtains the minimum.

For both situations, $\omega_I = \frac{4}{5}\alpha_I = 1, \omega_J = \alpha_I^{-1} = \frac{4}{5}$ satisfy the condition $2 < \alpha_I(3-\omega_I)$ in $\mathcal{D}^*$. Combining (4.16) and (4.17), we see that the optimal smoothing factor over $\mathcal{D}^*$ for λ_1, λ_2 is $\frac{3}{5}$. For the third eigenvalue, λ_*, because $\alpha_I = \frac{5}{4}\omega_I$ is a condition on $\mathcal{D}^*$, we always have $\max_{\boldsymbol{\theta}\in T^{\text{high}}} \left|1 - \omega_I \frac{m(\boldsymbol{\theta})}{\alpha_I}\right| = \frac{3}{5}$ as in BSR. Thus, we can draw the conclusion that the optimal smoothing factor for IBSR is

$$\min_{(\alpha_I,\omega_J,\omega_I)\in\mathcal{D}^*} \max_{\boldsymbol{\theta}\in T^{\text{high}}} \left\{ |1-\omega_I\lambda_*|,\, |1-\omega_I\lambda_1|,\, |1-\omega_I\lambda_2| \right\} = \frac{3}{5},$$

with $\alpha_I = \frac{5}{4}, \omega_I = 1$, and $\omega_J = \frac{4}{5}$. □

Remark 4.4.3. *For the optimal values $\alpha_I = \omega_J^{-1} = \frac{5}{4}$, and $\omega_I = 1$, (4.10) has real roots only for $m(\boldsymbol{\theta}) = 0, \frac{5}{4}$. For other $m(\boldsymbol{\theta}) \in [0, 2]$, the roots are complex.*

Remark 4.4.4. *It is interesting that the optimal parameter of $\alpha_I = \frac{5}{4}$ matches that found experimentally in [22] for solving the discretized Stokes problem using Taylor-Hood elements with Braess-Sarazin relaxation.*

Remark 4.4.5. *The definition of $\mathcal{D}^*$ makes use of the assumption that $\alpha_I = \omega_J^{-1}$, which is not strictly necessary, Thus, while the choice of parameters is unique over $\mathcal{D}^*$, it may not be globally unique. However, because our interest is whether IBSR can reach the same optimal smoothing factor as BSR, we do not consider this question further.*

Remark 4.4.6. *While exact GS-BSR is impractical, a reasonable inexact variant uses GS relaxation for the velocity equations and retains weighted Jacobi relaxation for the pressure correction, based on the same approximate Schur complement given in (4.8). Following similar reasoning as above, we can conclude the inexact variant cannot achieve a better smoothing factor than GS does for the velocity block, which is 0.5. While we have not analytically optimized the inexact GS-BSR parameters, numerical optimization shows that a smoothing factor of 0.5 can be achieved, and yields an LFA-predicted convergence factor of 0.48 with linear interpolation and 6-point restriction, and 0.41 with linear interpolation and 12-point restriction, for the optimal parameters found.*

Remark 4.4.7. *Similarly, a reasonable inexact SGS-BSR algorithm again uses weighted Jacobi relaxation for the pressure correction, based on the approximate Schur complement given in (4.8). Numerical optimization of the inexact SGS-BSR parameters yields a smoothing factor of 0.25, matching the lower bound given from the exact SGS-BSR case, and an LFA-predicted convergence factor of 0.20 with linear interpolation and 6-point restriction.*

Comparing Theorem 4.4.3 with Theorem 4.4.1, we note that IBSR and BSR obtain the same optimal smoothing factor, $\frac{3}{5}$, with the same choices $\alpha_I = \frac{5}{4}, \omega_I = 1$. The IBSR is simple to implement, avoiding the necessity of computing the exact inversion of the Schur complement. These properties make IBSR attractive as a smoother for general saddle-point problems.

4.5 Uzawa-type relaxation

Multigrid methods with Uzawa-type relaxation are a popular family of algorithms for solving saddle-point systems [14, 25]. Each step of the exact Uzawa algorithm requires the solution of a linear system with coefficient matrix A, as well as one with an approximation of the Schur complement, $-BA^{-1}B^T$. However, if this computation is replaced by approximate solutions produced by iterative methods then, with relatively modest requirements on the accuracy of the approximate solution, the resulting inexact Uzawa algorithm is convergent, with a convergence rate close to that of the exact algorithm [9, 14]. In order to distinguish the parameters from those used in Braess-Sarazin relaxation, we add the subscript U in the following. The Uzawa-type relaxation that we consider can be written as a simpler block solve than that used in BSR,

$$M_U \delta x = \begin{pmatrix} \alpha C & 0 \\ B & -S \end{pmatrix} \begin{pmatrix} \delta \mathcal{U} \\ \delta p \end{pmatrix} = \begin{pmatrix} r_{\mathcal{U}} \\ r_p \end{pmatrix}, \tag{4.18}$$

where αC is an approximation of A, and $-S$ is an approximation of the Schur complement, $-BA^{-1}B^T$.

Here, we discuss two cases. First, we consider an analogue to exact Braess-Sarazin with $C = \text{diag}(A), S = B(\alpha C)^{-1}B^T$. Then, we consider an algorithm with manageable cost, with $C = \text{diag}(A), S = \sigma^{-1}I$.

4.5.1 Schur-Uzawa relaxation

Here, we consider $C = \operatorname{diag}(A)$, $S = B(\alpha_{SU}C)^{-1}B^T$, giving the so-called Schur-Uzawa method. The amplification factor for this method is $\widetilde{\mathcal{S}}_{SU}(\alpha_{SU}, \omega_{SU}, \boldsymbol{\theta}) = I - \omega_{SU}\widetilde{M}_{SU}^{-1}\widetilde{\mathcal{L}}$ and the symbol of M_{SU} is given by

$$\widetilde{M}_{SU}(\theta_1, \theta_2) = \frac{1}{h^2}\begin{pmatrix} 4\alpha_{SU} & 0 & 0 \\ 0 & 4\alpha_{SU} & 0 \\ -i2h\sin\frac{\theta_1}{2} & -i2h\sin\frac{\theta_2}{2} & -\frac{m(\boldsymbol{\theta})}{\alpha_{SU}}h^2 \end{pmatrix}.$$

The determinant of $\widetilde{L} - \lambda\widetilde{M}_{SU}$ is then

$$\pi_{SU}(\lambda; \alpha_{SU}) = \frac{16\alpha_{SU}m(\boldsymbol{\theta})}{h^4}\big(\lambda - \frac{m(\boldsymbol{\theta})}{\alpha_{SU}}\big)\Big(\lambda^2 - \big(1 + \frac{m(\boldsymbol{\theta})}{\alpha_{SU}}\big)\lambda + 1\Big).$$

As discussed in Braess-Sarazin relaxation, the optimal smoothing factor for the modes $\lambda_{*_U} := \frac{m(\boldsymbol{\theta})}{\alpha_{SU}}$ is known to be

$$\big|1 - \frac{2\omega_{SU}}{\alpha_{SU}}\big| = \big|1 - \frac{\omega_{SU}}{2\alpha_{SU}}\big| = \frac{3}{5},$$

provided that $\frac{\omega_{SU}}{\alpha_{SU}} = \frac{4}{5}$.

To analyze the other eigenvalues of Schur-Uzawa relaxation, we denote λ_1, λ_2 as the roots of

$$g_{SU}(\lambda; \alpha_{SU}) = \lambda^2 - \big(1 + \frac{m(\boldsymbol{\theta})}{\alpha_{SU}}\big)\lambda + 1, \tag{4.19}$$

taking the discriminant of the quadratic function g_{SU} as

$$\Delta_{SU}(m(\boldsymbol{\theta}); \alpha_{SU}) = \big(1 + \frac{m(\boldsymbol{\theta})}{\alpha_{SU}}\big)^2 - 4.$$

Because the sign of the discriminant is undetermined and depends on the value of $m(\boldsymbol{\theta})$, we must consider three cases for the distribution of the eigenvalues. First, that all of the eigenvalues are real numbers. Second, that all of the eigenvalues are complex numbers. Finally, that some are real and some are complex. The main idea behind optimizing the smoothing factor is, simply, to optimize for each of the three cases respectively, then select the best one.

Theorem 4.5.1. *The optimal smoothing factor for Schur-Uzawa relaxation is*

$$\begin{aligned}\mu_{\text{opt},SU} &= \min_{(\alpha_{SU},\omega_{SU})}\max_{\boldsymbol{\theta}\in T^{\text{high}}}\left\{\left|\lambda(\widetilde{\mathcal{S}}_{SU}(\alpha_{SU},\omega_{SU},\boldsymbol{\theta}))\right|\right\}\\ &= \sqrt{\frac{33-3\sqrt{73}}{41-3\sqrt{73}}}\approx 0.6924,\end{aligned}$$

and is achieved if and only if

$$\alpha_{SU}=\frac{4}{\sqrt{73}-5},\ \omega_{SU}=\frac{4}{\sqrt{73}-3}.$$

Proof. Case 1: If $\Delta_{SU}(m(\boldsymbol{\theta});\alpha_{SU})\leq 0$ for all $m(\boldsymbol{\theta})$, then we must have $\alpha_{SU}\geq m(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$, so $\alpha_{SU}\geq 2$. In this case, we have two complex roots for all $m(\boldsymbol{\theta})$, whose magnitude, $\tau_{SU}(m(\boldsymbol{\theta}))$, is given by

$$\begin{aligned}\tau^2_{SU}(m(\boldsymbol{\theta})) &:= (1-\omega_{SU}\lambda_1)(1-\omega_{SU}\lambda_2),\\ &= 1-(\lambda_1+\lambda_2)\omega_{SU}+\lambda_1\lambda_2\omega^2_{SU},\\ &= 1-\omega_{SU}(1+\frac{m(\boldsymbol{\theta})}{\alpha_{SU}})+\omega^2_{SU}.\end{aligned}$$

The smoothing factor over these roots is given by

$$\begin{aligned}\mu_C(\alpha_{SU},\omega_{SU})^2: &= \max_{m(\boldsymbol{\theta})\in[\frac{1}{2},2]}\tau^2_{SU}(m(\boldsymbol{\theta}))=\tau^2_{SU}(\frac{1}{2})\\ &= \left(\omega_{SU}-(\frac{1}{2}+\frac{1}{4\alpha_{SU}})\right)^2+1-(\frac{1}{2}+\frac{1}{4\alpha_{SU}})^2.\end{aligned}\tag{4.20}$$

In order to minimize $\mu_C(\alpha_{SU},\omega_{SU})$, ω_{SU} must be equal to $\omega^*_{SU}=\frac{1}{2}+\frac{1}{4\alpha_{SU}}$. Because $\alpha_{SU}\geq 2$,

$$\min_{(\alpha_{SU}\geq 2,\omega_{SU})}\mu_C=\sqrt{1-(\frac{1}{2}+\frac{1}{4\times 2})^2}=\sqrt{\frac{39}{64}}\approx 0.7806,$$

provided that $\alpha_{SU}=2,\omega_{SU}=\frac{1}{2}+\frac{1}{4\alpha_{SU}}=\frac{5}{8}$.

Because there is another eigenvalue, $\frac{m(\boldsymbol{\theta})}{\alpha_{SU}}$, the optimal smoothing factor when $\Delta_{SU}(m(\boldsymbol{\theta});\alpha_{SU})\leq 0$ for all $\boldsymbol{\theta}$ is at least $\sqrt{\frac{39}{64}}$.

Case 2: If $\Delta_{SU}(m(\boldsymbol{\theta});\alpha_{SU})\geq 0$ for all $m(\boldsymbol{\theta})$, then we have $\alpha_{SU}\leq m(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$, so $\alpha_{SU} \leq \frac{1}{2}$. Denote the two eigenvalues of (4.19) as $\lambda_+(m(\boldsymbol{\theta})) > \lambda_-(m(\boldsymbol{\theta}))$. It is easy to check that λ_+ is an increasing function of $m(\boldsymbol{\theta})$, while λ_- is a decreasing function of $m(\boldsymbol{\theta})$. Set

$$\mu_R(\alpha_{SU}, \omega_{IU}) := \max_{m(\boldsymbol{\theta})\in[\frac{1}{2},2]} \{|1-\omega_{SU}\lambda|\} = \max\big\{|1-\omega_{SU}\lambda_+(2)|, |1-\omega_{SU}\lambda_-(2)|\big\}. \tag{4.21}$$

We know that to minimize this maximum, we need

$$\omega_{SU} = \frac{2}{\lambda_+(2)+\lambda_-(2)} = \frac{2}{\frac{2}{\alpha_{SU}}+1}, \tag{4.22}$$

and take $\omega_{SU}^{**} = \frac{2}{\frac{2}{\alpha_{SU}}+1}$. The smoothing factor for these modes is then given by

$$\begin{aligned}
\min \mu_R(\alpha_{SU}, \omega_{SU}) &= \min_{\alpha_{SU}\leq\frac{1}{2}} \left\{ \frac{\lambda_+(2)-\lambda_-(2)}{\lambda_+(2)+\lambda_-(2)} \right\} \\
&= \min_{\alpha_{SU}\leq\frac{1}{2}} \left\{ \sqrt{1-\frac{4\lambda_+(2)\lambda_-(2)}{\big(\lambda_+(2)+\lambda_-(2)\big)^2}} \right\} \\
&= \min_{\alpha_{SU}\leq\frac{1}{2}} \big\{ \sqrt{1-\omega_{SU}^2} \big\} \qquad (4.23) \\
&= \sqrt{\frac{21}{25}} \approx 0.9615,
\end{aligned}$$

because $\lambda_+(2)\lambda_-(2) = 1$ with the minimum achieved when $\alpha_{SU} = \frac{1}{2}$.

Because there is another eigenvalue, $\frac{m(\boldsymbol{\theta})}{\alpha_{SU}}$, the optimal smoothing factor when $\Delta_{SU}(m(\boldsymbol{\theta}); \alpha_{SU}) \geq 0$ for all $\boldsymbol{\theta}$ is at least $\sqrt{\frac{21}{25}}$.

Case 3: $\alpha_{SU} \in (\frac{1}{2}, 2)$. When $m(\boldsymbol{\theta}) \in (\frac{1}{2}, \alpha_{SU}]$, $\Delta_{SU}(m(\boldsymbol{\theta}); \alpha_{SU}) \leq 0$. From (4.20), we know that $\mu_C(\alpha_{SU}, \omega_{SU})$ is an increasing function of α_{SU}. When $m(\boldsymbol{\theta}) \in [\alpha_{SU}, 2)$, $\Delta_{SU}(m(\boldsymbol{\theta}); \alpha_{SU}) \geq 0$. From (4.22) and (4.23), we know that $\mu_R(\alpha_{SU}, \omega_{SU})$ is a decreasing function of α_{SU}. Set

$$\mu_{SU} = \min_{(\alpha_{SU},\omega_{SU})} \max\big\{ \max_{\alpha_{SU}\leq\boldsymbol{\theta}<2} \mu_R(\alpha_{SU}, \omega_{SU}), \max_{\frac{1}{2}\leq\boldsymbol{\theta}\leq\alpha_{SU}} \mu_C(\alpha_{SU}, \omega_{SU})\big\}.$$

In order to achieve the minimum, we must have $\mu_R(\alpha_{SU}, \omega_{SU}) = \mu_C(\alpha_{SU}, \omega_{SU})$ and

$\omega_{SU}^{*} = \omega_{SU}^{**}$. This gives $\alpha_{SU} = \frac{4}{\sqrt{73}-5}$, $\omega_{SU} = \frac{4}{\sqrt{73}-3}$, and

$$\mu_{SU} = \sqrt{1-\omega_{SU}^2} = \sqrt{\frac{33-3\sqrt{73}}{41-3\sqrt{73}}} \approx 0.6924.$$

Recall the third eigenvalue $\frac{m(\boldsymbol{\theta})}{\alpha_{SU}}$. Since $\alpha_{SU} = \frac{4}{\sqrt{73}-5}$ and $\omega_{SU} = \frac{4}{\sqrt{73}-3}$, we have

$$\max_{m(\boldsymbol{\theta})\in[\frac{1}{2},2]} \left\{ \left|1-\omega_{SU}\frac{m(\boldsymbol{\theta})}{\alpha_{SU}}\right| \right\} = \frac{70+2\sqrt{73}}{128} \approx 0.6804 < 0.6924.$$

From the three cases discussed above, we can clearly conclude that when $\alpha_{SU} = \frac{4}{\sqrt{73}-5}$ and $\omega_{SU} = \frac{4}{\sqrt{73}-3}$, we obtain the optimal smoothing factor $\mu_{SU} = \sqrt{\frac{33-3\sqrt{73}}{41-3\sqrt{73}}} \approx 0.6924$. □

We note that the convergence factor predicated for Schur-Uzawa is somewhat worse than for exact Braess-Sarazin. As we will see in the next section, further degradation occurs when we consider the more practical algorithm, σ-Uzawa.

4.5.2 σ-Uzawa relaxation

In Braess-Sarazin relaxation, we prefer to solve Schur complement system $(BC^{-1}B^T)\delta p = BC^{-1}r_{\mathcal{U}} - \alpha r_p$ by an inexact iteration such as weighted Jacobi for the pressure update. This idea can be adopted to the Schur-Uzawa relaxation, replacing the exact solution of $B(\alpha_{SU}C)^{-1}B^T\delta p = B\delta\mathcal{U} - r_p$ by the simple calculation of $\sigma^{-1}\delta p = B\delta\mathcal{U} - r_p$, which can be viewed as a weighted Jacobi iteration applied with the Schur-Uzawa solve, because the symbol of $\text{diag}(B(\alpha_{SU}C)^{-1}B^T)$ is α_{SU}^{-1}. Following the usual notation, we call the resulting parameter σ and the algorithm as σ-Uzawa relaxation. The symbol of the resulting approximation of $\mathcal{L}$, M_U, is given by

$$\widetilde{M}_U(\theta_1,\theta_2) = \frac{1}{h^2}\begin{pmatrix} 4\alpha_U & 0 & 0 \\ 0 & 4\alpha_U & 0 \\ -i2h\sin\frac{\theta_1}{2} & -i2h\sin\frac{\theta_2}{2} & -\sigma^{-1}h^2 \end{pmatrix}.$$

The determinant of $\widetilde{L}-\lambda\widetilde{M_U}$ is then

$$\pi_U(\lambda;\alpha_U,\sigma)=\frac{16\alpha_U^2}{\sigma h^4}\big(\lambda-\frac{m(\boldsymbol{\theta})}{\alpha_U}\big)\bigg(\lambda^2-\frac{1+\sigma}{\alpha_U}m(\boldsymbol{\theta})\lambda+\frac{m(\boldsymbol{\theta})\sigma}{\alpha_U}\bigg).$$

Because $\lambda_{*_U}:=\frac{m(\boldsymbol{\theta})}{\alpha_U}$ and $m(\boldsymbol{\theta})\in[\frac{1}{2},2]$ for high frequencies, the optimal smoothing factor for these modes is known to be

$$\big|1-\frac{2\omega_U}{\alpha_U}\big|=\big|1-\frac{\omega_U}{2\alpha_U}\big|=\frac{3}{5},$$

provided that $\frac{\omega_U}{\alpha_U}=\frac{4}{5}$.

To analyze the other eigenvalues of σ-Uzawa relaxation, we denote by λ_1,λ_2 the roots of

$$g_U(\lambda;\alpha_U,\sigma)=\lambda^2-\frac{(1+\sigma)m(\boldsymbol{\theta})}{\alpha_U}\lambda+\frac{m(\boldsymbol{\theta})\sigma}{\alpha_U}, \tag{4.24}$$

taking the discriminant of the quadratic function g_U as

$$\Delta_U(\alpha_U,\sigma)=\frac{m(\boldsymbol{\theta})(1+\sigma)^2}{\alpha_U^2}\bigg(m(\boldsymbol{\theta})-\frac{4\alpha_U\sigma}{(1+\sigma)^2}\bigg),$$

and take

$$m_1=0,\quad m_2=\frac{4\alpha_U\sigma}{(1+\sigma)^2}.$$

From (4.24), we have

$$\lambda_1+\lambda_2 = \frac{m(\boldsymbol{\theta})(1+\sigma)}{\alpha_U}>0, \tag{4.25}$$

$$\lambda_1\lambda_2 = \frac{m(\boldsymbol{\theta})\sigma}{\alpha_U}>0, \tag{4.26}$$

$$\lambda_{1,2} = \frac{(1+\sigma)m(\boldsymbol{\theta})}{2\alpha_U}\left(1\pm\sqrt{1-\frac{m_2}{m(\boldsymbol{\theta})}}\right). \tag{4.27}$$

The sign of $\Delta_U(\alpha_U,\sigma)$ (and, consequently, the value of m_2) plays an important role in the analysis of the smoothing factor. As before, we explore the optimal smoothing factor for three cases: only real eigenvalues, only complex eigenvalues, and when $\frac{1}{2}<m_2<2$, giving both real and complex eigenvalues. We first explore the case where only complex eigenvalues occur.

In order to discuss the complex eigenvalues, we take $\tau(m(\boldsymbol{\theta}))$ to be the magnitude of the two eigenvalues at frequency $\boldsymbol{\theta}$, giving

$$\begin{aligned}\tau^2(m(\boldsymbol{\theta})) &= (1-\omega_U\lambda_1)(1-\omega_U\lambda_2),\\ &= 1-(\lambda_1+\lambda_2)\omega_U+\lambda_1\lambda_2\omega_U^2,\\ &= 1+\frac{\omega_U}{\alpha_U}(\omega_U\sigma-\sigma-1)m(\boldsymbol{\theta}).\end{aligned}$$

For simplicity of discussion of the smoothing factor for complex eigenvalues, we give a general result that can be applied in the third case, when $\frac{1}{2} < m_2 < 2$.

Lemma 4.5.1. *Assume that $m_2 \geq \frac{1}{2}$ and let $\gamma = \min\{m_2, 2\}$. For $m(\boldsymbol{\theta}) \in [\frac{1}{2}, \gamma]$, eigenvalues λ_1 and λ_2 are complex conjugates and the smoothing factor for these modes over this range of $\boldsymbol{\theta}$ is*

$$SF_C = \max_{m(\boldsymbol{\theta})\in[\frac{1}{2},\gamma]} \tau(m(\boldsymbol{\theta})) = \sqrt{1+\frac{\omega_U(\omega_U\sigma-\sigma-1)}{\alpha_U}} \geq \sqrt{1-\frac{1}{2\gamma}},$$

with equality if and only if

$$\frac{\omega_U}{\alpha_U}(\omega_U\sigma-\sigma-1) = -\frac{1}{\gamma}.$$

Proof. Clearly, for $m(\boldsymbol{\theta}) \in [\frac{1}{2}, \gamma]$, $\Delta_U(\alpha_U, \sigma) \leq 0$ and $|1-\omega_U\lambda_1| = |1-\omega_U\lambda_2| = \tau(m(\boldsymbol{\theta}))$. In order to guarantee convergence, we require $\tau(m(\boldsymbol{\theta}))^2 < 1$ (with equality allowed for $\boldsymbol{\theta} = 0$). This requires that $\frac{\omega_U(\omega_U\sigma-\sigma-1)}{\alpha_U} < 0$. Because $\gamma = \min\{m_2, 2\}$, it is easily seen that

$$\begin{aligned}\tau^2(\gamma) &= 1+\frac{\omega_U}{\alpha_U}(\omega_U\sigma-\sigma-1)\gamma\\ &\geq 1+\frac{\omega_U}{\alpha_U}(\omega_U\sigma-\sigma-1)m_2\\ &= \left(1-\frac{2\omega_U\sigma}{1+\sigma}\right)^2 \geq 0,\end{aligned}$$

which gives

$$\frac{\omega_U}{\alpha_U}(\omega_U\sigma-\sigma-1) \geq -\frac{1}{\gamma}.$$

It follows that

$$\max_{m(\boldsymbol{\theta})\in[\frac{1}{2},\gamma]} \tau(m(\boldsymbol{\theta})) = \tau\left(\frac{1}{2}\right) = \sqrt{1+\frac{\omega_U}{2\alpha_U}(\omega_U\sigma-\sigma-1)} \geq \sqrt{1-\frac{1}{2\gamma}},$$

and that equality is achieved if and only if $\frac{\omega_U(\omega_U\sigma-\sigma-1)}{\alpha_U} = \frac{-1}{\gamma}$. □

Lemma 4.5.2. *If $m_2 = \frac{4\alpha_U\sigma}{(1+\sigma)^2} > 2$, then $\tau^2(2) = 1 + \frac{\omega_U}{\alpha_U}(\omega_U\sigma - \sigma - 1)2 > 0$.*

Proof. For contradiction, assume that $\tau(2) = 1 + \frac{\omega_U}{\alpha_U}(\omega_U\sigma - \sigma - 1)2 = 0$, which gives $\frac{\alpha_U}{\omega_U(\sigma+1-\omega_U\sigma)} = 2$. Because $m_2 > 2$, we have

$$\frac{4\alpha_U\sigma}{(1+\sigma)^2} > \frac{\alpha_U}{\omega_U(\sigma+1-\omega_U\sigma)},$$

which can be rewritten as

$$\left(\frac{\omega_U\sigma}{1+\sigma} - \frac{1}{2}\right)^2 < 0.$$

□

These results allow us to obtain a bound on the smoothing factor when $m_2 > 2$.

Theorem 4.5.2. *If $m_2 = \frac{4\alpha_U\sigma}{(1+\sigma)^2} > 2$, then the optimal smoothing factor for inexact Uzawa relaxation is larger than $\frac{\sqrt{3}}{2}$.*

Proof. From Lemma 4.5.1, we know the smoothing factor for the complex modes is $SF_C = \tau(\frac{1}{2}) \geq \sqrt{1 - \frac{1}{2\gamma}} = \frac{\sqrt{3}}{2}$ with equality if and only if $\tau^2(2) = 0$. However, from Lemma 4.5.2, we know when $m_2 > 2$, $\tau^2(2) \neq 0$. This implies that the optimal smoothing factor is larger than $\frac{\sqrt{3}}{2}$. □

We now consider the case where $m_2 \leq 2$. For $m(\boldsymbol{\theta}) \in [m_2, 2]$, the two roots are real. From (4.27), we have

$$\begin{aligned} |1-\omega_U\lambda_1| &= \left|1 - \frac{(1+\sigma)\omega_U}{2\alpha_U} m(\boldsymbol{\theta})\left(1 + \sqrt{1 - \frac{m_2}{m(\boldsymbol{\theta})}}\right)\right|, \\ |1-\omega_U\lambda_2| &= \left|1 - \frac{(1+\sigma)\omega_U}{2\alpha_U} m(\boldsymbol{\theta})\left(1 - \sqrt{1 - \frac{m_2}{m(\boldsymbol{\theta})}}\right)\right|. \end{aligned}$$

Let

$$\begin{aligned} R_+(m(\boldsymbol{\theta})) &= \frac{m(\boldsymbol{\theta})}{2}\left(1 + \sqrt{1 - \frac{m_2}{m(\boldsymbol{\theta})}}\right), \\ R_-(m(\boldsymbol{\theta})) &= \frac{m(\boldsymbol{\theta})}{2}\left(1 - \sqrt{1 - \frac{m_2}{m(\boldsymbol{\theta})}}\right). \end{aligned}$$

Function $R_+(m(\boldsymbol{\theta}))$ is an increasing function of $m(\boldsymbol{\theta})$ for $m(\boldsymbol{\theta}) \in [m_2, 2]$, giving

$$\begin{aligned} R_1 := R_+(m(\boldsymbol{\theta}))_{\max} &= R_+(2) = 1 + \sqrt{1 - \frac{m_2}{2}}, \\ R_+(m(\boldsymbol{\theta}))_{\min} &= R_+(m_2) = \frac{m_2}{2}. \end{aligned}$$

For function $R_-(m(\boldsymbol{\theta}))$, since it is a decreasing function of $m(\boldsymbol{\theta})$, where $m(\boldsymbol{\theta}) \in [m_2, 2]$, we have

$$\begin{aligned} R_-(m(\boldsymbol{\theta}))_{\max} &= R_-(m_2) = \frac{m_2}{2}, \\ R_2 := R_-(m(\boldsymbol{\theta}))_{\min} &= R_-(2) = 1 - \sqrt{1 - \frac{m_2}{2}}. \end{aligned}$$

Remark 4.5.1. *$R_-(m(\boldsymbol{\theta}))$ is a decreasing function of $m(\boldsymbol{\theta})$, because $R_-(m(\boldsymbol{\theta}))' = \frac{\sqrt{1-\frac{m_2}{m(\boldsymbol{\theta})}}+\frac{m_2}{2m(\boldsymbol{\theta})}-1}{2\sqrt{1-\frac{m_2}{m(\boldsymbol{\theta})}}} < 0$ for all $m(\boldsymbol{\theta}) \in (m_2, 2]$.*

From the above discussion, the smoothing factor for the two real eigenvalues in this case is

$$\begin{aligned} SF_R : &= \max_{\boldsymbol{\theta} \in T^{\text{high}}} \left|\lambda(\widetilde{\mathcal{S}}_U(\alpha_U, \omega_U, \sigma, \boldsymbol{\theta}))\right| \\ &= \max\left\{ \left|1 - \frac{(1+\sigma)\omega_U}{\alpha_U} R_1\right|, \left|1 - \frac{(1+\sigma)\omega_U}{\alpha_U} R_2\right| \right\}. \end{aligned}$$

We can simplify the above expression by noting that

$$SF_R = \begin{cases} \dfrac{(1+\sigma)\omega_U}{\alpha_U} R_1 - 1, & \text{if} \quad \dfrac{(1+\sigma)\omega_U}{\alpha_U} \geq 1 \\ 1 - \dfrac{(1+\sigma)\omega_U}{\alpha_U} R_2, & \text{if} \quad \dfrac{(1+\sigma)\omega_U}{\alpha_U} \leq 1 \end{cases} \tag{4.28}$$

This allows us to bound the smoothing factor for the case when $m_2 \leq \frac{1}{2}$.

Theorem 4.5.3. *If $m_2 = \frac{4\alpha_U \sigma}{(1+\sigma)^2} \leq \frac{1}{2}$, then the optimal smoothing factor for inexact Uzawa relaxation is at least $\frac{\sqrt{3}}{2}$.*

Proof. Because $m_2 \leq \frac{1}{2}$, the eigenvalues are all real. According to (4.28), the smoothing factor for $m(\boldsymbol{\theta}) \in [\frac{1}{2}, 2]$ is

$$SF_R = \begin{cases} \frac{(1+\sigma)\omega_U}{\alpha_U}(1+\frac{\sqrt{3}}{2}) - 1, & \text{if} \quad \frac{(1+\sigma)\omega_U}{\alpha_U} \geq 1 \\ 1 - \frac{(1+\sigma)\omega_U}{\alpha_U}(1-\frac{\sqrt{3}}{2}), & \text{if} \quad \frac{(1+\sigma)\omega_U}{\alpha_U} \leq 1 \end{cases}$$

It is easy to see that when $\frac{(1+\sigma)\omega_U}{\alpha} = 1$, SF_R reaches its minimum value of $\frac{\sqrt{3}}{2}$. Note that the conditions that $\frac{(1+\sigma)\omega_U}{\alpha} = 1$ and $m_2 \leq \frac{1}{2}$ might not be satisfied at the same time, so the optimal smoothing factor may be larger than $\frac{\sqrt{3}}{2}$. □

We now consider the case where $\frac{1}{2} \leq m_2 \leq 2$. The key parameter in the proof is $\frac{(1+\sigma)\omega_U}{\alpha_U}$, which determines which of bounds on the real eigenvalues is dominant.

Theorem 4.5.4. *When $m_2 \in [\frac{1}{2}, 2]$, the optimal smoothing factor for σ-Uzawa relaxation is*

$$\begin{aligned} \mu_{\text{opt},\sigma U} &= \min_{(\alpha_U,\omega_U,\sigma)} \max_{\boldsymbol{\theta}\in T^{\text{high}}} \left\{ \left|1 - \frac{2\omega_U}{\alpha_U}\right|, \left|1 - \frac{\omega_U}{2\alpha_U}\right|, SF_R, SF_C \right\} \\ &= \sqrt{1 - \frac{m_{\text{opt}}}{2}} = \sqrt{\frac{3}{5}} \approx 0.7746, \end{aligned}$$

if and only if $m_2 = m_{\text{opt}} = \frac{4}{5}$, and the parameters satisfy

$$\begin{aligned} \frac{1}{5(2\mu_{\text{opt},U} - 1)} \leq \ \omega_U \ &\leq \frac{2}{5(1-\mu_{\text{opt},U})}, \\ \alpha_U &= \frac{5\omega_U^2}{5\omega_U - 1}, \\ \sigma &= \frac{1}{5\omega_U - 1}. \end{aligned}$$

Proof. We first consider the case where $\frac{(1+\sigma)\omega_U}{\alpha_U} = 1$, and the two expressions in (4.28) coincide. In this case, $m_2 = \frac{4\alpha_U\sigma}{(1+\sigma)^2} = 4\frac{\omega_U^2\sigma}{\alpha_U}$, and, for $m(\boldsymbol{\theta}) \in [m_2, 2]$,

$$SF_R = \frac{(1+\sigma)\omega_U}{\alpha_U} R_1 - 1 = \sqrt{1 - \frac{m_2}{2}} = \sqrt{1 - 2\frac{\omega_U^2\sigma}{\alpha_U}}.$$

For $m(\boldsymbol{\theta}) \in [\frac{1}{2}, m_2]$, from Lemma 4.5.1, we have

$$SF_C = \sqrt{1 + \frac{\omega_U(\omega_U\sigma - \sigma - 1)}{2\alpha_U}} = \sqrt{\frac{1}{2} + \frac{\omega_U^2\sigma}{2\alpha_U}}.$$

Because SF_R is a decreasing function of $\frac{\omega_U^2\sigma}{\alpha_U}$ and SF_C is an increasing function of $\frac{\omega_U^2\sigma}{\alpha_U}$, the optimal smoothing factor over the modes bounded by these factor is achieved if and only if $SF_R = SF_C$ and is given by

$$\mu_{\text{opt},\sigma U} = \min_{(\alpha_U,\omega_U,\sigma)} \max_{m(\boldsymbol{\theta})\in[\frac{1}{2},2]} \left\{ \sqrt{1 - 2\frac{\omega_U^2\sigma}{\alpha_U}}, \sqrt{\frac{1}{2} + \frac{\omega_U^2\sigma}{2\alpha_U}} \right\} = \sqrt{\frac{3}{5}}, \tag{4.29}$$

with the minimum occurring when

$$\frac{\omega_U^2\sigma}{\alpha_U} = \frac{1}{5}, \tag{4.30}$$

$$\frac{(1+\sigma)\omega_U}{\alpha_U} = 1. \tag{4.31}$$

Furthermore, $m_{\text{opt}} := m_2 = 4\frac{\omega_U^2\sigma}{\alpha_U} = \frac{4}{5}$. We now show this is the best possible bound over these two modes before returning to consider the eigenvalues $1 - \omega_U\frac{m(\boldsymbol{\theta})}{\alpha_U}$.

In the following, take $x = \frac{(1+\sigma)\omega_U}{\alpha_U}$, and $y = \frac{\omega_U^2\sigma}{\alpha_U}$, then $m_2 = \frac{4\alpha_U\sigma}{(1+\sigma)^2} = \frac{4y}{x^2}$. Assume that $SF_C \leq \sqrt{\frac{3}{5}}$, that is,

$$\sqrt{1 + \frac{\omega_U(\omega_U\sigma - \sigma - 1)}{2\alpha_U}} = \sqrt{1 - \frac{x}{2} + \frac{y}{2}} \leq \sqrt{\frac{3}{5}},$$

which implies that

$$y \leq x - \frac{4}{5}. \tag{4.32}$$

If $x > 1$, from (4.28) and (4.32), we have

$$\begin{aligned} SF_R &= \frac{(1+\sigma)\omega_U}{\alpha_U}\left(1+\sqrt{1-\frac{m_2}{2}}\right)-1 \\ &= x+\sqrt{x^2-2y}-1 \\ &\geq x+\sqrt{x^2-2(x-\frac{4}{5})}-1 \\ &= x-1+\sqrt{(x-1)^2+\frac{3}{5}} \\ &> \sqrt{\frac{3}{5}}. \end{aligned}$$

Therefore, when $x > 1$, the optimal smoothing factor is larger than $\sqrt{\frac{3}{5}}$.

If $x < 1$, from (4.28) and (4.32), we have

$$\begin{aligned} SF_R &= 1-\frac{(1+\sigma)\omega_U}{\alpha_U}\left(1-\sqrt{1-\frac{m_2}{2}}\right) \\ &= 1-x+\sqrt{x^2-2y} \\ &\geq 1-x+\sqrt{x^2-2(x-\frac{4}{5})}-1 \\ &= 1-x+\sqrt{(x-1)^2+\frac{3}{5}} \\ &> \sqrt{\frac{3}{5}}. \end{aligned}$$

Therefore, when $x < 1$, the optimal smoothing factor is larger than $\sqrt{\frac{3}{5}}$.

Thus, over all choices of x, the optimal smoothing factor that over these modes is $\mu_{\text{opt},U} = \sqrt{\frac{3}{5}}$, achieved when $x = \frac{(1+\sigma_U)\omega_U}{\alpha_U} = 1$.

We now consider the eigenvalue $\lambda_{*,U} = \frac{m(\boldsymbol{\theta})}{\alpha_U}$. We know that $\min\limits_{(\alpha_U,\omega_U,\sigma)} \max\limits_{\boldsymbol{\theta}\in T^{\text{high}}} \big|1 - \omega_U \frac{m(\boldsymbol{\theta})}{\alpha_U}\big| = \frac{3}{5} < \mu_{\text{opt},U} = \sqrt{\frac{3}{5}}$. In order to have this mode not be reduced more slowly than the others, we need

$$|1-\frac{2\omega_U}{\alpha_U}| \leq \mu_{\text{opt},U} \text{ and } |1-\frac{\omega_U}{2\alpha_U}| \leq \mu_{\text{opt},U},$$

which imply that

$$2(1-\mu_{\text{opt},U})\frac{1}{\omega_U} \le \frac{1}{\alpha_U} \le \frac{1+\mu_{\text{opt},U}}{2}\frac{1}{\omega_U}. \tag{4.33}$$

Simplifying (4.30) and (4.31), we have

$$\alpha_U = \frac{5\omega_U^2}{5\omega_U - 1}, \tag{4.34}$$

$$\sigma = \frac{1}{5\omega_U - 1}. \tag{4.35}$$

Using (4.34) and (4.35), (4.33) can be simplified as

$$\frac{1}{5(2\mu_{\text{opt},U}-1)} \le \omega_U \le \frac{2}{5(1-\mu_{\text{opt},U})}. \tag{4.36}$$

Note that the set of values defined by (4.34), (4.35), and (4.36) is not empty, with parameters $\omega_U = 1, \alpha_U = \frac{5}{4}, \sigma = \frac{1}{4}$ in this set.

□

Corollary 4.5.1. *The optimal smoothing factor for σ-Uzawa relaxation over all possible parameters is $\sqrt{\frac{3}{5}}$.*

Comparing this to the optimal smoothing factor for both exact and inexact Braess-Sarazin, $\frac{3}{5}$, we note that Braess-Sarazin relaxation offers better smoothing performance, but requires more work per iteration. In the following, we compare the computational work of these two methods and distributive relaxation.

4.5.3 Comparing among IBSR, σ-Uzawa, and DWJ relaxation

To end this section, we turn our attention to an estimate of the computational work for multigrid methods with σ-Uzawa, IBSR and DWJ relaxation. Because $\mu^2_{\text{opt},\sigma U} = \mu_{\text{opt},I}$, one cycle of multigrid with IBSR brings about the same total reduction in error as 2 cycles using σ-Uzawa relaxation. However, for IBSR and DWJ relaxation, $\mu_{\text{opt},I} = \mu_{\text{opt},D}$.

Considering the cost per sweep of IBSR and Uzawa relaxation, we see that inexact Braess-Sarazin is expected to be slightly more efficient. Recall the IBSR (4.6), where

$C = \text{diag}(A)$, requires inexact solution of

$$\begin{aligned}(BC^{-1}B^T)\delta p &= BC^{-1}r_{\mathcal{U}} - \alpha r_p,\\ \delta\mathcal{U} &= \frac{1}{\alpha}C^{-1}(r_{\mathcal{U}} - B^T\delta p).\end{aligned}$$

Because we use the standard finite-difference discretizations, C is just a diagonal matrix and C^{-1} is very simple to compute. For the first equation, we use a single sweep of weighted Jacobi iteration, having precomputed the approximate Schur complement, $B(C)^{-1}B^T$. Thus, the total cost of a single sweep of IBSR is that of 2 applications of C^{-1}, one sweep of weighted Jacobi for δp, one matrix-vector product each with B and B^T, and some vector updates. In σ-Uzawa relaxation, Equation (4.18) is equivalent to computing updates as

$$\begin{aligned}\delta\mathcal{U} &= (\alpha C)^{-1}r_{\mathcal{U}},\\ S\delta p &= B\delta\mathcal{U} - r_p.\end{aligned}$$

Thus, the total cost of a single sweep is that of one application of C^{-1}, one diagonal scaling for δp, one matrix-vector product with B, and some vector updates. Thus, the cost of 2 sweeps of σ-Uzawa is slightly more than one sweep of inexact Braess-Sarazin and, in this case, inexact Braess-Sarazin is more efficient.

In distributive weighted-Jacobi relaxation, Equation (4.5) is equivalent to computing updates as

$$\begin{aligned}\delta\hat{\mathcal{U}} &= (\alpha C)^{-1}r_{\mathcal{U}},\\ \delta\hat{p} &= \big(\alpha\text{diag}(A_p)\big)^{-1}(r_p - B\delta\hat{\mathcal{U}}),\end{aligned}$$

followed by distribution to the original unknowns by computing

$$\begin{aligned}\delta\mathcal{U} &= \delta\hat{\mathcal{U}} + B^T\delta\hat{p},\\ \delta p &= -A_p\delta\hat{p}.\end{aligned}$$

Thus, the total cost of a single sweep is one application of $(\alpha C)^{-1}$, one sweep of Jacobi on A_p, one matrix-vector product with B^T and B, one application of A_p, and some vector updates. Comparing with IBSR, the cost of one sweep of DWJ relaxation is slightly more than the cost of one sweep of IBSR.

Remark 4.5.2. *Similar comparisons are possible between inexact (S)GS-BSR and published results for DGS and (S)GS-Uzawa. For (S)GS-based methods, the cost of an (S)GS sweep on the velocity (or pressure) equations is somewhat more expensive than the diagonal scaling discussed above. For the inexact (S)GS-BSR algorithms discussed in Remarks 4.4.6 and 4.4.7, the cost is now that of two sweeps of (S)GS on the velocity equations, one sweep of weighted Jacobi (diagonal scaling) for the pressure, one matrix-vector product each with B and B^T, and some vector updates. The DGS algorithm of [11, 30] requires a single sweep of GS on the velocity equations plus one on the pressure unknowns, one matrix-vector product each with B and B^T as well as one with A_p, and some vector updates. In [27], LFA predicts a two-grid convergence factor for DGS of 0.4 when using 6-point interpolation and 12-point restriction, essentially the same as that predicted in Remark 4.4.7 for GS-BSR with the same grid-transfer operators. As the cost of the extra operations for the pressure block in DGS is quite similar to that of the second sweep of GS on the velocity block, we conclude that LFA predicts essentially the same efficiency for these two approaches. In [16], LFA for GS-Uzawa predicts a two-grid convergence factor of 0.87 when 2 sweeps of GS are used on the velocity block in each sweep of Uzawa. While this algorithm is slightly less expensive per iteration than GS-BSR (due to the lack of a multiplication with B^T), the convergence predicted here for GS-BSR is clearly superior, although we note that [16] does not allow for weighted-GS relaxation on the velocities as we use in GS-BSR. Also in [16], LFA predictions for SGS-Uzawa show a smoothing factor of 0.5 for Uzawa using a single sweep of SGS, and an LFA-predicted two-grid convergence factor of 0.44. Comparing these to the predictions in Remark 4.4.7, we see that two sweeps of SGS-Uzawa should yield essentially the same LFA-predicted reduction per cycle as one of SGS-BSR, at a slightly higher cost per iteration (due to the use of one diagonal scaling operation on the pressure in each sweep of SGS-Uzawa).*

4.6 Numerical experiments

In this section, we present the optimized smoothing and LFA two-grid convergence factors for DWJ, Braess-Sarazin-type, and Uzawa-type relaxation. Furthermore, we validate these predictions against measured multigrid convergence factors using distributive weighted-Jacobi, inexact Braess-Sarazin, and σ-Uzawa relaxations. The numerical results show good agreement between predicted convergence and the true performance, although some dependence is seen on the boundary conditions imposed, as noted elsewhere in the literature.

4.6.1 LFA spectral radius of error-propagation symbols

In this section, we show the spectral radius of the error-propagation symbol for DWJ, Braess-Sarazin, and Uzawa-type relaxation, computed with $h = \frac{1}{64}$. Figure 4.2 gives the spectral radius of the error-propagation symbol for DWJ as a function of $\boldsymbol{\theta}$, showing that DWJ relaxation reduces errors over the high frequencies quickly. Figure 4.3 displays these for BSR and IBSR, showing that both reduce the error over the high frequencies at a fast speed. Figure 4.4 displays these for Schur-Uzawa and σ-Uzawa. Here, we see very flat profiles in the upper right quadrant, particularly for the case of σ-Uzawa, which reduces the error at a much slower speed over the high frequencies.

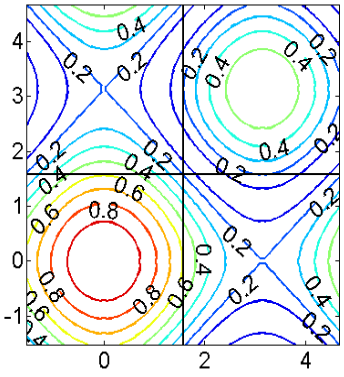

Figure 4.2: The spectral radius of the error-propagation symbol for DWJ, as a function of the Fourier mode, $\boldsymbol{\theta}$.

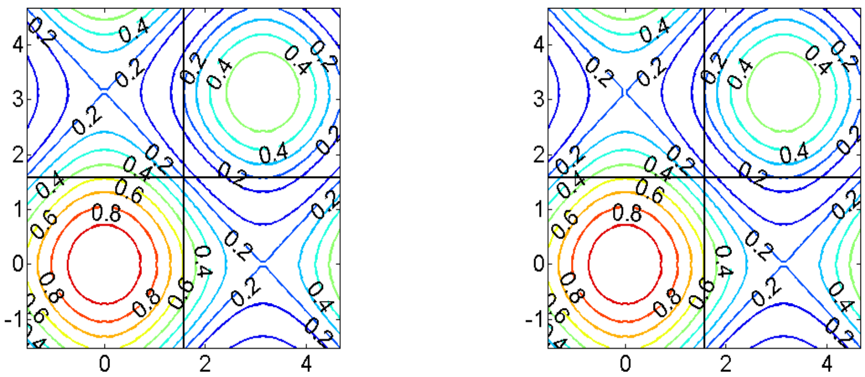

Figure 4.3: At left, the spectral radius of the error-propagation symbol for BSR, as a function of the Fourier mode, $\boldsymbol{\theta}$. At right, the spectral radius of the error-propagation symbol for IBSR.

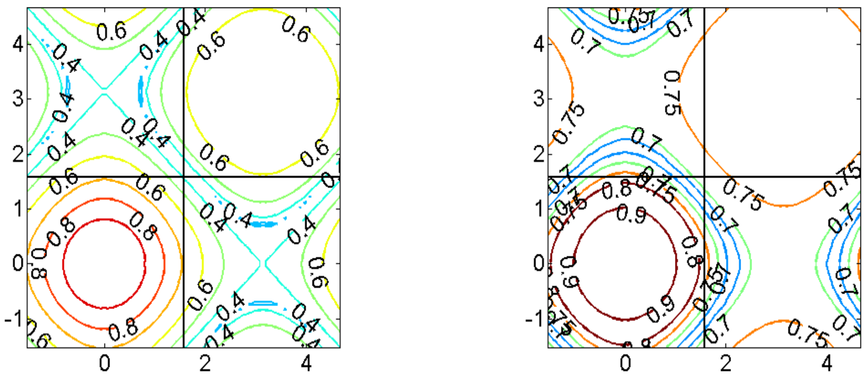

Figure 4.4: At left, the spectral radius of the error-propagation symbol for Schur-Uzawa, as a function of the Fourier mode, $\boldsymbol{\theta}$. At right, the spectral radius of the error-propagation symbol for σ-Uzawa.

4.6.2 LFA two-grid convergence factor

Let μ and ρ be the LFA-predicted smoothing and two-grid convergence factors, respectively, computed with $h = \frac{1}{64}$. For ρ, we first consider only one step of pre-smoothing (which gives the same results as one step of post-smoothing). At grid points corresponding to velocity unknowns, u and v, we consider six-point restrictions and at grid-points associated with pressure unknowns, p, a four-point cell-centered restriction is applied. For the prolongation of the corrections, we apply the corresponding adjoint operators multiplied by a factor of 4 or bilinear interpolation for velocity (12pts) and pressure (16pts) see, e.g., [27]. In Table 4.1, we give the choices of parameters for the relaxation schemes analyzed in the previous sections to present our LFA two-grid convergence factors. Note that parameter ω_J appears only in the IBSR algorithm, and σ only in σ-Uzawa.

Table 4.1: Relaxation parameter choices.

Relaxation parameter	DWJ	BSR	IBSR	Schur-Uzawa	σ-Uzawa
ω	1	1	1	$\frac{4}{\sqrt{73}-3}$	1
α	$\frac{5}{4}\omega = \frac{5}{4}$	$\frac{5}{4}\omega = \frac{5}{4}$	$\frac{5}{4}$	$\frac{4}{\sqrt{73}-5}$	$\frac{5\omega^2}{5\omega-1} = \frac{5}{4}$
ω_J or σ	\	\	$\frac{4}{5}$	\	$\frac{1}{5\omega-1} = \frac{1}{4}$
μ_{opt}	$\frac{3}{5}$	$\frac{3}{5}$	$\frac{3}{5}$	$\sqrt{\frac{33-3\sqrt{73}}{41-3\sqrt{73}}}$	$\sqrt{\frac{3}{5}}$

Figures 4.5-4.9 show the spectra of the two-grid error-propagation operators for different relaxation methods. In Figure 4.5, both linear and bilinear interpolation result in the same convergence factor $\mu = 0.600$, which is equal to the optimal smoothing factor for DWJ. In Figure 4.5, we see many eigenvalues with linear interpolation cluster around zero compared with the bilinear case. This might indicate that the linear interpolation operator produces an algorithm that reduces the error better. In Figure 4.6, we again have $\rho = \mu$ for both linear and bilinear interpolation for exact Braess-Sarazin relaxation, with some complex eigenvalues for the linear case, while all of the eigenvalues for bilinear interpolation are real. In Figure 4.7, we see some more significant differences between the distribution of the eigenvalues for the linear and bilinear cases, however the resulting spectral radii are the same. In Figure 4.8, for Schur Uzawa, we see that the two-grid spectral radius is larger than the smoothing factor with linear interpolation, but is the same as smoothing factor with bilinear interpolation. In Figure 4.9, both linear and bilinear interpolation for σ-Uzawa relaxation achieve the same convergence factor, $\rho = \sqrt{\frac{3}{5}}$, which is the same as the optimal smoothing factor, $\mu = \sqrt{\frac{3}{5}}$. All of these pictures confirm our theoretical optimal smoothing factors presented in previous sections, showing the (generally small) effect of the choice of interpolation.

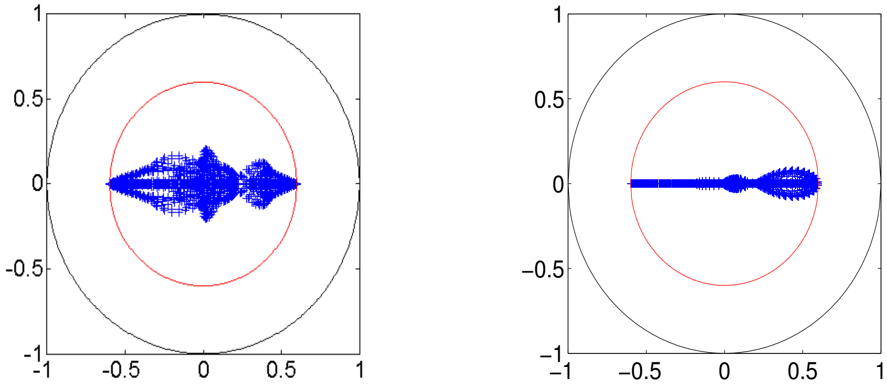

Figure 4.5: At left, the spectrum of the two-grid error-propagation operator for DWJ with linear interpolation. $\rho = \mu = 0.6000$. At right, the spectrum of the two-grid error-propagation operator for DWJ with bilinear interpolation. $\rho = \mu = 0.6000$.

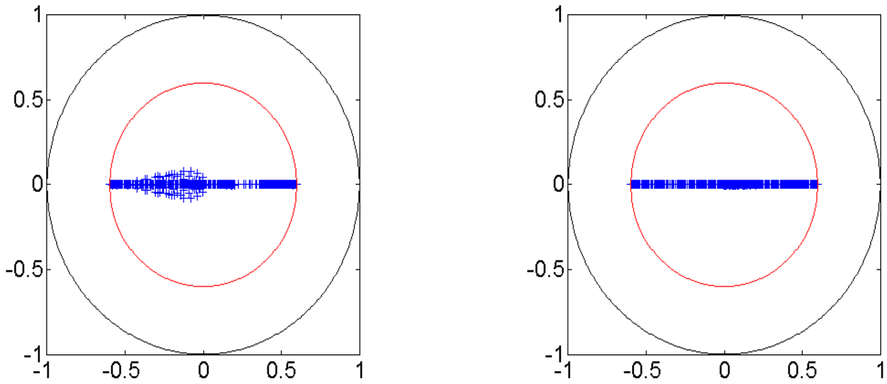

Figure 4.6: At left, the spectrum of the two-grid error-propagation operator for exact BSR with linear interpolation. $\rho = \mu = 0.6000$. At right, the spectrum of the two-grid error-propagation operator for exact BSR with bilinear interpolation. $\rho = \mu = 0.6000$.

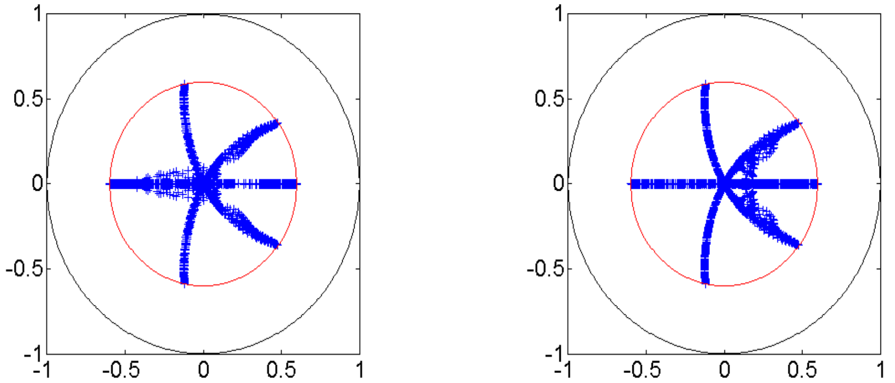

Figure 4.7: At left, the spectrum of the two-grid error-propagation operator for IBSR with linear interpolation. $\rho = \mu = 0.6000$. At right, the spectrum of the two-grid error-propagation operator for IBSR with bilinear interpolation. $\rho = \mu = 0.6000$.

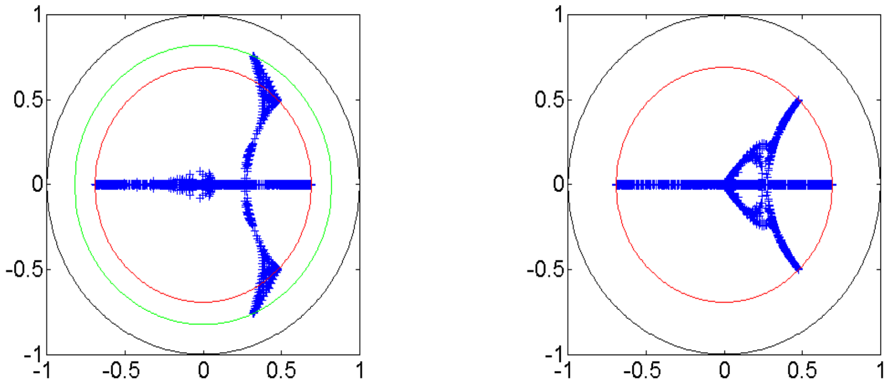

Figure 4.8: At left, the spectrum of the two-grid error-propagation operator for Schur-Uzawa with linear interpolation. $\rho = 0.8240, \mu = 0.6924$. At right, the spectrum of the two-grid error-propagation operator for Schur-Uzawa with bilinear interpolation. $\rho = \mu = 0.6924$.

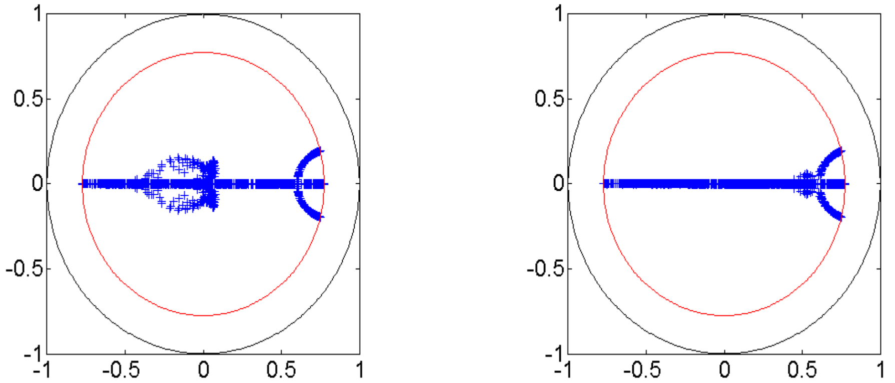

Figure 4.9: At left, the spectrum of the two-grid error-propagation operator for σ-Uzawa with linear interpolation. $\rho = \mu = \sqrt{\frac{3}{5}}$. At right, the spectrum of the two-grid error-propagation operator for σ-Uzawa with bilinear interpolation. $\rho = \mu = \sqrt{\frac{3}{5}}$.

4.6.3 Sensitivity of LFA-predicted convergence factors to parameter choice

In the analysis above, we give the optimal parameter choices for three block-structured relaxation schemes. Here, we present the LFA convergence factor for DWJ, IBSR, and σ-Uzawa as a function of these parameters, to show the sensitivity of performance to parameter choice. We consider the case of linear interpolation, where the LFA smoothing factor and predicted two-grid convergence factors match. Note that Theorem 4.3.1 demonstrates that the smoothing factor for DWJ is a function of $\frac{\omega_D}{\alpha_D}$ (but the same is not necessarily true for the convergence factor). In Figure 4.10, we plot the LFA smoothing and convergence factors for DWJ as a function of ω_D, with $\alpha_D = 1.0$, and see that these factors agree. To fix a single parameter for IBSR and σ-Uzawa, we consider choices motivated by their theoretical analysis, fixing $\omega_I = \frac{4}{5}\alpha_I$ for IBSR and $\sigma = \frac{1}{5\omega_U - 1}$ for σ-Uzawa. At the left of Figure 4.11, we present the LFA-predicted convergence factors for IBSR with variation in α_I and ω_J, seeing much stronger sensitivity to variations in ω_J than α_I, again with worse sensitivity to values larger than the optimal. At the right of Figure 4.11, we present the LFA-predicted convergence factors for σ-Uzawa as a function of α_U and ω_U. Here, we see great sensitivity for small values of α_U and large values of ω_U, but otherwise generally similar performance to the optimal parameter case.

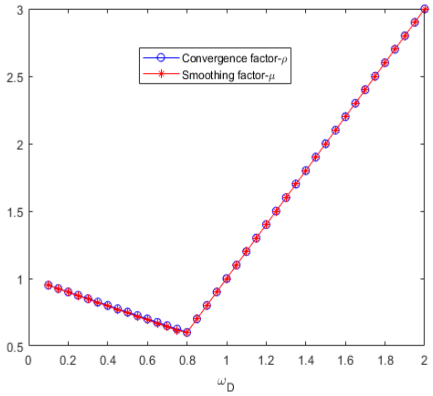

Figure 4.10: The two-grid LFA convergence and smoothing factors for DWJ, as a function of ω_D with $\alpha_D = 1$.

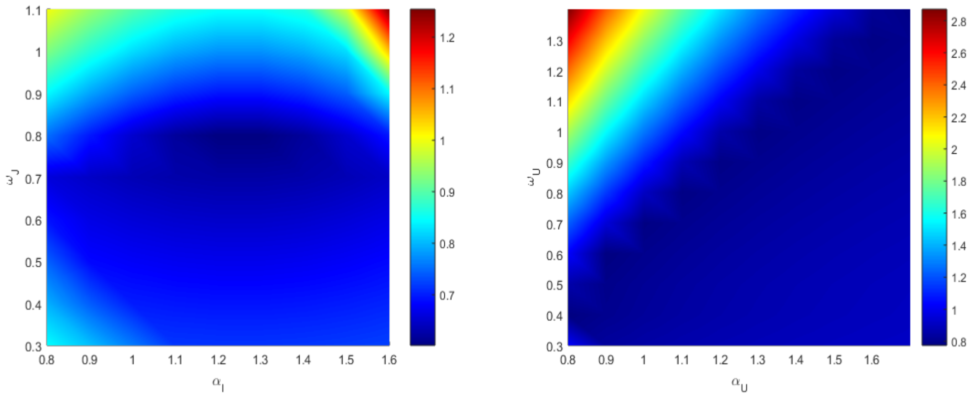

Figure 4.11: At left, the two-grid LFA convergence factor for IBSR, as a function of α_I and ω_J. At right, the two-grid LFA convergence factor for σ-Uzawa, as a function of α_U and ω_U.

4.6.4 Multigrid convergence factor

We now validate our LFA results against measured multigrid performance. We use the notation $W(\nu_1, \nu_2)$ to indicate the cycle type and the number of pre- and postsmoothing steps employed. Here, we use the defects (full system residuals in (4.3)) $d_h^{(k)} (k = 1, 2, \cdots)$ to experimentally measure the convergence factor as $\hat{\rho}_h^{(k)} = \sqrt[k]{\frac{\|d_h^{(k)}\|_2}{\|d_h^{(0)}\|_2}}$ (see [36]), with $k = 100$. We consider the homogeneous problem ($b = 0$) with discrete solution $x_h \equiv 0$, and start with a random initial guess $x^{(0)}$ to test the multigrid convergence factor. The coarsest grid is a 4×4 mesh. Rediscretization is used to define the coarse-grid operator. For comparison, we present the LFA predicated convergence factors, ρ_h, for two-grid cycles with ν_1 prerelaxation and ν_2 postrelaxation steps.

In Table 4.2, we present the multigrid performance of DWJ relaxation with Dirichlet boundary conditions. We see the same degradation in actual convergence behavior as was mentioned for DGS in [27] and note that performance is h-independent. Furthermore, as we increase the number of relaxation sweeps, we see degradation in even the LFA predication as compared to $\mu^{\nu_1+\nu_2}$ for bilinear interpolation. In order to see that boundary conditions play an important role in multigrid performance, we present the case of periodic boundary conditions in Table 4.3. These results show measured multigrid convergence factors that coincide with the LFA-predicated convergence factors. In both [10, 34], it is shown that additional boundary relaxation may be needed in order to achieve the convergence factors predicted by LFA, and this appears to be the case here for Dirichlet boundary conditions. We also note that [36] suggests the specific augmentation of Vanka-style box relaxation in place of distributed relaxation near the domain boundaries. Comparing linear and bilinear interpolation, these results indicate that linear interpolation outperforms bilinear interpolation in this case, matching some existing studies [30, 39, 40] for other relaxation schemes. Table 4.4 shows that the measured multigrid convergence factors again match well with the LFA-predicted two-grid convergence factors for IBSR with Dirichlet boundary conditions, and that the convergence is h-independent. We note no major differences in results between linear and bilinear interpolation, except a small one (that is captured by the LFA) for $W(2,2)$ cycles. Similar results are seen with periodic boundary conditions.

For the σ-Uzawa relaxation, there are many choices for ω_U, α_U, and σ, see Theorem 4.5.4. We tested a range of parameter values for the multigrid method with Dirichlet boundary conditions, and found that the choice of $\omega_U = \frac{1}{5(2\sqrt{3/5}-1)}$ is typically best. Thus, we use this value in our numerical results. In Table 4.5, the measured multigrid convergence factor degrades for $\nu_1 + \nu_2 > 1$ for both linear and bilinear interpolation with Dirichlet boundary conditions, and the same behavior was seen using a two-grid method. To confirm this is due to LFA doing a poor job of capturing the effects of boundary conditions, we tested the σ-Uzawa relaxation with periodic boundary conditions. In Table 4.6, we see no major difference between the measured convergence using linear and bilinear interpolation with periodic boundary conditions, and good agreement between the LFA-predicted convergence factor and the measured multigrid convergence factor. Comparing Table 4.6 with Table 4.5, we conclude that the degradation seen in Table 4.5 is, in fact, due to boundary conditions.

Remark 4.6.1. *We also tested the LFA-predicated two-grid convergence factors using Galerkin coarse-grid operators for the different relaxation schemes discussed in this paper. The convergence factors were almost the same as the ones obtained above using rediscretization coarse-grid operators for bilinear interpolation. However, for the case of linear interpolation, we see a large degradation in performance.*

Remark 4.6.2. *We see similar good performance for IBSR when using F-cycles; however, this is true only for Uzawa-type and distributive weighted-Jacobi relaxation on the problem with periodic boundary conditions. For $V(\nu_1, \nu_2)$-cycles with linear interpolation, when $\nu_1 + \nu_2 = 1$, both Braess-Sarazin-type and Uzawa relaxations are divergent. However, when $\nu_1 + \nu_2 > 1$, Braess-Sarazin relaxation works well for both Dirichlet and periodic boundary conditions, but Uzawa only works well for periodic boundary conditions. This is consistent with other studies of these relaxation schemes such as [16]. DWJ relaxation has similar behavior as Braess-Sarazin relaxation. For $V(\nu_1, \nu_2)$-cycles with bilinear interpolation, all of these three relaxation schemes are convergent with both Dirichlet and periodic boundary conditions, although there is a different degradation for each case, compared with the LFA predications.*

Table 4.2: Multigrid convergence factor for DWJ–Dirichlet BC.

Cycle $\hat{\rho}_h$	$W(0,1)$	$W(1,0)$	$W(1,1)$	$W(1,2)$	$W(2,1)$	$W(2,2)$
Linear interpolation						
$\rho_{h=1/256}$	0.600	0.600	0.360	0.216	0.216	0.130
$\hat{\rho}^{(100)}_{h=1/256}$	0.670	0.670	0.476	0.337	0.337	0.240
$\hat{\rho}^{(100)}_{h=1/128}$	0.673	0.672	0.475	0.338	0.337	0.240
Bilinear interpolation						
$\rho_{h=1/256}$	0.600	0.600	0.397	0.319	0.319	0.269
$\hat{\rho}^{(100)}_{h=1/256}$	0.668	0.668	0.474	0.340	0.340	0.270
$\hat{\rho}^{(100)}_{h=1/128}$	0.671	0.670	0.476	0.341	0.341	0.270

Table 4.3: Multigrid convergence factor for DWJ–Periodic BC.

Cycle $\hat{\rho}_h$	$W(0,1)$	$W(1,0)$	$W(1,1)$	$W(1,2)$	$W(2,1)$	$W(2,2)$
Linear interpolation						
$\rho_{h=1/256}$	0.600	0.600	0.360	0.216	0.216	0.130
$\hat{\rho}^{(100)}_{h=1/256}$	0.584	0.585	0.350	0.210	0.210	0.126
$\hat{\rho}^{(100)}_{h=1/128}$	0.584	0.585	0.350	0.211	0.210	0.127
Bilinear interpolation						
$\rho_{h=1/256}$	0.600	0.600	0.397	0.319	0.319	0.269
$\hat{\rho}^{(100)}_{h=1/256}$	0.584	0.584	0.381	0.303	0.302	0.253
$\hat{\rho}^{(100)}_{h=1/128}$	0.585	0.584	0.381	0.302	0.302	0.253

Table 4.4: Multigrid convergence factor for IBSR–Dirichlet BC.

Cycle $\hat{\rho}_h$	$W(0,1)$	$W(1,0)$	$W(1,1)$	$W(1,2)$	$W(2,1)$	$W(2,2)$
Linear interpolation						
$\rho_{h=1/256}$	0.600	0.600	0.360	0.216	0.216	0.130
$\hat{\rho}^{(100)}_{h=1/256}$	0.583	0.583	0.350	0.212	0.214	0.130
$\hat{\rho}^{(100)}_{h=1/128}$	0.583	0.582	0.350	0.214	0.213	0.130
Bilinear interpolation						
$\rho_{h=1/256}$	0.600	0.600	0.360	0.216	0.216	0.153
$\hat{\rho}^{(100)}_{h=1/256}$	0.582	0.581	0.349	0.209	0.209	0.146
$\hat{\rho}^{(100)}_{h=1/128}$	0.582	0.581	0.349	0.208	0.208	0.145

Table 4.5: $\omega_U = \frac{1}{5(2\sqrt{3/5}-1)}$: Multigrid convergence factor for σ-Uzawa–Dirichlet BC.

Cycle $\hat{\rho}_h$	$W(0,1)$	$W(1,0)$	$W(1,1)$	$W(1,2)$	$W(2,1)$	$W(2,2)$
Linear interpolation						
$\rho_{h=1/256}$	0.775	0.775	0.600	0.465	0.465	0.360
$\hat{\rho}^{(100)}_{h=1/256}$	0.767	0.777	0.646	0.533	0.532	0.447
$\hat{\rho}^{(100)}_{h=1/128}$	0.780	0.783	0.646	0.540	0.538	0.450
Bilinear interpolation						
$\rho_{h=1/256}$	0.775	0.775	0.600	0.465	0.465	0.360
$\hat{\rho}^{(100)}_{h=1/256}$	0.775	0.778	0.644	0.534	0.534	0.445
$\hat{\rho}^{(100)}_{h=1/128}$	0.781	0.780	0.648	0.537	0.537	0.446

Table 4.6: Multigrid convergence factor for σ-Uzawa–Periodic BC.

Cycle $\hat{\rho}_h$	$W(0,1)$	$W(1,0)$	$W(1,1)$	$W(1,2)$	$W(2,1)$	$W(2,2)$
Linear interpolation						
$\rho_{h=1/256}$	0.775	0.775	0.600	0.465	0.465	0.360
$\hat{\rho}^{(100)}_{h=1/256}$	0.752	0.752	0.580	0.449	0.449	0.347
$\hat{\rho}^{(100)}_{h=1/128}$	0.752	0.753	0.580	0.448	0.448	0.347
Bilinear interpolation						
$\rho_{h=1/256}$	0.775	0.775	0.600	0.465	0.465	0.360
$\hat{\rho}^{(100)}_{h=1/256}$	0.751	0.751	0.580	0.449	0.449	0.347
$\hat{\rho}^{(100)}_{h=1/128}$	0.753	0.751	0.579	0.448	0.448	0.347

4.7 Conclusions

In this paper, we develop an LFA for block-structured relaxation schemes for the Stokes equations. The convergence and smoothing theorems presented here provide us with optimized parameters for DWJ, Braess-Sarazin, and Uzawa relaxation. From the theory, the inexact Braess-Sarazin method has been proven to be as good as the exact iteration for solving the Stokes equations, with certain choices of parameters, and the convergence of the DWJ relaxation is as good as Braess-Sarazin with both offering slight improvement over Uzawa. For implementation, we consider the inexact cases, with weighted Jacobi iterations, as is suitable for use on modern in parallel and GPU architectures. In practice, we see much less sensitivity to boundary conditions for IBSR and, hence, generally recommend this as most efficient and robust of the approaches considered. Overall, the analysis presented here gives good insight into the use of block-structured relaxation for other types of saddle-point problems. The extensions of these block relaxation schemes to the Navier-Stokes equations in a nonlinear multigrid context is straightforward, but the analysis is not; this is a subject for future research. Developing LFA smoothing analysis to determine the optimal parameters in these relaxation schemes for finite-element discretization methods, for example, stable and stabilized rectangular elements for the Stokes Equation, will be a focus of our future research, as will be extensions to other saddle-point problems.

4.8 Appendix

In contrast to the results presented above, we now consider results for BSR using (symmetric) Gauss-Seidel relaxation for the velocity block.

4.8.1 BSR with (S)GS

First, we discuss the selection of C to be GS relaxation. One block stencil of C is

$$C_1 = \frac{1}{h^2}\begin{bmatrix} & 0 & \\ -1 & 4 & 0 \\ & -1 & \end{bmatrix}.$$

The symbol of C_1 is given by

$$\widetilde{C_1} = \frac{1}{h^2}(4 - e^{-i\theta_1} - e^{-i\theta_2}).$$

Taking $t = 4 - e^{-i\theta_1} - e^{-i\theta_2}$, then the stencil of $M_{G,E}$ is given by

$$\widetilde{M}_{G,E}(\theta_1,\theta_2) = \frac{1}{h^2}\begin{pmatrix} \alpha_{G,E}t & 0 & i2h\sin\frac{\theta_1}{2} \\ 0 & \alpha_{G,E}t & i2h\sin\frac{\theta_2}{2} \\ -i2h\sin\frac{\theta_1}{2} & -i2h\sin\frac{\theta_2}{2} & 0 \end{pmatrix}.$$

Furthermore,

$$\widetilde{L} - \lambda\widetilde{M}_{G,E} = \frac{1}{h^2}\begin{pmatrix} 4m(\boldsymbol{\theta}) - \alpha_{G,E}t\lambda & 0 & i2h\sin\frac{\theta_1}{2}(1-\lambda) \\ 0 & 4m(\boldsymbol{\theta}) - \alpha_{G,E}t\lambda & i2h\sin\frac{\theta_2}{2}(1-\lambda) \\ -i2h\sin\frac{\theta_1}{2}(1-\lambda) & -i2h\sin\frac{\theta_2}{2}(1-\lambda) & 0 \end{pmatrix}.$$

The determinant of $\widetilde{L} - \lambda\widetilde{M}_{G,E}$ is

$$\pi_{G,E}(\lambda;\alpha_{G,E}) = -\frac{4m(\boldsymbol{\theta})(4m(\boldsymbol{\theta}) - \alpha_{G,E}t\lambda)}{h^4}(1-\lambda)^2,$$

and the eigenvalues of $\widetilde{M}_{G,E}^{-1}\widetilde{\mathcal{L}}$ are given by

$$1, \quad 1, \quad \frac{4m(\boldsymbol{\theta})}{\alpha_{G,E}t}. \tag{4.37}$$

Considering the eigenvalue

$$\begin{aligned} \frac{4m(\boldsymbol{\theta})}{\alpha_{G,E}t} &= \frac{4 - e^{i\theta_1} - e^{i\theta_2} - e^{-i\theta_1} - e^{-i\theta_2}}{\alpha_{G,E}(4 - e^{-i\theta_1} - e^{-i\theta_2})} \\ &= \frac{1}{\alpha_{G,E}}\left(1 - \frac{e^{i\theta_1} + e^{i\theta_2}}{4 - e^{-i\theta_1} - e^{-i\theta_2}}\right), \end{aligned}$$

we can bound convergence using the inequality

$$\left|\frac{e^{i\theta_1} + e^{i\theta_2}}{4 - e^{-i\theta_1} - e^{-i\theta_2}}\right| \le \frac{1}{2}.$$

This inequality is strict when $\theta_1 = \frac{\pi}{2}, \theta_2 = \arccos(\frac{4}{5})$, as found in [39]. Thus, we can give the optimal smoothing factor for exact BSR with GS iteration.

Theorem 4.8.1. *The optimal smoothing factor for exact BSR with GS is*

$$\mu_{\text{opt}_{G,E}} = \min_{(\alpha_{G,E},\omega_{G,E})}\max_{\boldsymbol{\theta}\in T^{high}}\left\{\left|1 - \omega_{G,E}\frac{4m(\boldsymbol{\theta})}{\alpha_{G,E}t}\right|, \left|1 - \omega_{G,E}\right|\right\} = \frac{1}{2}.$$

Proof. Note that

$$\begin{aligned} & \big|1-\omega_{G,E}\frac{4m(\boldsymbol{\theta})}{\alpha_{G,E}t}\big| \\ = & \big|1-\frac{\omega_{G,E}}{\alpha_E}+\frac{\omega_{G,E}}{\alpha_{G,E}}\frac{e^{i\theta_1}+e^{i\theta_2}}{4-e^{-i\theta_1}-e^{-i\theta_2}}\big| \\ \leq & \big|1-\frac{\omega_{G,E}}{\alpha_{G,E}}\big|+\frac{1}{2}\cdot\frac{\omega_{G,E}}{\alpha_{G,E}}. \end{aligned}$$

Furthermore,

$$\min_{(\alpha_{G,E},\omega_{G,E})}\left(\big|1-\frac{\omega_{G,E}}{\alpha_{G,E}}\big|+\frac{\omega_{G,E}}{2\alpha_{G,E}}\right)=\frac{1}{2},$$

which is achieved for $\frac{\omega_{G,E}}{\alpha_{G,E}}=1$. It follows that the optimal smoothing factor for exact BSR with GS is as follows:

$$\mu_{\text{opt}_E}=\min_{(\alpha_{G,E},\omega_{G,E})}\max_{\boldsymbol{\theta}\in T^{high}}\left\{\big|1-\omega_{G,E}\frac{4m}{\alpha_{G,E}t}\big|,\big|1-\omega_{G,E}\big|\right\}=\frac{1}{2}, \tag{4.38}$$

and the choice $\frac{\alpha_{G,E}}{\omega_{G,E}}=1, \omega_{G,E}\in[\frac{1}{2},\frac{3}{2}]$ achieves the minimum. □

Remark 4.8.1. *Lexicographical GS for Laplace equation has the same smoothing factor of $\frac{1}{2}$, see [36].*

Now, we discuss the approximation of A by SGS; that is, $C_1=(D_A+L_A)D_A^{-1}(D_A+U_A)$, where D_A is the diagonal of the Laplace operator, L_A is the strict lower triangular part of the Laplace operator and U_A is the strict upper triangular part of the Laplace operator. The corresponding symbols of D_A, L_A and U_A are

$$\widetilde{D}_A=\frac{4}{h^2},\ \widetilde{L}_A=\frac{-e^{-i\theta_1}-e^{-i\theta_2}}{h^2},\ \widetilde{U}_A=\frac{-e^{i\theta_1}-e^{i\theta_2}}{h^2}.$$

Because $\widetilde{D}_A+\widetilde{L}_A=\frac{t}{h^2}$, $\widetilde{D}_A+\widetilde{U}_A=\frac{\bar{t}}{h^2}$. Furthermore, $\widetilde{C}_1=\frac{|t|^2}{4}$. So when we apply SGS to $M_{GS,E}$, we have the same two unit eigenvalues, as in (4.37). The third eigenvalue is now the same as applying SGS to the scalar Laplacian operator. It is well known that the smoothing factor of SGS is $\frac{1}{4}$. Thus, $\mu_{SGS,\text{opt}}=\frac{1}{4}$.

4.8.2 BSR with inexact (S)GS

The above relaxation scheme is impractical, since the Schur complement using the (S)GS approximation will be dense. Here, we replace that Schur complement by a simple diagonal scaling, giving an iteration with symbol

$$\widetilde{M}_{G,I}(\theta_1,\theta_2) = \frac{1}{h^2}\begin{pmatrix} \alpha_{G,I}t & 0 & i2h\sin\frac{\theta_1}{2} \\ 0 & \alpha_{G,I}t & i2h\sin\frac{\theta_2}{2} \\ -i2h\sin\frac{\theta_1}{2} & -i2h\sin\frac{\theta_2}{2} & h^2\beta_G \end{pmatrix},$$

where β_G is the symbol of $B(\alpha_{G,I}C)^{-1}B^T - \omega_{G,J}^{-1}I$,

$$\beta_G = (\alpha_{G,I}t)^{-1}4m(\boldsymbol{\theta}) - \omega_{G,J}^{-1}.$$

Now,

$$\widetilde{L} - \lambda\widetilde{M}_{G,I} = \frac{1}{h^2}\begin{pmatrix} 4m(\boldsymbol{\theta}) - \alpha_{G,I}t\lambda & 0 & i2h\sin\frac{\theta_1}{2}(1-\lambda) \\ 0 & 4m(\boldsymbol{\theta}) - \alpha_{G,I}t\lambda & i2h\sin\frac{\theta_2}{2}(1-\lambda) \\ -i2h\sin\frac{\theta_1}{2}(1-\lambda) & -i2h\sin\frac{\theta_2}{2}(1-\lambda) & -h^2\beta\lambda \end{pmatrix}.$$

The determinant of $\widetilde{L} - \lambda\widetilde{M}_{G,I}$ is

$$\begin{aligned} \pi_{G,I}(\lambda;\alpha_{G,I}) &= -\frac{4m(\boldsymbol{\theta}) - \alpha_{G,I}t\lambda}{(4m(\boldsymbol{\theta}) - \alpha_{G,I}\beta_G t)h^4}\big(\lambda^2 + \frac{(4\beta_G - 8)}{4m(\boldsymbol{\theta}) - \alpha_{G,I}t\beta_G}\lambda + \frac{4m(\boldsymbol{\theta})}{4m(\boldsymbol{\theta_G}) - \alpha_{G,I}t\beta_G}\big) \\ &= \frac{\omega_{G,J}}{h^4}\big(\lambda - \frac{4m(\boldsymbol{\theta})}{\alpha_{G,I}t}\big)\Big(\lambda^2 + \frac{4m(\boldsymbol{\theta})}{\alpha_{G,I}t}\big((\frac{4m(\boldsymbol{\theta})}{\alpha_{G,I}t} - 2)\omega_{G,J} - 1\big)\lambda + \frac{4m(\boldsymbol{\theta})}{\alpha_{G,I}t}\omega_{G,J}\Big). \end{aligned}$$

From the above equality, we know that one eigenvalue of $\widetilde{M}_{G,I}^{-1}\widetilde{\boldsymbol{\mathcal{L}}}$ is

$$\frac{4m(\boldsymbol{\theta})}{\alpha_{G,I}t},$$

which is the same eigenvalue as the exact GS-BSR.

From (4.38), we know

$$\min_{(\alpha_{G,I},\omega_{G,I})}\max_{\boldsymbol{\theta}\in T^{high}}\Big\{\big|1 - \omega_{G,I}\frac{4m(\boldsymbol{\theta})}{\alpha_{G,I}t}\big|\Big\} = \frac{1}{2},$$

providing that $\frac{\omega_{G,I}}{\alpha_{G,I}} = 1$. This tell us that the optimal smoothing factor for the IBSR is at least $\frac{1}{2}$, strictly providing that $\frac{\omega_{G,I}}{\alpha_{G,I}} = 1$. We do not know if the other two eigenvalues can be bounded accordingly.

Similarly, a reasonable inexact SGS-BSR algorithm again uses diagonal scaling for the pressure correction, based on the approximate Schur complement given in (4.8). It is easy to check that one eigenvalue of inexact SGS-BSR is the same as the one of exact SGS-BSR, which corresponds to the eigenvalue of SGS applied to scalar Laplacian operator. Thus, a lower bound on the optimal smoothing factor for inexact SGS-BSR is 0.25, but we do not know if a similar upper bound can be achieved.

Acknowledgements

The work of S.M. was partially funded by an NSERC Discovery Grant.

Bibliography

[1] J. Adler, T. R. Benson, E. Cyr, S. P. MacLachlan, and R. S. Tuminaro. Monolithic multigrid methods for two-dimensional resistive magnetohydrodynamics. *SIAM J. Sci. Comput.*, 38(1):B1–B24, 2016.

[2] 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.

[3] J. H. Adler, D. B. Emerson, S. P. MacLachlan, and T. A. Manteuffel. Constrained optimization for liquid crystal equilibria. *SIAM J. Sci. Comput.*, 38(1):B50–B76, 2016.

[4] C. Bacuta, P. S. Vassilevski, and S. Zhang. A new approach for solving Stokes systems arising from a distributive relaxation method. *Numerical Methods for Partial Differential Equations*, 27(4):898–914, 2011.

[5] R. E. Bank, B. D. Welfert, and H. Yserentant. A class of iterative methods for solving saddle point problems. *Numer. Math.*, 56(7):645–666, 1990.

[6] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numer.*, 14:1–137, 2005.

[7] D. Braess and W. Dahmen. A cascadic multigrid algorithm for the Stokes equations. *Numer. Math.*, 82(2):179–191, 1999.

[8] D. Braess and R. Sarazin. An efficient smoother for the Stokes problem. *Appl. Numer. Math.*, 23(1):3–19, 1997.

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

[10] A. Brandt. Rigorous quantitative analysis of multigrid, I. constant coefficients two-level cycle with L_2-norm. *SIAM J. Numer. Anal.*, 31(6):1695–1730, 1994.

[11] A. Brandt and N. Dinar. Multigrid solutions to elliptic flow problems. In *Numerical methods for partial differential equations*, volume 42 of *Publ. Math. Res. Center Univ. Wisconsin*, pages 53–147. Academic Press, New York-London, 1979.

[12] L. Chen. Multigrid methods for saddle point systems using constrained smoothers. *Comput. Math. Appl.*, 70(12):2854–2866, 2015.

[13] L. Chen, X. Hu, M. Wang, and J. Xu. A multigrid solver based on distributive smoother and residual overweighting for Oseen problems. *Numerical Mathematics: Theory, Methods and Applications*, 8(02):237–252, 2015.

[14] H. C. Elman and G. H. Golub. Inexact and preconditioned Uzawa algorithms for saddle point problems. *SIAM J. Numer. Anal.*, 31(6):1645–1661, 1994.

[15] 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.

[16] F. J. Gaspar, Y. Notay, C. W. Oosterlee, and C. Rodrigo. A simple and efficient segregated smoother for the discrete Stokes equations. *SIAM J. Sci. Comput.*, 36(3):A1187–A1206, 2014.

[17] B. Gmeiner, M. Huber, L. John, U. Rüde, and B. Wohlmuth. A quantitative performance study for Stokes solvers at the extreme scale. *J. Comput. Sci.*, 17(part 3):509–521, 2016.

[18] P. M. Gresho and R. L. Sani. On pressure boundary conditions for the incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 7(10):1111–1145, 1987.

[19] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8(12):2182–2189, 1965.

[20] L. John, U. Rüde, B. Wohlmuth, and W. Zulehner. On the analysis of block smoothers for saddle point problems. *arXiv preprint arXiv:1612.01333*, 2016.

[21] C. Keller, N. I. M. Gould, and A. J. Wathen. Constraint preconditioning for indefinite linear systems. *SIAM J. Matrix Anal. Appl.*, 21(4):1300–1317, 2000.

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

[23] P. Luo, C. Rodrigo, F. Gaspar, and C. Oosterlee. On an Uzawa smoother in multigrid for poroelasticity equations. *Numer. Linear Algebra Appl.*, 24(1), 2017.

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

[25] J.-F. Maitre, F. Musy, and P. Nigon. A fast solver for the Stokes equations using multigrid with a Uzawa smoother. In *Advances in multigrid methods (Oberwolfach, 1984)*, volume 11 of *Notes Numer. Fluid Mech.*, pages 77–83. Vieweg+Teubner Verlag, Wiesbaden, 1985.

[26] S. Manservisi. Numerical analysis of Vanka-type solvers for steady Stokes and Navier-Stokes flows. *SIAM J. Numer. Anal.*, 44(5):2025–2056, 2006.

[27] A. Niestegge and K. Witsch. Analysis of a multigrid Stokes solver. *Appl. Math. Comput.*, 35(3):291–303, 1990.

[28] M. A. Olshanskii. Multigrid analysis for the time dependent Stokes problem. *Math. Comp.*, 81(277):57–79, 2012.

[29] C. Oosterlee and F. Gaspar. Multigrid relaxation methods for systems of saddle point type. *Appl. Numer. Math.*, 58(12):1933–1950, 2008.

[30] C. W. Oosterlee and F. J. Gaspar. Multigrid methods for the Stokes system. *Computing in Science & Engineering*, 8(6):34–43, 2006.

[31] C. Rodrigo, F. J. Gaspar, and F. J. Lisbona. On a local Fourier analysis for overlapping block smoothers on triangular grids. *Appl. Numer. Math.*, 105:96–111, 2016.

[32] J. W. Ruge and K. Stüben. Algebraic multigrid. *Multigrid methods*, 3(13):73–130, 1987.

[33] J. Schöberl and W. Zulehner. On Schwarz-type smoothers for saddle point problems. *Numer. Math.*, 95(2):377–399, 2003.

[34] R. P. Stevenson. *On the validity of local mode analysis of multi-grid methods.* PhD thesis, Utrecht University, Utrecht, the Netherlands, 1990.

[35] S. Takacs. A robust multigrid method for the time-dependent Stokes problem. *SIAM J. Numer. Anal.*, 53(6):2634–2654, 2015.

[36] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid.* Academic Press, Inc., San Diego, CA, 2001. With contributions by A. Brandt, P. Oswald and K. Stüben.

[37] S. P. Vanka. Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. *J. Comput. Phys.*, 65(1):138–158, 1986.

[38] M. Wang and L. Chen. Multigrid methods for the Stokes equations using distributive Gauss-Seidel relaxations based on the least squares commutator. *J. Sci. Comput.*, 56(2):409–431, 2013.

[39] P. Wesseling. *An introduction to multigrid methods.* Pure and Applied Mathematics (New York). John Wiley & Sons, Ltd., Chichester, 1992.

[40] P. Wesseling and C. W. Oosterlee. Geometric multigrid with applications to computational fluid dynamics. *Journal of Computational and Applied Mathematics*, 128(1):311–334, 2001.

[41] G. Wittum. Multi-grid methods for Stokes and Navier-Stokes equations. *Numer. Math.*, 54(5):543–563, 1989.

[42] G. Wittum. On the robustness of ILU smoothing. *SIAM Journal on Scientific and Statistical Computing*, 10(4):699–717, 1989.

[43] G. Wittum. On the convergence of multi-grid methods with transforming smoothers. *Numer. Math.*, 57(1):15–38, 1990.

[44] W. Zulehner. A class of smoothers for saddle point problems. *Computing*, 65(3):227–246, 2000.

Chapter 5

Two-level Fourier analysis of multigrid for higher-order finite-element methods

Abstract

[1] In this paper, we employ local Fourier analysis (LFA) to analyze the convergence properties of multigrid methods for higher-order finite-element approximations to the Laplacian problem. We find that the LFA smoothing factor fails to accurately predict the observed multigrid performance. This failure of the LFA smoothing factor is explained, and we propose a modification to the analysis that yields a reasonable prediction to help choose the correct damping parameters for relaxation. Finally, we present two-grid and multigrid experiments, and the corrected parameter choice is shown to yield a significant improvement in the resulting two-grid and multigrid convergence factors.

Keywords: Finite-element method, higher-order elements, Jacobi iteration, local Fourier analysis, multigrid

AMS subject classification: 65M55, 65N30, 65Txx

[1] Authors: Y. He and S. P. MacLachlan
This work is submitted as *Two-level Fourier analysis of multigrid for higher-order finite-element methods*, to SIAM Journal on Matrix Analysis and Applications, 2018.

5.1 Introduction

Multigrid methods [2, 7, 19, 23, 24] are very popular to solve the linear systems that arise from the discretization of many PDEs. The choice of the multigrid components, such as grid transfer operators and the relaxation scheme, has a great influence on the performance of these algorithms. In this paper, we focus on the Laplace problem,

$$\begin{cases} -\Delta u(x) = f(x), & x \in \Omega, \\ \quad u(x) = g(x), & x \in \partial\Omega, \end{cases} \tag{5.1}$$

discretized using higher-order finite elements. In the literature, there are many efficient multigrid methods for problem (5.1), see [9, 21]. It is worthwhile, however, to understand how these methods work efficiently. LFA [21, 24] has proven a good tool for theoretical investigation and multigrid method design, including for the curl-curl equation [1, 15], parabolic partial differential equations [6, 22], the Stokes equations [10, 14, 15], and the Poisson equation [8, 17, 21].

Recently, some studies have reported that LFA fails to accurately predict some multigrid results, see [5, 6]. In [6], LFA does not offer its usual predictivity of the convergence behavior of the space-time diffusion equation and its generalizations. However, in [5], the authors develop new tools to make up for the failure of standard LFA to provide insight into the asymptotic convergence behaviour of multigrid methods for these problem. In [15], an LFA is presented for general problems, focusing on analyzing the complementarity between relaxation and coarse-grid correction (CGC) within multigrid solvers for systems of PDEs with finite-element discretizations. In that paper, the smoothing factor of LFA overestimates the two-grid convergence factor for the Taylor-Hood ($Q_2 - Q_1$) discretization of the Stokes equations. However, no further explanation is given. We show here that the failure might be related to the Q_2 approximation used for the velocity unknowns.

To our knowledge, the vast majority of existing LFA for the Poisson problem focuses on discretization using finite differences or linear finite elements [19, 21, 24]. In contrast, [8] studies the convergence of a multigrid method for the solution of a linear second-order elliptic equation by discontinuous Galerkin methods. In [17], the cell-centered finite-difference discretization on triangular grids is considered. A variant of LFA is applied to discretization matrices arising from Galerkin B-spline isogeometric analysis in [4], focusing on 2-level analysis in place of classical smoothing analysis. Here, we focus on standard higher-order finite-element discretizations of Poisson's equation with weighted Jacobi relaxation, and use LFA to understand performance. In contrast to the cases of standard finite-difference or (bi)linear finite-element discretizations, we will see that the LFA smoothing factor does not offer a good prediction of performance in the higher-order case.

In the literature, there are many studies about higher-order methods for different types of PDEs. The spectral element method for second-order problems was studied both numerically and theoretically in [16, 18], showing good smoothing properties of simple Jacobi relaxation for the Laplace problem. The impact of different higher-order finite-element discretizations for the Laplace problem on multigrid convergence, with Richardson and Jacobi relaxation, was considered in [13]. Comparison of different multigrid methods for higher-order finite-element discretizations, either as direct solvers or preconditioners, was reported in [20]. There, the convergence behaviour was seen to strongly depend on the polynomial order when multigrid is used as a preconditioner, but not for multigrid as a solver. Other studies of higher-order finite-element methods and multigrid include those for nonlinear problems [3] and the incompressible Navier-Stokes equations [11, 12].

Supporting numerical results demonstrate some key conclusions of our analysis. First, there is a notable gap between the classical LFA smoothing factor and the two-grid convergence factor for these elements. The standard LFA assumption of an "ideal" coarse-grid correction operator, which annihilates the low-frequency error components and leaves the high-frequency components unchanged is not true for higher-order finite-element discretizations, where our results show that the CGC reduces some high-frequency error quickly. Furthermore, minimizing the classical smoothing factor does not minimize the corresponding convergence factor.

The outline of the paper is as follows. In Section 5.2, we recall the standard definitions of LFA. In Section 5.3, we analyse the weighted Jacobi relaxation scheme for the Q_2 finite-element approximation in one dimension (1D) and show how to obtain optimal parameters to minimize the convergence factor. We extend this analysis to higher-order finite-elements in Section 5.4. In Section 5.5, two-grid LFA is presented for biquadratic Lagrangian elements in two dimensions (2D), and we discuss the optimal parameter choice. Conclusions are presented in Section 5.6.

5.2 Definitions and notations

In order to describe LFA for finite-element methods, we first introduce some terminology. More details can be found, for example, in [21]. We first consider one-dimensional infinite uniform grids, G_h. Let L_h be a scalar Toeplitz operator acting on $l^2(G_h)$

$$L_h \overset{\triangle}{=} [s_\kappa]_h \ (\kappa \in V); \ L_h w_h(x) = \sum_{\kappa \in V} s_\kappa w_h(x + \kappa h), \tag{5.2}$$

with constant coefficients $s_\kappa \in \mathbb{R}$ (or $\mathbb{C}$), where $w_h(x)$ is a function in $l^2(G_h)$. Here, V is taken to be a finite index set of integers, $V \subset \mathbb{Z}$. Note that since L_h is Toeplitz, it is diagonalized by the standard Fourier modes $\psi(\theta, x) = e^{\iota\theta\cdot x/h}$, where $\iota^2 = -1$.

Definition 5.2.1. *We call* $\widetilde{L}_h(\theta) = \sum_{\kappa \in V} s_\kappa e^{\iota\theta\kappa}$ *the symbol of* L_h.

Note that for all grid functions, $\psi(\theta, x)$,

$$L_h \psi(\theta, x) = \widetilde{L}_h(\theta)\psi(\theta, x).$$

Here, we consider multigrid methods for finite-element discretizations with standard geometric grid coarsening; that is, we construct a sequence of coarse grids by doubling the mesh size in each spatial direction. High and low frequencies for standard coarsening are given by

$$\theta \in T^{\text{low}} = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right), \theta \in T^{\text{high}} = \left[-\frac{\pi}{2}, \frac{3\pi}{2}\right) \setminus \left[-\frac{\pi}{2}, \frac{\pi}{2}\right).$$

The error-propagation operator for a relaxation scheme, represented similarly by a Toeplitz operator M_h, applied to a finite-element approximation is

$$\mathcal{S}_h(\omega, \theta) = I - \omega M_h^{-1} L_h,$$

where ω is an overall weighting factor.

Definition 5.2.2. *The error-propagation symbol,* $\widetilde{\mathcal{S}}_h(\theta)$, *for smoother* $\mathcal{S}_h$ *on the infinite grid* G_h *satisfies*

$$\mathcal{S}_h \psi(\theta, x) = \widetilde{\mathcal{S}}_h \psi(\theta, x), \ \theta \in \big[-\frac{\pi}{2}, \frac{3\pi}{2}\big),$$

for all $\psi(\theta, x)$, and the corresponding smoothing factor for $\mathcal{S}_h$ is given by

$$\mu_{\text{loc}} := \mu_{\text{loc}}(\mathcal{S}_h) = \max_{\theta \in T^{\text{high}}} \left\{ \left| \widetilde{\mathcal{S}}_h(\theta) \right| \right\}. \tag{5.3}$$

Definition 5.2.3. *Because the smoothing factor is a function of some parameters, let* $\mathbf{D}$ *be a bounded and closed set of allowable parameters and define the optimal smoothing factor over* $\mathbf{D}$ *as*

$$\mu_{\text{opt}} = \min_{\mathbf{D}} \mu_{\text{loc}}.$$

In what follows, we consider $(q \times q)$ linear systems of operators, which read

$$\mathbf{L}_h = \begin{pmatrix} L_h^{1,1} & \cdots & L_h^{1,q} \\ \vdots & \cdots & \vdots \\ L_h^{q,1} & \cdots & L_h^{q,q} \end{pmatrix}.$$

The $L_h^{i,j}(i, j = 1, 2, \ldots, q)$ are scalar Toeplitz operators. Each entry in $\widetilde{\mathbf{L}}_h$ is computed as the (scalar) symbol of the corresponding block of $L_h^{i,j}$, following Definition 5.2.1. For simplicity, we reuse the notation in (5.3) for the case of block symbols as described in the following.

On a collocated mesh, all blocks in $\mathbf{L}_h$ are diagonalized by the same transformation. However, in our setting, we consider $G_h = G_{h,N} \bigcup G_{h,C}$, for quadratic Lagrangian elements, with

$$G_{h,N} = \left\{ x_{k,N} := kh, k \in \mathbb{Z} \right\}, \text{and } G_{h,C} = \left\{ x_{k,C} := kh + h/2, k \in \mathbb{Z} \right\}. \tag{5.4}$$

Here G_h contains two types of meshpoints, the nodes of the mesh and the cell centres. The coarse grid, G_{2h}, is defined similarly. Each block $L_h^{i,j}$ in $\mathbf{L}_h$ for $i, j = 1, 2$ is defined as in (5.2), with V taken to be either a finite index set of integer (V_N) or half-integer (V_C) values, with $V_N \subset \mathbb{Z}$ and $V_C \subset \left\{ z + \frac{1}{2} | z \in \mathbb{Z} \right\}$. The operators discussed later are naturally treated as block operators, and the Fourier representation of each block can be calculated based on Definition 5.2.1, with Fourier bases adapted to account for the staggering of the mesh points. In Definition 5.2.2, the symbol $\widetilde{\mathcal{S}}_h(\theta)$ will be a matrix, thus, $\left| \widetilde{\mathcal{S}}_h(\theta) \right|$ is replaced by $\left| \lambda(\widetilde{\mathcal{S}}_h(\theta)) \right|$, the absolute value of the eigenvalues of $\widetilde{\mathcal{S}}_h(\theta)$, in (5.3).

The resulting Fourier functions are $\varphi(\theta, x_k) \in \text{span}\left\{ \varphi_N(\theta, x_k), \varphi_C(\theta, x_k) \right\}$ on G_h, in which

$$\varphi_N(\theta, x_k) \;= \begin{pmatrix} e^{\iota\theta\cdot x_{k,N}/h} & 0 \end{pmatrix}^T, \; \varphi_C(\theta, x_k) = \begin{pmatrix} 0 & e^{\iota\theta\cdot x_{k,C}/h} \end{pmatrix}^T,$$

where T denotes the (non-conjugate) transpose of the row vectors. Because $\varphi(\theta, x_k)$ is periodic in θ with period 2π, we consider the domain $\theta \in \big[-\frac{\pi}{2}, \frac{3\pi}{2}\big)$.

5.3 LFA for quadratics in 1D

Here, we consider the discretization of problem (5.1) in 1D, using quadratic (Q_2) finite elements, and nodal basis functions defined at the nodes of the mesh and cell centres (but the analysis could be modified for other bases), and will focus on weighted Jacobi relaxation.

5.3.1 Quadratic Lagrangian Elements

For these quadratic Lagrangian elements, the elementary contributions to the stiffness and mass matrices as 3×3 symmetric matrices are

$$EK = \frac{1}{3h}\begin{pmatrix} 7 & -8 & 1 \\ -8 & 16 & -8 \\ 1 & -8 & 7 \end{pmatrix}, \quad EM = \frac{h}{30}\begin{pmatrix} 4 & 2 & -1 \\ 2 & 16 & 2 \\ -1 & 2 & 4 \end{pmatrix},$$

respectively. We can decompose the resulting stencils into connections among and between the degrees of freedom (DOFs) located at the nodes of the mesh and those located at cell centres. The node-to-node connections yield the stencils

$$\frac{1}{3h}\begin{bmatrix} 1 & 14 & 1 \end{bmatrix} \text{ and } \frac{h}{30}\begin{bmatrix} -1 & 8 & -1 \end{bmatrix}.$$

The node-to-centre stencils are given by

$$\frac{1}{3h}\begin{bmatrix} -8 & \star & -8 \end{bmatrix} \text{ and } \frac{h}{30}\begin{bmatrix} 2 & \star & 2 \end{bmatrix},$$

with transposed connections between centres and nodes, where $\star$ stands for the degree-of-freedom position in the off-diagonal blocks. The centre-to-centre stencils are diagonal,

$$\frac{1}{3h}\left[16\right] \quad \text{and} \quad \frac{h}{30}\left[16\right].$$

On the infinite grid G_h, each of these stencils defines a Toeplitz operator on $\ell_2(G_{h,*})$ and, so, the block systems can be block diagonalized by considering the invariant subspace given by linear combinations of $\varphi_N(\theta, x)$ and $\varphi_C(\theta, x)$. The resulting block symbols of the stiffness and mass operators are

$$\widetilde{A}_h(\theta) = \frac{1}{3h}\begin{pmatrix} 14 + 2\cos\theta & -16\cos\frac{\theta}{2} \\ -16\cos\frac{\theta}{2} & 16 \end{pmatrix}, \; \widetilde{B}_h(\theta) = \frac{h}{30}\begin{pmatrix} 8 - 2\cos\theta & 4\cos\frac{\theta}{2} \\ 4\cos\frac{\theta}{2} & 16 \end{pmatrix}, \tag{5.5}$$

respectively. The error-propagation symbol of weighted Jacobi relaxation is given by

$$\widetilde{\mathcal{S}}_h(\theta) = I - \omega\widetilde{M}_h^{-1}(\theta)\widetilde{A}_h(\theta), \tag{5.6}$$

where $\widetilde{M}_h(\theta)$ is the symbol of the diagonal operator,

$$M_h = \frac{1}{3h}\begin{pmatrix} 14I & 0 \\ 0 & 16I \end{pmatrix}. \tag{5.7}$$

Using (5.5) and (5.7), we plot the distribution of eigenvalues of $\widetilde{M}_h^{-1}(\theta)\widetilde{A}_h(\theta)$, at the left of Figure 5.1. Note that as a block symbol, $\widetilde{M}_h^{-1}(\theta)\widetilde{A}_h(\theta)$ has 2 eigenvalues, each of which can be seen to be a continuous function of θ/π.

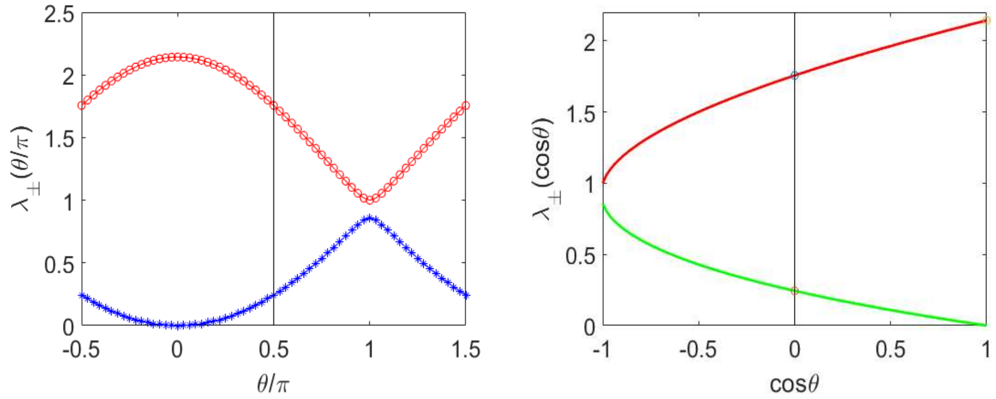

Figure 5.1: At left, the distribution of the two eigenvalues of $\widetilde{M}_h^{-1}(\theta)\widetilde{A}_h(\theta)$ as a function of θ/π. At right, the distribution of the two eigenvalues of $\widetilde{M}_h^{-1}(\theta)\widetilde{A}_h(\theta)$, as a function of $\cos\theta$.

To derive an analytical expression for the eigenvalues of $\widetilde{M_h}^{-1}(\theta)\widetilde{A}_h(\theta)$, we note that the determinant of $\widetilde{M_h}^{-1}(\theta)\widetilde{A}_h(\theta) - \lambda I$ is

$$(\lambda - 1)(\lambda - 1 - \frac{\cos\theta}{7}) - \frac{4}{7}(1 + \cos\theta).$$

Let λ_+ and λ_- be the eigenvalues of $\widetilde{M_h}^{-1}(\theta)\widetilde{A}_h(\theta)$; from above, we have

$$\lambda_\pm = \frac{14 + \cos\theta \pm \sqrt{\cos^2(\theta) + 112\cos\theta + 112}}{14}.$$

Taking $x = \cos\theta$, then we can write

$$\lambda_+(x) = \frac{14 + x + \sqrt{x^2 + 112x + 112}}{14}, \ \lambda_-(x) = \frac{14 + x - \sqrt{x^2 + 112x + 112}}{14}.$$

It is easy to check that

$$\begin{aligned} \lambda_+(x)_{\max} &= \lambda_+(1) = \frac{15}{7}, \qquad & \lambda_+(x)_{\min} &= \lambda_+(-1) = 1, \\ \lambda_-(x)_{\max} &= \lambda_+(-1) = \frac{6}{7}, \qquad & \lambda_-(x)_{\min} &= \lambda_-(1) = 0. \end{aligned}$$

We plot $\lambda_+(x), \lambda_-(x)$ at the right of Figure 5.1.

Throughout this paper, we denote $\lambda_{\text{max,H}}$ and $\lambda_{\text{min,H}}$ as the biggest and smallest eigenvalues over only the high frequency range, respectively. Since $\lambda_-(x) < \lambda_+(x)$, for high frequencies ($x \in [-1, 0]$), we have

$$\lambda_{\text{max,H}} = \lambda_+(0) = \frac{7 + 2\sqrt{7}}{7}, \quad \lambda_{\text{min,H}} = \lambda_-(0) = \frac{7 - 2\sqrt{7}}{7}.$$

Thus, the classical optimal choice of ω that minimizes the resulting smoothing factor for relaxation scheme (5.6) is given by

$$\omega^* = \frac{2}{\lambda_{\text{min,H}} + \lambda_{\text{max,H}}} = 1, \tag{5.8}$$

and the corresponding smoothing factor is

$$\mu_2^* = \min_{\omega} \max_{\theta \in T^{\text{high}}} \left|\lambda(\widetilde{\mathcal{S}}_h(\omega, \theta))\right| = \frac{2\sqrt{7}}{7} \approx 0.760.$$

Note, however, that this choice of ω^* leads to a diverging relaxation scheme, as $|1-\omega^*\lambda_+(1)| > 1$. While this might be acceptable assuming ideal CGC, it is worrisome from the perspective of robustness of the resulting multilevel algorithm. Thus, we consider another relaxation weight,

$$\omega^{**} = \frac{2}{\lambda^*_{\max} + \lambda_{\min,H}} = \frac{14}{22 - 2\sqrt{7}} \approx 0.838, \tag{5.9}$$

where $\lambda^*_{\max}$ is the biggest of all eigenvalues; that is $\lambda^*_{\max} = \lambda_+(1) = \frac{15}{7}$. For this choice, the corresponding smoothing factor is

$$\mu_2^{**} = \max_{\theta \in T^{\text{high}}} \left|\lambda(\widetilde{\mathcal{S}}_h(\omega^{**}, \theta))\right| = \frac{4+\sqrt{7}}{11-\sqrt{7}} \approx 0.795.$$

To understand and compare these choices, we now consider two-grid LFA and measured two-grid performance. We use the notation $TG(\nu_1, \nu_2)$ and $V(\nu_1, \nu_2)$ to indicate the cycle type and the number of pre- and postsmoothing steps employed. Here, we use the defects $d_h^{(k)}(k = 1, 2, \cdots$, with $d_h^{(k)} = b - A_h x_h^{(k)})$ to experimentally measure the convergence factor as $\hat{\rho}_h^{(k)} = \sqrt[k]{\frac{\|d_h^{(k)}\|_2}{\|d_h^{(0)}\|_2}}$ (see [21]), with $k = 100$. We consider the homogeneous problem, $A_h x_h = b = 0$, with discrete solution $x_h \equiv 0$, and start with a random initial guess, $x_h^{(0)}$, to test the multigrid convergence factor. The coarsest grid is a mesh with 4 elements. Rediscretization is used to define the coarse-grid operator (CGO). For comparison, we present the LFA-predicted convergence factors, ρ_h, for two-grid cycles with ν_1 prerelaxation and ν_2 postrelaxation steps (see (5.18)). We consider periodic boundary conditions.

In Table 5.1, we use ω^* as the weight. Note that the LFA convergence factor is larger than the smoothing factor. As noted earlier, while we see convergence for $\nu_1 + \nu_2 < 3$, we see divergence when $\nu_1 + \nu_2 = 3, 4$ for the two-grid method. Furthermore, even though the smoothing factor fails to predict the convergence factor, we see that the measured convergence factor matches well with the LFA-predicted two-grid convergence factor. For $\omega = \omega^{**}$, Table 5.2 shows a good improvement in the convergence factor compared with the choice of ω^*. We again see a good agreement between the measured convergence factor and the LFA-predicted convergence factor, but now the two-grid convergence factor is smaller than the smoothing factor, in contrast to the case of ω^*. Moreover, while the smoothing factor for the choice of ω^{**} is larger than that of ω^*, the two-grid factor is much better.

Table 5.1: Two-grid convergence factors for the Q_2 approximation with ω^* in 1D.

Cycle $\hat{\rho}_h$	$TG(0,1)$	$TG(1,0)$	$TG(1,1)$	$TG(1,2)$	$TG(2,1)$	$TG(2,2)$
$\omega = \omega^* = 1.000, \mu^* = 0.760$						
$\rho_{h=1/128}$	0.821	0.821	0.985	1.118	1.119	1.279
$\hat{\rho}^{(100)}_{h=1/128}$	0.813	0.815	0.974	1.096	1.102	1.255
$\hat{\rho}^{(100)}_{h=1/256}$	0.814	0.814	0.972	1.104	1.100	1.263

Table 5.2: Two-grid convergence factors for the Q_2 approximation with ω^{**} in 1D.

Cycle $\hat{\rho}_h$	$TG(0,1)$	$TG(1,0)$	$TG(1,1)$	$TG(1,2)$	$TG(2,1)$	$TG(2,2)$
$\omega = \omega^{**} = \frac{14}{22-2\sqrt{7}} \approx 0.838, \mu^{**} = 0.796$						
$\rho_{h=1/128}$	0.526	0.526	0.495	0.372	0.372	0.302
$\hat{\rho}^{(100)}_{h=1/128}$	0.522	0.521	0.491	0.365	0.366	0.296
$\hat{\rho}^{(100)}_{h=1/256}$	0.521	0.522	0.491	0.366	0.366	0.298

5.3.2 Two-grid LFA in 1D

Two natural questions are raised by these results. First, why is the LFA smoothing factor such a bad predictor of performance? Secondly, is ω^{**} the best choice for a weight, in terms of two-grid performance? To answer these questions, we consider two-grid LFA in more details.

Definition 5.3.1. *The $2h$-harmonics, $\mathcal{F}_{2h}(\theta)$, are given by*

$$\mathcal{F}_{2h}(\theta) = \text{span}\{\varphi_h(\theta^0, x), \varphi_h(\theta^1, x)\},$$

with $\theta = \theta^0 \in T^{\text{low}} := \Theta_{2h}$, and $\theta^\alpha = \theta + \alpha\pi$, where $\alpha = 0, 1$.

To apply LFA to the two-grid operator,

$$\mathcal{M}_h^{\text{TGM}} = S_h^{\nu_2} \mathcal{M}_h^{\text{CGC}} S_h^{\nu_1}, \tag{5.10}$$

we require the representation of the CGC operator,

$$\mathcal{M}_h^{\mathrm{CGC}} = I - PA_{2h}^{-1}RA_h.$$

Inserting the representations of S_h, A_h, A_{2h}, R, P into (5.10), we obtain the Fourier representation of two-grid error-propagation operator as

$$\hat{\mathcal{M}}_h^{\mathrm{TGM}}(\theta) = \hat{S}_h^{\nu_2}(\theta)\big(I - \hat{P}(\theta)(\widetilde{A}_{2h}(2\theta))^{-1}\hat{R}(\theta)\hat{A}_h(\theta)\big)\hat{S}_h^{\nu_1}(\theta),$$

where

$$\begin{aligned} \hat{A}_h(\theta) = \mathrm{diag}\left\{\widetilde{A}_h(\theta), \widetilde{A}_h(\theta+\pi)\right\}, \qquad \hat{S}_h(\theta) = \mathrm{diag}\left\{\widetilde{\mathcal{S}}_h(\theta), \widetilde{\mathcal{S}}_h(\theta+\pi)\right\}, \\ \hat{P}_h(\theta) = \left(\widetilde{P}_h(\theta); \widetilde{P}_h(\theta+\pi)\right), \qquad \hat{R}_h(\theta) = \left(\widetilde{R}_h(\theta), \widetilde{R}_h(\theta+\pi)\right), \end{aligned}$$

and

$$\widetilde{A}_{2h}(2\theta) = \frac{1}{6h}\begin{pmatrix} 14 + 2\cos(2\theta) & -16\cos\theta \\ -16\cos\theta & 16 \end{pmatrix},$$

in which diag$\{A, B\}$ stands for the block diagonal matrix with diagonal blocks, A and B.

The symbols $\widetilde{A}_h(\theta)$ and $\widetilde{A}_h(\theta+\pi)$ are as given above, while the symbols for relaxation are

$$\widetilde{\mathcal{S}}_h(\theta) = I - \omega\widetilde{M}_h^{-1}(\theta)\widetilde{A}_h(\theta),\ \widetilde{\mathcal{S}}_h(\theta+\pi) = I - \omega\widetilde{M}_h^{-1}(\theta+\pi)\widetilde{A}_h(\theta+\pi).$$

To derive symbols for the grid-transfer operators, we first consider an arbitrary restriction operator characterized by a constant coefficient stencil $R \overset{\wedge}{=} [r_\kappa]_h^{2h}$. Then, an infinite grid function $w_h : G_h \to \mathbb{R}$ (or $\mathbb{C}$) is transferred to the coarse grid, G_{2h}, in the following way:

$$(Rw_h)(x) \;\; = \;\; \sum_{\kappa\in V} r_\kappa w_h(x + \kappa h)\ (x \in G_{2h}).$$

In our case, we have two types of grid points on the fine and coarse grids, so the restriction operator can also be decomposed based on the partitioning of DOFs associated with nodes of the mesh and cell centres.

Let $\varphi_h(\theta^\alpha, x) = e^{\iota\theta^\alpha x/h}$. We have the following equality

$$\varphi_h(\theta^\alpha, x) = e^{\iota\alpha\pi x/h}\varphi_{2h}(2\theta^0, x), \text{ for all } x \in G_{2h}. \tag{5.11}$$

Note that $\varphi_h(\theta^\alpha, x)$ coincides on $G_{2h,N}$ with the respective grid function $\varphi_{2h}(2\theta^0, x)$, since $e^{\iota\alpha\pi x/h} \equiv 1$ in (5.11), when $x = 2jh$ for $j \in \mathbb{Z}$. However, $e^{\iota\alpha\pi x/h} = (-1)^\alpha$ when $x = 2(j + \frac{1}{2})h$ coincides with a point in $G_{2h,C}$.

Using this for $x \in G_{2h}$, we have

$$(R\varphi_h)(\theta^\alpha, \cdot)(x) \;=\; \sum_{\kappa\in V} r_\kappa e^{\iota(x+\kappa h)\theta^\alpha/h} = \sum_{\kappa\in V} r_\kappa e^{\iota\kappa\theta^\alpha} e^{\iota\alpha\pi x/h}\varphi_{2h}(2\theta^0, x).$$

Definition 5.3.2. *We call* $\widetilde{R}(\theta^\alpha) = \sum_{\kappa\in V} r_\kappa e^{\iota\kappa\theta^\alpha} e^{\iota\alpha\pi x/h} := \sum_{\kappa\in V} \widetilde{r}_\kappa$ *the restriction symbol of* R.

Remark 5.3.1. *If the restriction operator is defined on a collocated mesh, we have only* $G_{2h,N}$*, and* $e^{\iota\alpha\pi x/h} \equiv 1$ *in Definition 5.3.2, which coincides with the definition of the classical restriction symbol [24, Section 6.2.3].*

We consider biquadratic interpolation, and the corresponding adjoint operator for the restriction of the corrections. In stencil notation, the restriction operators are given by

$$R_N \overset{\wedge}{=} [(r_N)_\kappa]_h = \begin{bmatrix} 0 & -\frac{1}{8} & 0 & \frac{3}{8} & 1(\star) & \frac{3}{8} & 0 & -\frac{1}{8} & 0 \end{bmatrix}_h, \tag{5.12}$$

and

$$R_C \overset{\wedge}{=} [(r_C)_\kappa]_h = \begin{bmatrix} 0 & \frac{3}{4} & 1(\star) & \frac{3}{4} & 0 \end{bmatrix}_h, \tag{5.13}$$

where N, C stand for the node and centre points, respectively, and the $\star$ denotes the position (on the coarse grid) at which the discrete operator is applied. Note that these stencils include contributions from both fine-grid nodes and centers to the coarse-grid quantities. We illustrate these in Figure 5.2.

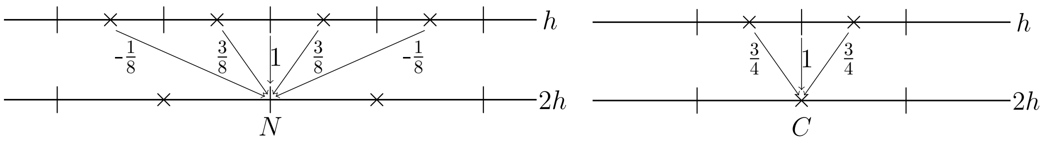

Figure 5.2: At left, R_N-restriction operator. At right, R_C-restriction operator.

As with the fine-grid matrix, both R_N and R_C require values from nodes and centres on the fine grid. We decompose R_N as $[R_N(N), R_N(C)]$ and R_C as $[R_C(N), R_C(C)]$ defined in the following

$$R_N(N) = [1],\ R_N(C) = [-\frac{1}{8} \quad \frac{3}{8} \ \star \ \frac{3}{8} \quad -\frac{1}{8}], \tag{5.14}$$

$$R_C(N) = [1],\ R_C(C) = [\frac{3}{4} \ \star \ \frac{3}{4}], \tag{5.15}$$

then apply Definition 5.3.2 to each piece separately to obtain the symbol of the restriction operator.

Theorem 5.3.1. *Define R as in (5.12) and (5.13). Then the Fourier representation of R is given by the (2×4)-matrix*

$$\begin{aligned}\hat{R}(\theta) &= \begin{pmatrix}\widetilde{R}(\theta^0) & \widetilde{R}(\theta^1)\end{pmatrix} \\ &= \begin{pmatrix}1 & \frac{3\cos(\frac{\theta}{2})-\cos(\frac{3\theta}{2})}{4} & 1 & \frac{-3\sin(\frac{\theta}{2})-\sin(\frac{3\theta}{2})}{4} \\ 1 & \frac{3\cos(\frac{\theta}{2})}{2} & -1 & \frac{3\sin(\frac{\theta}{2})}{2}\end{pmatrix}.\end{aligned}$$

Proof. Let $x \in G_{2h}$ and consider a fine-grid mode $\varphi(\theta^\alpha, y) = \beta_N\varphi_N(\theta^\alpha, y) + \beta_C\varphi_C(\theta^\alpha, y)$ for $y = x + \kappa h \in G_h$. Clearly the value of $[R\varphi(\theta^\alpha), \cdot](x)$ depends on whether x is a node on the coarse grid (and (5.12) is used) or x is a cell centre on the coarse grid (and (5.13) is used). From (5.14) and (5.15), we write the symbol for R in matrix form,

$$\widetilde{R}(\theta^\alpha) \quad = \quad \begin{pmatrix}\widetilde{R}_N(N, \theta^\alpha) & \widetilde{R}_N(C, \theta^\alpha) \\ \widetilde{R}_C(N, \theta^\alpha) & \widetilde{R}_C(C, \theta^\alpha)\end{pmatrix}, \tag{5.16}$$

acting on the vector $\begin{pmatrix}\beta_N & \beta_C\end{pmatrix}^T$, where T denotes the (non-conjugate) transpose of the row vectors.

From (5.14), (5.15), and Definition 5.3.2, we obtain the symbols

$$\begin{aligned}&\widetilde{R}_N(N, \theta^\alpha) = 1,\ \widetilde{R}_N(C, \theta^\alpha) = \frac{3}{4}\cos\big(\frac{\theta^\alpha}{2}\big) - \frac{1}{4}\cos\big(\frac{3\theta^\alpha}{2}\big), \\ &\widetilde{R}_C(N, \theta^\alpha) = (-1)^\alpha,\ \widetilde{R}_C(C, \theta^\alpha) = \frac{3}{2}\cos\big(\frac{\theta^\alpha}{2}\big)(-1)^\alpha.\end{aligned}$$

Concatenating $\hat{R}(\theta) = \begin{pmatrix}\widetilde{R}(\theta^0) & \widetilde{R}(\theta^1)\end{pmatrix}$ gives the symbol in the statement of the theorem. □

A similar calculation (see [15]) gives the symbol of biquadratic interpolation as

$$\hat{P}(\theta) = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{3\cos(\frac{\theta}{2})-\cos(\frac{3\theta}{2})}{8} & \frac{3\cos(\frac{\theta}{2})}{4} \\ \frac{1}{2} & -\frac{1}{2} \\ \frac{-3\sin(\frac{\theta}{2})-\sin(\frac{3\theta}{2})}{8} & \frac{3\sin(\frac{\theta}{2})}{4} \end{pmatrix}, \tag{5.17}$$

satisfying the usual relationship that $\hat{P}(\theta) = \frac{1}{2}(\hat{R}(\theta))^H$, where H denotes the conjugate transpose.

We again use rediscretization for the CGO, which matches the Galerkin CGO. The asymptotic two-grid convergence factor, $\rho_{\rm asp}$, is defined as

$$\rho_{\rm asp} = \sup\{\rho(\hat{\mathcal{M}}(\theta)^{\rm TGM}) : \theta \in \Theta_{2h}\}. \tag{5.18}$$

In what follows, we consider a discrete form of $\rho_{\rm asp}$, denoted by ρ_h, resulting from sampling $\rho_{\rm asp}$ over only finite set of frequencies. We consider only the case of a single relaxation; that is $\nu_1 + \nu_2 = 1$. Without loss of generality, let $\nu_1 = 1$, giving the two-grid representation as

$$\hat{\mathcal{M}}_h^{\rm TGM}(\theta) = \big(I - \hat{P}(\theta)(\widetilde{A}_{2h}(2\theta))^{-1}\hat{R}(\theta)\hat{A}_h(\theta)\big)\hat{S}_h(\theta). \tag{5.19}$$

5.3.3 A lower bound on convergence in 1D

To gain some insight and a lower bound on convergence, we consider now the limiting behavior when $\theta \to 0$. When $\theta = 0$, the two eigenvalues of

$$\widetilde{\mathcal{S}}_h(\theta+\pi) = I - \omega\widetilde{M}_h^{-1}(\theta+\pi)\widetilde{A}_h(\theta+\pi),$$

are $1-\omega, 1-\frac{6}{7}\omega$ and the eigenvector corresponding to $1-\omega$ is $v_1 = \begin{pmatrix} 0 & 1 \end{pmatrix}^T$.

From (5.17), when $\theta = 0$, we have the representation of interpolation

$$\hat{P}(0) = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{4} & \frac{3}{4} \\ \frac{1}{2} & -\frac{1}{2} \\ 0 & 0 \end{pmatrix},$$

and vector $\hat{v}_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}^T$ is not in the range of interpolation. Taken together, this tells us $\hat{v}_1$ is an eigenvector of $\hat{\mathcal{M}}_h^{\text{TGM}}(\theta)$ in the limit as $\theta \to 0$, allowing us to establish a lower bound on convergence.

Theorem 5.3.2. *For $\hat{\mathcal{M}}_h^{\text{TGM}}(\theta)$ defined as in (5.19),*

$$\text{trace}\big(\lim_{\theta\to 0} \hat{\mathcal{M}}_h^{\text{TGM}}(\theta)\big) = 2 - \frac{79}{28}\omega.$$

Proof. By standard calculation, we have

$$\lim_{\theta\to 0} \hat{\mathcal{M}}_h^{\text{TGM}}(\theta) = \begin{pmatrix} \frac{7-15\omega}{14} & \frac{-7+15\omega}{14} & \frac{-7+6\omega}{28} & 0 \\ -\frac{7-15\omega}{28} & -\frac{-7+15\omega}{28} & -\frac{-7+6\omega}{56} & 0 \\ -\frac{7-15\omega}{14} & -\frac{-7+15\omega}{14} & -\frac{-7+6\omega}{28} & 0 \\ 0 & 0 & 0 & 1-\omega \end{pmatrix}.$$

Thus, $\text{trace}\big(\lim_{\theta\to 0} \hat{\mathcal{M}}_h^{\text{TGM}}(\theta)\big) = \dfrac{7-15\omega}{14} - \dfrac{-7+15\omega}{28} - \dfrac{-7+6\omega}{28} + 1 - \omega = 2 - \dfrac{79}{28}\omega.$ □

Note that $\widetilde{P}(0)$ is full-rank, so there must be two zero eigenvalues of $\lim_{\theta\to 0} \hat{\mathcal{M}}_h^{\text{TGM}}(\theta)$. As $1-\omega$ is also an eigenvalue of $\lim_{\theta\to 0} \hat{\mathcal{M}}_h^{\text{TGM}}(\theta)$, Theorem 5.3.2 tells us that the other eigenvalue is $2 - \frac{79}{28}\omega - (1-\omega) = 1 - \frac{51}{28}\omega$. In order to minimize the spectral radius of $\lim_{\theta\to 0} \hat{\mathcal{M}}_h^{\text{TGM}}(\theta)$, we have the following result.

Lemma 5.3.1.

$$\min_{\omega} \left\{ \max\{|\lambda^*|\} : \lambda^* \in \lambda\big(\lim_{\theta\to 0} \hat{\mathcal{M}}_h^{\text{TGM}}(\theta)\big) \right\} = \frac{23}{79} \approx 0.291, \tag{5.20}$$

and only $\omega = \omega^{***} = \frac{56}{79}$ achieves the minimum.

Proof. Note that the four eigenvalues of $\lim_{\theta\to 0} \hat{\mathcal{M}}_h^{\text{TGM}}(\theta)$ are $0, 0, 1-\omega$, and $1 - \frac{51}{28}\omega$. Setting $|1-\omega| = |1 - \frac{51}{28}\omega|$, gives $\omega = \frac{56}{79}$. □

Corollary 5.3.1. *For any ω, the optimal two-grid convergence factor for a single relaxation (i.e., $\nu_1 + \nu_2 = 1$) is not less than $\frac{23}{79}$, and this factor can be achieved if and only if $\omega = \omega^{***}$.*

Corollary (5.3.1) only tells us that the two-grid convergence factor has a lower bound, but we do not know whether it can be achieved or not. We show this numerically.

For the remaining part of this paper, let μ and ρ be the LFA-predicted smoothing and two-grid convergence factors, respectively, computed with $h = \frac{1}{64}$. For ρ, we consider only one step of pre-smoothing (which gives the same results as one step of post-smoothing). We plot the predicted smoothing and convergence factors as a function of ω in 1D. The left of Figure 5.3 indicates that when the classical smoothing factor achieves its optimal value, the corresponding ω does not minimize the two-grid convergence factor. The choices of ω^* and ω^{**} in (5.8) and (5.9) both are clearly not the best choice. The left of Figure 5.3 shows that the optimal ω is $\omega^{***} = \frac{56}{79} \approx 0.709$, as proposed in Corollary 5.3.1. We explore the reasons for this below.

To see that the prediction of Lemma 5.3.1 is not a coincidence, we plot the two-grid convergence factor and $\max\left\{|1-\omega|, |1-\frac{51}{28}\omega|\right\}$ as a function of ω. Comparing the left and right of Figure 5.3 indicates that, for all ω, the two-grid convergence factor is given by $\max\left\{|1-\omega|, |1-\frac{51}{28}\omega|\right\}$.

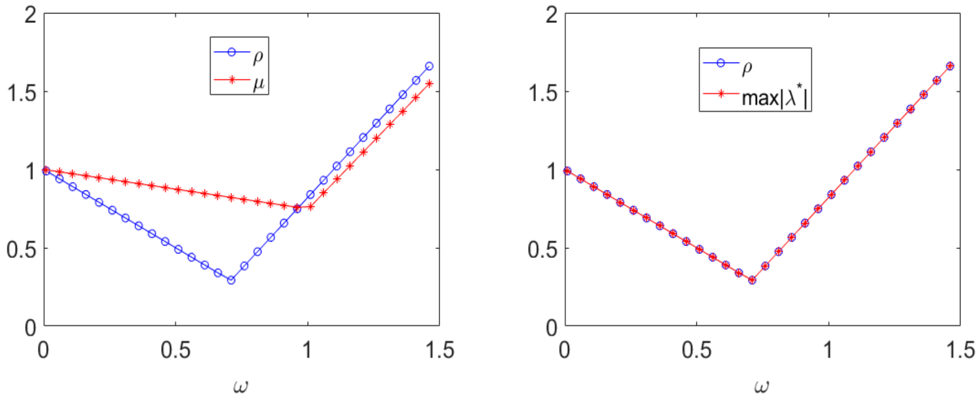

Figure 5.3: At left, LFA-predicted two-grid convergence and smoothing factors as a function of ω. At right, LFA-predicted two-grid convergence factor and $\max\{|\lambda^*|\}$ as a function of ω for the Q_2 approximation in 1D.

Two-grid and multigrid performance in 1D

Table 5.3 confirms that ω^{***} provides the best observed convergence factor, compared with the choices ω^* and ω^{**}, shown in Tables 5.1, 5.2. Table 5.3 also confirms that a single pre- or post-relaxation offers the most cost-effective cycle. Table 5.4 shows that similar convergence factors are obtained for full V-cycles.

Table 5.3: Two-grid convergence factors for the Q_2 approximation with ω^{***} in 1D.

Cycle $\hat{\rho}_h$	$TG(0,1)$	$TG(1,0)$	$TG(1,1)$	$TG(1,2)$	$TG(2,1)$	$TG(2,2)$
$\omega = \omega^{***} = \frac{56}{79} \approx 0.709, \mu = 0.822$						
$\rho_{h=1/128}$	0.291	0.291	0.249	0.090	0.090	0.064
$\hat{\rho}^{(100)}_{h=1/128}$	0.289	0.290	0.245	0.088	0.088	0.063
$\hat{\rho}^{(100)}_{h=1/256}$	0.289	0.289	0.246	0.088	0.088	0.063

Table 5.4: Multigrid convergence factors for the Q_2 approximation with ω^{***} in 1D.

Cycle $\hat{\rho}_h$	$V(0,1)$	$V(1,0)$	$V(1,1)$	$V(1,2)$	$V(2,1)$	$V(2,2)$
$\omega = \omega^{***} = \frac{56}{79} \approx 0.709, \mu = 0.822$						
$\rho_{h=1/128}$	0.291	0.291	0.249	0.090	0.090	0.064
$\hat{\rho}^{(100)}_{h=1/128}$	0.281	0.282	0.246	0.080	0.081	0.068
$\hat{\rho}^{(100)}_{h=1/256}$	0.284	0.280	0.246	0.083	0.082	0.068

5.3.4 A modified two-grid analysis

To better understand the failure of classical smoothing analysis for the Q_2 approximation, we first consider why the smoothing factor is a good predictor of performance for the Q_1 approximation. In the Q_1 case, we denote the CGC operator as $\hat{\mathcal{M}}^{\mathrm{CGC}}_{1,h}(\theta)$, and the symbol of the relaxation scheme as $\hat{S}_{1,h}(\theta)$, which are both 2×2 matrices. Here we use linear interpolation for P and $R = P^H$. By standard calculation, we have

$$\hat{\mathcal{M}}^{\mathrm{CGC}}_{1,h}(\theta) = \begin{pmatrix} \sin^2(\frac{\theta}{2}) & \cos^2(\frac{\theta}{2}) \\ \sin^2(\frac{\theta}{2}) & \cos^2(\frac{\theta}{2}) \end{pmatrix}.$$

In the standard LFA smoothing analysis, we assume an "ideal" CGC operator, $\mathcal{Q}_h$, in place of the true CGC, $\hat{\mathcal{M}}^{\mathrm{CGC}}_{1,h}(\theta)$, that annihilates the low-frequency error components and leaves the high-frequency components unchanged, see [21]. A natural choice for $\mathcal{Q}_h$ is as a projection operator,

$$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

To compute the convergence factor, we replace the CGC operator in (5.18) by $\mathcal{Q}_h$, giving

$$\sup\{\rho(\mathcal{Q}_h \hat{S}_{1,h}(\theta)) : \theta \in \Theta_{2h}\}. \tag{5.21}$$

Remark 5.3.2. *Note that (5.21) is equivalent to form (5.3).*

From the form of $\mathcal{Q}_h$ we can consider optimizing the smoothing factor by working only over the high frequencies as in Definition 5.2.3. In Figure 5.4, we plot the LFA-predicted two-grid convergence factor (5.18) and the smoothing factor as a function of ω and see that the smoothing factor perfectly captures the LFA-predicted two-grid convergence behavior.

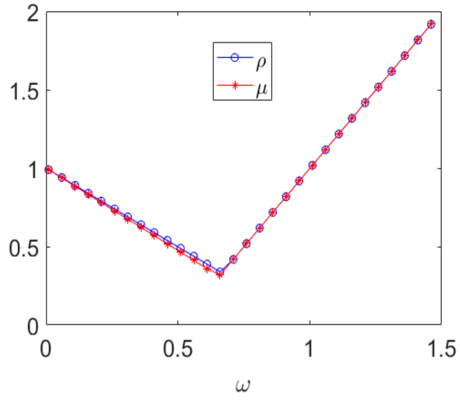

Figure 5.4: LFA-predicted two-grid convergence and smoothing factors as a function of ω for the Q_1 approximation in 1D.

However, as shown above in Subsection 5.3.1, generalizing $\mathcal{Q}_h$ to

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

does not give a good prediction of the two-grid convergence factor for the Q_2 approximation. Instead, we note that for the Q_1 case,

$$\lim_{\theta \to 0} \hat{\mathcal{M}}_{1,h}^{\text{CGC}}(\theta) = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix},$$

and, if we replace $\mathcal{Q}_h$ by this limit, then the eigenvalues of $\mathcal{Q}_h\hat{S}_{1,h}(\theta)$ do not change. This suggests that using $\lim\limits_{\theta\to 0}\hat{\mathcal{M}}_{1,h}^{\mathrm{CGC}}(\theta)$ as the ideal CGC operator may improve the robustness of the smoothing factor. We now extend this approximation for two-grid analysis of the Q_2 approximation.

Define

$$\mathcal{Q}_0 := \lim_{\theta\to 0}\big(I - \hat{P}(\theta)(\widetilde{A}_{2h}(2\theta))^{-1}\hat{R}(\theta)\hat{A}_h(\theta)\big). \tag{5.22}$$

By standard calculation,

$$\mathcal{Q}_0 = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} & -\frac{1}{4} & 0 \\ -\frac{1}{4} & \frac{1}{4} & \frac{1}{8} & 0 \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

To see how well $\mathcal{Q}_0$ works as an idealized CGC operator when predicting the two-grid convergence factor, let

$$\rho_0 = \rho_0(\omega) = \sup\{\rho(\mathcal{Q}_0\hat{S}_h(\theta)) : \theta \in \Theta_{2h}\}. \tag{5.23}$$

We plot ρ as a function of ω, compared with the LFA-predicted two-grid convergence factor ρ. Figure 5.5 shows that ρ_0 provides a much better prediction than the classical smoothing factor. Note that for smaller values of ω, ρ_0 slightly overpredicts the convergence factor, as $\mathcal{Q}_0$ captures poorly the true effects of CGC for values of θ near $\pm\frac{\pi}{2}$. We see that the optimal parameter of ρ_0 is very close to the optimal parameter for the two-grid convergence factor, ρ. Whether further improvement is possible is an open question.

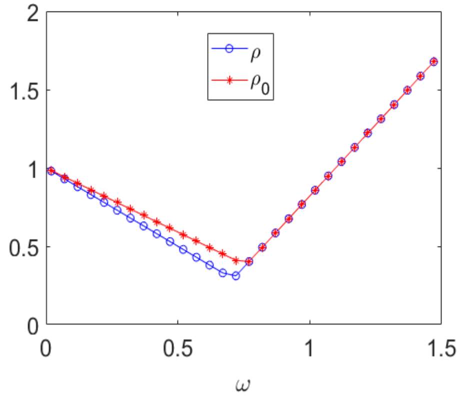

Figure 5.5: ρ and ρ_0, as a function of ω for the Q_2 approximation in 1D.

In (5.22), we compute the limit of the original CGC. Note that if we replace $\widetilde{A}_{2h}^{-1}(2\theta)$ by $(\widetilde{A}_{2h}(2\theta))^{\dagger}$, the Moore-Penrose pseudoinverse of matrix $\widetilde{A}_{2h}(2\theta)$, in (5.22), we can recover the same limit, but only indirectly. By a straightforward computation, we can consider the following operator

$$\mathcal{Q}_{\mathrm{MPP}} := \big(I - \hat{P}(0)(\widetilde{A}_{2h}(0))^{\dagger}\hat{R}(0)\hat{A}_h(0)\big). \tag{5.24}$$

By standard calculation,

$$\mathcal{Q}_{\mathrm{MPP}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{3}{4} & \frac{3}{8} & 0 \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Note that scalar multiples of $\begin{pmatrix}1 & 1 & 0 & 0\end{pmatrix}^T$ are in the null space of $\hat{R}(0) * \hat{A}_h(0)$ and, thus, $\mathcal{Q}_{\mathrm{MPP}}$ indicates that a constant error on the fine grid is not changed by this idealized CGC. To overcome this deficiency, we note that the singularity of $\widetilde{A}_{2h}(0)$ can be exploited to provide a correction in this direction of $\hat{P}(0)\begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}1 & 1 & 0 & 0\end{pmatrix}^T$. We find that

$$\mathcal{Q}_0 = \mathcal{Q}_{\mathrm{MPP}} + \mathcal{C}_2,$$

where

$$\mathcal{C}_2 = \begin{pmatrix}1 & 1 & 0 & 0\end{pmatrix}^T \begin{pmatrix}-\frac{1}{2} & -\frac{1}{2} & -\frac{1}{4} & 0\end{pmatrix}.$$

Note that the ideal CGC, $\mathcal{Q}_0$, has rank 2, while $\mathcal{Q}_{\text{MPP}}$ has rank 3. The column vector in the outer product, $\mathcal{C}_2$, naturally arises as the interpolation of a constant coarse-grid function, while the row vector is obtained as follows.

Let $v_c(\theta)$ be an eigenvector of $\widetilde{A}_{2h}(2\theta)$ such that $\widetilde{A}_{2h}(2\theta)v_c(\theta) = \lambda_c(\theta)v_c(\theta)$, and $\lim\limits_{\theta\to 0}\lambda_c(\theta) = 0$. Note that $\lim\limits_{\theta\to 0} v_c(\theta) = c\begin{pmatrix}1\\1\end{pmatrix}$ for arbitrary constant c. Now, define $\mathcal{D}$ such that

$$\lim_{\theta\to 0} v_c^T(\theta)\widetilde{A}_{2h}^{-1}(2\theta)\hat{R}(\theta)\hat{A}(\theta) = \lim_{\theta\to 0}\frac{v_c^T(\theta)\hat{R}(\theta)\hat{A}(\theta)}{\lambda_c(\theta)} = -\lim_{\theta\to 0} v_c^T(\theta)\mathcal{D}, \tag{5.25}$$

where $\mathcal{D}$ is a 2×4 matrix, with rank one. Noting that $\mathcal{D}$ is independent of θ, we find $\mathcal{D} = \begin{pmatrix}1\\1\end{pmatrix}\begin{pmatrix}-\frac{1}{2} & -\frac{1}{2} & -\frac{1}{4} & 0\end{pmatrix}$ giving the row vector in $\mathcal{C}_2$ above.

We now consider a modified two-grid error-propagation operator,

$$\hat{\mathcal{M}}^{\text{MTGM}}(\theta) := \mathcal{Q}_0\hat{S}(\theta),\;\; \theta\in\Theta_{2h},$$

which gives a good prediction for the convergence of multigrid for the Q_2 approximation. Now, we consider minimizing the spectral radius of $\hat{\mathcal{M}}^{\text{MTGM}}(\theta)$; that is, to minimize ρ_0.

By standard calculation, we have

$$\hat{S}(\theta) = \begin{pmatrix} 1-\omega(1+\frac{\cos(\theta)}{7}) & \frac{8}{7}\cos(\frac{\theta}{2})\omega & 0 & 0\\ \cos(\frac{\theta}{2})\omega & 1-\omega & 0 & 0\\ 0 & 0 & 1-\omega(1-\frac{\cos(\theta)}{7}) & -\frac{8}{7}\sin(\frac{\theta}{2})\omega\\ 0 & 0 & -\sin(\frac{\theta}{2})\omega & 1-\omega \end{pmatrix}.$$

Because $\mathcal{Q}_0$ has rank 2, $\hat{\mathcal{M}}^{\text{MTGM}}(\theta)$ has at most rank 2. By a straightforward calculation (done using a computer algebra system), the four eigenvalues of $\mathcal{Q}_0\hat{S}(\theta)$ are given by

$$\lambda(\theta) = 1 - g_{\pm}(\theta)\omega,\;\; 0,\;\; 0,$$

where $g_{\pm}(\theta)$ is

$$\frac{112+44\cos(\frac{\theta}{2})+2\cos(\theta)\pm\sqrt{2\big(1381+44(\cos(\frac{\theta}{2})+\cos(\frac{3\theta}{2}))-412\cos(\theta)+\cos(2\theta)\big)}}{112}.$$

We can check that $g_{\pm}(\theta)$ is an increasing function over $[-\frac{\pi}{2}, 0]$ and a decreasing function over $[0, \frac{\pi}{2}]$. We plot $g_{\pm}(\theta)$ as a function of θ over $[-\frac{\pi}{2}, \frac{\pi}{2}]$ in Figure 5.6.

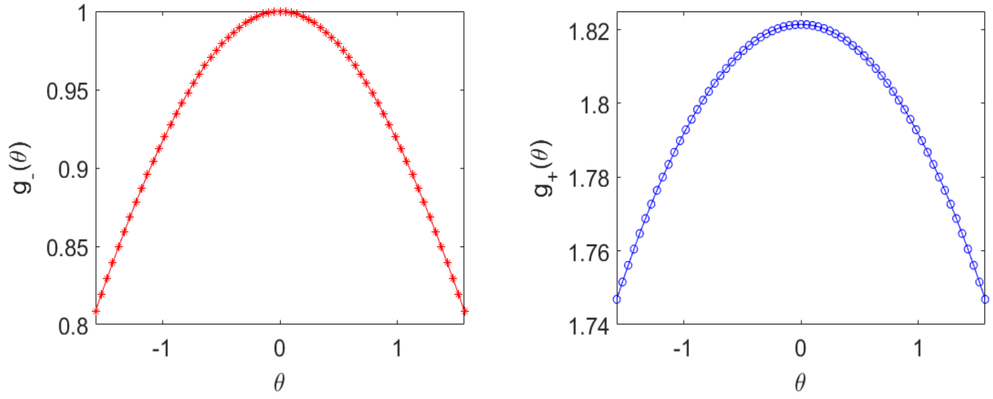

Figure 5.6: At left, $g_{-}(\theta)$ as a function of θ. At right, $g_{+}(\theta)$ as a function of θ.

The extreme values of $g_{\pm}(\theta)$ are obtained at $\theta = 0$ and $\theta = \pm\frac{\pi}{2}$; that is,

$$\begin{aligned} g_+(0) &= \frac{51}{28}, \quad g_-(0) = 1, \\ g_+(\pm\frac{\pi}{2}) &= \frac{56 + 11\sqrt{2} + \sqrt{690}}{56} < \frac{51}{28}, \\ g_-(\pm\frac{\pi}{2}) &= \frac{56 + 11\sqrt{2} - \sqrt{690}}{56} < 1. \end{aligned}$$

Thus,

$$\rho_0 = \sup\{\rho(\mathcal{Q}_0 \hat{S}_h(\theta)) : \theta \in \Theta_{2h}\} = \max\Big\{\big|1 - \frac{51}{28}\omega\big|, \big|1 - g_-(\pm\frac{\pi}{2})\omega\big|\Big\}.$$

Then, the optimal parameter minimizing ρ_0 is given by

$$\omega_{0,\text{opt}} = \frac{2}{\frac{51}{28} + \frac{56+11\sqrt{2}-\sqrt{690}}{56}} \approx 0.760,$$

and the corresponding predicted smoothing factor is

$$\rho_{0,\text{opt}} = \frac{\frac{51}{28} - \frac{56+11\sqrt{2}-\sqrt{690}}{56}}{\frac{51}{28} + \frac{56+11\sqrt{2}-\sqrt{690}}{56}} \approx 0.385.$$

Recall the optimal parameter and the true two-grid convergence factor are $\omega^{***} =$

0.709, $\rho = 0.291$, respectively. Compared with the true two-grid convergence, ρ_0 overpredicts the convergence factor based on the mode $\theta = \pm\frac{\pi}{2}$. However, this modified $\hat{\mathcal{M}}^{\text{MTGM}}(\theta)$ still offers useful information and a reasonable predictor of performance. Whether this "ideal" predictor can be used for other higher-order finite-element approximations will be explored in the following sections.

Remark 5.3.3. *Improved two-grid behavior can be achieved by considering different weights for the DOFs at nodes and those at cell centres for Jacobi relaxation; that is, putting distinct parameters in each diagonal block in the diagonal operator in (5.7). Then, the LFA shown above can be extended to this relaxation scheme to optimize the two-grid convergence factor, resulting in somewhat better convergence.*

5.4 Higher-order finite-element methods

In this section, we consider the finite-element spaces Q_p for $p = 3, 4$ and again examine the relationship between the LFA smoothing and two-grid convergence factors. In order to distinguish the block symbols for different p, we use superscripts in the matrices and block symbols in this section.

5.4.1 Cubic Lagrangian Elements

For cubic Lagrangian elements (Q_3), using nodal finite-element basis functions defined at the mesh nodes and the 1/3 and 2/3 points of the element, the elementary contributions to the stiffness matrix can be written as

$$EK_h^{(3)} = \frac{1}{40h}\begin{pmatrix} 296 & -189 & 54 & -13 \\ -189 & 432 & -297 & 54 \\ 54 & -297 & 432 & -189 \\ -13 & 54 & -189 & 296 \end{pmatrix}.$$

The corresponding symbol of stiffness operator is

$$\widetilde{A}_h^{(3)}(\theta) = \frac{1}{h}\begin{pmatrix} \frac{148-13\cos\theta}{20} & \frac{54e^{-\frac{2}{3}\iota\theta}-189e^{\frac{1}{3}\iota\theta}}{40} & \frac{54e^{\frac{2}{3}\iota\theta}-189e^{-\frac{1}{3}\iota\theta}}{40} \\ \frac{54e^{\frac{2}{3}\iota\theta}-189e^{-\frac{1}{3}\iota\theta}}{40} & \frac{54}{5} & -\frac{297e^{\frac{1}{3}\iota\theta}}{40} \\ \frac{54e^{-\frac{2}{3}\iota\theta}-189e^{\frac{1}{3}\iota\theta}}{40} & -\frac{297e^{-\frac{1}{3}\iota\theta}}{40} & \frac{54}{5} \end{pmatrix},$$

ordered as mesh nodes, then the 1/3 points and 2/3 points, respectively. The error-propagation symbol of weighted Jacobi relaxation is given by

$$\widetilde{\mathcal{S}}_h^{(3)}(\theta) = I - \omega\big(\widetilde{M}_h^{(3)}(\theta)\big)^{-1}\widetilde{A}_h^{(3)}(\theta), \tag{5.26}$$

where

$$\widetilde{M}_h^{(3)}(\theta) = \frac{1}{h}\begin{pmatrix} \frac{37}{5} & 0 & 0 \\ 0 & \frac{54}{5} & 0 \\ 0 & 0 & \frac{54}{5} \end{pmatrix}.$$

In Figure 5.7, we plot the eigenvalues of $\big(\widetilde{M}_h^{(3)}(\theta)\big)^{-1}\widetilde{A}_h^{(3)}(\theta)$. Considering the high frequencies, we see $\lambda_{\text{min,H}} = 0.085$ is obtained at $\theta = \frac{\pi}{2}$, and $\lambda_{\text{max,H}} = 2.394$ is obtained at $\theta = \pi$.

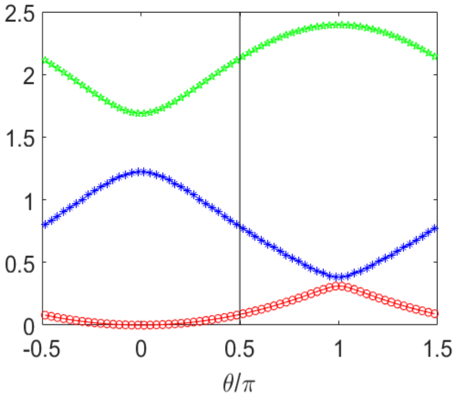

Figure 5.7: The distribution of eigenvalues of $\big(\widetilde{M}_h^{(3)}(\theta)\big)^{-1}\widetilde{A}_h^{(3)}(\theta)$ as a function of θ/π.

Thus, the classical optimal choice of ω for (5.26) is given by

$$\omega_3^* = \frac{2}{\lambda_{\text{min,H}} + \lambda_{\text{max,H}}} = 0.807,$$

and

$$\mu_3^* = \min_{\omega}\max_{\theta\in T^{\text{high}}}\big|\lambda(\widetilde{\mathcal{S}}_h^{(3)}(\omega,\theta))\big| = \frac{\lambda_{\text{max,H}} - \lambda_{\text{min,H}}}{\lambda_{\text{max,H}} + \lambda_{\text{min,H}}} \approx 0.931.$$

Denote the cubic finite-element interpolation operator as $R^{(3)}$ and the corresponding symbol as $\widetilde{R}^{(3)}$. Similarly to Theorem 5.3.1, we can write the symbol of restriction,

$R^{(3)}(\theta^\alpha)$, as

$$\widetilde{R}^{(3)}(\theta^\alpha) \;=\; \begin{pmatrix} 1-\frac{e^{\iota\theta^\alpha}}{16}-\frac{e^{-\iota\theta^\alpha}}{16} & \frac{5}{16}e^{\frac{1}{3}\iota\theta^\alpha}+\frac{1}{16}e^{-\frac{5}{3}\iota\theta^\alpha} & \frac{5}{16}e^{-\frac{1}{3}\iota\theta^\alpha}+\frac{1}{16}e^{\frac{5}{3}\iota\theta^\alpha} \\ \frac{9}{16}e^{\frac{1}{3}\iota\theta^\alpha}\beta & \frac{15}{16}e^{-\frac{1}{3}\iota\theta^\alpha}\beta & (1-\frac{5}{16}e^{\iota\theta^\alpha})\beta \\ \frac{9}{16}e^{-\frac{1}{3}\iota\theta^\alpha}\beta^2 & (1-\frac{5}{16}e^{-\iota\theta^\alpha})\beta^2 & \frac{15}{16}e^{\frac{1}{3}\iota\theta^\alpha}\beta^2 \end{pmatrix},$$

where $\beta = (e^{\frac{2}{3}\iota\pi})^\alpha$. Thus, the symbol of $R^{(3)}$ is the 3×6 matrix

$$\hat{R}^{(3)}(\theta) = \begin{pmatrix} \widetilde{R}^{(3)}(\theta^0) & \widetilde{R}^{(3)}(\theta^1) \end{pmatrix}, \text{ where } \theta = \theta^0 \in \Theta_{2h}.$$

The Fourier representation of $P^{(3)}$ is given by the 6×3 matrix,

$$\hat{P}^{(3)}(\theta) = \frac{1}{2}\big(\hat{R}^{(3)}(\theta)\big)^H.$$

We plot the smoothing factor and LFA-predicted two-grid convergence factor as a function of ω for cubic elements in 1D. Figure 5.8 indicates that when the smoothing factor achieves its optimal value, the corresponding ω does not minimize the two-grid convergence factor. From Figure 5.8, note that the optimal convergence factor, ρ, is 0.491 with $\omega = 0.650$, but the corresponding smoothing factor is 0.943, which is larger than the smoothing factor of 0.931 for $\omega_3^* = 0.807$ given above.

As the LFA smoothing factor again fails to predict the convergence factor, we extend the modification above to yield a new prediction based on $\hat{\mathcal{M}}^{\rm MTGM}(\theta)$, calculating $\mathcal{Q}_0$ again using the limit in (5.22). We plot ρ_0, compared with the true convergence factor at the right of Figure 5.8, and see that using $\mathcal{Q}_0$ accurately predicts the true convergence factor, except for a small overestimate for ω less than 0.65, as $\mathcal{Q}_0$ captures poorly the true effects of CGC for values of θ near $\pm\frac{\pi}{2}$. We observe that when $\theta = 0$, ρ_0 underestimates the true two-grid convergence factor. However, the optimal parameter of $\hat{\mathcal{M}}^{\rm MTGM}(\theta)$ is very close to the true optimal parameter for the two-grid convergence factor.

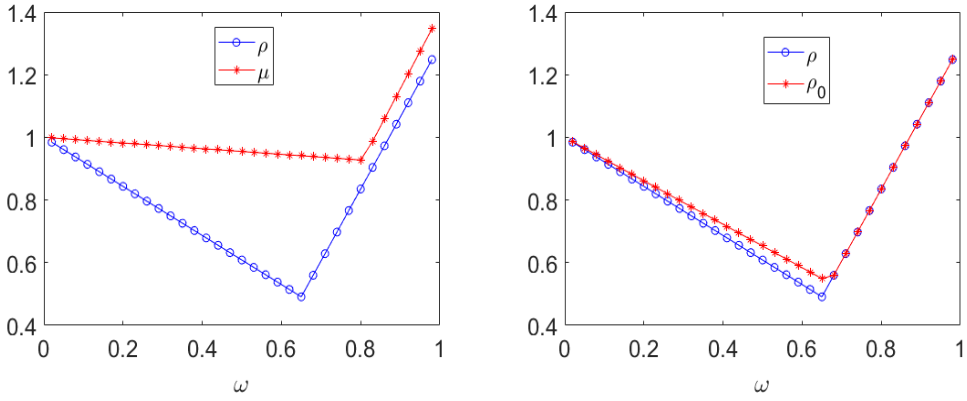

Figure 5.8: At left, the LFA-predicted two-grid convergence and smoothing factors as a function of ω. At right, ρ and ρ_0 as a function of ω for the Q_3 approximation in 1D.

5.4.2 Quartic Lagrangian Elements

For quartic Lagrangian elements (Q_4), using nodal finite-element basis functions defined at the mesh nodes and the $1/4, 1/2$, and $3/4$ points of the element, the elementary contributions to the stiffness matrix can be written as

$$EK_h^{(4)} = \frac{1}{945h}\begin{pmatrix} 9850 & -6848 & 3048 & -1472 & 347 \\ -6848 & 16640 & -14208 & 5888 & -1472 \\ 3048 & -14208 & 22320 & -14208 & 3048 \\ -1472 & 5888 & -14208 & 16640 & -6848 \\ 347 & -1472 & 3048 & -6848 & 9850 \end{pmatrix},$$

and the corresponding symbol of stiffness operator is

$$\widetilde{A}_h^{(4)}(\theta) = \frac{1}{h}\begin{pmatrix} \frac{9850+347(\eta^{-4}+\eta^{4})}{945} & -\frac{6848\eta+1472\eta^{-3}}{945} & \frac{1016\eta^{-2}+1016\eta^{2}}{315} & -\frac{6848\eta^{-1}+1472\eta^{3}}{945} \\ -\frac{6848\eta^{-1}+1472\eta^{3}}{945} & \frac{3328}{189} & -\frac{4736\eta}{315} & \frac{5888\eta^{2}}{945} \\ \frac{1016\eta^{2}+1016\eta^{-2}}{315} & -\frac{4736\eta^{-1}}{315} & \frac{496}{21} & -\frac{4736\eta}{315} \\ -\frac{6848\eta+1472\eta^{-3}}{945} & \frac{5888\eta^{-2}}{945} & -\frac{4736\eta^{-1}}{315} & \frac{3328}{189} \end{pmatrix},$$

where $\eta = e^{\frac{\iota\theta}{4}}$, with both ordered as mesh nodes, then the 1/4, 1/2, and 3/4 points of the mesh (followed by the right-hand node in $EK_h^{(4)}$).

The error-propagation symbol of weighted Jacobi relaxation is

$$\widetilde{\mathcal{S}}_h^{(4)}(\theta) = I - \omega\big(\widetilde{M}_h^{(4)}(\theta)\big)^{-1}\widetilde{A}_h^{(4)}(\theta),$$

where

$$\widetilde{M}_h^{(4)}(\theta) = \frac{1}{h}\begin{pmatrix} \frac{1970}{189} & 0 & 0 & 0 \\ 0 & \frac{3328}{189} & 0 & 0 \\ 0 & 0 & \frac{496}{21} & 0 \\ 0 & 0 & 0 & \frac{3328}{189} \end{pmatrix},$$

Using these symbols, we plot the distribution of eigenvalues of $\big(\widetilde{M}_h^{(4)}(\theta)\big)^{-1}\widetilde{A}_h^{(4)}(\theta)$ in Figure 5.9. From Figure 5.9, we see that the smallest eigenvalue over the high frequencies, $\lambda_{\text{min,H}} = 0.036$ is obtained at $\theta = \frac{\pi}{2}$ or $\frac{3\pi}{2}$. Similarly, $\lambda_{\text{max,H}} = 2.557$ is achieved with $\theta = \frac{\pi}{2}$ or $\frac{3\pi}{2}$.

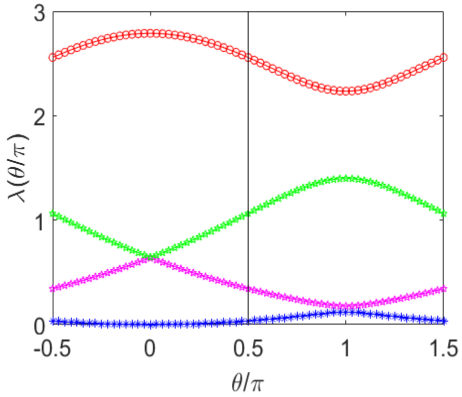

Figure 5.9: The distribution of eigenvalues of $\big(\widetilde{M}_h^{(4)}(\theta)\big)^{-1}\widetilde{A}_h^{(4)}(\theta)$ as a function of θ/π.

Thus, the optimal ω for the classical smoothing factor and the corresponding smoothing factor are

$$\omega_4^* = \frac{2}{\lambda_{\text{min,H}} + \lambda_{\text{max,H}}} = 0.772, \quad \mu_4^* = 0.973, \tag{5.27}$$

respectively.

As in the Q_2 case, the biggest eigenvalue over all frequencies is $\lambda_{\max}^* = 2.789 >$

$\lambda_{\mathrm{max,H}}$, obtained at $\theta = 0$. We, thus, consider the case of

$$\omega_4^{**} = \frac{2}{\lambda_{\mathrm{min,H}} + \lambda_{\mathrm{max}}^*} = 0.708.$$

Then, the corresponding smoothing factor is

$$\mu_4^{**} = \max_{\theta \in T^{\mathrm{high}}} \left|\lambda(\widetilde{\mathcal{S}}_h^{(4)}(\omega^{**}, \theta))\right| = \frac{\lambda_{\mathrm{max}}^* - \lambda_{\mathrm{min,H}}}{\lambda_{\mathrm{max}}^* + \lambda_{\mathrm{min,H}}} = 0.975. \tag{5.28}$$

Denote the quartic interpolation operator as $R^{(4)}$ and the corresponding symbol as $\widetilde{R}^{(4)}$. Similarly to Theorem 5.3.1, we can write the symbol of restriction, $R^{(4)}(\theta^\alpha)$, as

$$\widetilde{R}^{(4)}(\theta^\alpha) = \begin{pmatrix} 1 & \frac{35}{128}\xi + \frac{3}{128}\xi^5 - \frac{5}{128}\xi^{-7} - \frac{5}{128}\xi^{-3} & 0 & \frac{35}{128}\xi^{-1} + \frac{3}{128}\xi^{-5} - \frac{5}{128}\xi^{7} - \frac{5}{128}\xi^{3} \\ 0 & (\frac{35}{32}\xi^{-1} - \frac{5}{32}\xi^3)\gamma & \gamma & (\frac{15}{32}\xi + \frac{7}{32}\xi^5)\gamma \\ \gamma^2 & (-\frac{35}{64}\xi^{-3} + \frac{45}{64}\xi)\gamma^2 & 0 & (\frac{45}{64}\xi^{-1} - \frac{35}{64}\xi^3)\gamma^2 \\ 0 & (\frac{7}{32}\xi^{-5} + \frac{15}{32}\xi^{-1})\gamma^3 & \gamma^3 & (-\frac{5}{32}\xi^{-3} + \frac{35}{32}\xi)\gamma^3 \end{pmatrix},$$

where $\xi = e^{\frac{\iota\theta^\alpha}{4}}, \gamma = (e^{\frac{1}{2}\iota\pi})^\alpha$. Thus, the symbol of $R^{(4)}$ is the 4 × 8 matrix

$$\hat{R}^{(4)}(\theta) = \begin{pmatrix} \widetilde{R}^{(4)}(\theta^0) & \widetilde{R}^{(4)}(\theta^1) \end{pmatrix}, \text{ where } \theta = \theta^0 \in \Theta_{2h}.$$

The Fourier representation of $P^{(4)}$ is given by the 8 × 4 matrix,

$$\hat{P}^{(4)}(\theta) = \frac{1}{2}\big(\hat{R}^{(4)}(\theta)\big)^H.$$

We plot the LFA smoothing and two-grid convergence factors as a function of ω for this algorithm. At the left of Figure 5.10, we see that the LFA smoothing factor again fails to predict the two-grid convergence factor, and that the optimal convergence factor ρ is 0.608 with $\omega = 0.640$. The choices of ω in (5.27) and (5.28) both fail.

We present the results of the modified prediction using $\hat{\mathcal{M}}^{\mathrm{MTGM}}(\theta)$ here again defining Q_0 following (5.22). At the right of Figure 5.10, we compare ρ_0 with ρ, as a function of the relaxation parameter, ω, seeing that ρ_0 matches well with the true convergence, except for a small overestimation for small ω, as $\mathcal{Q}_0$ captures poorly the true effects of CGC for values of θ near $\pm\frac{\pi}{2}$. We also observe that when $\theta = 0$, ρ_0 is exactly the true two-grid convergence factor, which is the same as in the case of the Q_2 approximation.

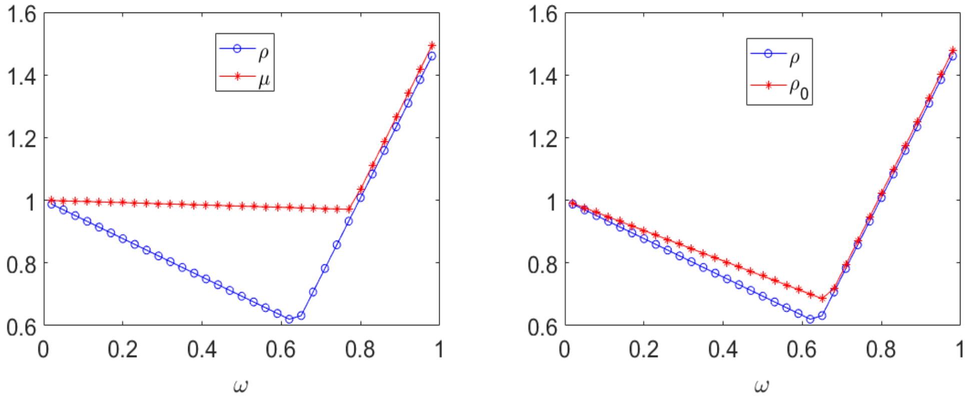

Figure 5.10: At right, LFA-predicted two-grid convergence and smoothing factors as a function of ω. At right, ρ and ρ_0 as a function of ω for the Q_4 approximation in 1D.

Remark 5.4.1. *We find that for the Q_3 and Q_4 approximations, we can again write*

$$\mathcal{Q}_0 = \mathcal{Q}_{\text{MPP}} + \mathcal{C},$$

where $\mathcal{Q}_{\text{MPP}}$ is defined using the Moore-Penrose pseudoinverse as in (5.24) and $\mathcal{C}$ is a rank-one matrix. In the Q_3 case, $\mathcal{C}$ is given as

$$\mathcal{C}_3 = \begin{pmatrix} 1 & 1 & 1 & 0 & & 0 & 0 \end{pmatrix}^T \begin{pmatrix} -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{9} & 0 & 0 \end{pmatrix},$$

and in the Q_4 case,

$$\mathcal{C}_4 = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}^T \begin{pmatrix} c_1 & c_2 \end{pmatrix},$$

where

$$\begin{aligned} c_1 &= \begin{pmatrix} -469/1536 & -53/192 & -73/512 & -53/192 \end{pmatrix}, \\ c_2 &= \begin{pmatrix} -367/3072 & -35/384(-1)^{1/4} & 0 & -35/384(-1)^{3/4} \end{pmatrix}. \end{aligned}$$

The column vector in the outer product, $\mathcal{C}_k (k = 3, 4)$, naturally arises as the interpolation of a constant coarse-grid function, while the row vector is obtained again by solving for a rank-one matrix $\mathcal{D}$ following (5.25).

5.5 LFA for the Q_2 approximation in 2D

In this section, we consider LFA for problem (5.1) in 2D, using biquadratic finite elements and the nodal basis functions defined at the mesh nodes, edge midpoints and element centres. We order the DOFs of the Q_2 approximation as nodes first, then midpoints of the edges parallel to the x-axis (the "x-edges"), followed by the midpoints of the edges parallel to the y-axis (the "y-edges"), and then the element centres. In this way, the grids in 2D are defined as

$$\boldsymbol{G_h} = G_{h_x} \bigoplus G_{h_y},$$

where

$$\boldsymbol{x} := (x, y) \in \boldsymbol{G_h} \text{ if and only if } x \in G_{h_x} \text{ and } y \in G_{h_y},$$

where G_{h_x} and G_{h_y} are defined as in 1D, see (5.4). Here, we consider $h_x = h_y = h$.

Thus, $\boldsymbol{G}_h$ can be rewritten as $\boldsymbol{G}_h = \boldsymbol{G}_h^1 \bigcup \boldsymbol{G}_h^2 \bigcup \boldsymbol{G}_h^3 \bigcup \boldsymbol{G}_h^4$ with

$$\boldsymbol{G}_h^j = \begin{cases} G_{h,N} \bigoplus G_{h,N} & \text{if} \quad j = 1, \\ G_{h,C} \bigoplus G_{h,N} & \text{if} \quad j = 2, \\ G_{h,N} \bigoplus G_{h,C} & \text{if} \quad j = 3, \\ G_{h,C} \bigoplus G_{h,C} & \text{if} \quad j = 4. \end{cases}$$

We refer to $\boldsymbol{G}_h^1, \boldsymbol{G}_h^2, \boldsymbol{G}_h^3$, and $\boldsymbol{G}_h^4$ as the NN-, CN-, NC-, and CC-type points on the grid $\boldsymbol{G}_h$, respectively.

5.5.1 Representation of the stiffness and mass operators

It is known that the stiffness and mass matrices for the Q_1 approximation in 2D can be written using tensor products of their 1D analogues. However, for the Q_2 approximation in 2D, we must carefully consider the ordering of the DOFs and the block structure of the resulting system. Assume that the stiffness and mass matrices in 1D are ordered by nodes and centres in 2×2-block matrices, given by

$$\mathcal{A}^{(2)} = \begin{pmatrix} A_{nn} & A_{nc} \\ A_{cn} & A_{cc} \end{pmatrix}, \ \mathcal{B}^{(2)} = \begin{pmatrix} B_{nn} & B_{nc} \\ B_{cn} & B_{cc} \end{pmatrix},$$

respectively. For the 2D case, we use the Tracy-Singh product to preserve block structuring in the product. Let $\mathbf{A}$ be an $(s \times t)$-block matrix, whose (i,j)-block is denoted by A_{ij}, and $\mathbf{B}$ be a $(p \times q)$-block matrix, whose (i,j)-block is denoted by B_{ij}. The Tracy-Singh product of $\mathbf{A}$ and $\mathbf{B}$ is defined by the pairwise Kronecker product for each pair of blocks in matrices $\mathbf{A}$ and $\mathbf{B}$, that is,

$$\mathbf{A}\circ\mathbf{B} = \begin{pmatrix} A_{11}\bar{\otimes}\,\mathbf{B} & \cdots & A_{1t}\bar{\otimes}\,\mathbf{B} \\ \vdots & \ddots & \vdots \\ A_{s1}\bar{\otimes}\,\mathbf{B} & \cdots & A_{st}\bar{\otimes}\,\mathbf{B} \end{pmatrix}, \text{where}\, A_{i,j}\bar{\otimes}\mathbf{B} = \begin{pmatrix} A_{ij}\otimes B_{11} & \cdots & A_{ij}\otimes B_{1q} \\ \vdots & \ddots & \vdots \\ A_{ij}\otimes B_{p1} & \cdots & A_{ij}\otimes B_{pq} \end{pmatrix},$$

where $\otimes$ is the standard Kronecker product. Then, the stiffness and mass matrices in 2D are given by

$$\mathcal{A}_2 = \mathcal{A}^{(2)} \circ \mathcal{B}^{(2)} + \mathcal{B}^{(2)} \circ \mathcal{A}^{(2)},\;\; \mathcal{B}_2 = \mathcal{B}^{(2)} \circ \mathcal{B}^{(2)},$$

respectively, and the ordering of the 4×4 block system corresponds to the indexing of the $\boldsymbol{G}_h^j$ given above. Similarly, if the biquadratic restriction matrix in 1D is given in block form as

$$\mathcal{R}^{(2)} = \begin{pmatrix} R_{nn} & R_{nc} \\ R_{cn} & R_{cc} \end{pmatrix},$$

then the corresponding restriction matrix in 2D is given by

$$\mathcal{R}_2 = \mathcal{R}^{(2)} \circ \mathcal{R}^{(2)},$$

with the same block ordering as the blocks in $\mathcal{A}_2$.

Using the Tracy-Singh product for the discretized operators allows us to compute symbols using standard Kronecker products. Given the symbols of the stiffness and mass operators for the Q_2 approximation in 1D, $\widetilde{A}_h(\theta)$ and $\widetilde{B}_h(\theta)$, respectively, the symbols of the stiffness and mass matrices in 2D are given by

$$\begin{aligned} \widetilde{A}_2(\theta_1,\theta_2) &= \widetilde{A}_h(\theta_2) \otimes \widetilde{B}_h(\theta_1) + \widetilde{B}_h(\theta_2) \otimes \widetilde{A}_h(\theta_1), \\ \widetilde{B}_2(\theta_1,\theta_2) &= \widetilde{B}_h(\theta_2) \otimes \widetilde{B}_h(\theta_1), \end{aligned}$$

respectively.

The above discussion is not limited to Q_2, and extends to Q_k as follows.

Remark 5.5.1. *The stiffness and mass matrices for the Q_k discretization in 2D can be written as*

$$\mathcal{A}_k = \mathcal{A}^{(k)} \circ \mathcal{B}^{(k)} + \mathcal{B}^{(k)} \circ \mathcal{A}^{(k)}, \ \ \mathcal{B}_k = \mathcal{B}^{(k)} \circ \mathcal{B}^{(k)},$$

respectively, where $\mathcal{A}^{(k)}$ and $\mathcal{B}^{(k)}$ are stiffness and mass matrices for the Q_k discretization in 1D, respectively.

Remark 5.5.2. *The symbols of the stiffness and mass matrices for the Q_k discretization in 2D are as follows*

$$\begin{aligned} \widetilde{A}_k(\theta_1,\theta_2) &= \widetilde{A}_h^{(k)}(\theta_2) \otimes \widetilde{B}_h^{(k)}(\theta_1) + \widetilde{B}_h^{(k)}(\theta_2) \otimes \widetilde{A}_h^{(k)}(\theta_1), \\ \widetilde{B}_k(\theta_1,\theta_2) &= \widetilde{B}_h^{(k)}(\theta_2) \otimes \widetilde{B}_h^{(k)}(\theta_1), \end{aligned}$$

respectively, where $\widetilde{A}_h^{(k)}$ and $\widetilde{B}_h^{(k)}$ are the stiffness and mass symbols for the Q_k discretization in 1D, respectively.

Remark 5.5.3. *The restriction matrix corresponding to the Q_k approximation in 2D is given by*

$$\mathcal{R}_k = \mathcal{R}^{(k)} \circ \mathcal{R}^{(k)},$$

with the same block ordering as $\mathcal{A}_k$ if $\mathcal{R}^{(k)}$ is ordered consistently with $\mathcal{A}^{(k)}$.

5.5.2 Fourier representation of grid transfer operators

Now we turn to the representation of biquadratic interpolation and its adjoint operator, restriction, in 2D. The extension of the restriction operator given in (5.12) and (5.13) from 1D to 2D with blocks ordered as mesh nodes, x-edge midpoints, y-edge midpoints, and cell centres can be written as $\boldsymbol{R} = \{\boldsymbol{R}_{NN}, \boldsymbol{R}_{CN}, \boldsymbol{R}_{NC}, \boldsymbol{R}_{CC}\}$, respectively. Let $\widetilde{\boldsymbol{R}}_{NN}, \widetilde{\boldsymbol{R}}_{CN}, \widetilde{\boldsymbol{R}}_{NC}$, and $\widetilde{\boldsymbol{R}}_{CC}$ be their Fourier representations. We show the representation of transfer operators is given by tensor products of their symbols in 1D.

Let

$$\begin{aligned} \boldsymbol{\alpha} &= (\alpha_1,\alpha_2) \in \big\{(0,0),(1,0),(0,1),(1,1)\big\}, \\ \boldsymbol{\theta}^{\boldsymbol{\alpha}} &= (\theta_1^{\alpha_1},\theta_2^{\alpha_2}) = (\theta_1+\alpha_1\pi,\ \theta_2+\alpha_2\pi),\ \boldsymbol{\theta} := \boldsymbol{\theta}^{(\mathbf{0},\mathbf{0})}. \end{aligned}$$

We use the ordering of $\boldsymbol{\alpha} = (0,0),(1,0),(0,1),(1,1)$ for the four harmonics.

Definition 5.5.1. *Assume that $T = [t_{\kappa_1}]$ and $S = [s_{\kappa_2}]$ are two stencil operators in 1D. The 2D stencil $S \bigotimes T$ is given by*

$$S \bigotimes T := [\boldsymbol{r}_{\boldsymbol{\kappa}}]_{\boldsymbol{h}}, \text{ with } \boldsymbol{r}_{\boldsymbol{\kappa}} = t_{\kappa_1} s_{\kappa_2}, \text{ and } \boldsymbol{\kappa} = (\kappa_1, \kappa_2),$$

so that R is the outer product of S and T.

We use this outer-product notation to simplify the computation of the symbol of the restriction operator in block form. Rewrite (5.12) and (5.13) as

$$R_N = \begin{bmatrix} -\frac{1}{8} & 0 & \frac{3}{8} & 1(\star) & \frac{3}{8} & 0 & -\frac{1}{8} \end{bmatrix}, \tag{5.29}$$

and

$$R_C = \begin{bmatrix} \frac{3}{4} & 1(\star) & \frac{3}{4} \end{bmatrix}, \tag{5.30}$$

respectively, by discarding the points outside the stencil of restriction. Then, the four restriction stencils in 2D for the Q_2 approximation can be denoted by

$$\boldsymbol{R}_{I_x I_y} = R_{I_y} \bigotimes R_{I_x} := [\boldsymbol{r}_{\boldsymbol{\kappa}}]_{I_x I_y}, \tag{5.31}$$

where $I_x, I_y \in \{N, C\}$.

We can extend Definition 5.3.2 to a "standard" restriction operator in 2D as follows.

Definition 5.5.2. *Let $\boldsymbol{T}(\boldsymbol{\theta}^{\boldsymbol{\alpha}}) = [\boldsymbol{t}_{\boldsymbol{\kappa}}]$ be a restriction stencil in 2D given as $\boldsymbol{T} = \mathcal{T}_2 \bigotimes \mathcal{T}_1$. We call*

$$\widetilde{\boldsymbol{T}}(\boldsymbol{\theta}^{\boldsymbol{\alpha}}) = \sum_{\boldsymbol{\kappa} \in \boldsymbol{V}} \boldsymbol{t}_{\boldsymbol{\kappa}} e^{\iota \boldsymbol{\kappa} \cdot \boldsymbol{\theta}^{\boldsymbol{\alpha}}} e^{\iota \pi \boldsymbol{\alpha} \cdot \boldsymbol{x}/h} := \sum_{\boldsymbol{\kappa} \in \boldsymbol{V}} \widetilde{\boldsymbol{t}}_{\boldsymbol{\kappa}} = \sum_{(\kappa_1, \kappa_2) \in \boldsymbol{V}} \widetilde{t}_{\kappa_1} \widetilde{t}_{\kappa_2}, \tag{5.32}$$

the restriction symbol of $\boldsymbol{T}$.

Here, by "standard", we mean the restriction operator is associated with only one type of meshpoint.

Remark 5.5.4. *It is easy to check that in (5.32),*

$$\widetilde{\boldsymbol{T}}(\boldsymbol{\theta}^{\boldsymbol{\alpha}}) = \sum_{(\kappa_1, \kappa_2) \in \boldsymbol{V}} \widetilde{t}_{\kappa_1} \widetilde{t}_{\kappa_2} = \sum_{\kappa_1} \sum_{\kappa_2} \widetilde{t}_{\kappa_1} \widetilde{t}_{\kappa_2} = \widetilde{\mathcal{T}}_1(\theta_1^{\alpha_1}) \widetilde{\mathcal{T}}_2(\theta_2^{\alpha_2}),$$

where $\widetilde{\mathcal{T}}_1(\theta_1^{\alpha_1})$ and $\widetilde{\mathcal{T}}_2(\theta_2^{\alpha_2})$ are the restriction symbols for $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively, due to the tensor product of $\mathcal{T}_2 \otimes \mathcal{T}_1$.

Note that $\boldsymbol{R}_{I_xI_y}$ draws values from four types of meshpoints on the fine grid. Similarly to 1D, the stencil $\boldsymbol{R}_{I_xI_y}$ can be split into 4 types of substencils, and the Fourier representation of $\boldsymbol{R}_{I_xI_y}$ can be written as a (1×4)-matrix as follows,

$$\widetilde{\boldsymbol{R}}_{I_xI_y}(\boldsymbol{\theta}^{\boldsymbol{\alpha}}) = \left(\widetilde{R}_{I_xI_y,NN}(\boldsymbol{\theta}^{\boldsymbol{\alpha}}) \quad \widetilde{R}_{I_xI_y,CN}(\boldsymbol{\theta}^{\boldsymbol{\alpha}}) \quad \widetilde{R}_{I_xI_y,NC}(\boldsymbol{\theta}^{\boldsymbol{\alpha}}) \quad \widetilde{R}_{I_xI_y,CC}(\boldsymbol{\theta}^{\boldsymbol{\alpha}}) \right). \tag{5.33}$$

The subscript J_xJ_y of $\widetilde{R}_{I_xI_y,J_xJ_y}(\boldsymbol{\theta}^{\boldsymbol{\alpha}})$ $(J_x, J_y \in \{N, C\})$ denotes the contributions of the J_xJ_y-type points on the fine grid to the I_xI_y points on the coarse grid.

Thus, we can use Definition 5.5.2 to calculate $\widetilde{R}_{I_xI_y,J_xJ_y}(\boldsymbol{\theta}^{\boldsymbol{\alpha}})$.

Theorem 5.5.1. *The entries in $\widetilde{\boldsymbol{R}}_{I_xI_y}(\boldsymbol{\theta}^{\boldsymbol{\alpha}})$ in (5.33) are given by,*

$$\widetilde{R}_{I_xI_y,J_xJ_y}(\boldsymbol{\theta}^{\boldsymbol{\alpha}}) = \widetilde{R}_{I_y}(J_y, \theta_2^{\alpha_2})\widetilde{R}_{I_x}(J_x, \theta_1^{\alpha_1}) \tag{5.34}$$

where $I_x, I_y, J_x, J_y \in \{N, C\}$. Note that the notation for the right-hand side of (5.34) is defined in the proof of Theorem 5.3.1.

Proof. Consider a 2D Fourier mode with frequency with $\boldsymbol{\theta}^{\boldsymbol{\alpha}}$, restricted to the coarse grid by the tensor product restriction operators given in (5.31). Because $\boldsymbol{R}_{I_xI_y} = R_{I_y} \bigotimes R_{I_x}$, $\boldsymbol{R}_{I_xI_y}$ can be split into four substencils $R_{I_xI_y,J_xJ_y}$, where $J_x, J_y \in \{N, C\}$, with corresponding symbol $\widetilde{R}_{I_xI_y,J_xJ_y}$. Since the tensor product preserves the stencil structure, $R_{I_xI_y,J_xJ_y} = R_{I_y}(J_y) \otimes R_{I_x}(J_x)$, where $R_{I_y}(J_y)$ stands for the substencil of R_{I_y} corresponding to the contributions from J_y-type points on the find grid, see (5.14) and (5.15). Thus, $\widetilde{R}_{I_xI_y,J_xJ_y}$ can be calculated based on Definition 5.5.2. According to Remark 5.5.4, $\widetilde{R}_{I_xI_y,J_xJ_y} = \widetilde{R}_{I_x}(J_x, \theta_1^{\alpha_1})\widetilde{R}_{I_y}(J_y, \theta_2^{\alpha_2})$. □

Corollary 5.5.1. *The symbol of restriction in 2D can be written as a tensor product of the restriction symbols in 1D, that is, $\widetilde{\boldsymbol{R}}(\boldsymbol{\theta}^{\boldsymbol{\alpha}})$ is the 4×4-matrix given by*

$$\widetilde{\boldsymbol{R}}(\boldsymbol{\theta}^{\boldsymbol{\alpha}}) = \widetilde{R}(\theta_2^{\alpha_2}) \otimes \widetilde{R}(\theta_1^{\alpha_1}),$$

ordered as mesh nodes, x-edge midpoints, y-edge midpoints, and cell centres.

Furthermore, the Fourier representation of $\boldsymbol{R}$ *is given by the* (1×4)*-block-matrix*

$$\hat{\boldsymbol{R}}(\boldsymbol{\theta}) = \left(\widetilde{\boldsymbol{R}}(\boldsymbol{\theta}^{(0,0)}) \quad \widetilde{\boldsymbol{R}}(\boldsymbol{\theta}^{(1,0)}) \quad \widetilde{\boldsymbol{R}}(\boldsymbol{\theta}^{(0,1)}) \quad \widetilde{\boldsymbol{R}}(\boldsymbol{\theta}^{(1,1)})\right).$$

The Fourier representation of $\boldsymbol{P}$ is given by a (16×4)-matrix and

$$\hat{\boldsymbol{P}}(\boldsymbol{\theta}) = \frac{1}{4}\big(\hat{\boldsymbol{R}}(\boldsymbol{\theta})\big)^H.$$

This approach can be extended to Q_k or any other nodal basis for Q_2 as long as the 2D node points are given as a tensor-product of 1D meshes.

Corollary 5.5.2. *The restriction symbol for the* Q_k *discretization in 2D can be written as a tensor product of the corresponding restriction symbols in 1D. That is,* $\widetilde{\boldsymbol{R}}^{(k)}(\boldsymbol{\theta}^{\boldsymbol{\alpha}})$ *is the* $k^2 \times k^2$*-matrix given by*

$$\widetilde{\boldsymbol{R}}^{(k)}(\boldsymbol{\theta}^{\boldsymbol{\alpha}}) = \widetilde{R}^{(k)}(\theta_2^{\alpha_2}) \otimes \widetilde{R}^{(k)}(\theta_1^{\alpha_1}),$$

ordered correspondingly to the order of $\widetilde{R}^{(k)}(\theta_1^{\alpha_1})$*. Furthermore,*

$$\hat{\boldsymbol{P}}^{(k)}(\boldsymbol{\theta}) = \frac{1}{4}\big(\hat{\boldsymbol{R}}^{(k)}(\boldsymbol{\theta})\big)^H.$$

5.5.3 A lower bound on convergence in 2D

Here, we also discuss the weighted Jacobi relaxation for the Q_2 approximation in 2D. The symbol of the two-grid error propagation operator is

$$\hat{\mathcal{M}}_h^{\mathrm{TGM}}(\boldsymbol{\theta}) = \left(I - \hat{\boldsymbol{P}}(\boldsymbol{\theta})\hat{A}_{2h}(2\boldsymbol{\theta})^{-1}\hat{\boldsymbol{R}}(\boldsymbol{\theta})\hat{\boldsymbol{A}}_{\boldsymbol{2}}(\boldsymbol{\theta})\right)\hat{\boldsymbol{S}}_{\boldsymbol{2}}(\boldsymbol{\theta}),$$

where

$$\begin{aligned}
\hat{A}_{2h}(2\boldsymbol{\theta}) &= \widetilde{A}_{2h}(2\theta_2) \otimes \widetilde{B}_{2h}(2\theta_1) + \widetilde{B}_{2h}(2\theta_2) \otimes \widetilde{A}_{2h}(2\theta_1),\\
\hat{\boldsymbol{A}}_{\mathbf{2}}(\boldsymbol{\theta}) &= \operatorname{diag}\left\{\widetilde{A}_2(\boldsymbol{\theta}^{(0,0)}), \widetilde{A}_2(\boldsymbol{\theta}^{(1,0)}), \widetilde{A}_2(\boldsymbol{\theta}^{(0,1)}), \widetilde{A}_2(\boldsymbol{\theta}^{(1,1)})\right\},\\
\hat{\boldsymbol{S}}_2(\boldsymbol{\theta}) &= \operatorname{diag}\left\{\widetilde{\mathcal{S}}(\boldsymbol{\theta}^{(0,0)}), \widetilde{\mathcal{S}}(\boldsymbol{\theta}^{(1,0)}), \widetilde{\mathcal{S}}(\boldsymbol{\theta}^{(0,1)}), \widetilde{\mathcal{S}}(\boldsymbol{\theta}^{(1,1)})\right\},\\
\hat{\boldsymbol{R}}(\boldsymbol{\theta}) &= \left(\widetilde{\boldsymbol{R}}(\boldsymbol{\theta}^{(0,0)}), \widetilde{\boldsymbol{R}}(\boldsymbol{\theta}^{(1,0)}), \widetilde{\boldsymbol{R}}(\boldsymbol{\theta}^{(0,1)}), \widetilde{\boldsymbol{R}}(\boldsymbol{\theta}^{(1,1)})\right),\\
\hat{\boldsymbol{P}}(\boldsymbol{\theta}) &= \frac{1}{4}\left(\hat{\boldsymbol{R}}(\boldsymbol{\theta})\right)^H,
\end{aligned}$$

in which

$$\widetilde{\mathcal{S}}(\boldsymbol{\theta}^{\boldsymbol{\alpha}}) = I - \omega \widetilde{M}_2^{-1} \widetilde{A}_2(\boldsymbol{\theta}^{\boldsymbol{\alpha}}), \text{ with}$$

$$\widetilde{M}_2 = \begin{pmatrix} \frac{112}{45} & 0 & 0 & 0\\ 0 & \frac{176}{45} & 0 & 0\\ 0 & 0 & \frac{176}{45} & 0\\ 0 & 0 & 0 & \frac{256}{45} \end{pmatrix}.$$

First, we take a look at the eigenvalues of $\widetilde{M}_2^{-1}\widetilde{A}_2(\boldsymbol{\theta})$. Figure 5.11 shows the eigenvalue distribution of $\widetilde{M}_2^{-1}\widetilde{A}_2(\boldsymbol{\theta})$ over $[-\frac{\pi}{2}, \frac{3\pi}{2}]^2$. Note that both the smallest and the biggest eigenvalues are achieved over the low frequencies, $[-\frac{\pi}{2}, \frac{\pi}{2}]^2$. As shown in Figure 5.11 and discussed in more detail below, the standard smoothing analysis fails to predict the two-grid convergence factor in this case as well.

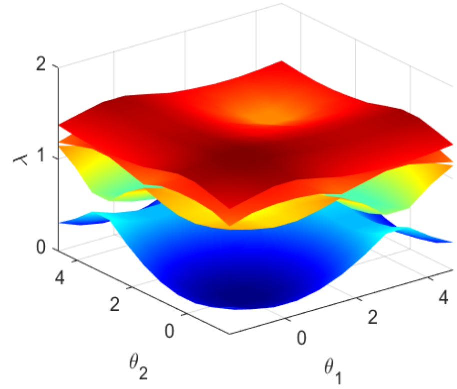

Figure 5.11: The distribution of eigenvalues, λ, of $\widetilde{M}_2^{-1}\widetilde{A}_2(\boldsymbol{\theta})$ as a function of $\boldsymbol{\theta} = (\theta_1, \theta_2)$.

Motivated by the analysis in Subsection 5.3.3, we consider the limiting behavior of $\hat{\mathcal{M}}_h^{\mathrm{TGM}}(\boldsymbol{\theta})$ when $\theta \to 0$. We first look at the range of the restriction operator when $\boldsymbol{\theta} = (0,0)$. From Corollary 5.5.1, we can calculate $\hat{\boldsymbol{R}}(\mathbf{0})$, given by

$$\widetilde{\boldsymbol{R}}(0,0) = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} \\ 1 & \frac{3}{2} & \frac{1}{2} & \frac{3}{4} \\ 1 & \frac{1}{2} & \frac{3}{2} & \frac{3}{4} \\ 1 & \frac{3}{2} & \frac{3}{2} & \frac{9}{4} \end{pmatrix}, \quad \widetilde{\boldsymbol{R}}(\pi,0) = \begin{pmatrix} 1 & 0 & \frac{1}{2} & 0 \\ -1 & 0 & -\frac{1}{2} & 0 \\ 1 & 0 & \frac{3}{2} & 0 \\ -1 & 0 & -\frac{3}{2} & 0 \end{pmatrix},$$

$$\widetilde{\boldsymbol{R}}(0,\pi) = \begin{pmatrix} 1 & \frac{1}{2} & 0 & 0 \\ 1 & \frac{3}{2} & 0 & 0 \\ -1 & -\frac{1}{2} & 0 & 0 \\ -1 & -\frac{3}{2} & 0 & 0 \end{pmatrix}, \quad \widetilde{\boldsymbol{R}}(\pi,\pi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Note that the dimensions of the null spaces of $\widetilde{\boldsymbol{R}}(\pi,0)$, $\widetilde{\boldsymbol{R}}(0,\pi)$ and $\widetilde{\boldsymbol{R}}(\pi,\pi)$ are 2, 2, and 3, respectively. Because $\hat{\boldsymbol{P}}(\mathbf{0}) = \frac{1}{4}\hat{\boldsymbol{R}}(\mathbf{0})^H$, we can easily identify seven vectors that are not treated by coarse-grid correction, and provide a lower bound on the two-grid convergence behavior.

To find the seven vectors (and the associated eigenvalues of $\lim\limits_{\boldsymbol{\theta}\to\mathbf{0}} \hat{\mathcal{M}}_h^{\mathrm{TGM}}(\boldsymbol{\theta})$), we consider the high frequencies corresponding to $(\theta_1^0, \theta_2^0) = (0,0)$. Let $T_2 = \widetilde{M_2}^{-1}\widetilde{A}_2(\pi,0)$, $T_3 = \widetilde{M_2}^{-1}\widetilde{A}_2(0,\pi)$, and $T_4 = \widetilde{M_2}^{-1}\widetilde{A}_2(\pi,\pi)$. By standard calculation, we have

$$T_2 = \begin{pmatrix} \frac{29}{28} & 0 & -\frac{1}{2} & 0 \\ 0 & 1 & 0 & -\frac{6}{11} \\ -\frac{7}{22} & 0 & 1 & 0 \\ 0 & -\frac{3}{8} & 0 & 1 \end{pmatrix}, \quad T_3 = \begin{pmatrix} \frac{29}{28} & -\frac{1}{2} & 0 & 0 \\ -\frac{7}{22} & 1 & 0 & 0 \\ 0 & 0 & 1 & -\frac{6}{11} \\ 0 & 0 & -\frac{3}{8} & 1 \end{pmatrix}, \quad T_4 = \begin{pmatrix} \frac{15}{14} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Standard calculation shows that T_2 has two eigenvalues, $\hat{\lambda}_{1,2} = 1 \pm \sqrt{\frac{9}{44}}$, with the corresponding eigenvectors $x_{1,2} = \begin{pmatrix} 0 & 1 & 0 & \pm\sqrt{\frac{11}{16}} \end{pmatrix}$, which are the in the null space of $\widetilde{\boldsymbol{R}}(\pi,0)^H$. Denote $\hat{x}_{1,2} = \begin{pmatrix} \mathrm{z} & x_{1,2} & \mathrm{z} & \mathrm{z} \end{pmatrix}^T$, where z stands for a zero vector with size 1×4. Similarly, it is easy to check that $\hat{\lambda}_{3,4} = 1 \pm \sqrt{\frac{9}{44}}$ are the two eigenvalues of T_3 corresponding to eigenvectors $x_{3,4} = \begin{pmatrix} 0 & 0 & 1 & \pm\sqrt{\frac{11}{16}} \end{pmatrix}$. Denote $\hat{x}_{3,4} = \begin{pmatrix} \mathrm{z} & \mathrm{z} & x_{3,4} & \mathrm{z} \end{pmatrix}^T$.

Finally, the structure of T_3 tells us that it has three eigenvalues: $\hat{\lambda}_{5,6,7} = 1$ and the corresponding eigenvectors are $x_5 = \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix}$, $x_6 = \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}$, $x_7 = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}$, which are in the null space of $\widetilde{\boldsymbol{R}}(\pi,\pi)^H$. Denote $\hat{x}_5 = \begin{pmatrix} \mathrm{z} & \mathrm{z} & \mathrm{z} & x_5 \end{pmatrix}^T$, $\hat{x}_6 = \begin{pmatrix} \mathrm{z} & \mathrm{z} & \mathrm{z} & x_6 \end{pmatrix}^T$, $\hat{x}_7 = \begin{pmatrix} \mathrm{z} & \mathrm{z} & \mathrm{z} & x_7 \end{pmatrix}^T$.

The above discussion gives seven eigenvalues of the two-grid operator $\lim\limits_{\boldsymbol{\theta}\to\mathbf{0}} \hat{\mathcal{M}}_h^{\mathrm{TGM}}(\boldsymbol{\theta})$, leading to the following results.

Lemma 5.5.1.

$$\min_{\omega}\left\{ \max\left\{|\lambda^{**}|\right\} : \lambda^{**} = 1 - \omega\hat{\lambda}_j, 1 \le j \le 7\right\} = \sqrt{\frac{9}{44}} \approx 0.453, \tag{5.35}$$

and only $\omega = \omega_2^* = 1$ achieves the minimum.

Proof. Since the smallest and largest values of $\hat{\lambda}_j (j = 1, 2, \cdots, 7)$ are $1 - \sqrt{\frac{9}{44}}$ and $1 + \sqrt{\frac{9}{44}}$, respectively, the optimal ω for (5.35) is $\omega_2^* = \frac{2}{1+\sqrt{\frac{9}{44}}+1-\sqrt{\frac{9}{44}}} = 1$. It follows $1 - \omega_2^*\left(1 - \sqrt{\frac{9}{44}}\right) = \sqrt{\frac{9}{44}}$. □

Corollary 5.5.3. *For any ω, the optimal convergence factor for the two-grid algorithm using a single weighted Jacobi relaxation (i.e., $\nu_1 + \nu_2 = 1$) on the Q_2 discretization in 2D, is not less than $\sqrt{\frac{9}{44}}$, and this factor can be achieved if and only if $\omega = \omega_2^*$.*

Two-grid and multigrid performance in 2D

In order to see how the parameter ω_2^* performs in practice in a multigrid method, we present two-grid and multigrid results. Table 5.5 shows that ω_2^* achieves the best possible results, with measured multigrid convergence factors that coincide with the LFA-predicted convergence factors. The same convergence factor is also obtained using full V-cycles, shown in Table 5.6.

Table 5.5: Two-grid convergence factors for the Q_2 approximation in 2D.

Cycle $\hat{\rho}_h$	$TG(0,1)$	$TG(1,0)$	$TG(1,1)$	$TG(1,2)$	$TG(2,1)$	$TG(2,2)$
$\omega = \omega_2^* = 1.000, \mu = 0.842$						
$\rho_{h=1/128}$	0.452	0.452	0.288	0.123	0.123	0.091
$\hat{\rho}^{(100)}_{h=1/128}$	0.442	0.442	0.280	0.119	0.119	0.088
$\hat{\rho}^{(100)}_{h=1/256}$	0.442	0.442	0.280	0.119	0.119	0.088

Table 5.6: Multigrid convergence factors for the Q_2 approximation in 2D.

Cycle $\hat{\rho}_h$	$V(0,1)$	$V(1,0)$	$V(1,1)$	$V(1,2)$	$V(2,1)$	$V(2,2)$
$\omega = \omega_2^* = 1.000, \mu = 0.842$						
$\rho_{h=1/128}$	0.452	0.452	0.288	0.123	0.123	0.091
$\hat{\rho}^{(100)}_{h=1/128}$	0.442	0.442	0.280	0.117	0.117	0.097
$\hat{\rho}^{(100)}_{h=1/256}$	0.442	0.442	0.281	0.116	0.117	0.097

5.5.4 A modified two-grid analysis for the Q_2 approximation in 2D

Considering the classical LFA smoothing and convergence factors, Figure 5.12 indicates that the optimal ω minimizing the two-grid convergence factor is 1, and that the LFA smoothing factor fails to predict the two-grid convergence factor for the Q_2 finite-element approximation in 2D.

In contrast, we plot the LFA-predicted two-grid convergence factor and $\max\{|\lambda^{**}|\}$ as defined in (5.35) as a function of ω, at the left of Figure 5.13. This shows that for all ω, the two-grid convergence factor is given by $\max\{|\lambda^{**}|\}$, and that convergence is dominated by the harmonic space associated with $\boldsymbol{\theta} = (0,0)$.

The modified prediction given by defining $\mathcal{Q}_0$ using the limit in (5.22) and ρ_0 as in (5.23) can also be extended to this case. We plot ρ_0, compared with the true convergence factor at the right of Figure 5.13. We see that ρ_0 again overpredicts the convergence factor, as $\mathcal{Q}_0$ captures poorly the true effects of CGC for values of (θ_1, θ_2)

near $(\pm\frac{\pi}{2}, \pm\frac{\pi}{2})$. However, ρ_0 still offers a reasonable prediction of convergence and of the optimal relaxation parameter.

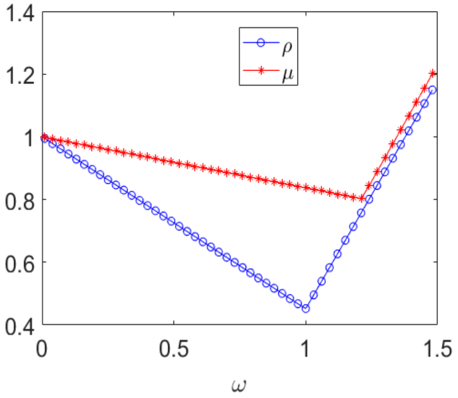

Figure 5.12: LFA-predicted two-grid convergence and smoothing factors as a function of ω for the Q_2 approximation in 2D.

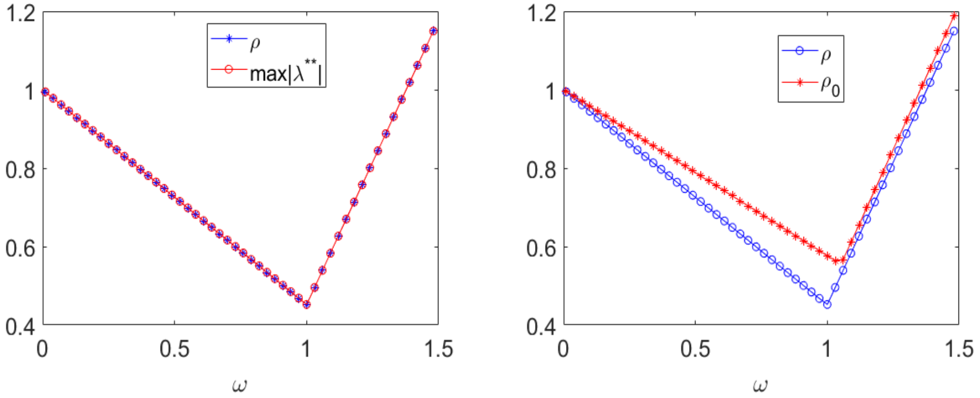

Figure 5.13: At left, LFA-predicted two-grid convergence factor and $\max\{|\lambda^{**}|\}$ as a function of ω. At right, LFA-predicted two-grid convergence factor and ρ_0, for the Q_2 approximation in 2D.

Remark 5.5.5. *For the Q_2 approximation in 1D, we see a improvement on two-grid behavior by considering different weights for the DOFs at nodes and those at cell centres for Jacobi relaxation. However, using different weights for DOFs at nodes, x-edges, y-edges, and element centres for the Q_2 approximation in 2D does not offer a better two-grid convergence factor.*

Remark 5.5.6. *We note that, in both 1D and 2D, the two-grid convergence factor gets worse with increasing polynomial degree of the finite-element approximation. This* *has been observed before in the literature [4, 9], and is commonly resolved by increasing the work done in the relaxation as the polynomial order is increased.*

As before, we can also relate $\mathcal{Q}_0$ to the Moore-Penrose pseudoinverese considered in (5.24).

Remark 5.5.7. *We find that*

$$\mathcal{Q}_0 = \mathcal{Q}_{\mathrm{MPP}} + \mathcal{C},$$

where $\mathcal{C}$ is a rank-one matrix given by

$$\mathcal{C} = \begin{pmatrix} e_1 & e_2 & e_2 & e_2 \end{pmatrix}^T \begin{pmatrix} d_1 & d_2 & d_3 & d_4 \end{pmatrix},$$

in which $e_1 = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$, $e_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}$, *and*

$$\begin{aligned} d_1 &= \begin{pmatrix} -11/48 & -13/48 & -13/48 & -11/48 \end{pmatrix}, d_2 = \begin{pmatrix} -11/96 & 0 & -13/96 & 0 \end{pmatrix}, \\ d_3 &= \begin{pmatrix} -11/96 & -13/96 & 0 & 0 \end{pmatrix}, d_4 = \begin{pmatrix} -5/64 & 0 & 0 & 0 \end{pmatrix}. \end{aligned}$$

The column vector in the outer product, $\mathcal{C}$, naturally arises as the interpolation of a constant coarse-grid function, while the row vector is obtained again by solving for the rank-one operator $\mathcal{D}$ following (5.25).

5.6 Conclusions

In this paper, we apply LFA to analyse and optimize the two-grid convergence factor for multigrid methods with higher-order finite-element approximations, especially focusing on optimal parameter choice for quadratic Lagrange elements in 1D and 2D. We find that minimizing the classical LFA smoothing factor fails to accurately predict the two-grid convergence factor. Ideal CGC operators are provided to overcome this failure, and optimal parameters that minimize the two-grid convergence factor are chosen based on the LFA results. With these parameters, we see good agreement between the measured convergence factor and predicted LFA convergence factor with periodic boundary conditions. Compared with the traditional parameter choice, based on minimizing the smoothing factor, we note a big improvement in performance with the corrected parameters. This may also explain why the LFA smoothing factor cannot predict the two-grid convergence factor for higher-order finite-element approximations to other types of PDEs, such as the $Q_2 - Q_1$ approximation to the Stokes equations, which was observed in [10].

5.7 Appendix

It is clear that the above analysis can be extended to many relaxation schemes. Here, we consider a slightly generalized form of Richardson relaxation that leads to improved results.

5.7.1 Richardson relaxation

The standard Richardson relaxation is given by $\mathcal{S}_h = I - \tau A_h$. Noting that we have node- and centre-type degrees of freedom, we consider a "generalized" Richardson relaxation with differening weights on the nodes and centres. The symbol for this relaxation is given by

$$\mathcal{S}_h(\theta) = I - M_r^{-1} A_h(\theta),$$

where

$$M_r = \frac{1}{3h}\begin{pmatrix} \frac{1}{\omega_1} & 0 \\ 0 & \frac{1}{\omega_2} \end{pmatrix}.$$

Similarly to Jacobi relaxation, we first look at the limiting behavior of the two-grid error-propagation operator $\hat{\mathcal{M}}_r^{\mathrm{TGM}}(\theta)$ when θ goes to zero. By standard calculation, we have

$$\lim_{\theta\to 0} \hat{\mathcal{M}_r}^{\mathrm{TGM}}(\theta) = \begin{pmatrix} \frac{1}{2} - 8(\omega_1+\omega_2) & -\frac{1}{2} + 8(\omega_1+\omega_2) & 3\omega_1 - \frac{1}{4} & 0 \\ -\frac{1}{4} + 4(\omega_1+\omega_2) & \frac{1}{4} + 4(\omega_1+\omega_2) & -\frac{3}{2}\omega_1 + \frac{1}{8} & 0 \\ -\frac{1}{2} + 8(\omega_1+\omega_2) & \frac{1}{2} - 8(\omega_1+\omega_2) & -3\omega_1 + \frac{1}{4} & 0 \\ 0 & 0 & 0 & 1 - 16\omega_2 \end{pmatrix}.$$

Thus,

$$\mathrm{trace}\big(\lim_{\theta\to 0} \hat{\mathcal{M}}_r^{\mathrm{TGM}}(\theta)\big) = 2 - 15\omega_1 - 28\omega_2.$$

We know $\lim\limits_{\theta\to 0}\hat{\mathcal{M}}_r^{\text{TGM}}(\theta)$ has two zero eigenvalues, and that $\lambda_{r1} = 1 - 16\omega_2$. Thus, the other nonzero eigenvalue is $\lambda_{r2} = 2 - 15\omega_1 - 28\omega_2 - (1 - 16\omega_2) = 1 - 15\omega_1 - 12\omega_2$.

5.7.2 Standard Richardson relaxation

Now, we consider the special case when $\omega_1 = \omega_2$.

Lemma 5.7.1. *When $\omega_1 = \omega_2$, the Richardson relaxation has a lower bound $\rho_r = \frac{11}{43} \approx 0.256$, achieved if and only if $\omega_1 = \omega_r = \frac{2}{43}$.*

Proof. Setting $|1 - 16\omega_1| = |1 - 15\omega_1 - 12\omega_2|$, gives $\omega_1 = \frac{2}{43}$. Then, $\rho_r = 1 - 16\omega_1 = \frac{11}{43} \approx 0.256$. □

Following this, we define ρ_0 as in (5.23) and $\rho_{00} = \sup\{\rho(\mathcal{Q}_0\hat{S}_h(0))\}$, and present the LFA-predicted two-grid convergence factor as a function of ω for Richardson relaxation. The left of Figure 5.14 indicates that the ideal (CGC) prediction, ρ_0, offers a good approximation in this case. The right of Figure 5.14, shows that the convergence behavior is dominated by the harmonics at zero frequency, as measured by ρ_{00}, which offers a perfect prediction. Recall the optimal convergence factor for Jacobi relaxation is 0.291; thus, Richardson relaxation is competitive with Jacobi relaxation.

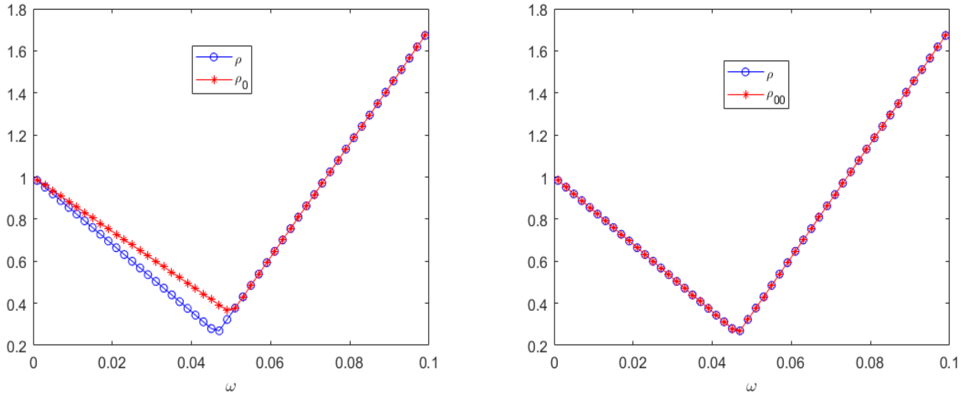

Figure 5.14: At left, LFA-predicted two-grid convergence factor and ρ_0 as a function of ω. At right, LFA-predicted two-grid convergence factor and ρ_{00} as a function of ω with Richardson relaxation for the Q_2 approximation in 1D.

In Table 5.7, we report the measured V-cycles multigrid convergence factor for Richardson relaxation with parameters $\omega_1 = \omega_2 = \omega_r$ obtained in Lemma 5.7.1.

Good agreement between the measured convergence factor with the LFA-predicted convergence factor is seen; however, little improvement occurs when adding relaxation to the V-cycle.

Table 5.7: Multigrid convergence factors for the Q_2 approximation with Richardson relaxation in 1D.

Cycle $\hat{\rho}_h$	$V(0,1)$	$V(1,0)$	$V(1,1)$	$V(1,2)$	$V(2,1)$	$V(2,2)$
$\omega_r = \frac{2}{43}, \mu = 0.825$						
$\rho_{h=1/128}$	0.256	0.256	0.233	0.066	0.066	0.058
$\hat{\rho}^{(100)}_{h=1/128}$	0.252	0.251	0.231	0.068	0.069	0.061
$\hat{\rho}^{(100)}_{h=1/256}$	0.261	0.260	0.230	0.070	0.069	0.061

5.7.3 Generalized Richardson relaxation

To potentially obtain better performance, we now consider when $\omega_1 \neq \omega_2$. To choose these weights, we seek to balance convergence for the harmonics at frequency zero with those at frequency $\frac{\pi}{2}$. By standard calculation,

$$\hat{\mathcal{M}_r}^{\text{TGM}}(\pi/2) = (\mathcal{M}_{r1}, \mathcal{M}_{r2}),$$

where

$$\mathcal{M}_{r1} = \begin{pmatrix} 3/8 - 4\omega_2 - 21/4\omega_1 & 3\sqrt{2}\omega_1 + \sqrt{2}(16\omega_2 - 1)/4 \\ 5\sqrt{2}\omega_2 + 11\sqrt{2}(14\omega_1 - 1)/32 & 5/8 - 10\omega_2 - 11/2\omega_1 \\ 21/4\omega_1 + 4\omega_2 - 3/8 & -3\sqrt{2}\omega_1 - \sqrt{2}(16\omega_2 - 1)/4 \\ -3\sqrt{2}\omega_2 - 5\sqrt{2}(14\omega_1 - 1)/32 & 5/2\omega_1 + 16\omega_2 - 3/8 \end{pmatrix}$$

and

$$\mathcal{M}_{r2} = \begin{pmatrix} 21/4\omega_1 + 4\omega_2 - 3/8 & 3\sqrt{2}\omega_1 + \sqrt{2}(16\omega_2 - 1)/4 \\ 3\sqrt{2}\omega_2 + 5\sqrt{2}(14\omega_1 - 1)/32 & 5/2\omega_1 + 6\omega_2 - 3/8 \\ 3/8 - 4\omega_2 - 21/4\omega_1 & -3\sqrt{2}\omega_1 - \sqrt{2}(16\omega_2 - 1)/4 \\ -5\sqrt{2}\omega_2 - 11\sqrt{2}(14\omega_1 - 1)/32 & 5/8 - 10\omega_2 - 11/2\omega_1 \end{pmatrix},$$

and, the four eigenvalues of $\hat{\mathcal{M}_r}^{\text{TGM}}(\pi/2)$ are

$$0,\ 0,\ \lambda_{r3} = 1 - \frac{27}{2}\omega_1 - 12\omega_2,\ \lambda_{r4} = 1 - 8\omega_1 - 16\omega_2.$$

Now, combining λ_{r3} and λ_{r4} with the two nonzero eigenvalues λ_{r1} and λ_{r2} from zero frequency, we can calculate a lower bound on the convergence factor of generalized Richardson relaxation over the modes $\theta = 0$ and $\theta = \frac{\pi}{2}$.

Lemma 5.7.2.

$$\rho_r^* := \min_{(\omega_1,\omega_2)} \Big\{ \max\{|\lambda_r|\} : \lambda_r \in \{\lambda_{ri}, i = 1,2,3,4\} \Big\} = \frac{2}{29} \approx 0.069, \tag{5.36}$$

and only $\omega_1 = \omega_{r1}^* = \frac{1}{58} \approx 0.0172$ *and* $\omega_2 = \omega_{r2}^* = \frac{27}{464} \approx 0.0582$ *achieve the minimum.*

Proof. Let $\varsigma_1 = 16\omega_2, \varsigma_2 = 15\omega_1 + 12\omega_2, \varsigma_3 = \frac{27}{2}\omega_1 + 12\omega_2$ and $\varsigma_4 = 8\omega_1 + 16\omega_2$. Thus, $\rho_r^* = \min\limits_{\omega_1,\omega_2} \big\{ \max|1 - \varsigma_i|, i = 1,2,3,4\big\}$. Note that $\varsigma_1 < \varsigma_4, \varsigma_3 < \varsigma_2$, and that $(\varsigma_2 - \varsigma_3) < (\varsigma_4 - \varsigma_1)$. The minimum is obtained if and only if

$$\varsigma_1 = \varsigma_3, \lambda_{r1} = -\lambda_{r4}. \tag{5.37}$$

From (5.37), we have $\omega_1 = \omega_{r1}^* = \frac{1}{58}$, $\omega_2 = \frac{27}{8}\omega_1 = \omega_{r2}^* = \frac{27}{464}$, and $\lambda_{r1} = \frac{2}{29}$. Under this condition, $|\lambda_{r3}| = 1 - 15\frac{1}{58} - 12\frac{27}{464} = \frac{5}{116} < \frac{2}{29}$. It follows $\rho_r^* = \frac{2}{29}$. □

Can we achieve the bound from (5.36)? The answer is yes! Using ω_{r1} and ω_{r2} in the LFA code, we see that the convergence factor is ρ_r^* over all low frequencies. Recall the optimal convergence factor, 0.291, for Jacobi relaxation, and note that $\rho_r^* \approx (0.291)^2$. Thus, generalized Richardson relaxation improves the convergence factor substantially.

We now exhibit the LFA-predicted two-grid convergence factor numerically as a function of ω_1 and ω_2, at the left of Figure 5.15. This shows the optimal convergence factor is

$$\rho_{r,\text{opt}} = 0.072,\ \text{with } \omega_1 = 0.0170, \omega_2 = 0.0585, \mu = 0.910,$$

consistent with Lemma 5.7.2, up to rounding errors. The right of Figure 5.15 presents the prediction ρ_0 as a function of ω_1 and ω_2. We find that the optimal convergence factor from this data is $\rho_{0,\text{opt}} = 0.217$ with $\omega_1 = 0.014$ and $\omega_2 = 0.0705$. Even though $\rho_{0,\text{opt}}$ overestimates the true optimal convergence factor, it still can be treated as a good prediction, particularly in contrast to the smoothing factor $\mu = 0.910$.

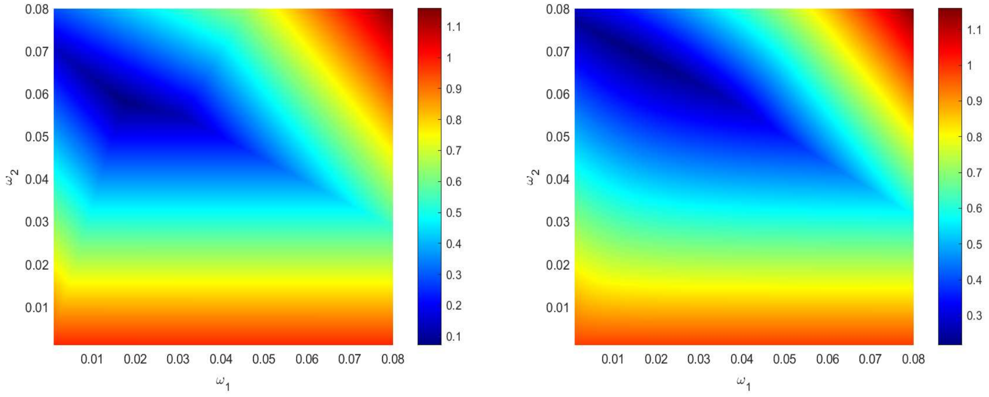

Figure 5.15: At left, ρ as a function of ω_1 and ω_2. At right, ρ_0 as a function of ω_1 and ω_2 with generalized Richardson relaxation for the Q_2 approximation in 1D.

Table 5.8 presents W-cycle performance using generalized Richardson relaxation with the parameters defined in Lemma 5.7.2. We see the measured multigrid convergence factor matches well with the LFA-predicted two-grid convergence factor, except for $\nu_1 + \nu_2 = 1$ with a slightly larger measured convergence factor. It shows that $\nu_1 + \nu_2 = 1$ is the most cost-effective cycle, compared with different numbers of pre- and postsmoothing steps.

Table 5.8: Multigrid convergence factors for the Q_2 approximation with generalized Richardson relaxation in 1D.

Cycle $\hat{\rho}_h$	$W(0,1)$	$W(1,0)$	$W(1,1)$	$W(1,2)$	$W(2,1)$	$W(2,2)$
$\omega_{r1} = \frac{1}{58}, \omega_{r2} = \frac{27}{464}, \mu = 0.909$						
$\rho_{h=1/128}$	0.069	0.069	0.189	0.120	0.120	0.106
$\hat{\rho}^{(100)}_{h=1/128}$	0.076	0.076	0.187	0.118	0.118	0.104
$\hat{\rho}^{(100)}_{h=1/256}$	0.076	0.076	0.188	0.118	0.118	0.104

Remark 5.7.1. *While the V-cycle convergence factor for the generalized Richardson relaxation does not match with the LFA-predicted convergence factor, the measured convergence factors are similar to the V-cycles results in Table 5.7.*

Remark 5.7.2. *We can also optimize the two-grid convergence factor with $(\nu_1, \nu_2) = (1, 1)$. However, the LFA-prediction shows that the optimal result of $\rho^{(\nu_1,\nu_2)}$ is larger than $(\rho_r^*)^2$. Thus, optimizing with a single relaxation is the best choice.*

Acknowledgements

The work of S.M. was partially funded by an NSERC Discovery Grant.

Bibliography

[1] T. Boonen, J. Van Lent, and S. Vandewalle. Local Fourier analysis of multigrid for the curl-curl equation. *SIAM Journal on Scientific Computing*, 30(4):1730–1755, 2008.

[2] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A multigrid tutorial*. SIAM, 2000.

[3] J. Brown. Efficient nonlinear solvers for nodal high-order finite elements in 3D. *J. Sci. Comput.*, 45(1-3):48–63, 2010.

[4] M. Donatelli, C. Garoni, C. Manni, S. Serra-Capizzano, and H. Speleers. Symbol-based multigrid methods for Galerkin B-spline isogeometric analysis. *SIAM J. Numer. Anal.*, 55(1):31–62, 2017.

[5] S. Friedhoff and S. MacLachlan. A generalized predictive analysis tool for multigrid methods. *Numerical Linear Algebra with Applications*, 22(4):618–647, 2015.

[6] S. Friedhoff, S. MacLachlan, and C. Borgers. Local Fourier analysis of space-time relaxation and multigrid schemes. *SIAM Journal on Scientific Computing*, 35(5):S250–S276, 2013.

[7] W. Hackbusch. *Multi-grid methods and applications*, volume 4. Springer Science & Business Media, 2013.

[8] P. W. Hemker, W. Hoffmann, and M. Van Raalte. Fourier two-level analysis for discontinuous Galerkin discretization with linear elements. *Numerical Linear Algebra with Applications*, 11(5-6):473–491, 2004.

[9] J. Heys, T. Manteuffel, S. McCormick, and L. Olson. Algebraic multigrid for higher-order finite elements. *Journal of Computational Physics*, 204(2):520–532, 2005.

[10] L. John, U. Rüde, B. Wohlmuth, and W. Zulehner. On the analysis of block smoothers for saddle point problems. *arXiv preprint arXiv:1612.01333*, 2016.

[11] V. John. Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier-Stokes equations. *Internat. J. Numer. Methods Fluids*, 40(6):775–798, 2002.

[12] V. John and G. Matthies. Higher-order finite element discretizations in a benchmark problem for incompressible flows. *International Journal for Numerical Methods in Fluids*, 37(8):885–903, 2001.

[13] M. Köster and S. Turek. The influence of higher order FEM discretisations on multigrid convergence. *Comput. Methods Appl. Math.*, 6(2):221–232, 2006.

[14] P. Luo, C. Rodrigo, F. J. Gaspar, and C. W. Oosterlee. On an Uzawa smoother in multigrid for poroelasticity equations. *Numer. Linear Algebra Appl.*, 24(1), 2017. e2074.

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

[16] Y. Maday and R. Muñoz. Spectral element multigrid. II. Theoretical justification. *J. Sci. Comput.*, 3(4):323–353, 1988.

[17] C. Rodrigo, P. Salinas, F. J. Gaspar, and F. J. Lisbona. Local Fourier analysis for cell-centered multigrid methods on triangular grids. *Journal of Computational and Applied Mathematics*, 259:35–47, 2014.

[18] E. M. Rønquist and A. T. Patera. Spectral element multigrid. I. Formulation and numerical results. *J. Sci. Comput.*, 2(4):389–406, 1987.

[19] K. Stüben and U. Trottenberg. Multigrid methods: Fundamental algorithms, model problem analysis and applications. *Multigrid Methods*, pages 1–176, 1982.

[20] H. Sundar, G. Stadler, and G. Biros. Comparison of multigrid algorithms for high-order continuous finite element discretizations. *Numer. Linear Algebra Appl.*, 22(4):664–680, 2015.

[21] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid.* Academic Press, Inc., San Diego, CA, 2001. With contributions by A. Brandt, P. Oswald and K. Stüben.

[22] S. Vandewalle and G. Horton. Fourier mode analysis of the multigrid waveform relaxation and time-parallel multigrid methods. *Computing*, 54(4):317–330, 1995.

[23] P. Wesseling. *An introduction to multigrid methods.* Pure and Applied Mathematics (New York). John Wiley & Sons, Ltd., Chichester, 1992.

[24] R. Wienands and W. Joppich. *Practical Fourier analysis for multigrid methods.* CRC press, 2004.

Chapter 6

Local Fourier analysis for mixed finite-element methods for the Stokes equations

Abstract

[1] We develop a local Fourier analysis of multigrid methods based on block-structured relaxation schemes for stable and stabilized mixed finite-element discretizations of the Stokes equations, to analyze their convergence behavior. Three relaxation schemes are considered: distributive, Braess-Sarazin, and Uzawa relaxation. From this analysis, parameters that minimize the local Fourier analysis smoothing factor are proposed for the stabilized methods with distributive and Braess-Sarazin relaxation. Considering the failure of the local Fourier analysis smoothing factor in predicting the true two-grid convergence factor for the stable discretization, we numerically optimize the two-grid convergence predicted by local Fourier analysis in this case. We also compare the efficiency of the presented algorithms with variants using inexact solvers. Finally, some numerical experiments are presented to validate the two-grid and multigrid convergence factors.

[1] Authors: Y. He and S. P. MacLachlan
This work is submitted as *Local Fourier analysis for mixed finite-element methods for the Stokes equations*, to the Journal of Computational and Applied Mathematics, 2018.

Keywords: Monolithic multigrid, Block-structured relaxation, local Fourier analysis, mixed finite-element methods, Stokes Equations

AMS subject classification: 65N55, 65F10, 65F08, 76M10

6.1 Introduction

In recent years, substantial research has been devoted to efficient numerical solution of the Stokes and Navier-Stokes equations, due both to their utility as models of (viscous) fluids and their commonalities with many other physical problems that lead to saddle-point systems (see, for example [14], and many of the other references cited here). In the linear (or linearized) case, solution of the resulting matrix equations is seen to be difficult, due to indefiniteness and the usual ill-conditioning of discretized PDEs. In the literature, block preconditioners (cf. [14] and the references therein) are widely used, due to their easy construction from standard multigrid algorithms for scalar elliptic PDEs, such as algebraic multigrid [30]. However, monolithic multigrid approaches [1, 3, 8, 26, 31] have been shown to outperform these preconditioners when relaxation parameters are properly chosen [2]. The focus of this work is on the analysis of such monolithic multigrid methods in the case of stable and stabilized finite-element discretizations of the Stokes equations.

Local Fourier analysis (LFA) [36, 41] has been widely used to predict the convergence behavior of multigrid methods, to help design relaxation schemes and choose algorithmic parameters. In general, the LFA smoothing factor provides a sharp prediction of actual multigrid convergence, see [36], under the assumption of an "ideal" coarse-grid correction scheme (CGC) that annihilates low-frequency error components and leaves high-frequency components unchanged. In practice, the LFA smoothing and two-grid convergence factors often exactly match the true convergence factor of multigrid applied to a problem with periodic boundary conditions [7, 34, 36]. Recently, the validity of LFA has been further analysed [29], extending this exact prediction to a wider class of problems. However, the LFA smoothing factor is also known to lose its predictivity of the true convergence in some cases [15, 19, 21]. In particular, the smoothing factor of LFA overestimates the two-grid convergence factor for the Taylor-Hood ($Q_2 - Q_1$) discretization of the Stokes equations with Vanka relaxation [21]. Even for the scalar Laplace operator, the LFA smoothing factor fails to predict the observed multigrid convergence factor for higher-order finite-element methods [19].

Two main questions interest us here. First, we look to extend the study of [21] to consider LFA of block-structured relaxation schemes for finite-element discretizations of the Stokes equations. Secondly, we consider if the LFA smoothing factor can predict the convergence factors for these relaxation schemes. Recently, LFA for multigrid based on block-structured relaxation schemes applied to the marker-and-cell (MAC) finite-difference discretization of the Stokes equations was shown to give a good prediction of convergence [18], in contrast to the results of [21]. Thus, a natural question to investigate is whether the contrasting results between [18] and [21] is due to the differences in discretization or those in the relaxation schemes considered. Here, we apply the relaxation schemes of [18] to the $Q_2 - Q_1$ discretization from [21], as well as an "intermediate" discretization using stabilized $Q_1 - Q_1$ approaches.

In recent decades, many block relaxation schemes have been studied and applied to many problems, including Braess-Sarazin-type relaxation schemes [1, 3, 5, 6, 43], Vanka-type relaxation schemes [1, 3, 21, 23, 28, 31, 37], Uzawa-type relaxation schemes [16, 17, 20, 22, 26], distributive relaxation schemes [4, 8, 27, 38, 42] and other types of methods [11, 35]. Even though LFA has been applied to distributive relaxation [25, 41], Vanka relaxation [21, 24, 28, 33], and Uzawa-type schemes [16] for the Stokes equations, most of the existing LFA has been for relaxation schemes using (symmetric) Gauss-Seidel (GS) approaches, and for simple finite-difference and finite-element discretizations. Considering modern multicore and accelerated parallel architectures, we focus on schemes based on weighted Jacobi relaxation with distributive, Braess-Sarazin, and Uzawa relaxation for common finite-element discretizations of the Stokes equations.

Some key conclusions of this analysis are as follows. First, while the LFA smoothing factor gives a good prediction of the true convergence factor for the stabilized discretizations with distributive weighted Jacobi and Braess-Sarazin relaxation, it does not for the Uzawa relaxation (in contrast to what is seen for the MAC discretization [18, 25]). For no cases, does the LFA smoothing factor offer a good prediction of the true convergence behaviour for the (stable) $Q_2 - Q_1$ discretization, suggesting that the discretization is responsible for the lack of predictivity, consistent with the results in [19, 21]. For both stable and stabilized discretizations, we see that distributive weighted Jacobi relaxation loses its high efficiency, in contrast to what is seen for the MAC scheme [18, 25]. Exact Braess-Sarazin relaxation is highly effective, with LFA-predicted $W(1,1)$ convergence factors of $\frac{1}{9}$ in the stabilized cases and $\frac{1}{4}$ in the stable case. To realize these rates with inexact cycles, however, requires nested W-cycles to solve the approximate Schur complement equation accurately enough in the stabilized case, although simple weighted Jacobi on the approximate Schur complement is observed to be sufficient in the stable case. For Uzawa-type relaxation, we see a notable gap between predicted convergence with exact inversion of the resulting Schur complement, versus inexact inversion, although some improvement is seen when replacing the approximate Schur complement with a mass matrix approximation, as is commonly used in block-diagonal preconditioners [32, 39, 40]. Overall, however, we see that Braess-Sarazin relaxation outperforms both distributive weighted Jacobi and Uzawa relaxation, for both stabilized and stable discretizations.

We organize this paper as follows. In Section 6.2, we introduce two stabilized Q_1-Q_1 and the stable $Q_2 - Q_1$ mixed finite-element discretizations of the Stokes equations in two dimensions (2D). In Section 6.3, we first review the LFA approach, then discuss the Fourier representation for these discretizations. In Section 6.4, LFA is developed for DWJ, BSR, and Uzawa-type relaxation, and optimal LFA smoothing factors are derived for the two stabilized $Q_1 - Q_1$ methods with DWJ and BSR. Multigrid performance is presented to validate the theoretical results. Section 6.5 exhibits optimized LFA two-grid convergence factors and measured multigrid convergence factors for the $Q_2 - Q_1$ discretization. Furthermore, a comparison of the cost and effectiveness of the relaxation schemes is given. Conclusions are presented in Section 6.6.

6.2 Discretizations

In this paper, we consider the Stokes equations,

$$\begin{aligned} -\Delta\vec{u} + \nabla p &= \vec{f}, \\ \nabla\cdot\vec{u} &= 0, \end{aligned} \tag{6.1}$$

where $\vec{u}$ is the velocity vector, p is the a scalar pressure of a viscous fluid, and $\vec{f}$ represents a (known) forcing term, together with suitable boundary conditions. Because of the nature of LFA, we validate our predictions against the problem with periodic boundary conditions on both $\vec{u}$ and p. Discretizations of (6.1) typically lead to a linear system of the following form:

$$Kx = \begin{pmatrix} A & B^T \\ B & -\beta C \end{pmatrix} \begin{pmatrix} \mathcal{U} \\ \mathrm{p} \end{pmatrix} = \begin{pmatrix} \mathrm{f} \\ 0 \end{pmatrix} = b, \tag{6.2}$$

where A corresponds to the discretized vector Laplacian, and B is the negative of the discrete divergence operator. If the discretization is naturally unstable, then $C \neq 0$ is the stabilization matrix, otherwise $C = 0$. In this paper, we discuss two stabilized $Q_1 - Q_1$ and the stable $Q_2 - Q_1$ finite-element discretizations.

Remark 6.2.1. *Here, we consider the vector Laplacian of the velocity in the Stokes equations, as is standard. For more general models, the divergence of the symmetric part of the gradient could be considered, affecting only the symbol of A in what follows.*

The natural finite-element approximation of Problem (6.1) is: Find $\vec{u}_h \in \mathcal{X}^h$ and $p_h \in \mathcal{H}^h$ such that

$$a(\vec{u}_h, \vec{v}_h) + b(p_h, \vec{v}_h) + b(q_h, \vec{u}_h) = g(\vec{v}_h), \text{ for all } \vec{v}_h \in \mathcal{X}_0^h \text{ and } q_h \in \mathcal{H}^h, \tag{6.3}$$

where

$$\begin{aligned} a(\vec{u}_h, \vec{v}_h) &= \int_\Omega \nabla \vec{u}_h : \nabla \vec{v}_h, \;\; b(p_h, \vec{v}_h) = -\int_\Omega p_h \nabla \cdot \vec{v}_h, \\ g(\vec{v}_h) &= \int_\Omega \vec{f}_h \cdot \vec{v}_h, \end{aligned}$$

and $\mathcal{X}^h \subset H^1(\Omega)$, $\mathcal{H}^h \subset L_2(\Omega)$ are finite-element spaces. Here, $\mathcal{X}_0^h \subset \mathcal{X}^h$ satisfies homogeneous Dirichlet boundary conditions in place of any non-homogenous essential boundary conditions on $\mathcal{X}^h$. Problem (6.3) has a unique solution only when $\mathcal{X}^h$ and $\mathcal{H}^h$ satisfy an inf-sup condition (see [9, 10, 13, 14]).

6.2.1 Stabilized $Q_1 - Q_1$ discretizations

The standard equal-order approximation of (6.3) is well-known to be unstable [10, 14]. To circumvent this, a scaled pressure Laplacian term can be added to (6.3); for a uniform mesh with square elements of size h, we subtract

$$c(p_h, q_h) = \beta h^2 (\nabla p_h, \nabla q_h),$$

for $\beta > 0$. With this, the resulting linear system is given by

$$\begin{pmatrix} A & B^T \\ B & -\beta h^2 A_p \end{pmatrix} \begin{pmatrix} \mathcal{U} \\ \mathrm{p} \end{pmatrix} = \begin{pmatrix} \mathrm{f} \\ 0 \end{pmatrix} = b,$$

where A_p is the Q_1 Laplacian operator for the pressure. Denote $S = BA^{-1}B^T$, and $S_\beta = BA^{-1}B^T + \beta C$, where $C = h^2 A_p$. From [14], the red-black unstable mode $\mathbf{p} = \pm\mathbf{1}$, can be moved from a zero eigenvalue to a unit eigenvalue (giving stability without loss of accuracy) by choosing β so that

$$\frac{\mathbf{p}^T S_\beta \mathbf{p}}{\mathbf{p}^T Q \mathbf{p}} = \beta \frac{\mathbf{p}^T C \mathbf{p}}{\mathbf{p}^T Q \mathbf{p}} = 1, \tag{6.4}$$

where Q is the mass matrix. Substituting the bilinear stiffness and mass matrices into (6.4), we find $\beta = \frac{1}{24}$. We refer to this method as the Poisson-stabilized discretization (PoSD).

An L_2 projection to stabilize the $Q_1 - Q_1$ discretization, proposed in [13], stabilizes with

$$C(p_h, q_h) = (p_h - \Pi_0 p_h, q_h - \Pi_0 q_h), \tag{6.5}$$

where Π_0 is the L_2 projection from $\mathcal{H}^h$ into the space of piecewise constant functions on the mesh. We refer to this method as the projection stabilized discretization (PrSD). The 4×4 element matrix C_4 of (6.5) is given by

$$C_4 = Q_4 - \mathbf{q}\mathbf{q}^T h^2,$$

where Q_4 is the 4×4 element mass matrix for the bilinear discretization and $\mathbf{q} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}^T$. In the projection stabilized method, we can write $C = Q - h^2 P$, where P is given by the 9-point stencil

$$P = \frac{1}{4} \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}.$$

Applying (6.4) to $C = Q - h^2 P$, we find that $\beta = 1$ is the optimal choice.

6.2.2 Stable $Q_2 - Q_1$ discretizations

In order to guarantee the well-posedness of the discrete system (6.2) with $C = 0$, the discretization of the velocity and pressure unknowns should satisfy an inf-sup condition,

$$\inf_{q_h \neq 0} \sup_{\vec{v}_h \neq \vec{0}} \frac{|b(q_h, \vec{v}_h)|}{\|\vec{v}_h\|_1 \|q_h\|_0} \geq \Gamma > 0,$$

where Γ is a constant. Taylor-Hood ($Q_2 - Q_1$) elements are well known to be stable [9, 14], where the basis functions associated with these elements are biquadratic for each component of the velocity field and bilinear for the pressure.

6.3 LFA preliminaries

6.3.1 Definitions and notations

In many cases, the LFA smoothing factor offers a good prediction of multigrid performance. Thus, we will explore the LFA smoothing factor and true (measured) multigrid convergence for the three types of relaxations considered here. We first introduce some terminology of LFA, following [36, 41]. We consider the following two-dimensional infinite uniform grids,

$$\mathbf{G}_h^j = \big\{\boldsymbol{x}^j := (x_1^j, x_2^j) = (k_1, k_2)h + \delta^j, (k_1, k_2) \in \mathbb{Z}^2\big\},$$

with

$$\delta^j = \begin{cases} (0,0) & \text{if} \quad j = 1, \\ (0, h/2) & \text{if} \quad j = 2, \\ (h/2, 0) & \text{if} \quad j = 3, \\ (h/2, h/2) & \text{if} \quad j = 4. \end{cases}$$

The coarse grids, $\mathbf{G}_{2h}^j$, are defined similarly.

Let L_h be a scalar Toeplitz operator defined by its stencil acting on $l^2(\mathbf{G}_h^j)$ as follows:

$$L_h \overset{\wedge}{=} [s_{\boldsymbol{\kappa}}]_h \ (\boldsymbol{\kappa} = (\kappa_1, \kappa_2) \in \boldsymbol{V}); \ L_h w_h(\boldsymbol{x}^j) = \sum_{\boldsymbol{\kappa} \in \boldsymbol{V}} s_{\boldsymbol{\kappa}} w_h(\boldsymbol{x}^j + \boldsymbol{\kappa} h), \qquad (6.6)$$

with constant coefficients $s_{\boldsymbol{\kappa}} \in \mathbb{R}$ (or $\mathbb{C}$), where $w_h(\boldsymbol{x}^j)$ is a function in $l^2(\mathbf{G}_h^j)$. Here, $\boldsymbol{V} \subset \mathbb{Z}^2$ is a finite index set. Because L_h is formally diagonalized by the Fourier modes $\varphi(\boldsymbol{\theta}, \boldsymbol{x}^j) = e^{i\boldsymbol{\theta}\cdot\boldsymbol{x}^j/\boldsymbol{h}} = e^{i\theta_1 x_1^j/h} e^{i\theta_2 x_2^j/h}$, where $\boldsymbol{\theta} = (\theta_1, \theta_2)$ and $i^2 = -1$, we use $\varphi(\boldsymbol{\theta}, \boldsymbol{x}^j)$ as a Fourier basis with $\boldsymbol{\theta} \in \big[-\frac{\pi}{2}, \frac{3\pi}{2}\big)^2$ (or any interval with length 2π). High and low frequencies for standard coarsening (as considered here) are given by

$$\boldsymbol{\theta} \in T^{\text{low}} = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^2, \, \boldsymbol{\theta} \in T^{\text{high}} = \left[-\frac{\pi}{2}, \frac{3\pi}{2}\right)^2 \Bigg\backslash \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^2.$$

Definition 6.3.1. *We call* $\widetilde{L}_h(\boldsymbol{\theta}) = \sum_{\boldsymbol{\kappa}\in\boldsymbol{V}} s_{\boldsymbol{\kappa}} e^{i\boldsymbol{\theta}\boldsymbol{\kappa}}$ *the symbol of* L_h.

Note that for all functions $\varphi(\boldsymbol{\theta}, \boldsymbol{x}^j)$,

$$L_h \varphi(\boldsymbol{\theta}, \boldsymbol{x}^j) = \widetilde{L}_h(\boldsymbol{\theta}) \varphi(\boldsymbol{\theta}, \boldsymbol{x}^j).$$

In what follows, we consider (3×3) linear systems of operators, which read

$$\mathcal{L}_h = \begin{pmatrix} L_h^{1,1} & L_h^{1,2} & L_h^{1,3} \\ L_h^{2,1} & L_h^{2,2} & L_h^{2,3} \\ L_h^{3,1} & L_h^{3,2} & L_h^{3,3} \end{pmatrix} = \begin{pmatrix} -\Delta_h & 0 & (\partial_x)_h \\ 0 & -\Delta_h & (\partial_y)_h \\ -(\partial_x)_h & -(\partial_y)_h & L_h^{3,3} \end{pmatrix}, \tag{6.7}$$

where $L_h^{3,3}$ depends on which discretization we use.

For the stabilized $Q_1 - Q_1$ approximations, the degrees of freedom for both velocity and pressure are only located on $\mathbf{G}_h^1$. In this setting, the $L_h^{k,\ell} (k, \ell = 1, 2, 3)$ in (6.7) are scalar Toeplitz operators. Denote $\widetilde{\mathcal{L}}_h$ as the symbol of $\mathcal{L}_h$. Each entry in $\widetilde{\mathcal{L}}_h$ is computed as the (scalar) symbol of the corresponding block of $L_h^{k,\ell}$, following Definition 6.3.1. Thus, $\widetilde{\mathcal{L}}_h$ is a 3×3 matrix. All blocks in $\mathcal{L}_h$ are diagonalized by the same transformation on a collocated mesh.

However, for the $Q_2 - Q_1$ discretization, the degrees of freedom for velocity are located on $\mathbf{G}_h = \bigcup_{j=1}^4 \mathbf{G}_h^j$, containing four types of meshpoints. The Laplace operator in (6.7) is defined by extending (6.6), with $\boldsymbol{V}$ taken to be a finite index set of values, $\boldsymbol{V} = V_N \bigcup V_X \bigcup V_Y \bigcup V_C$ with $V_N \subset \mathbb{Z}^2$, $V_X \subset \big\{(z_x + \frac{1}{2}, z_y) | (z_x, z_y) \in \mathbb{Z}^2\big\}$, $V_Y \subset \big\{(z_x, z_y + \frac{1}{2}) | (z_x, z_y) \in \mathbb{Z}^2\big\}$, and $V_C \subset \Big\{(z_x + \frac{1}{2}, z_y + \frac{1}{2}) | (z_x, z_y) \in \mathbb{Z}^2\Big\}$. With this, the (scalar) Q_2 Laplace operator is naturally treated as a block operator, and the Fourier representation of each block can be calculated based on Definition 6.3.1, with the Fourier bases adapted to account for the staggering of the mesh points. Thus, the symbols of $L_h^{1,1}$ and $L_h^{2,2}$ are 4×4 matrices. For more details of LFA for the Laplace operator using higher-order finite-element methods, refer to [19]. Similarly to the Laplace operator, both terms in the gradient, $(\partial_x)_h$ and $(\partial_y)_h$, can be treated as (4×1)-block operators. Then, the symbols of $L_h^{1,3}$ and $L_h^{2,3}$ are 4×1 matrices, calculated based on Definition 6.3.1 adapted for the mesh staggering. The symbols of $L_h^{3,1}$ and $L_h^{3,2}$ are the conjugate transposes of those of $L_h^{1,3}$ and $L_h^{2,3}$, respectively. Finally, $L_h^{3,3} = 0$. Accordingly, $\widetilde{\mathcal{L}}_h$ is a 9×9 matrix for $Q_2 - Q_1$ discretization.

Definition 6.3.2. *The error-propagation symbol, $\widetilde{\mathcal{S}}_h(\boldsymbol{\theta})$, for a block smoother $\mathcal{S}_h$ on the infinite grid $\mathbf{G}_h$ satisfies*

$$\mathcal{S}_h \varphi(\boldsymbol{\theta}, \boldsymbol{x}) = \widetilde{\mathcal{S}}_h \varphi(\boldsymbol{\theta}, \boldsymbol{x}), \ \boldsymbol{\theta} \in \Big[-\frac{\pi}{2}, \frac{3\pi}{2} \Big)^2,$$

for all $\varphi(\boldsymbol{\theta}, \boldsymbol{x})$, and the corresponding smoothing factor for $\mathcal{S}_h$ is given by

$$\mu_{\rm loc} = \mu_{\rm loc}(\mathcal{S}_h) = \max_{\boldsymbol{\theta} \in T^{\rm high}} \{ \big| \lambda(\widetilde{\mathcal{S}}_h(\boldsymbol{\theta})) \big| \},$$

where λ is an eigenvalue of $\widetilde{\mathcal{S}}_h(\boldsymbol{\theta})$.

In Definition 6.3.2, $\mathbf{G}_h = \mathbf{G}_h^1$ for the stabilized case (and $\widetilde{\mathcal{S}}_h(\boldsymbol{\theta})$ is a 3×3 matrix) and $\mathbf{G}_h = \bigcup_{j=1}^4 \mathbf{G}_h^j$ for the stable case (where $\widetilde{\mathcal{S}}_h(\boldsymbol{\theta})$ is a 9×9 matrix).

The error-propagation symbol for a relaxation scheme, represented by matrix M_h, applied to either the stabilized or stable scheme is written as

$$\widetilde{\mathcal{S}}_h(\boldsymbol{p}, \omega, \boldsymbol{\theta}) = I - \omega \widetilde{M}_h^{-1}(\boldsymbol{\theta}) \widetilde{\mathcal{L}}_h(\boldsymbol{\theta}),$$

where $\boldsymbol{p}$ represents parameters within M_h, the block approximation to $\mathcal{L}_h$, ω is an overall weighting factor, and $\widetilde{M}_h$ and $\widetilde{\mathcal{L}}_h$ are the symbols for M_h and $\mathcal{L}_h$, respectively. Note that $\mu_{\rm loc}$ is a function of some parameters in Definition 6.3.2. In this paper, we focus on minimizing $\mu_{\rm loc}$ with respect to these parameters, to obtain the optimal LFA smoothing factor.

Definition 6.3.3. *Let $\mathcal{D}$ be a bounded and closed set of allowable parameters and* *define the optimal smoothing factor over $\mathcal{D}$ as*

$$\mu_{\rm opt} = \min_{\mathcal{D}} \mu_{\rm loc}.$$

If the standard LFA assumption of an "ideal" CGC holds, then the two-grid convergence factor can be estimated by the smoothing factor, which is easy to compute. However, as expected, we will see that this idealized CGC does not lead to a good prediction for some cases we consider below. When the LFA smoothing factor fails to predict the true two-grid convergence factor, the LFA two-grid convergence factor can still be used. Thus, we give a brief introduction to the LFA two-grid convergence factor in the following.

Let

$$\begin{aligned}
\boldsymbol{\alpha} &= (\alpha_1, \alpha_2) \in \big\{(0,0),(1,0),(0,1),(1,1)\big\},\\
\boldsymbol{\theta}^{\boldsymbol{\alpha}} &= (\theta_1^{\alpha_1}, \theta_2^{\alpha_2}) = \boldsymbol{\theta} + \pi \cdot \boldsymbol{\alpha},\ \boldsymbol{\theta} := \boldsymbol{\theta}^{00} \in T^{\rm low}.
\end{aligned}$$

We use the ordering of $\boldsymbol{\alpha} = (0,0),(1,0),(0,1),(1,1)$ for the four harmonics. To apply LFA to the two-grid operator,

$$\boldsymbol{M}_h^{\rm TGM} = \mathcal{S}_h^{\nu_2} \mathcal{M}_h^{\rm CGC} \mathcal{S}_h^{\nu_1}, \tag{6.8}$$

we require the representation of the CGC operator,

$$\mathcal{M}_h^{\rm CGC} = I - P_h (\mathcal{L}_{2h}^*)^{-1} R_h \mathcal{L}_h,$$

where P_h is the multigrid interpolation operator and R_h is the restriction operator. The coarse-grid operator, $\mathcal{L}_{2h}^*$, can be either the Galerkin or rediscretization operator.

Inserting the representations of $\mathcal{S}_h, \mathcal{L}_h, \mathcal{L}_{2h}^*, P_h, R_h$ into (6.8), we obtain the Fourier representation of two-grid error-propagation operator as

$$\widetilde{\boldsymbol{M}}_h^{\rm TGM}(\boldsymbol{\theta}) = \widetilde{\boldsymbol{S}}_h^{\nu_2}(\boldsymbol{\theta}) \big(I - \widetilde{\boldsymbol{P}}_h(\boldsymbol{\theta}) (\widetilde{\mathcal{L}}_{2h}^*(2\boldsymbol{\theta}))^{-1} \widetilde{\boldsymbol{R}}_h(\boldsymbol{\theta}) \widetilde{\boldsymbol{L}}_h(\boldsymbol{\theta})\big) \widetilde{\boldsymbol{S}}_h^{\nu_1}(\boldsymbol{\theta}),$$

where

$$
\begin{aligned}
\widetilde{\boldsymbol{L}}_h(\boldsymbol{\theta}) &= \operatorname{diag}\left\{\widetilde{\mathcal{L}}_h(\boldsymbol{\theta}^{00}), \widetilde{\mathcal{L}}_h(\boldsymbol{\theta}^{10}), \widetilde{\mathcal{L}}_h(\boldsymbol{\theta}^{01}), \widetilde{\mathcal{L}}_h(\boldsymbol{\theta}^{11})\right\},\\
\widetilde{\boldsymbol{S}}_h(\boldsymbol{\theta}) &= \operatorname{diag}\left\{\widetilde{\mathcal{S}}_h(\boldsymbol{\theta}^{00}), \widetilde{\mathcal{S}}_h(\boldsymbol{\theta}^{10}), \widetilde{\mathcal{S}}_h(\boldsymbol{\theta}^{01}), \widetilde{\mathcal{S}}_h(\boldsymbol{\theta}^{11})\right\},\\
\widetilde{\boldsymbol{P}}_h(\boldsymbol{\theta}) &= \left(\widetilde{P}_h(\boldsymbol{\theta}^{00}); \widetilde{P}_h(\boldsymbol{\theta}^{10}); \widetilde{P}_h(\boldsymbol{\theta}^{01}); \widetilde{P}_h(\boldsymbol{\theta}^{11})\right),\\
\widetilde{\boldsymbol{R}}_h(\boldsymbol{\theta}) &= \left(\widetilde{R}_h(\boldsymbol{\theta}^{00}), \widetilde{R}_h(\boldsymbol{\theta}^{10}), \widetilde{R}_h(\boldsymbol{\theta}^{01}), \widetilde{R}_h(\boldsymbol{\theta}^{11})\right),
\end{aligned}
$$

in which $\operatorname{diag}\{T_1, T_2, T_3, T_4\}$ stands for the block diagonal matrix with diagonal blocks, T_1, T_2, T_3, and T_4.

Here, we use the standard finite-element interpolation operators and their transposes for restriction. For Q_1, the symbol is well-known [36] while, for the nodal basis for Q_2, the symbol is given in [19].

Definition 6.3.4. *The asymptotic two-grid convergence factor, ρ_{asp}, is defined as*

$$\rho_{\text{asp}} = \sup\{\rho(\widetilde{\boldsymbol{M}}_h(\boldsymbol{\theta})^{\text{TGM}}) : \boldsymbol{\theta} \in T^{\text{low}}\}.$$

In what follows, we consider a discrete form of ρ_{asp}, denoted by ρ_h, resulting from sampling ρ_{asp} over only a finite set of frequencies. For simplicity, we drop the subscript h throughout the rest of this paper, unless necessary for clarity.

6.3.2 Fourier representation of discretization operators

Fourier representation of the stabilized $Q_1 - Q_1$ discretization

By standard calculation, the symbols of the Q_1 stiffness and mass stencils are

$$
\begin{aligned}
\widetilde{A}_{Q_1}(\theta_1, \theta_2) &= \frac{2}{3}(4 - \cos\theta_1 - \cos\theta_2 - 2\cos\theta_1\cos\theta_2),\\
\widetilde{M}_{Q_1}(\theta_1, \theta_2) &= \frac{h^2}{9}(4 + 2\cos\theta_1 + 2\cos\theta_2 + \cos\theta_1\cos\theta_2),
\end{aligned}
$$

respectively. The stencils of the partial derivative operators $(\partial_x)_h$ and $(\partial_y)_h$ are

$$B_x^T = \frac{h}{12}\begin{bmatrix} -1 & 0 & 1 \\ -4 & 0 & 4 \\ -1 & 0 & 1 \end{bmatrix}, \ B_y^T = \frac{h}{12}\begin{bmatrix} 1 & 4 & 1 \\ 0 & 0 & 0 \\ -1 & -4 & -1 \end{bmatrix},$$

respectively, and the corresponding symbols are

$$\widetilde{B}_x^T(\theta_1, \theta_2) = \frac{ih}{3}\sin\theta_1(2+\cos\theta_2), \ \widetilde{B}_y^T(\theta_1, \theta_2) = \frac{ih}{3}(2+\cos\theta_1)\sin\theta_2,$$

where T denotes the conjugate transpose. Thus, the symbols of the stabilized finite-element discretizations of the Stokes equations are given by

$$\widetilde{\mathcal{L}}(\theta_1, \theta_2) = \begin{pmatrix} \widetilde{A}_{Q_1} & 0 & \widetilde{B}_x^T \\ 0 & \widetilde{A}_{Q_1} & \widetilde{B}_y^T \\ \widetilde{B}_x & \widetilde{B}_y & \widetilde{L}_h^{3,3} \end{pmatrix} := \begin{pmatrix} a & 0 & b_1 \\ 0 & a & b_2 \\ -b_1 & -b_2 & -c \end{pmatrix}.$$

For the Poisson-stabilized discretization, the symbol of $-L_h^{3,3}$ is $c = c_1 = a\beta h^2$. For the projection stabilized method, following (6.5), the symbol of $-L_h^{3,3}$ is

$$c_2 = \left(\frac{4 + 2\cos\theta_1 + 2\cos\theta_2 + \cos\theta_1\cos\theta_2}{9} - \frac{(1+\cos\theta_1)(1+\cos\theta_2)}{4}\right)h^2. \tag{6.9}$$

For convenience, we write $-C$ for the last block of Equation (6.2), and its symbol as $-c$ in the rest of this paper.

Fourier representation of stable $Q_2 - Q_1$ discretizations

The symbols of the stiffness and mass stencils for the Q_2 discretization using nodal basis functions in 1D are

$$\widetilde{A}_{Q_2}(\theta) = \frac{1}{3h}\begin{pmatrix} 14 + 2\cos\theta & -16\cos\frac{\theta}{2} \\ -16\cos\frac{\theta}{2} & 16 \end{pmatrix}, \ \widetilde{M}_{Q_2}(\theta) = \frac{h}{30}\begin{pmatrix} 8 - 2\cos\theta & 4\cos\frac{\theta}{2} \\ 4\cos\frac{\theta}{2} & 16 \end{pmatrix},$$

respectively [19]. Then, the Fourier representation of $-\Delta_h$ in 2D, can be written as a tensor product,

$$\widetilde{A}_2(\theta_1, \theta_2) \ = \ \widetilde{A}_{Q_2}(\theta_2) \otimes \widetilde{M}_{Q_2}(\theta_1) + \widetilde{M}_{Q_2}(\theta_2) \otimes \widetilde{A}_{Q_2}(\theta_1).$$

The tensor product preserves block structuring; that is, $\widetilde{A}_2(\theta_1, \theta_2)$ is a 4×4 matrix, ordered as mesh nodes, x-edge midpoints, y-edge midpoints, and cell centres. Each row of $\widetilde{A}_2(\theta_1, \theta_2)$ reflects the connections between one of the four types of degrees of freedom with each of these four types. Similarly, there are four types of stencils for $(\partial_x)_h$ and $(\partial_y)_h$.

The stencils and the symbols of $(\partial_x)_h$ for the nodal, x-edge, y-edge, and cell-centre degrees of freedom are

$$\begin{aligned}
B_N &= \frac{h}{18}\begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \ \widetilde{B}_N(\theta_1, \theta_2) = \frac{ih}{9}\sin\theta_1, \\
B_X &= \frac{h}{18}\begin{bmatrix} 0 & 0 \\ -4 & 4 \\ 0 & 0 \end{bmatrix}, \quad \widetilde{B}_X(\theta_1, \theta_2) = \frac{2ih}{9}\sin\frac{\theta_1}{2}, \\
B_Y &= \frac{h}{18}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \ \widetilde{B}_Y(\theta_1, \theta_2) = \frac{2ih}{9}\sin\theta_1\cos\frac{\theta_2}{2}, \\
B_C &= \frac{h}{18}\begin{bmatrix} -4 & 4 \\ -4 & 4 \end{bmatrix}, \quad \widetilde{B}_C(\theta_1, \theta_2) = \frac{8ih}{9}\sin\frac{\theta_1}{2}\cos\frac{\theta_2}{2},
\end{aligned}$$

respectively. Denote $\widetilde{B}_{Q_2,x}(\theta_1, \theta_2)^T = [\widetilde{B}_N; \widetilde{B}_X; \widetilde{B}_Y; \widetilde{B}_C]$.

Similarly to $\widetilde{B}_{Q_2,x}(\theta_1, \theta_2)^T$, the symbol of the stencil of $(\partial_y)_h$ can be written as

$$\widetilde{B}_{Q_2,y}(\theta_1, \theta_2)^T = [\widetilde{B}_N(\theta_2, \theta_1); \widetilde{B}_Y(\theta_2, \theta_1); \widetilde{B}_X(\theta_2, \theta_1); \widetilde{B}_C(\theta_2, \theta_1)].$$

Thus, the Fourier representation of the $Q_2 - Q_1$ finite-element discretization of the Stokes equations can be written as

$$\widetilde{\mathcal{L}}_h(\theta_1, \theta_2) = \begin{pmatrix} \widetilde{A}_2(\theta_1, \theta_2) & 0 & \widetilde{B}_{Q_2,x}(\theta_1, \theta_2)^T \\ 0 & \widetilde{A}_2(\theta_1, \theta_2) & \widetilde{B}_{Q_2,y}(\theta_1, \theta_2)^T \\ \widetilde{B}_{Q_2,x}(\theta_1, \theta_2) & \widetilde{B}_{Q_2,y}(\theta_1, \theta_2) & 0 \end{pmatrix}.$$

Note that the Fourier symbol for the $Q_2 - Q_1$ discretization is a 9×9 matrix, and that the LFA smoothing factor for the Q_2 approximation generally fails to predict the true two-grid convergence factor [19, 21]. The same behavior is seen for the relaxation schemes considered here. Therefore, we do not present smoothing factor analysis for this case and only optimize two-grid LFA predictions numerically.

6.4 Relaxation for $Q_1 - Q_1$ discretizations

6.4.1 DWJ relaxation

Distributive GS (DGS) relaxation [8, 27] is well known to be highly efficient for the MAC finite-difference discretization [36], and other discretizations [11, 12]. Its sequential nature is often seen as a significant drawback. However, Distributive weighted Jacobi (DWJ) relaxation was recently shown to achieve good performance for the MAC discretization [18]. Thus, we consider DWJ relaxation for the finite-element discretizations considered here. The discretized distribution operator can be represented by the preconditioner

$$\mathcal{P} = \begin{pmatrix} I_h & 0 & (\partial_x)_h \\ 0 & I_h & (\partial_y)_h \\ 0 & 0 & \Delta_h \end{pmatrix}.$$

Then, we apply blockwise weighted-Jacobi relaxation to the distributed operator

$$\mathcal{LP} \approx \mathcal{L}^* = \begin{pmatrix} -\Delta_h & 0 & 0 \\ 0 & -\Delta_h & 0 \\ -(\partial_x)_h & -(\partial_y)_h & -(\partial_x)_h^2 - (\partial_y)_h^2 + L^{3,3}\Delta_h \end{pmatrix}, \qquad (6.10)$$

where we note that the operators $(\partial_x)_h^2$ and $(\partial_y)_h^2$ are formed by taking products of the gradient operators and, thus, do not satisfy the identity $(\partial_x)_h^2 + (\partial_y)_h^2 = \Delta_h$.

The discrete matrix form of $\mathcal{P}$ is

$$\mathcal{P} = \begin{pmatrix} I & B^T \\ 0 & -A_p \end{pmatrix},$$

where A_p is the Laplacian operator discretized at the pressure points. For distributive weighted-Jacobi relaxation (with weights α_1, α_2), we need to solve a system of the form

$$M_D \delta\hat{x} = \begin{pmatrix} \alpha_1 \text{diag}(A) & 0 \\ B & \alpha_2 h^2 I \end{pmatrix} \begin{pmatrix} \delta\hat{\mathcal{U}} \\ \delta\hat{p} \end{pmatrix} = \begin{pmatrix} r_{\mathcal{U}} \\ r_p \end{pmatrix}, \qquad (6.11)$$

then distribute the updates as $\delta x = \mathcal{P}\delta\hat{x}$. We use h^2 in the $(2,2)$ block of (6.11), because the diagonal entries of the $(2,2)$ block will be of the form of a constant times h^2 (up to boundary conditions), for both stabilization terms. The error propagation operator for the scheme is, then, $I - \omega\mathcal{P}M_D^{-1}\mathcal{L}$.

The symbol of the blockwise weighted-Jacobi operator, M_D, is

$$\widetilde{M}_D(\theta_1,\theta_2) = \begin{pmatrix} \frac{8}{3}\alpha_1 & 0 & 0 \\ 0 & \frac{8}{3}\alpha_1 & 0 \\ -b_1 & -b_2 & h^2\alpha_2 \end{pmatrix}.$$

By standard calculation, the eigenvalues of the error-propagation symbol, $\widetilde{\mathcal{S}}_D(\alpha_1,\alpha_2,\omega,\boldsymbol{\theta}) = I - \omega\widetilde{\mathcal{P}}\widetilde{M}_D^{-1}\widetilde{\mathcal{L}}$, are

$$1 - \frac{\omega}{\alpha_1}y_1,\ \ 1 - \frac{\omega}{\alpha_1}y_1, 1 - \frac{\omega}{\alpha_2}y_2, \tag{6.12}$$

where $y_1 = \frac{3a}{8}$ and $y_2 = \frac{-b_1^2-b_2^2+ac}{h^2}$.

Noting that $y_1 = \frac{3a}{8}$ is very simple, we first consider a lower bound on the optimal LFA smoothing factor corresponding to y_1.

Lemma 6.4.1.

$$\mu^* := \min_{(\alpha_1,\omega)} \max_{\boldsymbol{\theta}\in T^{\text{high}}} \left\{ \left|1 - \frac{\omega}{\alpha_1}y_1\right| \right\} = \frac{1}{3},$$

and this value is achieved if and only if $\frac{\omega}{\alpha_1} = \frac{8}{9}$.

Proof. It is easy to check that $a = \frac{2(4-\cos\theta_1-\cos\theta_2-2\cos\theta_1\cos\theta_2)}{3} \in [2,4]$ for $\boldsymbol{\theta} \in T^{\text{high}}$. The minimum of y_1 is $y_{1,\min} = \frac{3}{4}$ with $(\cos\theta_1,\cos\theta_2) = (0,1)$ or $(1,0)$ and the maximum is $y_{1,\max} = \frac{3}{2}$ with $(\cos\theta_1,\cos\theta_2) = (1,-1)$ or $(-1,1)$. Thus, $\mu^* = \frac{y_{1,\max}+y_{1,\min}}{y_{1,\max}-y_{1,\min}} = \frac{1}{3}$ under the condition $\frac{\omega}{\alpha_1} = \frac{2}{y_{1,\min}+y_{1,\max}} = \frac{8}{9}$. □

Remark 6.4.1. *The optimal smoothing factor for damped Jacobi relaxation for the* Q_1 *finite-element discretization of the Laplacian is* $\frac{1}{3}$ *with* $\frac{\omega}{\alpha} = \frac{8}{9}$. *Thus, this offers an intuitive lower bound on the possible performance of block relaxation schemes that include this as a piece of the overall relaxation.*

From (6.12), we see that the only difference between the eigenvalues of DWJ relaxation for the Poisson-stabilized and projection stabilized methods is in the third eigenvalue, which depends on y_2 and, consequently, on the stabilization term.

Poisson-stabilized discretization with DWJ relaxation

For the Poisson-stabilized case, $y_2 = \frac{-b_1^2-b_2^2+ac}{h^2}$ with $c = \beta\alpha h^2$ and $\beta = \frac{1}{24}$. By standard calculation, $y_{2,\min} = \frac{8}{27}$, with $\big(\cos\theta_1, \cos\theta_2\big) = (-1,-1)$, and $y_{2,\max} = \frac{64}{51}$ with $\big(\cos\theta_1, \cos\theta_2\big) = (\frac{8}{17}, 0)$ or $(0, \frac{8}{17})$.

Theorem 6.4.1. *The optimal smoothing factor for the Poisson-stabilized discretization with DWJ relaxation is* $\frac{55}{89}$*, that is,*

$$\mu_{\text{opt}} = \min_{(\alpha_1,\omega,\alpha_2)} \max_{\boldsymbol{\theta}\in T^{\text{high}}} \left\{ \left|\lambda(\widetilde{\mathcal{S}}(\alpha_1,\alpha_2,\omega,\boldsymbol{\theta}))\right| \right\} = \frac{55}{89} \approx 0.618,$$

and is achieved if and only if

$$\frac{\omega}{\alpha_2} = \frac{459}{356}, \; \frac{136}{267} \le \frac{\omega}{\alpha_1} \le \frac{96}{89}. \tag{6.13}$$

Proof. $\min\limits_{(\alpha_2,\omega)} \max\limits_{\boldsymbol{\theta}\in T^{\text{high}}} \left\{ \left|1 - \frac{\omega}{\alpha_2} y_2\right| \right\} = \frac{y_{2,\max} - y_{2,\min}}{y_{2,\max} + y_{2,\min}} = \frac{55}{89}$ with the condition that $\frac{\omega}{\alpha_2} = \frac{2}{y_{2,\max}+y_{2,\min}} = \frac{459}{356}$. Because $\frac{55}{89} > \frac{1}{3}$, we need to require $|1 - \frac{\omega}{\alpha_1} y_1| \le \frac{55}{89}$ for all y_1 to achieve this factor. It follows that $\frac{136}{267} \le \frac{\omega}{\alpha_1} \le \frac{96}{89}$. □

Projection stabilized discretization with DWJ relaxation

For the projection stabilized discretization, y_2 depends on c_2 given in (6.9), and standard calculation gives $y_{2,\min} = \frac{8}{27}$ with $\big(\cos\theta_1, \cos\theta_2\big) = (-1,-1)$ and $y_{2,\max} = \frac{3}{2}$ with $(\cos\theta_1, \cos\theta_2) = (-\frac{1}{2}, 1)$ or $(1, -\frac{1}{2})$.

Theorem 6.4.2. *The optimal smoothing factor for the projection stabilized discretization with DWJ relaxation is* $\frac{65}{97}$*, that is,*

$$\mu_{\text{opt}} = \min_{(\alpha_1,\omega,\alpha_2)} \max_{\boldsymbol{\theta}\in T^{\text{high}}} \left\{ \left|\lambda(\widetilde{\mathcal{S}}(\alpha_1,\alpha_2,\omega,\boldsymbol{\theta}))\right| \right\} = \frac{65}{97} \approx 0.670,$$

and is achieved if and only if

$$\frac{\omega}{\alpha_2} = \frac{108}{97}, \; \frac{128}{291} \le \frac{\omega}{\alpha_1} \le \frac{108}{97}. \tag{6.14}$$

Proof. $\min\limits_{(\alpha_2,\omega)} \max\limits_{\boldsymbol{\theta}\in T^{\text{high}}} \left\{ \left|1 - \frac{\omega}{\alpha_2} y_2\right| \right\} = \frac{y_{2,\max} - y_{2,\min}}{y_{2,\max} + y_{2,\min}} = \frac{65}{97}$ with the condition that

$\frac{\omega}{\alpha_2} = \frac{2}{y_{2,\max}+y_{2,\min}} = \frac{108}{97}$. Since $\frac{65}{97} > \frac{1}{3}$, we need to require $|1 - \frac{\omega}{\alpha_1}y_1| \leq \frac{65}{97}$ for all y_1 to achieve this factor, which leads to $\frac{128}{291} \leq \frac{\omega}{\alpha_1} \leq \frac{108}{97}$. □

Comparing the Poisson-stabilized and projection stabilized discretizations using DWJ, we see that the optimal LFA smoothing factor for the Poisson-stabilized discretization slightly outperforms that of the projection stabilized discretization. In both cases, a stronger relaxation on the $(3, 3)$ block of (6.10) would be needed in order to improve performance to match the lower bound on the convergence factor of $\frac{1}{3}$.

6.4.2 Braess-Sarazin relaxation

Although DWJ relaxation is efficient, we see clearly in the above that it "underpeforms" in relaxation to weighted Jacobi relaxation for the scalar Poisson problem. Furthermore, proper construction of the preconditioner, $\mathcal{P}$, is not always possible or straightforward, especially for other types of saddle-point problems. Considering these obstacles, we also analyse other block-structured relaxation schemes. Braess-Sarazin-type algorithms were originally developed as a relaxation scheme for the Stokes equations [6], requiring the solution of a greatly simplified but global saddle-point system. The (exact) BSR approach was first introduced in [6], where it was shown that a multigrid convergence rate of $O(k^{-1})$ can be achieved, where k denotes the number of smoothing steps on each level. As a relaxation scheme for the system in (6.2), one solves a system of the form

$$M_E\delta_x = \begin{pmatrix} \alpha D & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \delta\mathcal{U} \\ \delta p \end{pmatrix} = \begin{pmatrix} r_{\mathcal{U}} \\ r_p \end{pmatrix}, \tag{6.15}$$

where D is an approximation to A, the inverse of which is easy to apply, for example I, or diag(A). Solutions of (6.15) are computed in two stages as

$$\begin{aligned} S\delta p &= \frac{1}{\alpha}BD^{-1}r_{\mathcal{U}} - r_p, \\ \delta\mathcal{U} &= \frac{1}{\alpha}D^{-1}(r_{\mathcal{U}} - B^T\delta p), \end{aligned}$$

where $S = \frac{1}{\alpha}BD^{-1}B^T + C$, and $\alpha > 0$ is a chosen weight for D to obtain a better approximation to A. We consider an additional weight, ω, for the global update, δx, to improve the effectiveness of the correction to both the velocity and pressure unknowns.

There is a significant difficulty in practical use of exact BSR because it requires an exact inversion of the approximate Schur complement, S, which is typically very expensive. A broader class of iterative methods for the Stokes problem is discussed in [43], which demonstrated that the same $O(k^{-1})$ performance can be achieved as with exact BSR when the pressure correction equation is not solved exactly. In practice, an approximate solve is sufficient for the Schur complement system, such as with a few sweeps of weighted Jacobi relaxation or a few multigrid cycles. In what follows, we take $D = \text{diag}(A)$ and analyze exact BSR; to see what convergence factor can be achieved. In numerical experiments, we then consider whether it is possible to achieve the same convergence factor using an inexact solver. The symbol of M_E is given by

$$\widetilde{M}_E(\theta_1, \theta_2) = \begin{pmatrix} \frac{8}{3}\alpha & 0 & b_1 \\ 0 & \frac{8}{3}\alpha & b_2 \\ -b_1 & -b_2 & -c \end{pmatrix}.$$

The symbol of the error-propagation matrix for weighted exact BSR is $\widetilde{\mathcal{S}}_E(\alpha, \omega, \boldsymbol{\theta}) = I - \omega\widetilde{M}_E^{-1}\widetilde{\mathcal{L}}$. A standard calculation shows that the determinant of $\widetilde{\mathcal{L}} - \lambda\widetilde{M}_E$ is

$$\pi_E(\lambda; \alpha) = (1-\lambda)(a - \frac{8}{3}\alpha\lambda)\Big[(1-\lambda)(b_1^2 + b_2^2) + (\frac{8}{3}\alpha\lambda - a)c\Big]. \tag{6.16}$$

We first establish a lower bound on the LFA smoothing factor for the stabilized method with BSR.

Theorem 6.4.3. *The optimal LFA smoothing factor for the Poisson-stabilized and projection stabilized discretizations with exact BSR is not less than* $\frac{1}{3}$.

Proof. From (6.16), two eigenvalues of $\widetilde{M}_E^{-1}\widetilde{\mathcal{L}}$ are given by

$$\lambda_1 = 1, \ \lambda_2 = \frac{3a}{8\alpha},$$

which are independent of the stabilization term, c. From Lemma 6.4.1, we know that for λ_2, the optimal smoothing factor is $\frac{1}{3}$, under the condition that $\frac{\omega}{\alpha} = \frac{8}{9}$. Note that if $|1 - \omega\lambda_1| \leq \frac{1}{3}$, then $\frac{2}{3} \leq \omega \leq \frac{4}{3}$. Because there is another eigenvalue, λ_3, the optimal LFA smoothing factor is not less than $\frac{1}{3}$. □

Similarly to DWJ, we see that the Jacobi relaxation for the Laplacian discretization places a limit on the overall performance of BSR. From (6.16), the third eigenvalue of

$\widetilde{M}_E^{-1}\widetilde{\mathcal{L}}$ is $\lambda_3 = \frac{ac+b}{\frac{8}{3}\alpha c+b}$, where $b = -(b_1^2 + b_2^2) \geq 0$ (because both b_1 and b_2 are imaginary). Thus, we only need to check whether we can choose α and ω so that $|1-\omega\lambda_3| \leq \frac{1}{3}$ over all high frequencies, while also ensuring $|1-\omega\lambda_1| \leq \frac{1}{3}$ and $|1-\omega\lambda_2| \leq \frac{1}{3}$.

Theorem 6.4.4. *The optimal smoothing factor for both the Poisson-stabilized and projection stabilized discretizations with exact BSR is*

$$\mu_{\text{opt}} = \min_{(\alpha,\omega)} \max_{\boldsymbol{\theta}\in T^{\text{high}}} \left|\lambda(\widetilde{\mathcal{S}}(\alpha,\omega,\boldsymbol{\theta}))\right| = \frac{1}{3},$$

if and only if

$$\frac{\omega}{\alpha} = \frac{8}{9},\ \frac{3}{4} \leq \alpha \leq \frac{3}{2}.$$

Proof. Note that $a \in [2,4]$, and choose α such that $2 = a_{\min} \leq \frac{8}{3}\alpha \leq a_{\max} = 4$. If c is positive, the following always holds

$$\frac{3}{4\alpha} = \frac{a_{\min}}{\frac{8}{3}\alpha} \leq \frac{a_{\min}c+b}{\frac{8}{3}\alpha c+b} \leq \frac{ac+b}{\frac{8}{3}\alpha c+b} \leq \frac{a_{\max}c+b}{\frac{8}{3}\alpha c+b} \leq \frac{a_{\max}}{\frac{8}{3}\alpha} = \frac{3}{2\alpha}.$$

Furthermore, if $\frac{\omega}{\alpha} = \frac{8}{9}$, we have

$$\frac{2}{3} = \frac{3}{4\alpha}\cdot\frac{8}{9}\alpha \leq \omega\lambda_3 \leq \frac{3}{2\alpha}\cdot\frac{8}{9}\alpha = \frac{4}{3}. \tag{6.17}$$

For both discretizations, we can check that $c > 0$ over the high frequencies. From (6.17), it is easy to see that $|1-\omega\lambda_3| \leq \frac{1}{3}$, with $\alpha = \frac{9}{8}\omega \in [\frac{3}{4}, \frac{3}{2}]$. □

6.4.3 Numerical experiments for stabilized discretizations

We now present LFA predictions, validating DWJ, (inexact) BSR, and the related Uzawa iteration against measured multigrid performance for these schemes. We consider the homogeneous problem in (6.1), with periodic boundary conditions, and a random initial guess, $x_h^{(0)}$.

Convergence is measured using the averaged convergence factor, $\hat{\rho}_h^{(k)} = \sqrt[k]{\frac{\|d_h^{(k)}\|_2}{\|d_h^{(0)}\|_2}}$, with $k = 100$, and $d_h^{(k)} = b - Kx_h^{(k)}$. The LFA predictions are made with $h = 1/128$, for both the smoothing factor, μ, and two-grid convergence factor, ρ_h. For testing, we use standard $W(\nu_1,\nu_2)$ cycles with bilinear interpolation for Q_1 variables and biquadratic interpolation for Q_2 variables, and their adjoints for restriction. We consider both rediscretization and Galerkin coarsening, noting that they coincide for all terms except the stabilization terms that include a scaling of h^2. The coarsest grid is a mesh with 4 elements.

PoSD with DWJ

From the range of parameters allowed in (6.13), we select $\alpha_1 = 1.451$, $\alpha_2 = 1.000$, and $\omega = 1.290$ (for convenience, satisfying the equality in (6.13)) to compute the LFA predictions. Figure 6.1 shows the spectrum of the two-grid error-propagation operators for DWJ relaxation with rediscretization and Galerkin coarsening. Note that the two-grid convergence factor is the same as the optimal smoothing factor for rediscretization, but not for Galerkin coarsening.

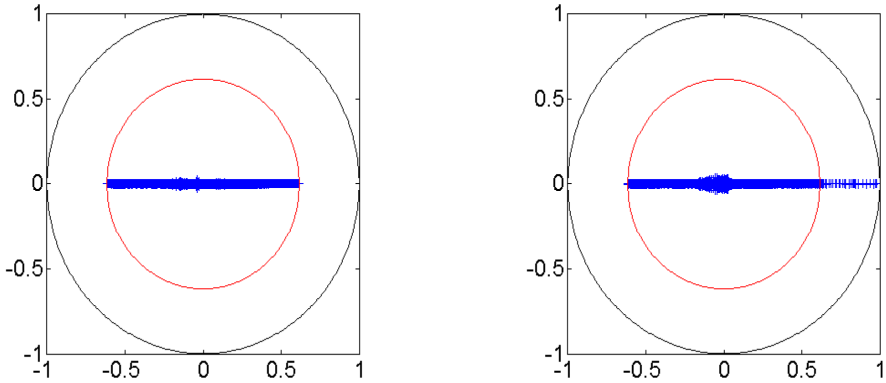

Figure 6.1: The spectrum of the two-grid error-propagation operator using DWJ for PoSD. Results with rediscretization are shown at left, while those with Galerkin coarsening are at right. In both figures, the inner circle has radius equal to the LFA smoothing factor.

In order to see the sensitivity of performance to parameter choice, we consider the two-grid LFA convergence factor with rediscretization coarsening. From (6.13), we know that there are many optimal parameters. To fix a single parameter for DWJ, we consider the case of $\omega = \frac{459}{356}$ and, at the left of Figure 6.2, we present the LFA-predicted two-grid convergence factors for DWJ with variation in α_1 and α_2. Here, we see strong sensitivity to "too small" values of both parameters, for $\alpha_1 < 1$ and $\alpha_2 < 0.9$, including a notable portion of the optimal range of values predicted by the LFA smoothing factor. At the right of Figure 6.2, we fix $\alpha_2 = \frac{356}{459}\omega$ and vary ω and α_1. The two lines are the lower and upper bounds from (6.13), between which LFA predicts the optimal convergence factor should be achieved. Note that not all of the allowed parameters obtain the optimal convergence factor. Here, we see great sensitivity for large values of ω, but a large range with generally similar performance as in the optimal parameter case.

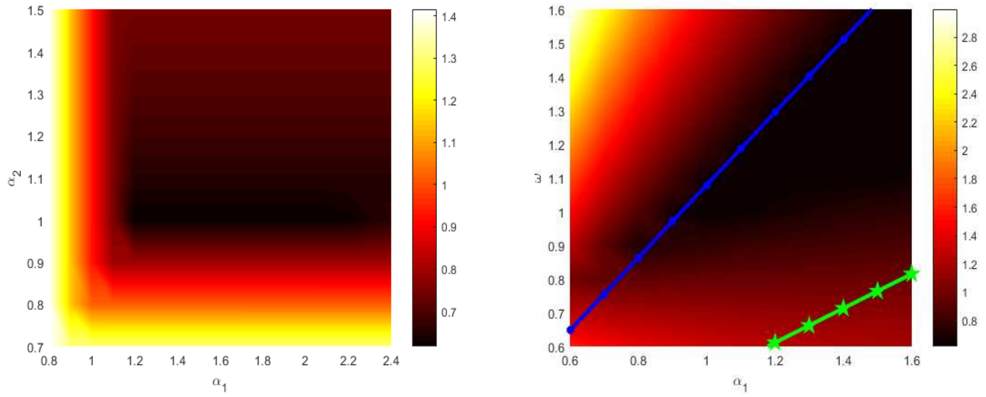

Figure 6.2: The two-grid LFA convergence factor for the PoSD using DWJ and rediscretization. At left, we fix $\omega = \frac{459}{356}$ and vary α_1 and α_2. At right, we fix $\alpha_2 = \frac{356}{459}\omega$ and vary ω and α_1.

In Table 6.1, we present the multigrid performance of DWJ with W-cycles for rediscretization coarsening. These results show measured multigrid convergence factors that coincide with the LFA-predicted two-grid convergence factors. Similar results are seen for V-cycles with rediscretization. For Galerkin coarsening, nearly identical W-cycle results are seen when $\nu_1 + \nu_2 > 2$, but divergence is seen for W-cycles with $\nu_1 + \nu_2 = 1$ or 2, and for all V-cycles tested.

Table 6.1: W-cycle convergence factors for DWJ with rediscretization for PoSD.

Cycle $\hat{\rho}_h$	$W(0,1)$	$W(1,0)$	$W(1,1)$	$W(1,2)$	$W(2,1)$	$W(2,2)$
$\alpha_1 = 1.451,\ \alpha_2 = 1.000,\ \omega = 1.290,\ \mu = 0.618$						
$\rho_{h=1/128}$	0.618	0.618	0.382	0.236	0.236	0.146
$\hat{\rho}^{(100)}_{h=1/64}$	0.564	0.568	0.349	0.215	0.214	0.133
$\hat{\rho}^{(100)}_{h=1/128}$	0.561	0.568	0.348	0.215	0.214	0.132

PrSD with DWJ

From the range of parameters allowed in (6.14), we choose $\alpha_1 = 1,\ \alpha_2 = 1,\ \omega = \frac{108}{97}$. Figure 6.3 shows that the smoothing factor provides a good prediction for the two-grid convergence factor with rediscretization, but not with Galerkin coarsening.

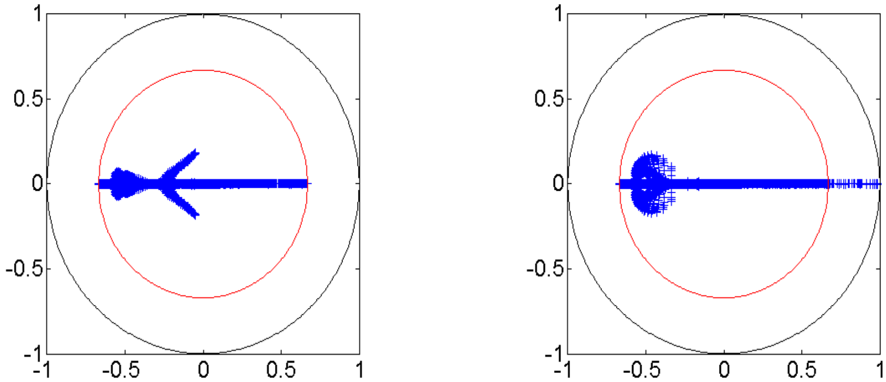

Figure 6.3: The spectrum of the two-grid error-propagation operator using DWJ for PrSD. Results with rediscretization are shown at left, while those with Galerkin coarsening are at right. In both figures, the inner circle has radius equal to the LFA smoothing factor.

Similarly to the discussion above, we consider the sensitivity to parameter choice for DWJ applied to PrSD. To fix a single parameter for DWJ, we consider the case of $\omega = \frac{108}{97}$. At the left of Figure 6.4, we present the LFA-predicted convergence factors for DWJ with variation in α_1 and α_2, again seeing a strong sensitivity to “too small” values of the parameters. At the right of Figure 6.4, we fix $\alpha_2 = \frac{97}{108}\omega$. The two lines are the lower and upper bounds from (6.14), between which LFA predicts the optimal convergence factor should be achieved. Note that not all of the parameters in this range obtain the optimal convergence factor. We see that, for small α_1, the convergence factor is very sensitive to large values of ω.

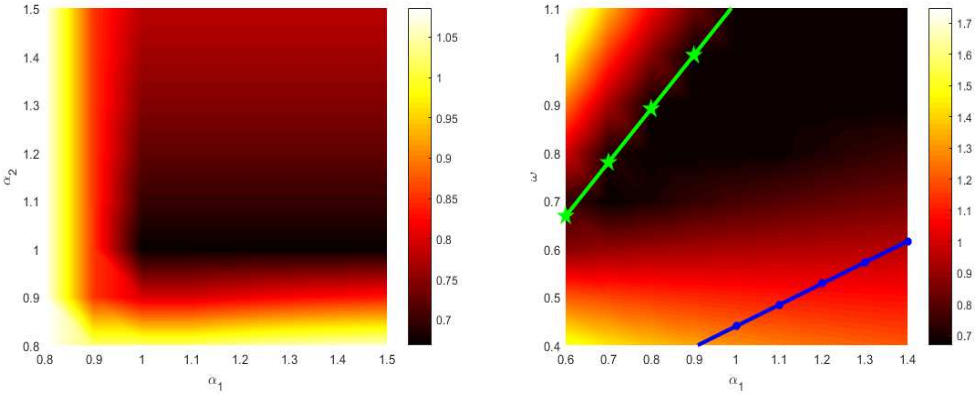

Figure 6.4: The two-grid LFA convergence factor for the PrSD using DWJ and rediscretization. At left, we fix $\omega = \frac{108}{97}$ and vary α_1 and α_2. At right, we fix $\alpha_2 = \frac{97}{108}\omega$ and vary ω and α_1.

In Table 6.2, we present the multigrid performance of DWJ relaxation with W-cycles for rediscretization coarsening. We see that the measured multigrid convergence factors match well with the LFA-predicted two-grid convergence factors. For Galerkin coarsening, as in the case of PoSD, we see divergence when $\nu_1+\nu_2 \leq 2$, but performance matching that of rediscretization for $\nu_1 + \nu_2 > 2$. Here, V-cycle results are similar to the W-cycle results for both rediscretization and Galerkin coarsening approaches.

Table 6.2: W-cycle convergence factors for DWJ with rediscretization for PrSD.

Cycle $\hat{\rho}_h$	$W(0,1)$	$W(1,0)$	$W(1,1)$	$W(1,2)$	$W(2,1)$	$W(2,2)$
$\alpha_1 = 1,\ \alpha_2 = 1,\ \omega = 108/97,\ \mu = 0.670$						
$\rho_{h=1/128}$	0.670	0.670	0.449	0.300	0.300	0.201
$\hat{\rho}^{(100)}_{h=1/64}$	0.652	0.652	0.436	0.291	0.292	0.196
$\hat{\rho}^{(100)}_{h=1/128}$	0.651	0.652	0.435	0.291	0.291	0.195

PoSD with BSR

Next, we consider BSR for PoSD, first displaying the two-grid LFA convergence factor as a function of α for rediscretization coarsening with $\omega = \frac{8}{9}\alpha$ in Figure 6.5. Comparing the convergence factor with μ^2, for $\nu_1 = \nu_2 = 1$, we see a good match over the interior of the interval $\frac{3}{4} \leq \alpha \leq \frac{3}{2}$ predicted by Theorem 6.4.4. For larger values of $\nu_1 + \nu_2$, this agreement deteriorates as is typical when the behavior of coarse-grid correction becomes dominant.

At the right of Figure 6.5, we see good agreement between ρ and μ when $\nu_1 + \nu_2 = 1$ with fixed $\alpha = 1$. In both cases, similar behaviour is seen with Galerkin coarsening.

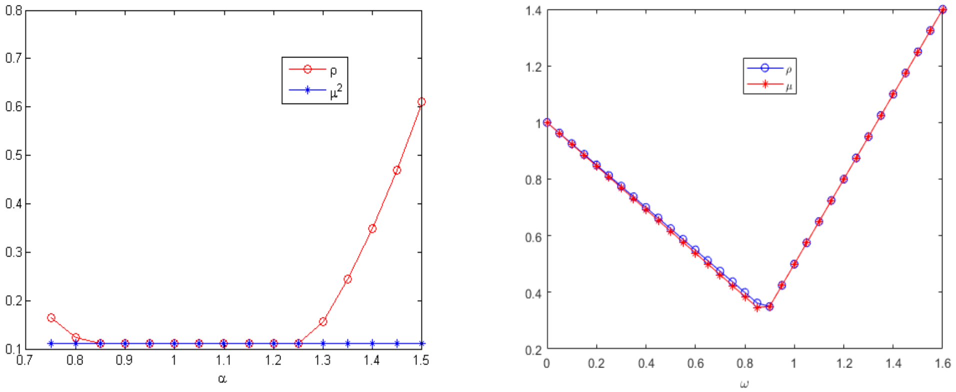

Figure 6.5: Two-grid and smoothing factors for BSR with rediscretization for PoSD. At left, comparing ρ with μ^2 for $\nu_1 = \nu_2 = 1$ with $\omega = \frac{8}{9}\alpha$. At right, comparing ρ with μ for $\nu_1 + \nu_2 = 1$ with $\alpha = 1$.

Motivated by the above, we use $\alpha = 1$ and $\omega = \frac{8}{9}$ for multigrid experiments with rediscretization, solving the Schur complement equation exactly. Table 6.3 shows that the measured multigrid convergence factors match well with the LFA-predicted two-grid convergence factors for W-cycles with rediscretization coarsening, and similar results are seen for Galerkin coarsening.

Table 6.3: W-cycle convergence factors for BSR with rediscretization for PoSD.

Cycle $\hat{\rho}_h$	$W(0,1)$	$W(1,0)$	$W(1,1)$	$W(1,2)$	$W(2,1)$	$W(2,2)$
$\rho_{h=1/128}$	0.333	0.333	0.111	0.079	0.079	0.062
$\hat{\rho}^{(100)}_{h=1/64}$	0.324	0.323	0.112	0.075	0.075	0.058
$\hat{\rho}^{(100)}_{h=1/128}$	0.323	0.323	0.112	0.075	0.075	0.058

For practical use, we consider solving the Schur complement system inexactly, using a few sweeps of Jacobi. For a two-grid method, similar performance to Table 6.3 can be obtained with only 2 sweeps of relaxation per Schur complement solve, but degradation is seen for W-cycles, particularly as $\nu_1 + \nu_2$ increases.

To maintain the performance observed for exact BSR, we could simply use more Jacobi iterations on the Schur complement system; however, experiments showed that this did not lead to a scalable algorithm. Instead, we consider solving the Schur complement system by applying a multigrid $W(1,1)$-cycle using weighted relaxation with weight ω_I, shown in Table 6.4. Following [43], we refer to this as inexact Braess-Sarazin relaxation (IBSR). From Table 6.4, we observe that using only 1 or 2 $W(1,1)$-cycles on the approximate Schur complement achieves convergence factors matching those in Table 6.3, and that the $W(1,1)$ cycle is the most cost effective.

Table 6.4: W-cycle convergence factor for IBSR with inner $W(1,1)$-cycle for the PoSD. In brackets, minimum value of the number of inner $W(1,1)$-cycles that achieves the same convergence factors as those of LFA predictions for exact BSR.

Cycle $\hat{\rho}_h$	$W(0,1)$	$W(1,0)$	$W(1,1)$	$W(1,2)$	$W(2,1)$	$W(2,2)$
$(\alpha,\omega,\omega_I)=(1,8/9,1)$						
$\rho_{h=1/128}$(LFA)	0.333	0.333	0.111	0.079	0.079	0.062
$\hat{\rho}^{(100)}_{h=1/64}$	0.368(2)	0.346(2)	0.131(2)	0.075(2)	0.075(2)	0.059(1)
$\hat{\rho}^{(100)}_{h=1/128}$	0.343(2)	0.351(2)	0.111(2)	0.075(2)	0.075(2)	0.063(1)

PrSD with BSR

We now consider BSR for the PrSD. At the left of Figure 6.6, we see a good agreement between the two-grid convergence factor and μ^2 for $\nu_1=\nu_2=1$ for some parameters in the range defined in Theorem 6.4.4 when using rediscretization. A larger interval of agreement is seen for the corresponding results for Galerkin coarsening. In both cases, agreement between the two-grid convergence factor and $\mu^{\nu_1+\nu_2}$ degrades as $\nu_1+\nu_2$ increases, as expected.

Note that Theorem 6.4.4 demonstrates that the smoothing factor for BSR is a function of $\frac{\omega}{\alpha}$ (but the same is not necessarily true for the convergence factor). In Figure 6.6, we plot the LFA smoothing and convergence factors for BSR with rediscretization as a function of ω, with $\alpha=0.8$ and see that these factors generally agree, although the smoothing factor slightly underestimates the convergence factor. As two-grid convergence is, however, sensitive to the choice of α, the smoothing factor generally underestimates the convergence factor for other values of α.

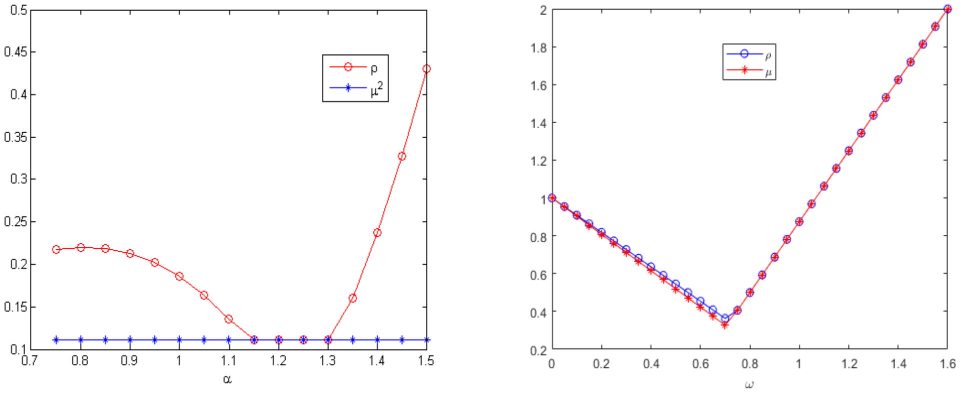

Figure 6.6: Two-grid and smoothing factors for BSR with rediscretization for PrSD. At left, comparing ρ with μ^2 for $\nu_1 = \nu_2 = 1$ with $\omega = \frac{8}{9}\alpha$. At right, comparing ρ with μ for $\nu_1 + \nu_2 = 1$ with $\alpha = \frac{4}{5}$.

Fixing $\omega = \frac{8}{9}\alpha$ with $\alpha = 1.2$ (as suggested by Figure 6.6 for $\nu_1 = \nu_2 = 1$), Table 6.5 shows that the measured multigrid convergence factors again match well with the LFA-predicted two-grid convergence factors for W-cycles with rediscretization coarsening. Note, however, the degradation for $\nu_1 + \nu_2 = 1$, where the smoothing factor analysis predicts a convergence factor of $\frac{1}{3}$ that is not realized. However, the convergence factor of $\frac{1}{3}$ can be achieved by choosing $\alpha = \frac{4}{5}$ and $\omega = \frac{8}{9}\alpha$ in the BSR scheme with either $W(1,0)$ or $W(0,1)$ cycles, but these choices lead to a slight degradation with $\nu_1 + \nu_2 > 1$. Similar results are seen for Galerkin coarsening with $\alpha = 1$ and $\omega = \frac{8}{9}\alpha$ with the notable exception that the smoothing factor prediction was matched by both the two-grid LFA convergence factor and true W-cycle convergence in this case for all experiments.

Table 6.5: W-cycle convergence factors for BSR with rediscretization for PrSD.

Cycle $\hat{\rho}_h$	$W(0,1)$	$W(1,0)$	$W(1,1)$	$W(1,2)$	$W(2,1)$	$W(2,2)$
$\rho_{h=1/128}$	0.673	0.673	0.111	0.079	0.079	0.062
$\hat{\rho}^{(100)}_{h=1/64}$	0.585	0.585	0.112	0.075	0.075	0.058
$\hat{\rho}^{(100)}_{h=1/128}$	0.584	0.584	0.112	0.075	0.075	0.058

For practical use, we again consider solving the Schur complement system inexactly, using the Jacobi iteration. As was the case for PoSD, we can recover performance consistent with the exact BSR results in Table 6.5 only for the case of two-grid cycles with $\nu_1 + \nu_2 = 1$ when using 3 Jacobi iterations on the Schur complement.

Again we consider solving the Schur complement system by applying a multigrid $W(1,1)$-cycle. Table 6.6 shows that this IBSR is seen to be effective, requiring 1 to 4 $W(1,1)$ cycles on the Schur complement system to math the convergence seen in Table 6.5. Again, $W(1,1)$ cycles are seem to be most cost effective.

Table 6.6: W-cycle convergence factor for IBSR with inner $W(1,1)$-cycle for the PrSD. In brackets, minimum value of the number of inner $W(1,1)$-cycles that achieves the same convergence factors as those of LFA predictions for exact BSR.

Cycle / $\hat{\rho}_h$	$W(0,1)$	$W(1,0)$	$W(1,1)$	$W(1,2)$	$W(2,1)$	$W(2,2)$
$(\alpha, \omega, \omega_I) = (6/5, 16/15, 1.1)$						
$\rho_{h=1/128}$ (LFA)	0.673	0.673	0.111	0.079	0.079	0.062
$\hat{\rho}^{(100)}_{h=1/64}$	0.680(4)	0.677(1)	0.112(3)	0.075(2)	0.075(2)	0.059(1)
$\hat{\rho}^{(100)}_{h=1/128}$	0.659(1)	0.662(1)	0.112(3)	0.075(2)	0.075(2)	0.067(1)

6.4.4 Stabilized discretizations with Uzawa relaxation

Multigrid methods using Uzawa relaxation schemes [16, 17, 26] are popular approaches due to their low cost per iteration. We consider Uzawa relaxation as a simplification of BSR, determining the update as the (weighted) solution of

$$M\delta x = \begin{pmatrix} \alpha D & 0 \\ B & -\hat{S} \end{pmatrix} \begin{pmatrix} \delta\mathcal{U} \\ \delta p \end{pmatrix} = \begin{pmatrix} r_{\mathcal{U}} \\ r_p \end{pmatrix},$$

where αD is an approximation to A and $-\hat{S}$ is an approximation of the Schur complement, $-BA^{-1}B^T - C$.

Here, we consider an analogue to exact BSR with $D = \text{diag}(A)$. The choice of $\hat{S}$ is discussed later. In this setting, we observe that minimizing the LFA smoothing factor does not minimize the LFA convergence factor. Thus, we consider minimizing the two-grid convergence factor numerically for $\nu_1 + \nu_2 = 1$ and $\nu_1 = \nu_2 = 1$ with rediscretization coarsening, and compare with measured multigrid performance.

We consider three approximations to the Schur complement, starting from the true approximate Schur complement, $C + B(\alpha\text{diag}(A))^{-1}B^T$. Motivated by the stable finite-element case, we also consider replacing $B(\alpha\text{diag}(A))^{-1}B^T$ in this matrix by a weighted mass matrix, yielding $\hat{S} = C + \delta Q$. Finally, motivated by the finite-difference case and efficiency of implementation, we consider taking $\hat{S} = \sigma h^2 I$, for a scalar weight, σ, to be optimized by the LFA. Note that, due to the constant-coefficient stencils assumed by LFA, this corresponds to using a single sweep of Jacobi to approximate solution of either of the two above approximations.

For the case of $C + B(\alpha\text{diag}(A))^{-1}B^T$, the optimized LFA two-grid convergence factors for $\nu_1 + \nu_2 = 1$ with rediscretization coarsening are 0.428 for PoSD and 0.436 for PrSD. These are notably worse than the BSR smoothing factor of $\frac{1}{3}$, which is achieved for $W(1,0)$ or $W(0,1)$ cycles. Here, $W(1,0)$ cycles reflect this convergence, achieving measured convergence factor rates of 0.417 for PoSD and 0.526 for PrSD. Increasing the number of relaxation sweeps per iteration yields some improvement in the predicted LFA convergence factors when optimizing parameters again, but not enough to outperform repeated $W(1,0)$ cycles.

For the mass-matrix-based approximation, $\hat{S} = C + \delta Q$, the optimized two-grid convergence factors for $\nu_1 + \nu_2 = 1$ with rediscretization coarsening are 0.5 for PoSD and 0.417 for PrSD. While poorer convergence might be expected in both cases, the addition of an extra parameter, δ, allows the (slight) improvement for PrSD. In both cases, we observe consistent performance with numerical experiments, achieving convergence factors of 0.493 for PoSD and 0.392 for PrSD using $W(0,1)$ or $W(1,0)$ cycles.

Finally, for the diagonal approximation $\hat{S} = \sigma h^2 I$, we achieve notably better performance optimizing with $\nu_1 = \nu_2 = 1$ than for $\nu_1 + \nu_2 = 1$. For PoSD, the optimized two-grid LFA convergence factor is 0.382, while it is 0.497 for PrSD. In practice, we achieve slightly worse convergence factors using $W(1,1)$ cycles with rediscretization coarsening, of 0.531 for PoSD and 0.543 for PrSD. These are both significantly worse than the convergence factors of $\frac{1}{9}$ observed using inexact BSR; however, it must be noted that W-cycles on the Schur complement system were needed in that case. A better approximation to inverting the true approximate Schur complement would be to apply multigrid to it, just as was done for IBSR above. Here, we observe that significant work may be needed to achieve convergence similar to that of Uzawa where the Schur complement is exactly inverted, requiring 10 $W(1,1)$-cycles on the approximate Schur complement to achieve a convergence factor of 0.416 for PoSD and 0.522 for PrSD, suggesting that the Jacobi version of Uzawa is ultimately more efficient.

6.4.5 Comparing cost and performance

As discussed in [18], the costs per iteration of DWJ and inexact BSR are roughly equal, so long as the cost of iteration on the BSR approximate Schur complement is close to that of a single Jacobi step. In contrast, 2 sweeps of Uzawa, with $\hat{S} = \sigma h^2 I$, have cost comparable to a single sweep of inexact BSR. Thus, for both PoSD and PrSD, inexact BSR is seen to be most cost effective, with $W(1,1)$ convergence factors of $\frac{1}{9}$, compared to about 0.25 for 2 $W(1,1)$ cycles of Uzawa and 0.35 or 0.44 for a single $W(1,1)$ cycle of DWJ. While the added cost of W-cycles on the Schur complement are significant, they clearly pay off in this case.

6.5 Relaxation for $Q_2 - Q_1$ discretization

As explored in [19], classical LFA smoothing factor analysis is unreliable for Q_2 discretizations, making it unsuitable for analysis of the standard stable $Q_2 - Q_1$ discretization of the Stokes equations. Thus, we consider only numerical ("brute force") optimization of two-grid LFA convergence factors in this setting.

For DWJ, we find optimal convergence factors of 0.619 for $\nu_1 + \nu_2 = 1$ and 0.558 for $\nu_1 = \nu_2 = 1$. While the former is quite comparable to convergence predicted and achieved for both stabilized discretizations with $\nu_1 + \nu_2 = 1$, we see a significant lack of improvement with increased relaxation, in contrast to the equal-order case. The same is observed for multigrid W-cycle performance, with $W(1,0)$ convergence measured at 0.620 and $W(1,1)$ convergence measured at 0.510.

For exact BSR, we find optimal convergence factors of 0.551 for $\nu_1 + \nu_2 = 1$ and 0.250 for $\nu_1 = \nu_2 = 1$. While these are slightly larger than the comparable factors of $\frac{1}{3}$ and $\frac{1}{9}$, respectively, for the stabilized discretizations, they still reflect good performance of the underlying method.

At left of Figure 6.7, we show the spectral radius of the error-propagation symbol for exact BSR as a function of Fourier frequency, $\boldsymbol{\theta}$, noting that predicted reduction over the high frequencies is not as good as would be needed to equal two-grid convergence in the equal-order case. In order to see how the convergence factor changes with the parameters α and ω, we display the convergence factor as a function of α and ω at the right of Figure 6.7. The optimal choice, of $\alpha = 1.1$ and $\omega = 1.05$, occurs in a narrow band of ω values, but larger range of α values lead to reasonable results.

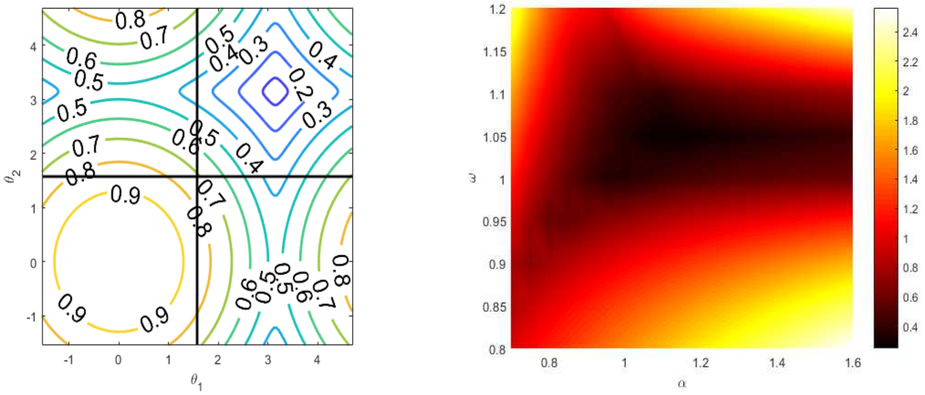

Figure 6.7: At left, the spectral radius of the error-propagation symbol for exact BSR applied to the $Q_2 - Q_1$ discretization, as a function of the Fourier mode, $\boldsymbol{\theta}$. At right, the LFA-predicted two-grid convergence factor for BSR applied to the $Q_2 - Q_1$ discretization as a function of α and ω, with $(\nu_1, \nu_2) = (1, 1)$.

As always, an inexact solve of the Schur complement system is needed to yield a practical variant of BSR. While 2 sweeps of Jacobi appears sufficient to achieve scalable W-cycle convergence when $\nu_1 + \nu_2 > 2$, we find 3 sweeps are needed to achieve $W(1, 1)$ convergence factors of 0.240, in contrast to results in [43] and for the equal-order discretizations considered here, where a much stronger iteration was needed. Similar results were seen for $V(1, 1)$ cycles when 3 sweeps of Jacobi were used for the Schur complement system.

Finally, we consider the same three variants of Uzawa relaxation as examined above for the equal-order case. For $\hat{S} = B(\alpha\text{diag}(A))^{-1}B^T$, the best convergence factor found for $\nu_1 + \nu_2 = 1$ was 0.729, while better convergence was predicted for $\hat{S} = \delta Q$, with factor 0.554. This is to be expected, perhaps, since the Q_1 mass matrix is well-known to be a better approximation of the true Schur complement than the classical BSR approximate Schur complement. However, approximating either by a single sweep of Jacobi, yielding $\hat{S} = \sigma h^2 I$, gives a convergence factor 0.717. While 2-grid cycles with

$\nu_1 + \nu_2 = 1$ match the predicted convergence factor, W-cycles did not converge for these parameters.

Comparing, then, the efficiency of inexact BSR and DWJ for the $Q_2 - Q_1$ discretization, we see that inexact BSR, where $W(1,1)$ cycles achieve a convergence factor of 0.24 provides roughly the same reduction as 3 cycles with 1 DWJ sweep per cycle, where LFA predicts $\rho = 0.619$. Noting that inexact BSR is relatively more expensive in this case, with cost dominated by the two diagonal scalings per sweep on the Q_2 velocity degrees of freedom, we suggest a proper implementation study is needed to determine which, if either, provides best performance in practice.

6.6 Conclusions

In this paper, LFA is presented for block-structured relaxation schemes for stabilized and stable finite-element discretizations of the Stokes equations. The convergence and smoothing factors exhibited here provide optimized parameters for DWJ and BSR for the stabilized discretizations. The convergence of (inexact) BSR clearly outperforms multigrid with both DWJ and Uzawa relaxation. While the LFA smoothing factor loses its predictivity of the two-grid convergence factor for the stable $Q_2 - Q_1$ discretization and for Uzawa relaxation for both stabilized and stable discretizations, the two-grid LFA convergence factor can still provide useful predictions. We consider as well the inexact case for BSR, with Jacobi iterations or multigrid cycles used to approximate solution of the Schur complement system, as is suitable for use on modern parallel and graphics processing unit (GPU) architectures. From numerical experiments, we see that inexact BSR can be as good as the exact iteration for solving the Stokes equations, with the same choices of parameters and, hence, generally recommend this as the most efficient and robust of the approaches considered. The analysis and LFA predictions demonstrated here offer good insight into the use of block-structured relaxation for other types of saddle-point problems, which will be considered in future work.

Acknowledgements

The work of S.M. was partially supported by an NSERC discovery grant.

Bibliography

[1] J. Adler, T. R. Benson, E. Cyr, S. P. MacLachlan, and R. S. Tuminaro. Monolithic multigrid methods for two-dimensional resistive magnetohydrodynamics. *SIAM J. Sci. Comput.*, 38(1):B1–B24, 2016.

[2] 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.

[3] J. H. Adler, D. B. Emerson, S. P. MacLachlan, and T. A. Manteuffel. Constrained optimization for liquid crystal equilibria. *SIAM J. Sci. Comput.*, 38(1):B50–B76, 2016.

[4] C. Bacuta, P. S. Vassilevski, and S. Zhang. A new approach for solving Stokes systems arising from a distributive relaxation method. *Numerical Methods for Partial Differential Equations*, 27(4):898–914, 2011.

[5] D. Braess and W. Dahmen. A cascadic multigrid algorithm for the Stokes equations. *Numer. Math.*, 82(2):179–191, 1999.

[6] D. Braess and R. Sarazin. An efficient smoother for the Stokes problem. *Appl. Numer. Math.*, 23(1):3–19, 1997.

[7] A. Brandt. Rigorous quantitative analysis of multigrid, I. constant coefficients two-level cycle with L_2-norm. *SIAM Journal on Numerical Analysis*, 31(6):1695–1730, 1994.

[8] A. Brandt and N. Dinar. Multigrid solutions to elliptic flow problems. In *Numerical methods for partial differential equations*, volume 42 of *Publ. Math. Res. Center Univ. Wisconsin*, pages 53–147. Academic Press, New York-London, 1979.

[9] 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.

[10] F. Brezzi and J. Douglas. Stabilized mixed methods for the Stokes problem. *Numer. Math.*, 53(1):225–235, 1988.

[11] L. Chen. Multigrid methods for saddle point systems using constrained smoothers. *Comput. Math. Appl.*, 70(12):2854–2866, 2015.

[12] L. Chen, X. Hu, M. Wang, and J. Xu. A multigrid solver based on distributive smoother and residual overweighting for Oseen problems. *Numerical Mathematics: Theory, Methods and Applications*, 8(02):237–252, 2015.

[13] C. R. Dohrmann and P. B. Bochev. A stabilized finite element method for the Stokes problem based on polynomial pressure projections. *International Journal for Numerical Methods in Fluids*, 46(2):183–201, 2004.

[14] 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.

[15] S. Friedhoff, S. MacLachlan, and C. Borgers. Local Fourier analysis of space-time relaxation and multigrid schemes. *SIAM Journal on Scientific Computing*, 35(5):S250–S276, 2013.

[16] F. J. Gaspar, Y. Notay, C. W. Oosterlee, and C. Rodrigo. A simple and efficient segregated smoother for the discrete Stokes equations. *SIAM J. Sci. Comput.*, 36(3):A1187–A1206, 2014.

[17] B. Gmeiner, M. Huber, L. John, U. Rüde, and B. Wohlmuth. A quantitative performance study for Stokes solvers at the extreme scale. *J. Comput. Sci.*, 17(part 3):509–521, 2016.

[18] Y. He and S. P. MacLachlan. Local Fourier analysis of block-structured multigrid relaxation schemes for the Stokes equations. *Numerical Linear Algebra with Applications*, 25(3), 2018. e2147.

[19] Y. He and S. P. MacLachlan. Two-level Fourier analysis of multigrid for higher-order finite-element methods. *submitted*, 2018.

[20] L. John, U. Rüde, B. Wohlmuth, and W. Zulehner. On the analysis of block smoothers for saddle point problems. *arXiv preprint arXiv:1612.01333*, 2016.

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

[22] J.-F. Maitre, F. Musy, and P. Nigon. A fast solver for the Stokes equations using multigrid with a Uzawa smoother. In *Advances in multigrid methods (Oberwolfach, 1984)*, volume 11 of *Notes Numer. Fluid Mech.*, pages 77–83. Vieweg+Teubner Verlag, Wiesbaden, 1985.

[23] S. Manservisi. Numerical analysis of Vanka-type solvers for steady Stokes and Navier-Stokes flows. *SIAM J. Numer. Anal.*, 44(5):2025–2056, 2006.

[24] J. Molenaar. A two-grid analysis of the combination of mixed finite elements and Vanka-type relaxation. In *Multigrid methods, III (Bonn, 1990)*, volume 98 of *Internat. Ser. Numer. Math.*, pages 313–323. Birkhäuser, Basel, 1991.

[25] A. Niestegge and K. Witsch. Analysis of a multigrid Stokes solver. *Appl. Math. Comput.*, 35(3):291–303, 1990.

[26] M. A. Olshanskii. Multigrid analysis for the time dependent Stokes problem. *Math. Comp.*, 81(277):57–79, 2012.

[27] C. W. Oosterlee and F. J. Gaspar. Multigrid methods for the Stokes system. *Computing in Science & Engineering*, 8(6):34–43, 2006.

[28] C. Rodrigo, F. J. Gaspar, and F. J. Lisbona. On a local Fourier analysis for overlapping block smoothers on triangular grids. *Appl. Numer. Math.*, 105:96–111, 2016.

[29] C. Rodrigo, F. J. Gaspar, and L. T. Zikatanov. On the validity of the local Fourier analysis. *arXiv preprint arXiv:1710.00408*, 2017.

[30] J. W. Ruge and K. Stüben. Algebraic multigrid. *Multigrid methods*, 3(13):73–130, 1987.

[31] J. Schöberl and W. Zulehner. On Schwarz-type smoothers for saddle point problems. *Numer. Math.*, 95(2):377–399, 2003.

[32] D. Silvester and A. Wathen. Fast iterative solution of stabilised Stokes systems part II: using general block preconditioners. *SIAM J. Numer. Anal.*, 31(5):1352–1367, 1994.

[33] S. Sivaloganathan. The use of local mode analysis in the design and comparison of multigrid methods. *Computer Physics Communications*, 65(1-3):246–252, 1991.

[34] R. P. Stevenson. *On the validity of local mode analysis of multi-grid methods.* PhD thesis, 1990.

[35] S. Takacs. A robust multigrid method for the time-dependent Stokes problem. *SIAM J. Numer. Anal.*, 53(6):2634–2654, 2015.

[36] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid.* Academic Press, Inc., San Diego, CA, 2001. With contributions by A. Brandt, P. Oswald and K. Stüben.

[37] S. P. Vanka. Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. *J. Comput. Phys.*, 65(1):138–158, 1986.

[38] M. Wang and L. Chen. Multigrid methods for the Stokes equations using distributive Gauss-Seidel relaxations based on the least squares commutator. *J. Sci. Comput.*, 56(2):409–431, 2013.

[39] A. Wathen and D. Silvester. Fast iterative solution of stabilised Stokes systems. part I: Using simple diagonal preconditioners. *SIAM J. Numer. Anal.*, 30(3):630–649, 1993.

[40] A. J. Wathen and T. Rees. Chebyshev semi-iteration in preconditioning for problems including the mass matrix. *Electronic Transactions on Numerical Analysis*, 34(125-135):125–135, 2009.

[41] R. Wienands and W. Joppich. *Practical Fourier analysis for multigrid methods.* CRC press, 2004.

[42] G. Wittum. Multi-grid methods for Stokes and Navier-Stokes equations. *Numer. Math.*, 54(5):543–563, 1989.

[43] W. Zulehner. A class of smoothers for saddle point problems. *Computing*, 65(3):227–246, 2000.

Chapter 7

Local Fourier analysis of BDDC-like algorithms

Abstract

[1] Local Fourier analysis is a commonly used tool for the analysis of multigrid and other multilevel algorithms, providing both insight into observed convergence rates and predictive analysis of the performance of many algorithms. In this paper, we adapt local Fourier analysis to examine variants of two- and three-level BDDC algorithms, to better understand the eigenvalue distributions and condition number bounds on these preconditioned operators. This adaptation is based on a new choice of basis for the space of Fourier harmonics that greatly simplifies the application of local Fourier analysis in this setting. The local Fourier analysis is validated by considering the two dimensional Laplacian and predicting the condition numbers of the preconditioned operators with different sizes of subdomains. Several variants are analyzed, showing the two- and three-level performance of the "lumped" variant can be greatly improved when used in multiplicative combination with a weighted diagonal scaling preconditioner, with weight optimized through the use of LFA.

Keywords: BDDC, domain decomposition, local Fourier analysis, multiplicative methods

[1] Authors: J. Brown, Y. He and S. P. MacLachlan
This work is submitted as *Local Fourier analysis of BDDC-Like Algorithms*, to SIAM Journal on Scientific Computing, 2018.

AMS subject classification: 65N22, 65N55, 65F08

7.1 Introduction

Domain decomposition methods are well-studied approaches for the numerical solution of partial differential equations both experimentally and theoretically [1, 10, 12, 27], due to their efficiency and robustness for many large-scale problems, and the need for parallel algorithms. Among the main families of domain decomposition algorithms are Neumann-Neumann [27], FETI [13], Schwarz [12, 27], and Optimized Schwarz [10, 15]. Balancing domain decomposition by constraints (BDDC) is one family of non-overlapping domain decomposition method. While BDDC was first introduced by Dohrmann in [6], several variants have recently been proposed. BDDC-like methods have been successfully applied to many PDEs, including elliptic problems [18, 22], the incompressible Stokes equations [17, 19], H(curl) problems [9], flow in porous media [29], and the incompressible elasticity problem [7, 8]. Theoretical analysis of BDDC has primarily been based on finite-element approximation theory [4, 7, 11, 23, 24]. It has been shown that the condition number of the preconditioned BDDC operator can be bounded by a function of $\frac{H}{h}$(where h is the meshsize, and H is the subdomain size), independent of the number of subdomains [29]. A nonoverlapping domain decomposition method for discontinuous Galerkin based on the BDDC algorithm is presented in [3], and the condition number of the preconditioned system is shown to be bounded by similar estimates as those for conforming finite element methods. BDDC methods in three- or multilevel forms have also been developed [25, 30, 31].

Since BDDC algorithms are widely used to solve many problems with high efficiency and parallelism, better understanding of how this methodology works is useful in the design of new algorithms. Local Fourier analysis (LFA), first introduced by Brandt [2] and well-studied for multigrid methods [5, 26, 28, 32, 33], is an analysis framework that provides predictive performance estimates for many multilevel iterations and preconditioners. However, to our knowledge, there has been no research applying local Fourier analysis to BDDC-like algorithms. The same is true of the closely related finite element tearing and interconnect (FETI) methodology [13, 14, 16]. Because LFA can reflect both the distribution of eigenvalues and associated eigenvectors of a preconditioned operator, here, we adopt LFA to analyze variants of the common "lumped" and "Dirichlet" BDDC algorithms, based on [20], to guide construction of these methods. To do this, we introduce a novel basis for the Fourier analysis that is well-suited for application to domain decomposition preconditioners.

Applying the two-level BDDC algorithm requires the solution of a Schur complement equation (coarse problem), which usually poses some difficulty with increasing problem size. Two- and three-level variants are, thus, considered in this paper. However, as is well-known in the literature, the performance of BDDC degrades sharply from two-level to three-level methods, particularly for large values of H/h. Since our analysis shows that the largest eigenvalues of the preconditioned operator for the lumped BDDC algorithm are associated with oscillatory modes, we propose variants of BDDC based on multiplicative preconditioning and multigrid ideas. From the condition numbers offered by LFA, we can easily compare the efficiency of these variants. Furthermore, LFA can provide optimal parameters for these multiplicative methods, helping tune and understand sensitivity to the parameter choice.

This paper is organized as follows. In Section 7.2, we introduce the finite element discretization of the Laplace problem in two dimensions and the lumped and Dirichlet preconditioners. Two- and three-level preconditioned operators are developed in Section 7.3. In Section 7.4, we discuss the Fourier representation of the preconditioned operators. Section 7.5 reports LFA-predicted condition numbers of the BDDC variants considered here. Conclusions are presented in Section 7.6.

7.2 Discretization

We consider the two-dimensional Laplace problem in weak form: Find $u \in H_0^1(\Omega) := V$ such that

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \, d\Omega = \langle f, v \rangle, \forall v \in V, \tag{7.1}$$

where $\Omega \subset \mathbb{R}^2$ is a bounded domain with Lipschitz boundary $\partial\Omega$. Here, we consider the Ritz-Galerkin approximation over V_h, the space of piecewise bilinear functions on a uniform rectangular mesh of $\Omega = [0,1]^2$. The corresponding linear system of equations is given as

$$Ax = b. \tag{7.2}$$

We partition the domain, Ω, into N nonoverlapping subdomains, $\Omega_i, i = 1, 2, \cdots, N$, where each subdomain is a union of shape regular elements and the nodes on the boundaries of neighboring subdomains match across the interface $\Gamma = \bigcup \partial\Omega_i \backslash \partial\Omega$. The interface of subdomain Ω_i is defined by $\Gamma_i = \partial\Omega_i \bigcap \Gamma$. Here, we consider $\Omega = [0, 1]^2$, with both a discretization mesh (with meshsize h) and subdomain mesh (with meshsize $H = ph$) given by uniform grids with square elements or subdomains.

The finite-element space V_h can be rewritten as $V_h = V_{I,h} \bigoplus V_{\Gamma,h}$, where $V_{I,h}$ is the sum of the subdomain interior variable spaces $V_{I,h}^{(i)}$. Functions in $V_{I,h}^{(i)}$ are supported in the subdomain Ω_i and vanish on the subdomain interface Γ_i. $V_{\Gamma,h}$ is the space of traces on Γ of functions in V_h. Then, we can write the subdomain problem with Neumann boundary conditions on Γ_i as

$$A^{(i)} x^{(i)} = \begin{pmatrix} A_{II}^{(i)} & A_{\Gamma I}^{(i)^T} \\ A_{\Gamma I}^{(i)} & A_{\Gamma\Gamma}^{(i)} \end{pmatrix} \begin{pmatrix} x_I^{(i)} \\ x_\Gamma^{(i)} \end{pmatrix} = \begin{pmatrix} b_I^{(i)} \\ b_\Gamma^{(i)} \end{pmatrix}, \tag{7.3}$$

where $x^{(i)} = (x_I^{(i)}, x_\Gamma^{(i)}) \in V_h^{(i)} = (V_{I,h}^{(i)}, V_{\Gamma,h}^{(i)})$, and T denotes the conjugate transpose. Then, the global problem (7.2) can be assembled from the subdomain problems (7.3) as

$$A = \sum_{i=1}^{N} R^{(i)^T} A^{(i)} R^{(i)}, \text{ and } b = \sum_{i=1}^{N} R^{(i)^T} b^{(i)},$$

where $R^{(i)}$ is the restriction operator from a global vector to a subdomain vector on Ω_i.

7.2.1 A partially subassembled problem

In order to describe variants of the BDDC methods, we first introduce a partially subassembled problem, following [20], and the corresponding space of partially subassembled variables,

$$\hat{V}_h = V_{\Pi,h} \bigoplus V_{r,h}, \tag{7.4}$$

where $V_{\Pi,h}$ is spanned by the subdomain vertex nodal basis functions (the coarse degrees of freedom). The complementary space, $V_{r,h}$, is the sum of the subdomain spaces $V_{r,h}^{(i)}$, which correspond to the subdomain interior and interface degrees of freedom and are spanned by the basis functions which vanish at the coarse-grid degrees of freedom. For a 4 × 4 mesh, the degrees of freedom in $V_{\Pi,h}$ are those corresponding to the circled nodes at the left of Figure 7.1, while the degrees of freedom in $V_{r,h}$ correspond to all interior nodes, plus duplicated (broken) degrees freedom along subdomain boundaries.

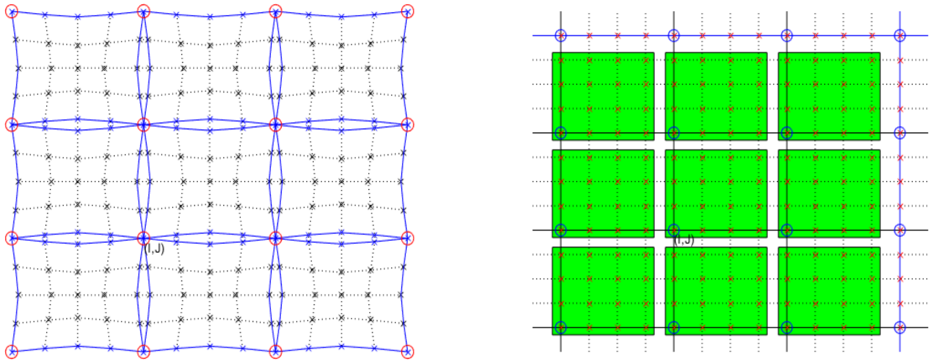

Figure 7.1: At left, the partially broken decomposition given in Equation (7.4), with circled degrees of freedom corresponding to $V_{\Pi,h}$ and all others corresponding to $V_{r,h}$. This matches the periodic array of subdomains induced by the subsets $\mathfrak{S}^*_{I,J}$ introduced in Equation (7.23) for $p = 4$. At right, a non-overlapping decomposition into subdomains of size $p \times p$ for $p = 4$, corresponding to the subsets $\mathfrak{S}_{I,J}$ introduced in Equation (7.19), where LFA works on an infinite grid and characterizes operators by their action in terms of the non-overlapping partition denoted in green.

The partially subassembled problem matrix, corresponding to the variables in the space $\hat{V}_h$, is obtained by assembling the subdomain matrices (7.3) only with respect to the coarse-level variables; that is,

$$\hat{A} = \sum_{i=1}^{N} (\bar{R}^{(i)})^T A^{(i)} \bar{R}^{(i)}, \tag{7.5}$$

where $\bar{R}^{(i)}$ is a restriction from space $\hat{V}_h$ to $V_h^{(i)}$.

7.2.2 Lumped and Dirichlet preconditioners

In order to define the preconditioners under consideration for (7.2), we introduce a positive scaling factor, $\delta_i(\boldsymbol{x})$, for each node $\boldsymbol{x}$ on the interface Γ_i of subdomain Ω_i. Let $\mathcal{N}_{\boldsymbol{x}}$ be the set of indices of the subdomains that have $\boldsymbol{x}$ on their boundaries. Define $\delta_i(\boldsymbol{x}) = 1/|\mathcal{N}_{\boldsymbol{x}}|$, where $|\mathcal{N}_{\boldsymbol{x}}|$ is the cardinality of $\mathcal{N}_{\boldsymbol{x}}$. The scaled injection operator, $\mathcal{R}_1$, is defined so that each column of $\mathcal{R}_1$ corresponds to a degree of freedom of the global problem (7.2). For subdomain interior and coarse-level variables, the corresponding column of $\mathcal{R}_1$ has a single entry with value 1. Columns that correspond to an interface degree of freedom $\boldsymbol{x} \in \Gamma_{i,h}$ (the set of nodes in Γ_i) have $|\mathcal{N}_{\boldsymbol{x}}|$ non-zero entries each of $\delta_i(\boldsymbol{x})$.

Based on the partially subassembled problem, the first preconditioner introduced for solving (7.2) is

$$M_1^{-1} = \mathcal{R}_1^T \hat{A}^{-1} \mathcal{R}_1.$$

The preconditioned operator $M_1^{-1}A$ has the same eigenvalues as the preconditioned FETI-DP operator with a lumped preconditioner, except for some eigenvalues equal to 0 and 1 [14, 20]. We refer to M_1 as the *lumped* preconditioner.

A similar preconditioner for A augments this using discrete harmonic extensions in the restriction and interpolation operators [20], giving

$$M_2^{-1} = (\mathcal{R}_1^T - \mathcal{H} J_D)\hat{A}^{-1} \underbrace{(\mathcal{R}_1 - J_D^T \mathcal{H}^T)}_{:=\mathcal{R}_2}, \tag{7.6}$$

where $\mathcal{H}$ is the direct sum of $\mathcal{H}^{(i)} = -(A_{II}^{(i)})^{-1}(A_{\Gamma I}^{(i)})^T$, which maps the jump over a subdomain interface (given by J_D) to the interior of the subdomain by solving a local Dirichlet problem, and gives zero for other values. For any given $v \in \hat{V}_h$, the component of $J_D^T v$ on subdomain Ω_i is given by

$$\left(J_D^T v(\boldsymbol{x})\right)^{(i)} = \sum_{j \in \mathcal{N}_{\boldsymbol{x}}} \left(\delta_j(\boldsymbol{x}) v^{(i)}(\boldsymbol{x}) - \delta_i(\boldsymbol{x}) v^{(j)}(\boldsymbol{x})\right), \ \forall \boldsymbol{x} \in \Gamma_{i,h}. \tag{7.7}$$

Extending the interface values using the discrete harmonic extension minimizes the energy norm of the resulting vector [27], giving a better stability bound. Furthermore, the preconditioned operator $M_2^{-1}A$ has the same eigenvalues as the BDDC operator [18], except for some eigenvalues equal to 1 [20]. We refer to M_2 as the *Dirichlet* preconditioner.

Standard bounds (see, e.g., [20]) on the condition numbers of the preconditioned operators are that, for $M_1^{-1}A$, there exists $\mathfrak{C}_{1,0} \geq 0$ such that $\kappa \leq \mathfrak{C}_{1,0}\frac{H}{h}(1 + \log\frac{H}{h})$ and, for $M_2^{-1}A$, there exists $\mathfrak{C}_{2,0} \geq 0$ such that $\kappa \leq \mathfrak{C}_{2,0}(1 + \log\frac{H}{h})^2$.

7.3 Two- and three-level variants

In both of the above preconditioned operators, we need to solve the following partially subassembled problem, now written in block form

$$\hat{A}\hat{x} = \begin{pmatrix} A_{rr} & \hat{A}_{\Pi r}^T \\ \hat{A}_{\Pi r} & A_{\Pi\Pi} \end{pmatrix} \begin{pmatrix} \hat{x}_r \\ \hat{x}_\Pi \end{pmatrix} = \begin{pmatrix} A_{rr} & 0 \\ \hat{A}_{\Pi r} & \hat{S}_\Pi \end{pmatrix} \begin{pmatrix} I & A_{rr}^{-1}\hat{A}_{\Pi r}^T \\ 0 & I \end{pmatrix} \begin{pmatrix} \hat{x}_r \\ \hat{x}_\Pi \end{pmatrix} = \begin{pmatrix} \hat{d}_r \\ \hat{d}_\Pi \end{pmatrix} = \hat{d}, \tag{7.8}$$

where $\hat{S}_\Pi = A_{\Pi\Pi} - \hat{A}_{\Pi r}A_{rr}^{-1}\hat{A}_{\Pi r}^T$, $\hat{x}_r$ contains the subdomain interior and interface degrees of freedom, and $\hat{x}_\Pi$ corresponds to the coarse-level degrees of freedom, which are located at the corners of the subdomains. We write $\hat{A}$ in (7.8) in factorization form to easily separate the action on the coarse degrees of freedom, and to find the corresponding symbol of $\hat{A}^{-1}$. If we define

$$P = \begin{pmatrix} -A_{rr}^{-1}\hat{A}_{\Pi r}^T \\ I \end{pmatrix},$$

then the Schur complement is the Galerkin coarse operator, $\hat{S}_\Pi = P^T\hat{A}P$, and block-factorization solve for $\hat{A}$ can be seen to be equivalent to a two-level additive multigrid method with exact F-relaxation using

$$S_F = \begin{pmatrix} A_{rr}^{-1} & 0 \\ 0 & 0 \end{pmatrix}.$$

In the partially subassembled problem (7.8), we need to solve a coarse problem related to $\hat{S}_\Pi$. We can either solve this coarse problem exactly (corresponding to a two-level method, where the Schur complement is inverted exactly) or inexactly (as a three-level method), where the lumped and Dirichlet preconditioners defined above are used recursively to solve this problem.

7.3.1 Exact and inexact solve for the Schur complement

Let

$$\mathcal{K}_1 = \begin{pmatrix} A_{rr} & 0 \\ \hat{A}_{\Pi r} & \hat{S}_\Pi \end{pmatrix}, \ \mathcal{K}_2 = \begin{pmatrix} I & A_{rr}^{-1}\hat{A}_{\Pi r}^T \\ 0 & I \end{pmatrix},$$

and note that the product of $\mathcal{K}_1$ and $\mathcal{K}_2$ is $\hat{A}$. For $i = 1, 2, j = 0, 1, 2$, let $G_{i,j}$ denote the preconditioned operators for two- and three-level variants of BDDC, where i and j denote using M_i and $M_{s,j}$ (with $M_{s,0} := \hat{S}_\Pi$) as preconditioners for the fine and coarse problems, respectively, where $M_{s,j}^{-1}$ stands for applying the preconditioner M_j to the Schur complement problem. By standard calculation, we can write

$$G_{i,j} = \mathcal{R}_i^T \mathcal{K}_2^{-1} \mathcal{P}_j \mathcal{K}_1^{-1} \mathcal{R}_i A,$$

with

$$\mathcal{P}_j = \begin{pmatrix} I & 0 \\ 0 & M_{s,j}^{-1} \hat{S}_\Pi \end{pmatrix}.$$

Remark 7.3.1. *When $j = 0$, $G_{i,j}$ is a two-level method, solving the Schur complement problem exactly, as $\mathcal{P}_0 \equiv I$. Note that, for the three-level variants ($j = 1, 2$),*

$$\mathcal{P}_j \mathcal{K}_1^{-1} = \begin{pmatrix} I & 0 \\ 0 & M_{s,j}^{-1} \hat{S}_\Pi \end{pmatrix} \begin{pmatrix} A_{rr}^{-1} & 0 \\ -\hat{S}_\Pi^{-1} \hat{A}_{\Pi r} A_{rr}^{-1} & \hat{S}_\Pi^{-1} \end{pmatrix} = \begin{pmatrix} A_{rr}^{-1} & 0 \\ -M_{s,j}^{-1} \hat{A}_{\Pi r} A_{rr}^{-1} & M_{s,j}^{-1} \end{pmatrix}.$$

Thus, $G_{i,j}$ can be applied without directly applying the inverse of $\hat{S}_\Pi$.

Standard bounds (see, e.g., [31]) on the condition numbers of the three-level preconditioned operators are that there exists $\mathfrak{C}_{i,j}$ such that $\kappa(G_{i,j}) \leq \mathfrak{C}_{i,j} \Upsilon_i \Upsilon_j$, where $\Upsilon_1 = \frac{H}{h}(1 + \log \frac{H}{h})$ and $\Upsilon_2 = (1 + \log \frac{H}{h})^2$.

7.3.2 Multiplicative preconditioners

As we shall see, the bounds above are relatively sharp and the performance of both preconditioners degrades with subdomain size and number of levels. To attempt to counteract this, we consider multiplicative combinations of these preconditioners with a simple diagonal scaling operator, mimicking the use of weighted Jacobi relaxation in classical multigrid methods. We use $G_{i,j}^f$ to denote the multiplicative preconditioned operator based on $G_{i,j}$ with diagonal scaling on the fine level. Here,

$$G_{i,j}^f = G_{i,j} + \omega D^{-1} A (I - G_{i,j}), \; i = 1, 2, \; j = 0, 1, 2, \tag{7.9}$$

where D is the diagonal of A and ω is a chosen relaxation parameter.

Another variant is the use of multiplicative preconditioning on the coarse level with a similar diagonal scaling. We use $G^c_{i,j}$ to denote the resulting multiplicative preconditioner. Here,

$$G^c_{i,j} = \mathcal{R}_i^T \mathcal{K}_2^{-1} \mathcal{P}_j^c \mathcal{K}_1^{-1} \mathcal{R}_i A, \ i, j = 1, 2, \tag{7.10}$$

where

$$\mathcal{P}_j^c = \begin{pmatrix} I & 0 \\ 0 & G_{c,j} \end{pmatrix},$$

in which

$$G_{c,j} = M_{s,j}^{-1} \hat{S}_\Pi + \omega D_s^{-1} \hat{S}_\Pi (I - M_{s,j}^{-1} \hat{S}_\Pi),$$

where D_s is the diagonal of $\hat{S}_\Pi$.

Instead of using a single sweep of Jacobi in $G_{c,j}$, we can consider a symmetrized Jacobi operator $G^s_{c,j}$, where $I - G^s_{c,j} = (I - \omega_1 D_s^{-1} \hat{S}_\Pi)(I - M_{s,j}^{-1} \hat{S}_\Pi)(I - \omega_2 D_s^{-1} \hat{S}_\Pi)$; that is,

$$G^s_{c,j} = G_{c,j} + \omega_2 (I - G_{c,j}) D_s^{-1} \hat{S}_\Pi,$$

then $G^c_{i,j}$ changes to

$$G^{s,c}_{i,j} = \mathcal{R}_i^T \mathcal{K}_2^{-1} \mathcal{P}_j^{s,c} \mathcal{K}_1^{-1} \mathcal{R}_i A, \ i, j = 1, 2. \tag{7.11}$$

When $\omega_1 = \omega_2$, $G^{s,c}_{i,j}$ is a symmetric preconditioner for A, although we note that our LFA predicts a positive real spectrum for the nonsymmetric forms, $G^f_{i,j}$ and $G^c_{i,j}$, as well.

Finally, we can also apply the multiplicative operators based on diagonal scaling on both the fine and coarse levels. We denote this as

$$G^{f,c}_{i,j} = G^c_{i,j} + \omega_2 D^{-1} A (I - G^c_{i,j}), \ i = 1, 2, \ j = 1, 2, \tag{7.12}$$

where D is the diagonal of A and ω_2 is a chosen relaxation parameter.

In the following, we focus on analyzing the spectral properties of the above preconditioned operators by local Fourier analysis [28]. The main focus of this work is on the operators $\mathcal{K}_1, \mathcal{K}_2$, and $\mathcal{P}_j$, because the Fourier representations of other operators are just combinations of these three and some simple additional terms.

7.4 Local Fourier analysis

To apply LFA to the BDDC-like methods proposed here, we first review some terminology of classical LFA. We consider a two-dimensional infinite uniform grid, $\mathbf{G}_h$, with

$$\mathbf{G}_h = \big\{ \boldsymbol{x}_{i,j} := (x_i, x_j) = (ih, jh), (i,j) \in \mathbb{Z}^2 \big\}, \tag{7.13}$$

and Fourier functions $\psi(\boldsymbol{\theta}, \boldsymbol{x}_{i,j}) = e^{\iota \boldsymbol{\theta} \cdot \boldsymbol{x}_{i,j}/h}$ on $\mathbf{G}_h$, where $\iota^2 = -1$ and $\boldsymbol{\theta} = (\theta_1, \theta_2)$. Let L_h be a Toeplitz operator acting on $l^2(\mathbf{G}_h)$ as

$$L_h \overset{\wedge}{=} [s_{\boldsymbol{\kappa}}]_h \ (\boldsymbol{\kappa} = (\kappa_1, \kappa_2) \in \mathbb{Z}^2); \ L_h w_h(\boldsymbol{x}) = \sum_{\boldsymbol{\kappa} \in \boldsymbol{V}} s_{\boldsymbol{\kappa}} w_h(\boldsymbol{x} + \boldsymbol{\kappa} h),$$

with constant coefficients $s_{\boldsymbol{\kappa}} \in \mathbb{R}$ (or $\mathbb{C}$), where $w_h(\boldsymbol{x})$ is a function in $l^2(\mathbf{G}_h)$. Here, $\boldsymbol{V}$ is taken to be a finite index set. Note that since L_h is Toeplitz, it is diagonalized by the Fourier modes $\psi(\boldsymbol{\theta}, \boldsymbol{x})$.

Definition 7.4.1. *We call* $\widetilde{L}_h(\boldsymbol{\theta}) = \sum_{\boldsymbol{\kappa} \in \boldsymbol{V}} s_{\boldsymbol{\kappa}} e^{\iota \boldsymbol{\theta} \boldsymbol{\kappa}}$ *the symbol of* L_h.

Note that for all grid functions, $\psi(\boldsymbol{\theta}, \boldsymbol{x})$,

$$L_h \psi(\boldsymbol{\theta}, \boldsymbol{x}) = \widetilde{L}_h(\boldsymbol{\theta}) \psi(\boldsymbol{\theta}, \boldsymbol{x}).$$

Remark 7.4.1. *In Definition 7.4.1, the operator* L_h *acts on a single function on* $\mathbf{G}_h$*, so* $\widetilde{L}_h$ *is a scalar. For an operator mapping vectors on* $\mathbf{G}_h$ *to vectors on* $\mathbf{G}_h$*, the symbol will be extended to be a matrix.*

7.4.1 Change of Fourier basis

Here, we discuss domain decomposition methods. While the classical basis set for LFA, denoted $\mathbf{E}_h$ below, could be used, we find it is substantially more convenient to make use of a transformed "sparse" basis, introduced here as $\mathbf{F}_H$. This basis allows a natural expression of the periodic structures in domain decomposition preconditioners. We treat each subdomain problem as one macroelement patch, and each subdomain block in the global problem is diagonalized by a coupled set of Fourier modes introduced in the following. Because each subdomain has the same size, $p \times p$, we consider the high and low frequencies for coarsening by factor p, given by

$$\boldsymbol{\theta} \in T^{\text{low}} = \left[-\frac{\pi}{p}, \frac{\pi}{p}\right)^2, \boldsymbol{\theta} \in T^{\text{high}} = \left[-\frac{\pi}{p}, \frac{(2p-1)\pi}{p}\right)^2 \Big\backslash \left[-\frac{\pi}{p}, \frac{\pi}{p}\right)^2.$$

Let $\boldsymbol{\theta}^{(q,r)} = (\theta_1^{(q)}, \theta_2^{(r)})$, where $\theta_1^{(q)} = \theta_1^{(0)} + \frac{2\pi q}{p}$ and $\theta_2^{(r)} = \theta_2^{(0)} + \frac{2\pi r}{p}$ for $0 \leq q, r < p$. For any given $\boldsymbol{\theta}^{(0,0)} \in T^{\text{low}}$, we define the p^2-dimensional space

$$\mathbf{E}_h(\boldsymbol{\theta}^{(0,0)}) := \text{span}\{\psi(\boldsymbol{\theta}^{(q,r)}, \boldsymbol{x}_{s,t}) = e^{\iota\boldsymbol{\theta}^{(q,r)}\cdot\boldsymbol{x}_{s,t}/h} : q, r = 0, 1, \cdots, p-1\}, \tag{7.14}$$

as the classical space of Fourier harmonics for factor p coarsening.

For any $\boldsymbol{x}_{s,t} \in \mathbf{G}_h$, we consider a grid function defined as a linear combination of the p^2 basis functions for $\mathbf{E}_h(\boldsymbol{\theta}^{(0,0)})$ with frequencies $\{\boldsymbol{\theta}^{(q,r)}\}_{q,r=0}^{p-1}$ and coefficients $\{\beta_{q,r}\}_{q,r=0}^{p-1}$ as

$$e_{s,t} := \sum_{q,r=0}^{p-1} \beta_{q,r}\psi(\boldsymbol{\theta}^{(q,r)}, \boldsymbol{x}_{s,t}).$$

We note that any index (s,t) has a unique representation as $(pm+k, pn+\ell)$ where $(m,n) \in \mathbb{Z}^2$ and $k, \ell \in \{0, 1, \cdots, p-1\}$. From (7.14), we have

$$\begin{aligned} e_{pm+k,pn+\ell} &= \sum_{q,r=0}^{p-1} \beta_{q,r} e^{\iota(\theta_1^{(0)}+\frac{2\pi q}{p})x_s/h} e^{\iota(\theta_2^{(0)}+\frac{2\pi r}{p})x_t/h} \\ &= \sum_{q,r=0}^{p-1} \beta_{q,r} e^{\iota\theta_1^{(0)}x_s/h} e^{\iota\frac{2\pi q(pm+k)}{p}} e^{\iota\theta_2^{(0)}x_t/h} e^{\frac{2\pi r(pn+\ell)}{p}} \\ &= \sum_{q,r=0}^{p-1} \beta_{q,r} e^{\iota\frac{2\pi qk}{p}} e^{\iota\theta_1^{(0)}x_s/h} e^{\frac{2\pi r\ell}{p}} e^{\iota\theta_2^{(0)}x_t/h} \\ &= \left(\sum_{q,r=0}^{p-1} \beta_{q,r} e^{\iota\frac{2\pi qk}{p}} e^{\frac{2\pi r\ell}{p}}\right)\left(e^{\iota\boldsymbol{\theta}^{(0,0)}\cdot\boldsymbol{x}_{s,t}/h}\right). \end{aligned}$$

Thus, we can write

$$e_{pm+k,pn+\ell} = \hat{\beta}_{k,\ell} e^{\iota\boldsymbol{\theta}\cdot\boldsymbol{x}_{s,t}/H}, \tag{7.15}$$

with

$$\boldsymbol{\theta} = p\boldsymbol{\theta}^{(0,0)}, \text{ and } \hat{\beta}_{k,\ell} = \sum_{q,r=0}^{p-1} \beta_{q,r} e^{\iota\frac{2\pi qk}{p}} e^{\frac{2\pi r\ell}{p}}. \tag{7.16}$$

Thus, for any point (s,t) with $\mathrm{mod}(s,p)=k$ and $\mathrm{mod}(t,p)=\ell$, $e_{s,t}$ can be reconstructed from a single Fourier mode with coefficient $\hat{\beta}_{k,\ell}$. Thus, on the mesh $\mathbf{G}_h$ defined in (7.13), the periodicity of the basis functions in $\mathbf{E}_h(\boldsymbol{\theta}^{(0,0)})$ can also be represented by a pointwise basis on each $p\times p$-block.

Based on (7.15), we consider a "sparse" p^2-dimensional space as follows

$$\mathbf{F}_H(\boldsymbol{\theta}) := \mathrm{span}\{\varphi_{k,\ell}(\boldsymbol{\theta},\boldsymbol{x}_{s,t}) = e^{\iota\boldsymbol{\theta}\cdot\boldsymbol{x}_{s,t}/H}\chi_{k,\ell}(\boldsymbol{x}_{s,t}) : k,\ell=0,1,\cdots,p-1\}, \qquad (7.17)$$

where $\boldsymbol{\theta}\in[-\pi,\pi)$ and

$$\chi_{k,\ell}(\boldsymbol{x}_{s,t}) = \begin{cases} 1, & \text{if } \mathrm{mod}(s,p)=k, \text{ and } \mathrm{mod}(t,p)=\ell, \\ 0, & \text{otherwise.} \end{cases}$$

Note that, with this notation, (7.15) can be rewritten as

$$e_{pm+k,pn+\ell} = \hat{\beta}_{k,\ell}\varphi_{k,\ell}(\boldsymbol{\theta},\boldsymbol{x}_{s,t}). \qquad (7.18)$$

Theorem 7.4.1. *$\mathbf{E}_h(\boldsymbol{\theta}^{(0,0)})$ and $\mathbf{F}_H(p\boldsymbol{\theta}^{(0,0)})$ are the same.*

Proof. While the derivation above shows directly that $\mathbf{E}_h(\boldsymbol{\theta}^{(0,0)})\subset\mathbf{F}_H(p\boldsymbol{\theta}^{(0,0)})$, we revisit this calculation now to show that the mapping $\{\beta_{q,r}\}\rightarrow\{\hat{\beta}_{k,\ell}\}$ is invertible and, hence, $\mathbf{F}_H(p\boldsymbol{\theta}^{(0,0)})\subset\mathbf{E}_h(\boldsymbol{\theta}^{(0,0)})$ as well.

Let $\mathcal{X}$ be an arbitrary vector with size $p^2\times 1$, denoted as

$$\mathcal{X} \;=\; \begin{pmatrix}\mathcal{X}_0 & \mathcal{X}_1 & \cdots & \mathcal{X}_{p-2} & \mathcal{X}_{p-1}\end{pmatrix}^T,$$

where

$$\mathcal{X}_r \;=\; \begin{pmatrix}\beta_{0,r} & \beta_{1,r} & \cdots & \beta_{p-2,r} & \beta_{p-1,r}\end{pmatrix}, \quad r=0,1,\cdots,p-1.$$

Then, we define a $p^2\times 1$ vector, $\hat{\mathcal{X}}$, based on (7.16), as follows

$$\hat{\mathcal{X}} \;=\; \begin{pmatrix}\hat{\mathcal{X}}_0 & \hat{\mathcal{X}}_1 & \cdots & \hat{\mathcal{X}}_{p-2} & \hat{\mathcal{X}}_{p-1}\end{pmatrix}^T,$$

where

$$\hat{\mathcal{X}}_\ell \;=\; \left(\hat{\beta}_{0,\ell} \;\; \hat{\beta}_{1,\ell} \;\; \cdots \;\; \hat{\beta}_{p-2,\ell} \;\; \hat{\beta}_{p-1,\ell}\right), \quad \ell = 0, 1, \cdots, p-1,$$

in which

$$\hat{\beta}_{k,\ell} = \sum_{r=0}^{p-1} \Big(\sum_{q=0}^{p-1} \beta_{q,r} e^{\iota\frac{2\pi qk}{p}} \Big) e^{\frac{2\pi r\ell}{p}}, \quad q, r = 0, 1, \cdots, p-1.$$

Let $\mathcal{T}$ be the matrix of this transformation, $\hat{\mathcal{X}} = \mathcal{T}\mathcal{X}$, and

$$\mathcal{T}_1 = \begin{pmatrix} (e^{\iota\frac{2\pi}{p}0})^0 & (e^{\iota\frac{2\pi}{p}1})^0 & (e^{\iota\frac{2\pi}{p}2})^0 & \cdots & (e^{\iota\frac{2\pi}{p}(p-1)})^0 \\ (e^{\iota\frac{2\pi}{p}0})^1 & (e^{\iota\frac{2\pi}{p}1})^1 & (e^{\iota\frac{2\pi}{p}2})^1 & \cdots & (e^{\iota\frac{2\pi}{p}(p-1)})^1 \\ (e^{\iota\frac{2\pi}{p}0})^2 & (e^{\iota\frac{2\pi}{p}1})^2 & (e^{\iota\frac{2\pi}{p}2})^2 & \cdots & (e^{\iota\frac{2\pi}{p}(p-1)})^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (e^{\iota\frac{2\pi}{p}0})^{p-2} & (e^{\iota\frac{2\pi}{p}1})^{p-2} & (e^{\iota\frac{2\pi}{p}2})^{p-2} & \cdots & (e^{\iota\frac{2\pi}{p}(p-1)})^{p-2} \\ (e^{\iota\frac{2\pi}{p}0})^{p-1} & (e^{\iota\frac{2\pi}{p}1})^{p-1} & (e^{\iota\frac{2\pi}{p}2})^{p-1} & \cdots & (e^{\iota\frac{2\pi}{p}(p-1)})^{p-1} \end{pmatrix}.$$

Note that $\mathcal{T}_1\mathcal{X}_r$ defines a vector whose $(k+1)$-th entry is $\displaystyle\sum_{q=0}^{p-1} \beta_{q,r} e^{2\pi qk/p}$ and, thus, we see that $\mathcal{T} = \mathcal{T}_1 \otimes \mathcal{T}_1$.

Note that $\mathcal{T}_1$ is a $p \times p$ Vandermonde matrix based on values $d_k = e^{\iota\frac{2\pi k}{p}}$, where $k = 0, 1, 2, \cdots, p-1$. It is obvious that $d_j \neq d_k$ if $j \neq k$. Consequently, $\det(\mathcal{T}_1) \neq 0$. Thus, $\mathcal{T}_1$ is invertible, and so is $\mathcal{T}$. It follows that $\mathbf{E}_h(\boldsymbol{\theta}^{(0,0)})$ and $\mathbf{F}_H(p\boldsymbol{\theta}^{(0,0)})$ are equivalent. □

Remark 7.4.2. *Let $z = e^{\iota 2\pi/p}$, be the primitive p-th root of unity, and note that $(\mathcal{T}_1)_{i,j} = z^{(j-1)(i-1)}$. Thus, $\widetilde{\mathcal{T}}_1 = \frac{1}{\sqrt{p}}\mathcal{T}_1$ is the unitary discrete Fourier transform (DFT) matrix with $\widetilde{\mathcal{T}}_1^{-1} = \widetilde{\mathcal{T}}_1^{T}$, where T denotes the conjugate transpose. Thus, $\mathcal{T}_1^{-1} = \frac{1}{p}\mathcal{T}_1^T$. Similarly, $\mathcal{T}$ is a scaled version of the two-dimensional unitary Fourier transform matrix, and $\mathcal{T}^{-1} = \frac{1}{p^2}\mathcal{T}^T$.*

In the rest of this paper, we use the basis of $\mathbf{F}_H$ as the foundation for local Fourier analysis on the $p \times p$ periodic structures of the BDDC operators. The "sparse" (or "pointwise") nature of the basis in $\mathbf{F}_H$ allows a natural expression of the operators in BDDC and, as such, is more convenient than the equivalent "global" basis in $\mathbf{E}_h$.

Note that the presentation above assumes that the original Fourier space, $\mathbf{E}_h$, is considered with harmonic frequencies in domain $[-\frac{\pi}{p}, \frac{(2p-1)\pi)}{p})^2$, and the sparse basis in $\mathbf{F}_H$ considers a single mode, $\boldsymbol{\theta} \in [-\pi, \pi)^2$. In both cases, it is clear that any frequency set covering an interval of length 2π in both x and y components can be used instead.

7.4.2 Representation of the original problem

On $\mathbf{G}_h$, we call each node, (I, J), where $\mathrm{mod}(I, p) = 0$ and $\mathrm{mod}(J, p) = 0$ a coarse-level point index. We construct a collective grid set associated with (I, J) for each subdomain as

$$\mathfrak{S}_{I,J} = \left\{ \boldsymbol{x}_{(I+k,J+\ell)} : k, \ell = 0, 1, \cdots, p-1 \right\}. \tag{7.19}$$

The degrees of freedom in A can be divided into subsets, $\mathfrak{S}_{I,J}$, whose union provides a disjoint cover for the set of degrees of freedom on the infinite mesh $\mathbf{G}_h$. Throughout the rest of this paper, the index (I, J) corresponds to the coarse point at the lower-left corner of the subdomain under consideration, unless stated otherwise. The left of Figure 7.2 shows the meshpoints for this decomposition for $p = 4$.

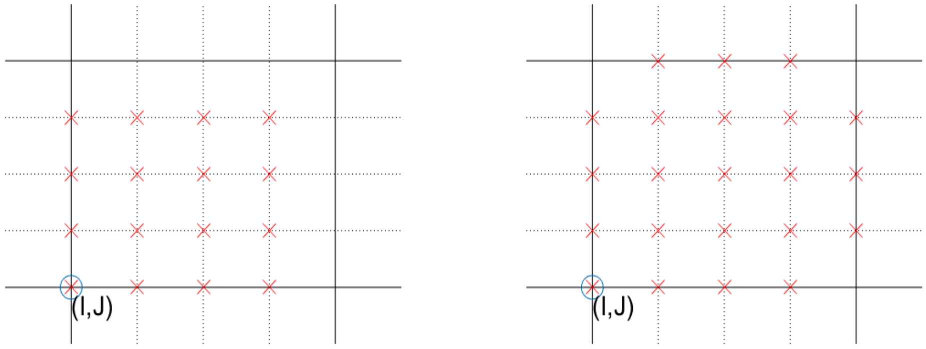

Figure 7.2: At left, the location of degrees of freedom in $\mathfrak{S}_{I,J}$ defined in Equation (7.19) for one subdomain with $p = 4$. At right, the location of degrees of freedom in $\mathfrak{S}^*_{I,J}$ defined in Equation (7.23) for one subdomain with $p = 4$.

For each $\mathfrak{S}_{I,J}$, we use a row-wise ordering of the grid points (lexicographical ordering). This will fix the ordering of the symbols in the following; for any other ordering, a permutation operator would need to be applied. In the following, we do not show the specific position of each element in a vector or matrix, and they are assumed to be consistent with the ordering of the grid points. Based on the set $\mathfrak{S}_{I,J}$, we define the p^2-dimensional space

$$\mathcal{E}(\boldsymbol{\theta}) = \text{span}\big\{\boldsymbol{\varphi}_{k,\ell}(\boldsymbol{\theta}) : k, \ell = 0, 1, \cdots, p-1\big\}, \tag{7.20}$$

where $\boldsymbol{\varphi}_{k,\ell}(\boldsymbol{\theta}) = \big(\varphi_{k,\ell}(\boldsymbol{\theta}, \boldsymbol{x}_{I+s,J+t})\big)_{s,t=0}^{p-1}$ is a $p^2 \times 1$ vector with only one nonzero element, defined in (7.17), in the position corresponding to $(I+k, J+\ell)$. For both $\mathcal{E}(\boldsymbol{\theta})$ and $\boldsymbol{\varphi}_{k,\ell}(\boldsymbol{\theta})$, we have simply taken the infinite mesh representation of $\mathbf{F}_H$ and truncated it to a single $p \times p$ block of the mesh, which is sufficient to define the symbol of A in this basis. Let Φ_h be a $p^2 \times p^2$ diagonal matrix, whose diagonal elements are functions $\varphi(\boldsymbol{\theta}, \boldsymbol{x}) = e^{\iota\boldsymbol{\theta}\cdot\boldsymbol{x}/H}$, where $\boldsymbol{x} \in \mathfrak{S}_{I,J}$, so $\mathcal{E}(\boldsymbol{\theta}) = \text{Range}(\Phi_h)$.

Note that each subdomain contains p^2 degrees of freedom, and that the corresponding symbol is not a scalar due to the definition of the Fourier basis in (7.20). We treat the block symbol as a system, presented as a $p^2 \times p^2$ matrix. Let $A_{I,J}$ be the periodic Laplace operator on $\mathfrak{S}_{I,J}$. Then, its symbol $\widetilde{A}$ satisfies

$$A_{I,J}\boldsymbol{\phi}(\boldsymbol{\theta}, \boldsymbol{x}) = \widetilde{A}(\boldsymbol{\theta})\boldsymbol{\phi}(\boldsymbol{\theta}, \boldsymbol{x}),\ \forall \boldsymbol{\phi}(\boldsymbol{\theta}, \boldsymbol{x}) \in \mathcal{E}(\boldsymbol{\theta}), \tag{7.21}$$

where $\widetilde{A}$ is a $p^2 \times p^2$ matrix. Equation (7.21) is equivalent to

$$A_{I,J} \sum_{0 \le k,\ell \le p-1} \alpha_{k,\ell}\boldsymbol{\varphi}_{k,\ell} = \Phi_h \widetilde{A}\boldsymbol{\alpha}, \tag{7.22}$$

for any vector $\boldsymbol{\alpha}$, whose elements are denoted as $\alpha_{k,\ell}$. Since (7.22) holds for any $\alpha_{k,\ell}$, we have $\widetilde{A} = \Phi_h^T A_{I,J} \Phi_h$, where T is the (conjugate) transpose. Note that $\Phi_h^{-1} = \Phi_h^T$ and the entries in these matrices have the same form, $e^{\pm\iota\boldsymbol{\theta}\cdot\boldsymbol{x}_{I,J}/H}$.

We consider the action of $\widetilde{A}(\boldsymbol{\theta})$ on a vector in terms of the coefficients of the Fourier basis functions. Considering a point in $\mathfrak{S}_{I,J}$, if the values of a function at neighbouring points are expressed by $\alpha_{k,\ell}\boldsymbol{\varphi}_{k,\ell}$, the entries in $\widetilde{A}(\boldsymbol{\theta})\boldsymbol{\alpha}$ give the coefficients of the Fourier expansion of the original operator A on G_h acting on the function in $\mathcal{E}(\boldsymbol{\theta})$ with coefficient $\boldsymbol{\alpha}$. We note that a similar approach was employed for LFA for vector finite-element discretizations in [21].

7.4.3 Representation of preconditioned operators

Now we turn to calculating the Fourier representations of M_1^{-1} and M_2^{-1}. First, we define a collective grid set associated with (I,J) for the partially subassembled problem for each subdomain as

$$\mathfrak{S}^*_{I,J} = \{\boldsymbol{x}_{(I+k,J+\ell)} : k,\ell = 0,1,\cdots,p\} \setminus \{\boldsymbol{x}_{(I+p,J)}, \boldsymbol{x}_{(I,J+p)}, \boldsymbol{x}_{(I+p,J+p)}\}, \qquad (7.23)$$

see the right of Figure 7.2. We first consider the stencil of M_1^{-1} acting on one subdomain, $\mathfrak{S}^*_{I,J}$.

Recall the scaling operator, $\mathcal{R}_1$, where each column of $\mathcal{R}_1$ corresponding to a degree of freedom of the global problem in the interiors and at the coarse-grid points has a single nonzero entry with value 1, and each column of $\mathcal{R}_1$ corresponding to an interface degree of freedom has two nonzero entries, each with value $\frac{1}{2}$. Since we consider periodic Fourier modes on each subdomain, the interface degrees of freedom share the same values scaled by an exponential shift. For example, at the left of Figure 7.2, the degrees of freedom located at the left boundary and the right boundary have the same coefficient of the (shifted) exponential, as do the degrees of freedom located at the bottom and top. Thus, $\mathcal{R}_1$ is its own Fourier representation, since the neighborhoods do not contribute to each other. Note that $\mathcal{R}_1$ maps the p^2-dimensional Fourier basis from $\mathcal{E}(\boldsymbol{\theta})$, used to express $\widetilde{A}(\boldsymbol{\theta})$ onto a $(p+1)^2-3$ dimensional space with similar sparse basis on $\mathfrak{S}^*_{I,J}$ that is suitable for expressing the symbol of $\hat{A}$ and its inverse.

We now focus on $\hat{A}$ presented on one subdomain. Let $\hat{A}^{(I,J)}$ be a $(p+1)^2\times(p+1)^2$ matrix, which is the partially subassembled problem on one subdomain including its four neighbouring coarse-grid degrees of freedom, as

$$\hat{A}^{(I,J)} = \begin{pmatrix} A_{rr}^{(I,J)} & \left(\hat{A}_{\Pi r}^{(I,J)}\right)^T \\ \hat{A}_{\Pi r}^{(I,J)} & A_{\Pi\Pi}^{(I,J)} \end{pmatrix} = \begin{pmatrix} A_{rr}^{(I,J)} & 0 \\ \hat{A}_{\Pi r}^{(I,J)} & \hat{S}_{\Pi}^{(I,J)} \end{pmatrix} \begin{pmatrix} I & \left(A_{rr}^{(I,J)}\right)^{-1}\left(\hat{A}_{\Pi r}^{(I,J)}\right)^T \\ 0 & I \end{pmatrix}, \qquad (7.24)$$

where $A_{rr}^{(I,J)}$ is a $\big((p+1)^2-4\big)\times\big((p+1)^2-4\big)$ matrix corresponding to the interior and interface degrees of freedom on the subdomain and $A_{\Pi\Pi}^{(I,J)}$ corresponds to the four coarse-level variables on one subdomain. Note that $A_{\Pi\Pi}^{(I,J)} = \frac{2}{3}I$ and $\hat{S}_{\Pi}^{(I,J)} = A_{\Pi\Pi}^{(I,J)} - \hat{A}_{\Pi r}^{(I,J)}(A_{rr}^{(I,J)})^{-1}(\hat{A}_{\Pi r}^{(I,J)})^T$. We use index (I,J) as a superscript in order to distinguish this from the matrix in (7.8), but note that it is independent of the particular subdomain, (I, J), under consideration. Let $\widetilde{\widetilde{A}}$ be the Fourier representation of the partially subassembled problem with the corresponding symbol being a $\big((p+1)^2-3\big) \times \big((p+1)^2-3\big)$ matrix,

$$\widetilde{\widetilde{A}} = \begin{pmatrix} \widetilde{A}_{rr} & 0 \\ \widetilde{A}_{\Pi r} & \widetilde{S}_{\Pi} \end{pmatrix} \begin{pmatrix} \widetilde{I} & (\widetilde{A}_{rr})^{-1}\widetilde{A}_{\Pi r}^{T} \\ 0 & \widetilde{I} \end{pmatrix} = \widetilde{\mathcal{K}}_1 \widetilde{\mathcal{K}}_2,$$

where $\widetilde{A}_{rr}$ is a $\big((p+1)^2-4\big) \times \big((p+1)^2-4\big)$ Fourier representation of $A_{rr}^{(I,J)}$ computed as was done for $\widetilde{A}$ above and $\widetilde{S}_{\Pi}$ is the representation of the global Schur complement, $\hat{S}_{\Pi}$. Let $S_0 = \hat{A}_{\Pi r}^{(I,J)}(A_{rr}^{(I,J)})^{-1}(\hat{A}_{\Pi r}^{(I,J)})^T$ be a 4×4 matrix corresponding to the vertices adjacent to one subdomain, representing one macroelement of the coarse-level variables. Direct calculation shows this matrix has the same nonzero structure as the element stiffness matrix for a symmetric second-order differential operator on a uniform square mesh, with equal values for the connections from each node to itself (denoted s_1), its adjacent vertices (s_2), and its opposite corner (s_3). Since $\hat{S}_{\Pi}^{(I,J)} = \frac{2}{3}I - S_0$ gives the macroelement stiffness contribution, assembling the coarse-level stiffness matrix over 2×2 macroelement patches yields $\widetilde{S}_{\Pi}$ as the symbol of the 9-point stencil given by

$$\begin{bmatrix} -s_3 & -2s_2 & -s_3 \\ -2s_2 & \frac{8}{3} - 4s_1 & -2s_2 \\ -s_3 & -2s_2 & -s_3 \end{bmatrix},$$

acting on the coarse points.

$\widetilde{A}_{\Pi r}$ is the representation of the contribution from interior and interface degrees of freedom to the coarse degrees of freedom, and has only 12-nonzero elements per subdomain, with 3 contributing to each corner of the subdomain. We take the coarse-level point $\boldsymbol{x}_{I,J}$ as an example. At the right of Figure 7.2, $\boldsymbol{x}_{I,J}$ obtains contributions from the points $\boldsymbol{x}_{I+1,J}, \boldsymbol{x}_{I+1,J+1}, \boldsymbol{x}_{I,J+1}$ and the corresponding stencils are

$$\begin{bmatrix} * & -\frac{1}{6} \end{bmatrix}, \begin{bmatrix} & -\frac{1}{3} \\ * & \end{bmatrix}, \begin{bmatrix} -\frac{1}{6} \\ * \end{bmatrix},$$

where $*$ denotes the position on the grid at which the discrete operator is applied, namely $\boldsymbol{x}_{I,J}$. The symbols of these three stencils are given by $-\frac{1}{6}e^{\iota\theta_1/p}$, $-\frac{1}{3}e^{\iota(\theta_1+\theta_2)/p}$, $-\frac{1}{6}e^{\iota\theta_2/p}$, respectively. Since $\boldsymbol{x}_{I,J}$ is adjacent to three other subdomains, the coarse degree of freedom at $\boldsymbol{x}_{I,J}$ also obtains contributions from those subdomains, and the other 9 contributing stencils are computed similarly. Finally, the representation of $M_1^{-1}A$ is given by

$$\widetilde{G}_{1,0}(\boldsymbol{\theta}) = \widetilde{\mathcal{R}}_1^T(\widetilde{\hat{A}})^{-1}\widetilde{\mathcal{R}}_1\widetilde{A} = \widetilde{\mathcal{R}}_1^T\widetilde{\mathcal{K}}_2^{-1}\widetilde{\mathcal{K}}_1^{-1}\widetilde{\mathcal{R}}_1\widetilde{A}.$$

For the Dirichlet preconditioner in (7.6), we also need to know the LFA representation of the operators J_D and $\mathcal{H}$. Since J_D is a pointwise scaling operator, its symbol in the pointwise basis of $\mathbf{F}_H$ is itself. According to the definition of $\mathcal{H}$, the symbol of $\mathcal{H}$ is given by $\widetilde{\mathcal{H}} = \widetilde{A}_{rr,I}^{-1}\widetilde{A}_{\Gamma,I}^T$, where $\widetilde{A}_{rr,I}$ is the submatrix of $\widetilde{A}_{rr}$ corresponding to the interior degrees of freedom, and $\widetilde{A}_{\Gamma,I}^T$ is the submatrix of $\widetilde{\hat{A}}$ corresponding to the contribution of the interface degrees of freedom to the interior degrees of freedom. Both of these are computed in a similar manner to $\widetilde{A}$ and $\widetilde{\hat{A}}$ as described above. Thus, the LFA representation of $M_2^{-1}A$ can be written as

$$\widetilde{G}_{2,0}(\boldsymbol{\theta}) = (\widetilde{\mathcal{R}}_1^T - \widetilde{\mathcal{H}}\widetilde{J}_D)\widetilde{\mathcal{K}}_2^{-1}\widetilde{\mathcal{K}}_1^{-1}(\widetilde{\mathcal{R}}_1 - \widetilde{J}_D^T\widetilde{\mathcal{H}}^T)\widetilde{A}.$$

The details of the 3-level variants of LFA are similar to those given above. We now consider a segment of the infinite mesh given, on the fine level, by a $p \times p$ array of subdomains, with each subdomain of size $p \times p$ elements. On the first coarse level (corresponding to the Schur complement $\hat{S}_\Pi$ in (7.8)), we then consider a single $p \times p$ subdomain of the infinite coarse mesh, and apply the same technique recursively. To accommodate this, we adapt the fine-level Fourier modes to be $\varphi^*(\boldsymbol{\theta}, \boldsymbol{x}) := e^{\iota\boldsymbol{\theta}\cdot\boldsymbol{x}/H'}$, where $H' = p^2h$. The coarse-level Fourier modes are then the same as (7.20). Thus, $\widetilde{G_{i,j}}(\boldsymbol{\theta})$ is a $p^4 \times p^4$ matrix for the three-level variants.

7.5 Numerical results

7.5.1 Condition numbers of two-level variants

In the LFA setting, $\boldsymbol{\theta} = (\theta_1, \theta_2) \in [-\pi, \pi)^2$. Here we take $d\theta = \pi/n$ as the discrete stepsize and sample the Fourier space at $2n$ evenly distributed frequencies in θ_1 and θ_2 with offset $\pm d\theta/2$ from $\theta_1 = \theta_2 = 0$ to avoid the singularity at zero frequency. For each frequency on the mesh, we compute the eigenvalues of the two-level operators, and define $\kappa := \frac{e_{\max}}{e_{\min}}$, where $e_{\min}$ and $e_{\max}$ are the smallest and biggest eigenvalues over all frequencies.

Table 7.1 shows the condition numbers for the two-level preconditioners with variation in both subdomain size, p, and sampling frequency, n. When $n = 2$, the condition number prediction is notably inaccurate, but we obtain a consistent prediction for $n \geq 4$ (and very consistent for $n \geq 8$). For $\widetilde{G}_{1,0}$, the condition number increases quickly with p as expected. Compared with $\widetilde{G}_{1,0}$, $\widetilde{G}_{2,0}$ has a much smaller condition number that grows more slowly with p. For $\widetilde{G}_{1,0}$, we know there exists $\mathfrak{C}_{1,0}$ such that the true condition number of the preconditioned system (on a finite grid) is bounded by $\mathfrak{C}_{1,0}\frac{H}{h}(1 + \log\frac{H}{h})$ [20]; from this data, we see that our LFA prediction is consistent with this, with constant $\mathfrak{C}_{1,0} \approx 0.6$. For $\widetilde{G}_{2,0}$, we know there exists $\mathfrak{C}_{2,0}$ such that the true condition number of the preconditioned system (on a finite grid) is bounded by $\mathfrak{C}_{2,0}(1 + \log\frac{H}{h})^2$ [20]; from this data, again we see that our LFA prediction is consistent with this, with constant $\mathfrak{C}_{2,0} \approx 0.4$.

Table 7.1: LFA-predicted condition numbers of two-level preconditions as a function of subdomain size, p, and sampling frequency, n.

	$\widetilde{G}_{1,0}$				$\widetilde{G}_{2,0}$			
p n	4	8	16	32	4	8	16	32
2	4.14	11.11	27.95	67.55	2.23	3.02	3.94	5.01
4	4.36	11.94	30.27	73.44	2.32	3.15	4.13	5.26
8	4.42	12.18	30.94	75.16	2.34	3.19	4.17	5.32
16	4.44	12.25	31.12	75.61	2.35	3.19	4.19	5.33
32	4.44	12.26	31.16	75.72	2.35	3.20	4.19	5.34
64	4.44	12.27	31.17	75.75	2.35	3.20	4.19	5.34
128	4.44	12.27	31.18	75.76	2.35	3.20	4.19	5.34
$\mathfrak{C}_{i,0}(n = 32)$	0.47	0.50	0.52	0.53	0.41	0.34	0.29	0.27

Optimizing the weight parameters for $\widetilde{G}^f_{1,0}$ and $\widetilde{G}^f_{2,0}$ by systematic search with different n and p, we see that the optimal parameter ω is dependent on p, but largely independent of n. Table 7.2 shows that significant improvement can be had for the M_1 preconditioner, but not for M_2, see Table 7.3. We again see small n (e.g., $n = 4$ or 8) is enough to obtain a consistent prediction for these condition numbers.

Table 7.2: Condition numbers for two-level lumped preconditioner with fine-grid multiplicative combination with diagonal scaling, $\widetilde{G}^f_{1,0}$. In brackets, value of weight parameter, ω, that minimizes condition number.

p n	4	8	16	32
2	2.06(2.1)	3.18(2.3)	5.43(2.5)	9.71(2.6)
4	2.17(1.5)	3.29(2.3)	5.64(2.5)	9.99(2.6)
8	2.18(1.4)	3.32(2.3)	5.70(2.5)	10.08(2.6)
16	2.18(1.4)	3.32(2.3)	5.72(2.5)	10.10(2.6)
32	2.18(1.4)	3.33(2.3)	5.72(2.5)	10.10(2.6)
64	2.18(1.4)	3.33(2.3)	5.72(2.5)	10.10(2.6)

Table 7.3: Condition numbers for two-level Dirichlet preconditioner with fine-grid multiplicative combination with diagonal scaling, $\widetilde{G}^f_{2,0}$. In brackets, value of weight parameter, ω, that minimizes condition number.

p n	4	8	16	32
2	1.82(2.2)	2.36(1.7)	3.12(2.0)	4.20(1.8)
4	2.03(1.1)	2.54(1.6)	3.33(2.0)	4.44(1.8)
8	2.07(1.1)	2.59(1.6)	3.39(2.0)	4.50(1.8)
16	2.08(1.1)	2.60(1.6)	3.40(2.0)	4.52(1.8)
32	2.08(1.1)	2.60(1.6)	3.40(2.0)	4.52(1.8)
64	2.08(1.1)	2.61(1.6)	3.40(2.0)	4.52(1.8)

In order to see the sensitivity of performance to parameter choice, we consider the condition numbers for the two-level lumped and Dirichlet preconditioners in multiplicative combination with diagonal scaling on the fine grid with $p = 8$, as a function of ω, in Figure 7.3. We see that the condition number of $\widetilde{G}^f_{1,0}$ shows strong sensitivity to small values of ω. For $\widetilde{G}^f_{2,0}$, however, many allowable parameters obtain a good condition number.

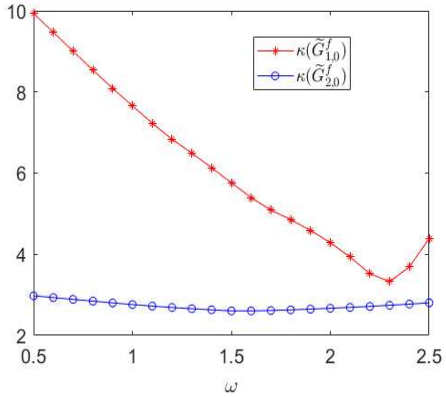

Figure 7.3: Condition numbers for two-level lumped and Dirichlet preconditioners in multiplicative combination with diagonal scaling on the fine grid with $p = 8$, as a function of relaxation parameter, ω.

7.5.2 Eigenvalue distribution of two-level variants

In this section, we take $n = 32$, yielding $2n$ points in each dimension and $(2n)^2 = 4096$ values of $\boldsymbol{\theta}$, although similar results are seen for smaller values of n. We also consider only $p = 8$, although similar results are seen for other values of p. For $\widetilde{G}^f_{1,0}$ and $\widetilde{G}^f_{2,0}$, we use the optimal values of ω, shown in the tables above. The histograms in Figure 7.4 show the density of eigenvalues for the two-level preconditioned operators. For these values of n and p, our LFA computes a total of 262144 eigenvalues, giving 64 eigenvalues for each of 4096 sampling points. For all cases, the eigenvalues around 1 (represented in two bins in the histogram, covering the interval from 0.9 to 1.1) appear with dominating multiplicity, accounting for about 200,000 of the computed eigenvalues.

Note that there is a gap in the spectrum of $\widetilde{G}_{1,0}$ that increases in size with p (not shown here). A notable difference between $\widetilde{G}_{1,0}$ and $\widetilde{G}_{2,0}$ is that, while there is still a small gap in the spectrum of $\widetilde{G}_{2,0}$, it is not very prominent. Note also that the spectra are real-valued, with only roundoff-level errors in the imaginary component. Comparing the eigenvalues for $\widetilde{G}^f_{1,0}$ and $\widetilde{G}^f_{2,0}$ with those for $\widetilde{G}_{1,0}$ and $\widetilde{G}_{2,0}$, we see that the eigenvalues are much more tightly clustered for $\widetilde{G}^f_{1,0}$, but still exhibit a gap in the spectrum. The eigenvalues of $\widetilde{G}^f_{2,0}$, in contrast, appear to lie in a continuous interval. We note that little improvement is seen in the spectrum of $\widetilde{G}^f_{2,0}$, in comparison with $\widetilde{G}_{2,0}$. Also interesting to note is that, in contrast to all other cases, the smallest eigenvalue of $\widetilde{G}^f_{1,0}$ is less than 1.

Remark 7.5.1. *As the LFA predicts both eigenvectors and eigenvalues, we can examine the frequency composition of the eigenvectors associated with these eigenvalues. The largest eigenvalue of $\widetilde{G}_{1,0}$ is found to be dominated by oscillatory modes, but this is not true for $\widetilde{G}_{2,0}$. This motivates the proposed multiplicative method based on simple diagonal scaling, which is well known to effectively damp oscillatory errors in the classical multigrid setting.*

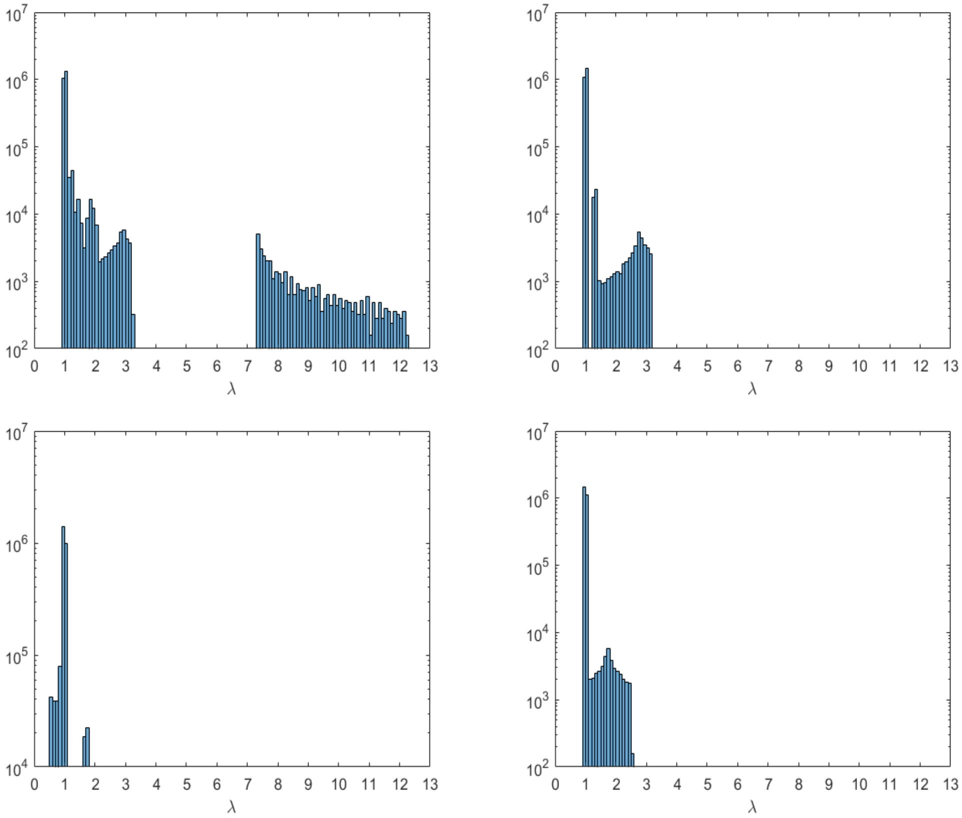

Figure 7.4: Histograms showing density of eigenvalues for two-level preconditioned operators with $p = 8$. Top left: $\widetilde{G}_{1,0}$, Top right: $\widetilde{G}_{2,0}$, Bottom left: $\widetilde{G}^f_{1,0}$, Bottom right: $\widetilde{G}^f_{2,0}$.

7.5.3 Condition numbers of three-level variants

For the three-level preconditioned operators, we need to find all the eigenvalues of a $p^4 \times p^4$ matrix for each sampled value of $\boldsymbol{\theta}$. For the two-level variants, we saw that sampling with $n = 4$ is sufficient to give useful accuracy of the LFA predictions. Here, we also see similar behavior in Table 7.4, which shows the condition numbers of $\widetilde{G}_{i,j}(i, j = 1, 2)$ for varying p an n. We see that, as expected from the theory, these condition numbers show degradation from the two-level case. It is not surprising that $\widetilde{G}_{2,2}$ has the smallest condition number of these variants, since M_2 is applied to both fine and coarse levels.

Table 7.5 presents the condition number of variants $\widetilde{G}^f_{i,j}$ and $\widetilde{G}^c_{i,j}$, based on the multiplicative combination with diagonal scaling on the fine level and coarse level, respectively, and some improvement is offered. For fixed p, the optimal ω is found to be robust to n (not shown here). In general, we see better performance for $\widetilde{G}^f_{i,j}$ in comparison to $\widetilde{G}^c_{i,j}$, and $\widetilde{G}^f_{1,1}$ offers significant improvement over $\widetilde{G}_{1,1}$. For other values of i, j, however, only small improvements are seen.

Table 7.4: Condition numbers of three-level preconditioners with no multiplicative relaxation.

p	$\widetilde{G}_{1,1}$	$\widetilde{G}_{1,2}$	$\widetilde{G}_{2,1}$	$\widetilde{G}_{2,2}$
$4(n = 2)$	9.18	5.43	7.27	4.24
$4(n = 4)$	9.65	5.68	7.63	4.47
$4(n = 8)$	9.79	5.74	7.73	4.53
$4(n = 16)$	9.82	5.76	7.76	4.54
$4(n = 32)$	9.83	5.76	7.77	4.55
$8(n = 2)$	46.66	15.46	24.73	7.55
$8(n = 4)$	50.00	16.15	26.53	7.94
$8(n = 8)$	50.96	16.33	27.05	8.04

Table 7.5: Condition numbers of three-level preconditioners with fine-scale or coarse-scale multiplicative preconditioning. All results were computed with $n = 4$, and the experimentally optimized weight, ω, is shown in brackets.

p	$\widetilde{G}^f_{1,1}$	$\widetilde{G}^f_{1,2}$	$\widetilde{G}^f_{2,1}$	$\widetilde{G}^f_{2,2}$
4	6.80(1.4)	4.28(1.4)	6.14(1.6)	4.04(1.1)
8	28.75(1.7)	9.16(1.7)	20.94(1.6)	6.73(1.5)

p	$\widetilde{G}^c_{1,1}$	$\widetilde{G}^c_{1,2}$	$\widetilde{G}^c_{2,1}$	$\widetilde{G}^c_{2,2}$
4	6.04(1.6)	5.47(1.1)	4.67(1.6)	4.30(1.0)
8	31.91(2.0)	15.17(1.4)	15.57(2.1)	7.46(1.2)

In order to see the sensitivity of performance to parameter choice, we consider three-level preconditioners with weighted multiplicative preconditioning on both fine and coarse scales, $\widetilde{G}^{f,c}_{1,1}$ and $\widetilde{G}^{f,c}_{2,2}$, with $p = 4$ and $n = 4$. At the left of Figure 7.5, we present the LFA-predicted condition number for $\widetilde{G}^{f,c}_{1,1}$ with variation in ω_1 and ω_2. Here, we see strong sensitivity to "small" values of ω_1, for example $\omega_1 < 1.5$, and also to large values of ω_1 with small values of ω_2. We note general improvement, though, in the optimal performance for large ω_1 with suitably chosen ω_2, albeit with diminishing returns as ω_1 continues to increase. Fixing $\omega_1 = 4$, we find $\omega_2 = 1.7$ offers best performance, with optimal condition number of 2.66. At the right of Figure 7.5, we consider $\widetilde{G}^{f,c}_{2,2}$ as a function of ω_1 and ω_2. Here, we see stronger sensitivity to large values of ω_2, and to large values of ω_1 and small values of ω_2, but a large range of parameters that give generally similar performance. Fixing $\omega_1 = 4$, we find that $\omega_2 = 1.2$ achieves the optimal condition number of 3.72. Similar performance was seen for $\widetilde{G}^{f,c}_{1,2}, \widetilde{G}^{f,c}_{2,1}$, and $\widetilde{G}^{s,c}_{i,j}$. Slight improvements can be seen by allowing even larger values of ω_1, giving an LFA-predicted condition number for $\widetilde{G}^{f,c}_{1,1}$ of 2.25 with $\omega_1 = 5.0$ and $\omega_2 = 2.0$, but a much smaller band of values of ω_2 leads to near-optimal performance as ω_1 increases. For $\widetilde{G}^{f,c}_{2,2}$, this sensitivity does not arise, but the improvements are even more marginal, achieving an LFA-predicted condition number of 3.63 for $\omega_1 = 5.8$ and $\omega_2 = 1.3$.

Motivated by Figure 7.5, we fix $\omega_1 = 4$ with $n = 4$, and optimize the condition numbers for the three-level preconditioners with two multiplicative preconditioning steps per iteration, either both on the coarse level, $\widetilde{G}^{s,c}_{i,j}$, or one on each level, $\widetilde{G}^{f,c}_{i,j}$, with respect to ω_2. From Table 7.6, notable improvement is seen for all i, j with $\widetilde{G}^{f,c}_{i,j}$, particularly for $\widetilde{G}^{f,c}_{1,1}$ and $\widetilde{G}^{f,c}_{2,1}$. We also note that there is little variation in the optimal parameter for each preconditioner between the $p = 4$ and $p = 8$ cases. It is notable that we are able to achieve similar performance for the multiplicative preconditioner based on M_1 as seen for M_2, and that both show significant improvement from the classical three-level results shown in Table 7.4, when used in combination with multiplicative preconditioning on both fine and coarse levels.

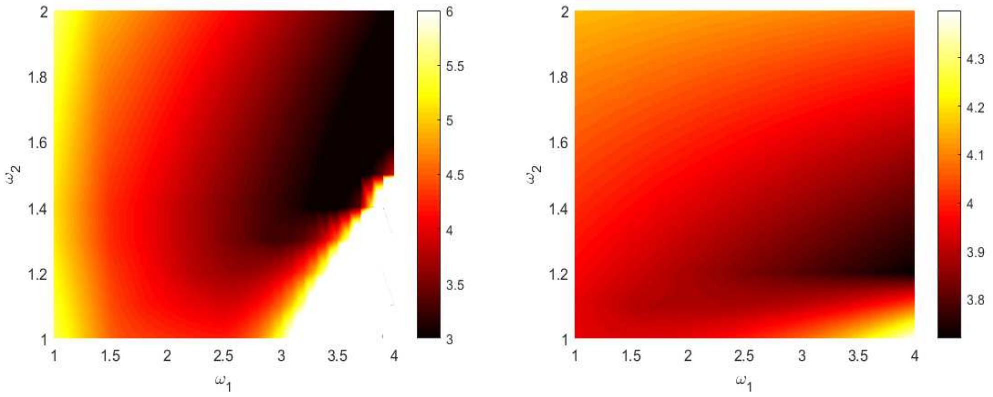

Figure 7.5: Condition number of three-level preconditioners with multiplicative preconditioning on both the fine and coarse scales as a function of ω_1 and ω_2, with $p = 4$ and $n = 4$. At left, condition number for $\widetilde{G}^{f,c}_{1,1}$; at right, condition number for $\widetilde{G}^{f,c}_{2,2}$.

Table 7.6: Condition numbers of three-level preconditioners with symmetric weighting of multiplicative preconditioning on the coarse scale, $\widetilde{G}^{s,c}_{i,j}$, and weighting of multiplicative preconditioning on both fine and coarse scales, $\widetilde{G}^{f,c}_{i,j}$. All results were computed with $n = 4$, and the experimentally optimized weight, ω_2, is shown in brackets.

p	$\widetilde{G}^{s,c}_{1,1}$	$\widetilde{G}^{s,c}_{1,2}$	$\widetilde{G}^{s,c}_{2,1}$	$\widetilde{G}^{s,c}_{2,2}$
4	5.43(1.4)	5.34(0.9)	4.22(1.3)	4.18(0.9)
8	17.45(1.2)	14.13(1.0)	8.31(1.1)	6.88(0.9)

p	$\widetilde{G}^{f,c}_{1,1}$	$\widetilde{G}^{f,c}_{1,2}$	$\widetilde{G}^{f,c}_{2,1}$	$\widetilde{G}^{f,c}_{2,2}$
4	2.66(1.7)	3.85(1.3)	3.24(1.8)	3.72(1.2)
8	5.16(1.8)	7.59(1.7)	4.88(1.8)	5.70(1.5)

7.6 Conclusions

In this paper, we quantitatively estimate the condition numbers of variants of BDDC algorithms, using local Fourier analysis. A new choice of basis is proposed to simplify the LFA, and we believe this choice will prove useful in analysing many domain decomposition algorithms in the style used here. Multiplicative preconditioners with these two domain decomposition methods are discussed briefly, and both lumped and Dirichlet variants can be improved in this way. The coarse problem involved in these domain decomposition methods can be solved by similar methods. LFA analysis of three-level variants is also considered. Degradation in convergence is well known when moving from two-level to three-level variants of these algorithms. We show that the LFA presented above, in combination with the use of multiplicative preconditioners on the coarse and fine levels provide ways to mitigate this performance loss. Future work includes extending these variants of the preconditioned operators, using LFA to optimize the resulting algorithms, and considering other types of problems with similar preconditioners.

Acknowledgements

The work of J.B. was partially funded by U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Award Number DE-SC0016140. The work of S.M. was partially funded by an NSERC Discovery Grant.

Bibliography

[1] J. H. Bramble, J. E. Pasciak, J. P. Wang, and J. Xu. Convergence estimates for product iterative methods with applications to domain decomposition. *Math. Comp.*, 57(195):1–21, 1991.

[2] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comp.*, 31(138):333–390, 1977.

[3] S. C. Brenner, E.-H. Park, and L.-Y. Sung. A BDDC preconditioner for a

symmetric interior penalty Galerkin method. *Electron. Trans. Numer. Anal.*, 46:190–214, 2017.

[4] S. C. Brenner and L.-Y. Sung. BDDC and FETI-DP without matrices or vectors. *Computer Methods in Applied Mechanics and Engineering*, 196(8):1429–1435, 2007.

[5] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A multigrid tutorial.* SIAM, 2000.

[6] C. R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. *SIAM Journal on Scientific Computing*, 25(1):246–258, 2003.

[7] C. R. Dohrmann. An approximate BDDC preconditioner. *Numerical Linear Algebra with Applications*, 14(2):149–168, 2007.

[8] C. R. Dohrmann. Preconditioning of saddle point systems by substructuring and a penalty approach. In *Domain decomposition methods in science and engineering XVI*, volume 55 of *Lect. Notes Comput. Sci. Eng.*, pages 53–64. Springer, Berlin, 2007.

[9] C. R. Dohrmann and O. B. Widlund. A BDDC algorithm with deluxe scaling for three-dimensional H(curl) problems. *Communications on Pure and Applied Mathematics*, 69(4):745–770, 2016.

[10] V. Dolean, P. Jolivet, and F. Nataf. *An introduction to domain decomposition methods: Algorithms, theory, and parallel implementation.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2015.

[11] M. Dryja, J. Galvis, and M. Sarkis. BDDC methods for discontinuous Galerkin discretization of elliptic problems. *J. Complexity*, 23(4-6):715–739, 2007.

[12] M. Dryja and O. B. Widlund. Towards a unified theory of domain decomposition algorithms for elliptic problems. In *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Houston, TX, 1989)*, pages 3–21. SIAM, Philadelphia, PA, 1990.

[13] M. Dryja and O. B. Widlund. A FETI-DP method for a mortar discretization of elliptic problems. *Lecture Notes in Computational Science and Engineering*, 23:41–52, 2002.

[14] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. FETI-DP: a dual–primal unified FETI method Part I: A faster alternative to the two-level FETI method. *International Journal for Numerical Methods in Engineering*, 50(7):1523–1544, 2001.

[15] M. J. Gander, F. Magoulès, and F. Nataf. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM J. Sci. Comput.*, 24(1):38–60, 2002.

[16] J. Li. A dual-primal FETI method for incompressible Stokes equations. *Numerische Mathematik*, 102(2):257–275, 2005.

[17] J. Li and O. Widlund. BDDC algorithms for incompressible Stokes equations. *SIAM Journal on Numerical Analysis*, 44(6):2432–2455, 2006.

[18] J. Li and O. Widlund. FETI-DP, BDDC, and block Cholesky methods. *International Journal for Numerical Methods in Engineering*, 66(2):250–271, 2006.

[19] J. Li and O. Widlund. A BDDC preconditioner for saddle point problems. In *Domain decomposition methods in science and engineering XVI*, volume 55 of *Lect. Notes Comput. Sci. Eng.*, pages 413–420. Springer, Berlin, 2007.

[20] J. Li and O. Widlund. On the use of inexact subdomain solvers for BDDC algorithms. *Computer Methods in Applied Mechanics and Engineering*, 196(8):1415–1428, 2007.

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

[22] J. Mandel and M. Brezina. Balancing domain decomposition for problems with large jumps in coefficients. *Math. Comp.*, 65(216):1387–1401, 1996.

[23] J. Mandel and C. R. Dohrmann. Convergence of a balancing domain decomposition by constraints and energy minimization. *Numerical Linear Algebra with Applications*, 10(7):639–659, 2003.

[24] J. Mandel, C. R. Dohrmann, and R. Tezaur. An algebraic theory for primal and dual substructuring methods by constraints. *Applied Numerical Mathematics*, 54(2):167–193, 2005.

[25] J. Mandel, B. r. Sousedík, and C. R. Dohrmann. Multispace and multilevel BDDC. *Computing*, 83(2-3):55–85, 2008.

[26] K. Stüben and U. Trottenberg. Multigrid methods: Fundamental algorithms, model problem analysis and applications. *Multigrid Methods*, pages 1–176, 1982.

[27] A. Toselli and O. Widlund. *Domain decomposition methods: Algorithms and theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2005.

[28] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, Inc., San Diego, CA, 2001. With contributions by A. Brandt, P. Oswald and K. Stüben.

[29] X. Tu. *Domain decomposition algorithms: methods with three levels and for flow in porous media*. PhD thesis, New York University, Graduate School of Arts and Science, 2006.

[30] X. Tu. Three-level BDDC in three dimensions. *SIAM J. Sci. Comput.*, 29(4):1759–1780, 2007.

[31] X. Tu. Three-level BDDC in two dimensions. *Internat. J. Numer. Methods Engrg.*, 69(1):33–59, 2007.

[32] P. Wesseling. *An introduction to multigrid methods*. Pure and Applied Mathematics (New York). John Wiley & Sons, Ltd., Chichester, 1992.

[33] R. Wienands and W. Joppich. *Practical Fourier analysis for multigrid methods*. CRC press, 2004.

Chapter 8

Conclusions and future work

In this thesis, to address the lack of existing research on analysis of vector potential formulations of MHD, we have provided a theoretical analysis for the existence and uniqueness of solutions of both the continuum two-dimensional resistive magnetohydrodynamics model and its discretization, closing the open question of existence and uniqueness of solutions. Furthermore, under moderate conditions, we have proved that Newton's method yields well-posed linearizations and converges to the solution of the weak formulation.

To better understand the performance of monolithic multigrid methods for solving saddle-point problems, we have employed LFA to analyze common block-structured relaxation schemes, including Braess-Sarazin, Uzawa, and distributive Jacobi relaxation, for the Stokes equations. Both the Marker-and-Cell (MAC), and finite-element discretizations (stable and stabilized) have been discussed. LFA helps us understand and optimise these relaxations when solving such saddle-point system with multigrid methods. Comparisons have been made among these relaxations. All in all, inexact Braess-Sarazin relaxation generally outperforms both Uzawa and distributive weighted Jacobi relaxations for the discretizations considered here.

To improve the validity of LFA smoothing analysis, we have designed a modified two-grid LFA for higher-order finite-element discretizations of the Laplace problem, remedying the failure of classical smoothing analysis. Proper parameters have been proposed for the Jacobi relaxation scheme in this setting. This study has shown how coarse-grid correction works for these discretizations, not only reducing low-frequency error components, but also some of those with high-frequency. These findings add to our understanding of the poor predictivity of smoothing analysis for Taylor-Hood elements for the Stokes equations as well, and might be useful for other types of relaxation schemes. We note that this work has some limitations. This study has only examined the weighted Jacobi relaxation, and further investigation into general relaxation schemes for higher-order finite-element discretizations needs to be performed.

To enrich the applicability of extant LFA, we have developed LFA for BDDC, one of the nonoverlapping domain decomposition methods, to close a gap where there is no such LFA research. Our study has provided a framework for LFA with an innovative Fourier basis, which greatly simplifies the analysis. Quantitative estimates of the condition number of the preconditioned systems have been presented. From this LFA, improved performance has been achieved for some two- and three-level variants of BDDC.

The results presented here show that LFA could be applied to other problems to develop efficient algorithms of both multigrid and domain decomposition type. Further research is proposed in the following areas:

1. Many types of problems lead to saddle-point structure. Thus, possible extensions of our LFA work include:

    - The same approach used to analyse the MAC scheme for the Stokes equations can be adapted to analyze many optimal control problems. The construction and analysis of fast numerical methods for control problems governed by PDEs are in urgent demand. Often, the discretization of control problems leads to saddle-point systems, and analyzing this type of problem using LFA has potential to yield value insight.
    - In some approaches to the eigenvalue problem, for example, using Newton's method, saddle-point systems naturally arise. Thus, it is likely that these tools can be effectively applied to eigenvalue problems.
    - Not much research on all-at-once solution of time-dependent problems with LFA exists. However, time-dependent problems commonly arise in science and engineering applications, and receive much attention. Extending the application of LFA to this field offers promise, particularly for "parallel in time" approaches, such as parareal and multigrid reduction in time (MGRIT).

- We are confident that our analysis of BDDC using LFA, especially the use of a "sparse" Fourier basis, will serve as a fundamental tool for analysis of other types of domain decomposition methods, as well as for other approaches such as the classical nested dissection factorization algorithm. Note that there is no existing LFA research for such direct solvers, which we view through the lens of inexact solution of the subdomain problems. We hope that LFA will be valuable in constructing and analysing good preconditioners from many classes of algorithms.

2. Existing work on the analysis of higher-order finite-element methods using LFA mostly focuses on pointwise relaxation. However, in practical use, collective relaxation has been developed for many PDEs. There is a need to understand the solution of these discretizations using collective relaxation, especially for multivariate problems. Furthermore, nowadays, modern parallelism is a trend within scientific computing. Thus, a natural addition to the work presented here is the solution of higher-order finite-element discretizations with multiplicative and additive Schwarz smoothers, including Q_2 and P_2 elements for the Laplace problem, and $P_2 - P_1$ elements for the Stokes equations, with Vanka-type relaxation. These discretizations are tractable with a combination of existing LFA tools and the extensions presented here. It would be interesting to better understand additive Schwarz smoothers in particular to develop efficient algorithms for modern parallel architectures.

3. Another interesting direction for future work is the design of efficient LFA algorithms. In practical use of LFA, we sample in both frequency, $\boldsymbol{\theta}$, and over parameters to optimize eigenvalues of the two-grid error-propagation operator. Note that this needs much computational work; for example, for three-level BDDC, for each frequency and set of parameters, we solve an eigenvalue problem of dimension p^4. The same will be true for other types of domain decomposition methods. Thus, a more efficient LFA strategy is needed. One approach is to use gradient-based optimization (suggested by Jed Brown), based on smoothed approximations of the condition number to isolate the dominant modes that are responsible for the smallest and largest eigenvalues of the operators in the LFA framework. This should reduce the number of frequencies needing to be sampled while optimizing the parameters, as well as the work needed for this optimization, while preserving the accuracy of the algorithm.