Algebraic Multigrid(AMG) for Graph Laplacian Linear Systems: Extensions of AMG for Signed, Undirected and Unsigned, Directed Graphs

by

A. L. Fox

B.A., Loyola University of Maryland, 2012

M.S., University of Colorado Boulder, 2015

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Applied Mathematics

2017

This thesis entitled:
Algebraic Multigrid(AMG) for Graph Laplacian Linear Systems: Extensions of AMG for Signed, Undirected and Unsigned, Directed Graphs
written by A. L. Fox
has been approved for the Department of Applied Mathematics

Prof. Tom Manteuffel

Dr. Geoff Sanders

Dr. John Ruge

Prof. Christian Ketelsen

Prof. Francois Meyer

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Fox, A. L. (Ph.D., Applied Mathematics)

Algebraic Multigrid(AMG) for Graph Laplacian Linear Systems: Extensions of AMG for Signed, Undirected and Unsigned, Directed Graphs

Thesis directed by Prof. Tom Manteuffel

Relational datasets are often modeled as an unsigned, undirected graph due the nice properties of the resulting graph Laplacian, but information is lost if certain attributes of the graph are not represented. This thesis presents two generalizations of Algebraic Multigrid (AMG) solvers with graph Laplacian systems for different graph types: applying Gremban's expansion to extend unsigned graph Laplacian solvers to signed graph Laplacian systems and generalizing techniques in Lean Algebraic Multigrid (LAMG) to a new multigrid solver for unsigned, directed graph Laplacian systems.

Signed graphs extend the traditional notion of connections and disconnections to include both favorable and adverse relationships, such as friend-enemy social networks or social networks with "likes" and "dislikes." Gremban's expansion is used to transform the signed graph Laplacian into an unsigned graph Laplacian with twice the number of unknowns. By using Gremban's expansion, we extend current unsigned graph Laplacian solvers' to signed graph Laplacians. This thesis analyzes the numerical stability and applicability of Gremban's expansion and proves that the error of the solution of the original linear system can be tightly bounded by the error of the expanded system.

In directed graphs, some subset of relationships are not reciprocal, such as hyperlink graphs, biological neural networks, and electrical power grids. A new algebraic multigrid algorithm, Nonsymmetric Lean Algebraic Multigrid (NS-LAMG), is proposed, which uses ideas from Lean Algebraic Multigrid, nonsymmetric Smoothed Aggregation, and multigrid solvers for Markov chain stationary distribution systems. Low-degree elimination, introduced in Lean Algebraic Multigrid for undirected graphs, is redefined for directed graphs.

A semi-adaptive multigrid solver, inspired by low-degree elimination, is instrumented in the setup phase, which can be adapted for Markov chain stationary distributions systems. Numerical results shows that NS-LAMG out performs GMRES(k) for real-world, directed graph Laplacian linear systems. Both generalizations enable more choices in modeling decisions for graph Laplacian systems.

Due the successfulness of NS-LAMG and other various nonsymmetric AMG (NS-AMG) solvers, a further study of theoretical convergence properties are discussed in this thesis. In particular, a necessary condition known as "weak approximation property", and a sufficient one, referred to as "strong approximation property" as well as the "super strong approximation property" are generalized to nonsymmetric matrices and the various relationships between the approximation properties are proved for the nonsymmetric case. In NS-AMG, if $P \neq R$ the two-grid error propagation operator for the coarse-grid correction is an oblique projection with respect to any reasonable norm, which can cause the error to increase. A main focal point of this paper is a discussion on the conditions in which the error propagation operator is bounded, as the stability of the error propagation operator and the approximation properties play an important role in proving convergence of the two-grid method for NS-AMG, which was studied in [37].

Dedication

To my wonderful partner in crime Dan Weidner.

Acknowledgements

A great many of people have supported not only the work contained in this thesis but my overall growth as a researcher, writer, and human being. I would first like to thank Tom Manteuffel, my thesis advisor, for his incredible mind and willingness to work with me. And Geoff Sanders, my co-advisor, for providing multiple research opportunities at Lawrence Livermore National Lab. Both Tom and Geoff were amazing mentors who have showed me how to not doubt my abilities and how to live a balanced life. Christian Ketelsen was also a huge part of my thesis. He not only would take time to read my work but also provided a friendship that I am forever grateful for. I am thankful for my fellow graduate students, Hillary Fairbanks and Meredith Plumley, whom together we formed a supportive, understanding community. Van Henson, Steve McCormick and John Ruge were great resources and were always fun to work with. Finally, the group of graduate student and post-docs at *Grandview* where my closest and most stimulating friendships. Toby Jones persuaded me to join the Grandview group and introduced me to Tom and Geoff. Chris Leibs, Ben O'Neill, Jeff Allen, Wayne Mitchell, Ben Southworth, Steffen Muenzenmaier, Delyan Kalchev, and Tristan Konolige have been amazing friends and colleagues.

My family has been very instrumental in my career at CU. I am forever grateful for my mother and father for instilling in me my determination to pursue any goal. My brother and sister have been tremendous companions. My partner Dan Weidner has been by my side every step in this amazing journey and I can not even begin to describe what his support and love have given me. I have become apart of his family who have also been incredibly supportive. Thank you all.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

Graphs are used to mathematically model the relationships between data in many real-world networks. Examples include, relationships between people, roads between intersections, connections between neurons, and interactions between protein structures. In a social network, such as Facebook, people are labeled as *vertices* and friendships between people are labeled as *edges*. Graphs are key in understanding and leveraging big data and became popular in information retrieval when Google used large graph-based models to rank web pages. By representing relationships between data as a graph, we can find answers to various network analysis tasks, such as data mining or simulation. In this work, we consider real-world graphs which tend to be scale-free and small-world. The traditional definition of a scale-free graph is that the distribution of the vertices' *degree*, the number of edges connected to the vertex, follow a power-law distribution. This implies, that there are a few highly connected "hubs", while most vertices are connected to only one or two other vertices. Examples of "hubs" in hyperlink graphs are Google and Yahoo since they are connected to many webpages. A graph is small-world when the average path length, the mean distance between any two vertices, is relatively small. For example, in a social network, your friends are typically friends with each other. Scale-free, small-world graphs are known to have extremely difficult topologies that traditional linear solvers tend to struggle with as the number of edges grows. Efficient computational tools for real-world graphs are crucial in order to advance analysis tools for big data.

1.0.1 Motivation for Efficient Linear Solvers

Often these analysis tasks require solutions to linear systems posed on the underlying graphs, and efficient solution of these linear equations is a priority. A prototypical example of these tasks is topological graph ranking, where a set of known data is to be expanded into a ranked list of all the other data, based on the connectivity with the known set. A *linear-solver-based approach* to this problem is to build a matrix associated with the graph (e.g. the graph Laplacian L, a matrix representation of the connectivity of the graph) and a right-hand-side vector associated with the known set of data, $\mathbf{b}$. The solution to $L\mathbf{x} = \mathbf{b}$ for a ranking vector $\mathbf{x}$ ranks the vertices according to their size. The vector $\mathbf{x}$ is used to order and prioritize data for follow-up analysis. Prominent examples of this flavor of graph applications are seen in the literature for unsigned graphs models, where all data relationships are considered similarities ([1],[7],[10],[32],[41]). Other graph applications that require a solution to a linear system include ([24],[39],[42],[45],[13]).

In aggregate, these works apply linear-solver-based ranking to both *undirected graph models*, where all relationships are considered bi-directional, and *directed graph models*, where some relationships are not bi-directional. Linear-solver-based ranking is also of interest for *signed graphs*, where negative relationships represent dissimilarities ([19],[50]). Due to the vast size of real-world graphs, many direct solvers become unacceptably slow as the number of vertices and links in a graph grows. For example, classical Gaussian elimination takes time $O(n^3)$, where n is the number of vertices. Iterative solvers can be used instead, and multigrid is an attractive option since it has been shown to be a very robust and scalable solver for many PDE-type problems. The goal of this work is to apply Algebraic Multigrid (AMG) techniques to efficiently solve

$$L\mathbf{x} = \mathbf{b}, \tag{1.1}$$

where L is a graph Laplacian for large real-world signed and directed graphs. Due to the vast number of applications and continuing growth of the size of graphs, robust and scalable solvers are needed for graph analysis tasks for various representations of graphs beyond unsigned, undirected graphs. Before this work, Multigrid algorithms have been demonstrated as robust for unsigned, undirected graph Laplacian linear systems but have not been shown to be robust for signed or directed graph models. This thesis presents two extensions to AMG solvers: an extension of AMG solvers for unsigned, undirected graph Laplacians to signed, undirected graph Laplacian linear systems using Gremban's expansion and a new AMG solver for unsigned, directed graph Laplacian linear systems, called Nonsymmetric Lean Algebraic Multigrid (NS-LAMG). By combining Gremban's Expansion and NS-LAMG, signed, directed graph Laplacian linear systems could then be solved. In this work, we use acronyms for the four main classes of graphs we consider: unsigned-undirected (UU), signed-undirected (SU), unsigned-directed (UD), and signed-directed (SD).

1.0.2 Four Graph Classes

Unsigned, Undirected Graph Solvers: Unsigned, Undirected (UU) graph systems are known to be the simplest of the four problems that are in the scope of this thesis since the graph Laplacian results in a symmetric, diagonally dominant matrix with a known null-space. For UU graph Laplacian linear systems, Algebraic multigrid (AMG) provides an attractive framework due to the robustness and scalability of many multigrid algorithms and the lack of geometrical structure available for a graph. The success of AMG algorithms comes from combining two complementary error reducing processes: local *relaxation* (also known as *smoothing*) and *coarse-grid correction.* Local relaxation quickly reduces the local oscillatory components of the error. The error that remains after relaxation is usually referred to as *algebraically smooth error.* AMG methods use coarse-grid correction to enhance local relaxation. The aim of the coarse-grid correction is to construct coarse spaces so that the smooth error left by the relaxation process is accurately represented in the coarse space while ensuring that the coarse subspace is of a much smaller dimension than the fine space. The *restriction* and *prolongation* operators, which define the coarse-grid correction, map the fine space to the coarse space and the coarse space to the fine space, respectively. A multilevel method is obtained by recursively applying relaxation and coarse-grid correction. The heart of any multigrid algorithm is constructing the balance between the relaxation process that quickly attenuates oscillatory components of the error and constructing the coarse subspaces with adequate approximation properties. Geometric multigrid algorithms form the coarse subspaces based on the underlying geometry of the problem and inherently have good approximation properties. However, for most graph applications, the underlying geometry is either unknown or is lacking altogether. Thus, the coarse subspace needs to be constructed solely based off of the problem matrix. This case defines Algebraic Multigrid. There exists a number of AMG solvers for UU graph Laplacian systems, which include Lean Algebraic Multigrid (LAMG) [23], Combinatorial Multigrid (CMG) [15] and Degree-Aware Rooted Aggregation ([27], [28]).

Signed, Undirected Graph Solvers Unlike a UU graph, a SU graph has both positive edges and negative edges. A positive edge of a signed graph is seen as a similarity or proximity between two vertices, while a negative edge can show dissimilarity or distance. In the signed graph Laplacian representation, the negative edges correspond to positive off-diagonal entries while positive edges correspond to negative off-diagonal entries. As an example, a social network may be represented as a signed graph where positive edges represent “friends” and negative edges represent “foes.” Or, alternatively, a negative edge may represent two people whose grouping together should be avoided, as it may cause a disruption.

Chapter 3 analyzes the extension of unsigned graph Laplacian linear system solvers to signed graph Laplacian linear systems using Gremban’s expansion. For the signed variants, SU and SD, Gremban’s expansion [11] allows one to recast signed graph systems into unsigned graph systems with twice the number of unknowns and matrix non-zeros. The expansion decomposes any diagonally dominant matrix, $S = M + P$, into a diagonally dominant Z-matrix, M, and a non-negative matrix, P. Then $S\mathbf{x} = \mathbf{b}$ maybe solved by using an expanded system:

$$G\mathbf{w} = \begin{bmatrix} M & -P \\ -P & M \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ -\mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \end{bmatrix} = \mathbf{z}, \tag{1.2}$$

in which G is a diagonally dominant Z-matrix and, therefore, amenable to fast solvers designed for this class of matrices. In the case where S is an SU graph Laplacian, G is a UU graph Laplacian, and fast UU graph Laplacian solvers like LAMG are useful. Under suitable hypotheses on S, the system G is either nonsingular or has a trivial null space. In either case, the solution to the original system is found by solving the expanded system and then extracting $\mathbf{x}$. By applying Gremban's expansion, the larger linear system, $G\mathbf{w} = \mathbf{z}$, can be solved using any UU graph Laplacian solver. This thesis shows that Gremban's expansion gives theoretical insight into the interpretability of linear-solver-based ranking approaches to signed graphs. For SU and SD graphs, we show Gremban's expansion is numerically stable; in other words, the error of the original linear system is bounded by the error of the linear system for the expansion. We also demonstrate that the condition number of the expansion matrix, G, is at least the condition number of S, implying a robust UU solver is needed for the expanded system to yield a robust SU solver. We also empirically demonstrate that LAMG in conjunction with Gremban's expansion yields an efficient solution technique for several real-world SU graphs: one large set that has real-world topologies and synthetic sign structure, and another set where the sign structure comes from real-world movie rankings. By applying Gremban's expansion in conjunction with a robust UU graph Laplacian solver (e.g. LAMG), the sign structure does not affect the performance of the solver. This work is novel because we have provided a clean and simple tool to apply solvers that are already used in the graph community for unsigned graphs to signed graphs. Data scientists are no longer restricted by solver performance when determining the type of relationships, positive or negative, they wish to represent.

Unsigned, Directed Graph Solvers A graph is directed if an edge has a direction; in other words, information flows in one direction from vertex to vertex. For example, the World Wide Web is a directed graph, where the vertices are webpages and the edges represent hyperlinks from one page to another. The direction of edges plays an extremely important role in the dynamics between the vertices in UD graphs, and important information is lost if the directionality of the relationship is not represented. Despite the "directedness" of many real-world graphs, many linear solvers only focus on undirected graphs matrix representations. Unlike the graph Laplacian of an undirected graph, the graph Laplacian of a directed graph is nonsymmetric. Not only are real-world, UD graph Laplacians nonsymmetric, they are typically scale-free and small-world as well, which results in extremely difficult topologies. The current methods used in the graph community for directed graph Laplacian systems are direct solvers such as SuperLU [21] and QR Factorization [8] and iterative solvers such as Generalized Minimal Residual Method with Restart (GMRES(k)) [34] and Generalized Least Squares QR Method (LSQR) [33]. Multilevel iterative solvers for scale-free, UD graph Laplacian systems have not been widely studied, but the success of LAMG and CMG for UU graphs shows promise of developing a robust multilevel solver for UD graphs. A UD graph Laplacian has a zero column sum but possibly a nonzero row sum. Due to the zero column sum, the linear systems involving UD graphs, as there are many other linear systems that are related to directed graphs, are associated with Markov chain stationary distribution systems. Many techniques from Markov chain solvers can be applied in conjunction with ideas from LAMG. Again, many applications make use of efficient linear solvers, yet due to the lack of robust multilevel linear solvers for UD graph Laplacians, many applications for

UU graph have not been extended to UD graphs.

Chapter 4 describes and discusses our new AMG method, Nonsymmetric Lean Algebraic Multigrid (NS-LAMG), for UD graph Laplacians. Our proposed AMG algorithm pulls ideas from Adaptive Markov Chain AMG [41], Nonsymmetric Adaptive Smoothed Aggregation [37], and LAMG [23]. *Low-degree elimination* exploits the scale-free property and eliminates low-degree vertices from the linear system. Low-degree elimination in [23] is examined for UD graphs, and in many cases it drastically reduces the problem size. AMG is then applied to the reduced linear system, and the solution is projected back to the fine space with no error. This thesis numerically shows that low-degree elimination enhances AMG performance without significantly increasing the complexity of the solver. Due to the nonsymmetry of directed graph Laplacian systems, we adapt the restriction and prolongation operators from Nonsymmetric Smooth Aggregation in [37]. The operators in [37] are defined using near null-space vectors, yet, due to the zero column sum of a UD graph Laplacian, the Laplacian is singular with a known left null-space vector and a unique, strictly positive, right null-space vector. The left and right null-space vectors are then used to define the intergrid transfer operators. Thus, an adequate approximation to the right null-space vector needs to be found. We propose a semi-adaptive Markov chain stationary distribution system solver to find the right null-space vector, as $\mathcal{L}\mathbf{v} = \mathbf{0}$, where $\mathcal{L}$ is the normalized directed graph Laplacian, is reformulated as a Markov chain stationary distribution system. The semi-adaptive solver utilizes low-degree elimination and stationary AMG aggregation, and thus, is more parallelizable than traditional full-adaptive AMG Markov chain solvers. Numerically, this thesis shows that the semi-adaptive solvers perform similarly to the full-adaptive solvers due to low-degree elimination. The hierarchy of coarse operators are created by alternating low-degree elimination and coarse-graph aggregation. Using the constructed hierarchy, the right null-space vector is approximated using the semi-adaptive Markov chain solver proposed in Chapter 4, and the setup phase is concluded. This thesis compares GMRES(k) to NS-LAMG numerically and shows that for real-world graphs NS-LAMG out performs

GMRES(k). NS-LAMG provides a robust solver for UD graph Laplacians linear systems and is the first algebraic multigrid solver for such systems. NS-LAMG is superior to traditional nonsymmetric iterative solvers. By providing a robust, scalable solver for UD graph Laplacian systems, data scientists will have the capability to explore UD graphs without simplifying to an UU graph or restricting themselves to a subset of real-world graphs with a small number of vertices.

Signed, Directed Graph Solvers SD graphs are also of interest in the graph community. SD graphs are a combination between signed graphs and directed graphs, which typically make their associated linear systems difficult to solve. A simple example of a SD graph is a rating scheme where individuals of a team rate other members. If each team member was asked to rate every other team member's ability with respect to their own ability, then links between the vertices would be signed, as well as directed with respect to each member. This thesis also shows how Gremban's expansion can be generalized to any (possibly nonsymmetric) diagonally dominant matrix. Gremban's expansion could then expand SD graphs to UD graphs. By combining Gremban's expansion with NS-LAMG we can efficently solve SD graph Laplacian linear systems.

Data scientists would like to solve graph Laplacian linear systems involving all four types of graphs: UU, SU, UD, and SD graphs. As the size and availability of data continues to grow, traditional methods for solving graph Laplacian systems are infeasible and robust, scalable linear solvers are needed for many data analysis applications. The two extensions of AMG provided in this thesis are useful in the graph community since they provide robust solvers for different representations of graph Laplacians linear systems than just UU graphs. By providing robust solvers for different types of graph representations, new applications and avenues in the graph community can be scalably explored. Since UU graph Laplacian solvers were the main graph Laplacian solver available, data scientists were restricted in their modeling choices. Now, since this thesis has developed robust linear solvers for SU, SD, and UD graph Laplacians there is an opportunity in the graph community for UU graph applications, such as graph ranking, centerpiece subgraph problems, and clustering to be adapted for different modeling choices.

1.0.3 Nonsymmetric AMG Theory

Due to the successfulness of NS-LAMG for directed graphs, described in Chapter 4, Chapter 5 fills in gaps in the theory behind AMG for nonsymmetric problems. Algebraic multigrid is traditionally motivated for symmetric positive definite (SPD) linear systems and the convergence theory of AMG for SPD matrices is also relatively well-understood. However, significantly less work has been done studying the convergence of nonsymmetric AMG (NS-AMG). In this chapter, we build the theory for general, nonsingular, nonsymmetric matrix. The theory can easily be applied to UD graph Laplacian systems where the kernel can be projected from the linear system. In some cases, applying standard (symmetric) AMG approach to nonsymmetric problems can be effective. However, often a direct application of symmetric AMG on a nonsymmetric problem may fail without appropriate modifications of AMG components, as in [6]. Specialized nonsymmetric AMG techniques have been successfully developed for a number of specific applications. For example, Markov chain transition matrices, though nonsymmetric, have a number of nice properties, and iterative methods for Markov-chain stationary distribution systems have been widely studied. Various AMG techniques for Markov chains include [40, 4, 46]. Theoretical convergence results for iterative aggregation/disaggregation methods (IAD) are well known [17, 18, 25, 12], and [17] showed that IAD is equivalent to the two-grid nonsymmetric AMG (NS-AMG) method [40]. Various other efforts have been made to develop robust AMG solvers applicable to nonsymmetric linear systems [29, 35, 26, 30, 51, 49, 38]. However, despite a number of effective AMG methods for nonsymmetric problems, there remains a lack of rigorous motivation or theory explaining why and when the algorithms converge.

Typically the coarse-grid correction for NS-AMG is an oblique (non-orthogonal) projection with respect to any reasonable inner product. Meaning that, in some cases, the coarse-grid correction can actually increase the global error. This makes the convergence theory much more difficult to achieve, as relaxation must overcome any increase in error from the coarse-grid correction. Two-grid convergence of NS-AMG in [6] is considered by studying the application of AMG to an equivalent linear system with respect to the singular values. Let H be any nonsingular, nonsymmetric matrix. Let $H := U\Sigma V^*$ be the singular value decomposition (SVD) of H and define the orthogonal matrix $Q := VU^*$. Then the original system $H\mathbf{x} = \mathbf{b}$ can be reformulated as an SPD system in two ways:

$$QH\mathbf{x} = Q\mathbf{b} \tag{1.3}$$

$$HQ\mathbf{y} = \mathbf{b} \text{ for } \mathbf{x} = Q\mathbf{y}. \tag{1.4}$$

The matrices $QH = \sqrt{H^*H}$ and $HQ = \sqrt{HH^*}$ are SPD matrices that have the same singular-value distribution as H. Analysis of NS-AMG is performed using the energy norm with respect to QH ($\|\cdot\|_{QH} = \sqrt{\langle QH\cdot, \cdot\rangle}$). In practice, Q is not formed since this would require computing an SVD of H, but simply provides a framework with which to consider AMG convergence. Sufficient conditions for two-grid convergence in the energy-norm with respect to QH is broken into two parts [6]: (i) the prolongation operators satisfies the nonsymmetric strong approximation property, a generalization of the well-known strong approximation property with respect to the energy norm of QH, and (ii) coarse-grid correction is *stable*, i.e the error propagation operator of the coarse-grid correction is bounded, independent of problem size. This thesis fills in gaps in nonsymmetric AMG convergence theory, building off of the theory presented in [6]. In particular, the standard AMG approximation properties are generalized to the nonsymmetric setting by introducing approximation properties on both the restriction and prolongation operators. Known approximation-property relations for the SPD case, for example, the strong approximation property implies the weak approximation property, are then proved in the nonsymmetric setting. The focal point of

Chapter 5 is quantifying when the error propagation operator of the coarse-grid correction is stable. A quality measure of the relation between the restriction and prolongation operators is introduced that ensures stability and, therefore, two-grid convergence, when coupled with approximation properties. It is the goal of this work to further the theoretical development of NS-AMG, in the hopes to better understand when and why NS-AMG converges.

Chapter 2

Background

2.1 The Graph Laplacian for Unsigned, Undirected Graphs

A graph is most commonly represented by a unsigned, undirected (UU) relationship as it results in symmetric matrix representations. A UU graph, $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$, relates a set of n vertices, $\mathcal{V}$, by m connections or edges in the set $\mathcal{E}$ with weights, w. An edge $(i, j) \in \mathcal{E}$ between two vertices i and j is an undirected or symmetric relationship, meaning (j, i) is also in $\mathcal{E}$ and $w_{ij} = w_{ji}$. Additionally, edge (i, j) is a positive relationship, with $w_{ij} > 0$, as its size is proportional to the strength of affinity between i and j. Assume $\mathcal{G}$ is connected, i.e. there exist a path from any two vertices, with no self loops, that is $(i, i) \notin \mathcal{E}$. The Laplacian associated with a UU $\mathcal{G}$ is represented by

$$L = D - A \in \mathbb{R}^{n \times n}, \tag{2.1}$$

where the adjacency matrix, $A \in \mathbb{R}^{n \times n}$, and diagonal degree matrix, $D \in \mathbb{R}^{n \times n}$, are defined as

$$A_{ij} = \begin{cases} w_{ji} & (i, j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}, \tag{2.2}$$

and

$$D_{ij} = \begin{cases} d_i & i = j \\ 0 & \text{otherwise} \end{cases}, \tag{2.3}$$

respectively, and $d_i = \sum_i w_{ij}$ is known as the *degree* of vertex i.

The matrix L is singular with known kernel, $L\mathbf{1} = \mathbf{0}$. For the linear system, $L\mathbf{x} = \mathbf{b}$, if L is symmetric, $\mathbf{b}$ is in the range of L only if $\mathbf{1}^t\mathbf{b} = 0$. If $\mathbf{b}$ is not in the range, we are interested in $L^\dagger\mathbf{b}$, where where $L^\dagger$ represents the Moore-Penrose pseudo-inverse of L. Because L has a known one-dimensional kernel, $L^\dagger$ can be calculated by using projections and a SPD solve,

$$\mathbf{x} = \left(L + \frac{\alpha}{n}\mathbf{1}\mathbf{1}^t\right)^{-1}\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^t\right)\mathbf{b}, \tag{2.4}$$

for any $\alpha > 0$. This is an important discussion for the signed case, as the matrices of interest are not always singular, which will be demonstrated later in this section.

For UU graphs, matrix L can be symmetrically factored using its incidence matrix, $E \in \mathbb{R}^{n\times m}$, which maps the edges to the vertices. Let the e-th edge be $(i, j) \in \mathcal{E}$. Then e is oriented as follows for $i < j$:

$$E_{k,e} = \begin{cases} +1 & k = i \\ -1 & k = j \\ 0 & otherwise \end{cases}. \tag{2.5}$$

Let W be a diagonal matrix such that $W_{e,e} = w_{ij}$. This yields the factorization

$$L = EWE^t. \tag{2.6}$$

This factorization and the quadratic form,

$$\mathbf{x}^t L\mathbf{x} = \sum_{(i,j)\in\mathcal{E}} w_{ij}(x_i - x_j)^2, \tag{2.7}$$

are powerful tools used for deriving properties of solutions to $L\mathbf{x} = \mathbf{b}$, designing numerical solvers, and performing numerical analysis of such solvers. While conventionally used for UU graphs, these formulation tools can be generalized to signed graphs as well as directed graphs. As mentioned in the Section 1, there exists many AMG methods for UU graph Laplacain linear systems, i.e LAMG [23], CMG [15] and Degree-Aware Rooted Aggregation ([27], [28]). The next section outlines the basics of AMG and LAMG, as LAMG is used in the numerical testing of Section 3 and is the basis of NS-LAMG, outlined in Section 4.

2.2 Basics of AMG

Multigrid was first developed for PDE-type problems discretized on uniform grids. It is known for its robustness and scalability, which makes multigrid an attractive option for graphs since the size of graphs have become increasingly large due to the vast collection and accessibility of data. Multigrid was first formulated for symmetric, positive-definite (SPD) matrices. For UU graph Laplacians, L is symmetric, possibly indefinite, but has a known kernel that can be easily projected from the system. Let B be an $n \times n$ SPD matrix. The two main components of any multigrid algorithm are smoothing and coarse-grid correction. First, smoothing quickly attenuates the oscillatory components of the error and is typically a stationary residual-based iterative method, which is typically of the form

$$\mathbf{x} \leftarrow \mathbf{x} + M^{-1}\mathbf{r} \qquad \mathbf{r} := \mathbf{b} - B\mathbf{x},$$

where M^{-1} is a computation inexpensive preconditioner that approximates B^{-1} and $\mathbf{r}$ is the residual. Most stationary (residual-based) iterative methods, e.g. weighted-Jacobi and

Gauss-Seidel, are easy to implement and require a minimal amount of operations per iteration. Second, coarse-grid correction projects the smooth error onto a coarse subspace to efficiently reduce smooth error components. The coarse-grid correction involves *transfer operators*, i.e. the prolongation and restriction operators, that transfer information between fine and coarse grids. Define the fine and coarse grids as the space $\mathbb{R}^n$ and $\mathbb{R}^{n_c}$, respectively. The interpolation operator, $P \in \mathbb{R}^{n \times n_c}$, maps the coarse grid to the fine grid while the restriction operator, $R^t \in \mathbb{R}^{n_c \times n}$, maps the fine grid to the coarse grid. If B is symmetric and $R = P$, then the coarse operator, $B_c = R^t B P$, is symmetric. The two-grid method for solving $B\mathbf{x} = \mathbf{b}$ is then defined in Algorithm 1. In Algorithm 1, $\mathbf{e}$ is the error in the current iterate and $\mathbf{r} = \mathbf{b} - B\mathbf{x}$ is the residual. The fine level problem matrix and the residual are restricted to the coarse subspace and the coarse approximation of the error, $\mathbf{e}_c$, is then solved for. The approximated error is then interpolated to the fine grid to obtain an approximation to the fine grid error, $\tilde{\mathbf{e}} = P\mathbf{e}_c$. The two-level coarse-grid correction is

$$\mathbf{x} \leftarrow \mathbf{x} + P(R^t B P)^{-1} R^t \mathbf{r} \tag{2.8}$$

and has the associated *error propagation operator*

$$\mathbf{e} \leftarrow (I - P(R^t B P)^{-1} R^t B)\mathbf{e}. \tag{2.9}$$

Define the error propagation operator as $(I - \Pi) := (I - P(R^t B P)^{-1} R^t B)$. If B is a symmetric positive definite matrix and $R = P$, then the error propagation operator, $(I - \Pi)$, is an orthogonal projection in the B-norm ($\| \cdot \|_B = \sqrt{\langle B \cdot, \cdot \rangle}$), meaning

$$(I - \Pi)^2 \;=\; (I - \Pi) \tag{2.10}$$

Algorithm 1 Two-Level AMG($\mathbf{x}, \mathbf{b}, \nu_1, \nu_2$)

Input: Known right-hand side, $\mathbf{b}$, current iterate, $\mathbf{x}$, number of pre-smoothing iteration, ν_1, and number of post-smoothing iterations, ν_2. Implicit input is the two-level hierarchy (B, P, R).

Output: Updated iterate, $\mathbf{x}$.

```
repeat
    Do $\nu_1$ smoothing steps on $B\mathbf{x} = \mathbf{b}$.
    Compute residual $\mathbf{r} = \mathbf{b} - B\mathbf{x} = B\mathbf{e}$.
    Solve $B_c\mathbf{e}_c = R^t\mathbf{r}$, where $B_c = R^tBP$.
    Correct $\mathbf{x} \leftarrow \mathbf{x} + P\mathbf{e}_c$
    Do $\nu_2$ smoothing steps on $B\mathbf{x} = \mathbf{b}$.
until convergence criterion satisfied
```

and

$$\langle B(I-\Pi)\mathbf{y},\mathbf{z}\rangle = \langle B\mathbf{y},(I-\Pi)\mathbf{z}\rangle. \tag{2.11}$$

The error propagation operator projects the error in the direction of the range of the interpolation operator onto the space spanned by the null-space of R^tB. To form a full V-cycle, Algorithm 1 is recursively called. Multilevel notation for the hierarchy involves K levels, with $k=1$ as the finest, and $k=K$ as the coarsest level. The interpolation and restriction operators between the levels are denoted by P^k_{k+1} and $(R^k_{k+1})^t$, respectively, and the coarse problem matrices can be written as $B_{k+1}=(R^k_{k+1})^tB_kP^k_{k+1}$ on each level. The multilevel method for solving $B\mathbf{x}=\mathbf{b}$ is then defined in Algorithm 2.

AMG differs from classical geometric multigrid in the way that the interpolation operators are formed. Geometric multigrid forms P based on the underlying geometry of the problem. However, for graphs, the underlying geometry is unknown. Thus, P needs to be defined solely based on the problem matrix B. The coarse-grid correction needs to be designed to complement the given relaxation process and must attenuated the error that remains. Classical AMG is based off of the assumption that geometrically smooth error is in or near the null-space of B. Assuming the problem matrix B is SPD and has been scaled so that its largest eigenvalue equals one, let $\mathbf{e}$ be a small normalized eigenvalue of B such that $B\mathbf{e}=\lambda\mathbf{e}$, thus,

$$\lambda = \mathbf{e}^tB\mathbf{e} \ll 1. \tag{2.12}$$

The constant vector, $\mathbf{1}$, is a geometrically smooth vector, and from the assumption that smooth error must be in the near null-space of B, we can assume that B has a row sum of zero. By using this assumption, we can expand $\mathbf{e}^tB\mathbf{e}$,

Algorithm 2 MultiLevel AMG($\mathbf{x}_k, \mathbf{b}_k, \nu_1, \nu_2$)

```
Input: Known right-hand side, $\mathbf{b}_k$, current iterate, $\mathbf{x}_k$, number of pre-smoothing iteration, $\nu_1$, and number of post-smoothing iterations, $\nu_2$. Implicit input is the multilevel hierarchy $(B_k, P^k_{k+1}, (R^k_{k+1})^t)$.
Output: Updated iterate, $x_k$.
  repeat
    Do $\nu_1$ smoothing steps on $B_k\mathbf{x}_k = \mathbf{b}_k$.
    Compute residual $\mathbf{b}_{k+1} = (R^k_{k+1})^t(\mathbf{b}_k - B_k\mathbf{x}_k)$
    if $k = K - 1$ then Solve $B_K\mathbf{x}_K = \mathbf{b}_K$
    else $\mathbf{x}_{k+1} \leftarrow$ MultiLevel AMG $(\mathbf{x}_k, \mathbf{b}_k, \nu_1, \nu_2)$
    end if
    Correct $\mathbf{x}_k \leftarrow \mathbf{x}_k + P^k_{k+1}\mathbf{x}_{k+1}$
    Do $\nu_2$ smoothing steps on $B_k\mathbf{x}_k = \mathbf{b}_k$.
  until convergence criterion satisfied
```

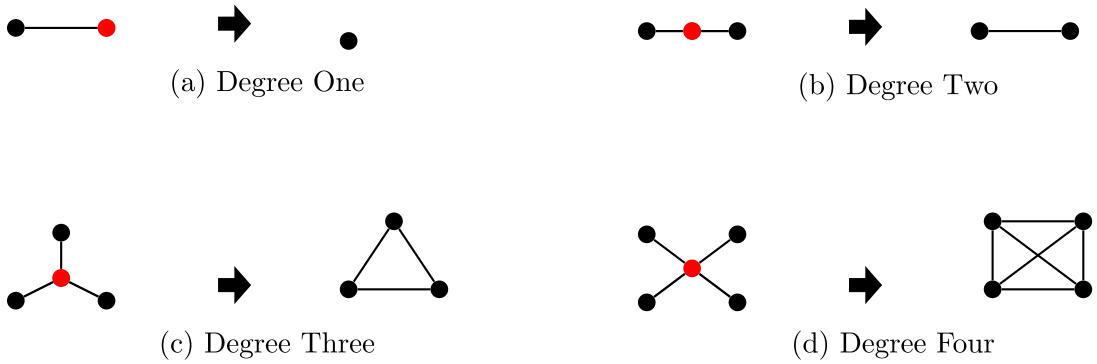
$$\mathbf{e}^t B\mathbf{e} = \sum_{i<j}(-b_{ij})(e_i - e_j)^2 \ll 1, \tag{2.13}$$

which helps define the *strength of connection* between vertices as smooth error varies slowly in the direction of relatively large coefficients of the problem matrix. There are many ways to approach coarsening, but for many AMG algorithms, coarsening is based on a *strength-of-connection matrix*. The strength-of-connection matrix, S, is usually defined by a function on the matrix B, with small connections thrown away by using a strength threshold parameter. Coarse grid aggregates, where the fine grid degrees of freedom are associated with a one coarse grid degree of freedom are chosen, such that the grid is coarsened in the direction of strong connections, i.e. the vertices i and j are aggregated together if the entry S_{ij} is large.

2.3 Lean Algebraic Multigrid

Livne and Brandt modified AMG specifically for UU graph Laplacians to create LAMG. LAMG was able to exploit information about UU graph Laplacians to create a robust and scalable multigrid algorithm for a wide variety of UU graphs, including scale-free and mesh-like graphs. The three main contributions of LAMG are low-degree elimination, caliber-one interpolation with an energy correction step, and the use of affinity to compute a new type of strength-of-connection matrix. Many of LAMG's methods are applied in our NS-LAMG algorithm for UD graph Laplacians linear systems, which will be discussed Chapter 4.

LAMG's low-degree elimination phase projects low-degree vertices from the linear system using the Schur complement. When a vertex is eliminated, the neighboring vertices get connected to each other if they were not already connected. The goal of the low-degree elimination phase is to reduce the problem size while not considerably increasing the number of nonzeros of the coarsened problem matrix. Recall, the off-diagonal nonzeros represent the edges in the graph. LAMG eliminates any vertex with degree four or less, meaning that the vertex is symmetrically attached to at most four vertices. Figure (2.1) depicts the elimination of a vertex that has degree one, two, three, and four. The left is the original system, and the right is after the red vertex is projected out of the system. In Figure (2.1d), two additional edges were created by eliminating a vertex of degree four. Real-world graphs are often scale-free, meaning that they have many low-degree vertices. By eliminating low-degree vertices, the size of the coarsened problem is drastically reduced, while the number of nonzeros in the problem matrix is maintained. Additionally, after the elimination, the interpolation of the error from the coarse graph to the fine graph is exact. As a result, if elimination is performed before each aggregation level, it will significantly reduce the cost of each cycle with no distortion of the solution.


(a) Degree One

(b) Degree Two

(c) Degree Three

(d) Degree Four

Figure 2.1: Low-Degree Elimination for UU Graphs

For the aggregation phase, for PDE-type problems, it is known that a second order or higher interpolation operator [5], meaning a fine vertex is represented on the coarse subspace by more than one coarse vertex, improves convergence. For graphs, a second order interpolation operator creates fill-in on the coarser graphs, which is unintuitive. Fill-in represents adding more edges or connections between the aggregates that cause the coarse graphs Laplacians to become dense, requiring more storage for the coarser levels than if a caliber-one operator was used. To remove this hindrance, a caliber-one or order-one operator is employed, in which each fine vertex is represented on the coarse subspace by only one coarse vertex. LAMG is an accelerated, caliber-one, aggregation-AMG algorithm that utilizes an iterate recombination phase, also know as k-cycles [31], to overcome the inaccuracy of the order-one interpolation operator.

An effective coarse grid hinges on the choice of good aggregates. LAMG creates an affinity matrix to describe the "distance" of the vertices in the aggregation phase. Classical AMG uses the entires in the adjacency matrix A, but makes incorrect decisions for non-local graphs. The affinity matrix contains the algebraic distance between the vertices, which is trying to capture aggregates that are strongly coupled together in the low energy or smooth vectors. To form S, a set of K test vectors, $\mathbf{x}_1, \ldots, \mathbf{x}_K$, is generated. Each test vector is the result of taking a random vector, randomly containing entries in $[-1, 1]$, and applying ν relaxation sweeps to $L\mathbf{x}_i = \mathbf{0}$. Let $X \in \mathbb{R}^{n \times K}$ be a matrix whose columns are $\mathbf{x}_1, \ldots, \mathbf{x}_K$, i.e.

$$X = \begin{bmatrix} \mathbf{x}_1 & \ldots & \mathbf{x}_K \end{bmatrix}. \tag{2.14}$$

Let X^j be the jth row of X and let (X^j, X^i) be the inner product between two such rows. Then define the affinity between vertex u an v, which is denoted by s_{uv}, as

$$S_{uv} = 1 - \frac{|(X^u, X^v)|^2}{(X^u, X^u)(X^v, X^v)}. \tag{2.15}$$

Note that $S_{uv} \in [0, 1]$, and it is assumed $S_{uv} = 0$ if $L_{uv} = 0$. These key modifications have been shown to be efficient and robust for most topologies of UU graphs. Our own implementation of LAMG, implemented in the programming language Julia [36], is an efficient and robust solver for UU graphs. grid

Chapter 3

Gremban's Expansion for Signed Graphs

3.1 The Graph Laplacian for Signed Undirected Graphs

Provided in this chapter are the details of Gremban's expansion. Gremban's expansion extends unsigned graph Laplacian solvers to signed graph Laplacian systems with no changes to the solver. For SU graphs denoted by $\mathcal{G}^{\pm}(\mathcal{V}, \mathcal{E}^{+} \cup \mathcal{E}^{-}, w)$, there are both positively weighted edges, $w_{ij} > 0$ for $(i, j) \in \mathcal{E}^{+}$, and negatively weighted edges, $w_{ij} < 0$ for $(i, j) \in \mathcal{E}^{-}$. For simplicity, we assume there are no contradictory edges; that is, $(i, j) \in \mathcal{E}^{+} \Rightarrow (i, j) \notin \mathcal{E}^{-}$, and, conversely, $(p, q) \in \mathcal{E}^{-} \Rightarrow (p, q) \notin \mathcal{E}^{+}$. Thus, $\mathcal{E}^{+} \cap \mathcal{E}^{-} = \emptyset$, and $\mathcal{E}^{+} \cup \mathcal{E}^{-} = \mathcal{E}$. We will also assume the signed graphs are connected. For SU graphs to be connected, the underlying UU graph must be *connected*, i.e. there exists a path from any two vertices. For SD graphs the underlying UD must be *strongly connected*, i.e there exists a directed path from any two vertices. The matrix forms associated with $\mathcal{G}^{\pm}$, for SU graphs, are easily generalized from Section 2.1. The signed Laplacian associated of graph $\mathcal{G}^{\pm}$ is represented by

$$L^{\pm} = D^{\pm} - A^{\pm} \in \mathbb{R}^{n \times n}, \tag{3.1}$$

where the signed adjacency matrix, $A^{\pm} \in \mathbb{R}^{n \times n}$, and signed diagonal degree matrix, $D^{\pm} \in \mathbb{R}^{n \times n}$, are defined as

$$A^{\pm}_{ij} = \begin{cases} w_{ji} & (i,j) \in \mathcal{E}^{+} \cup \mathcal{E}^{-} \\ 0 & \text{otherwise} \end{cases}, \tag{3.2}$$

and

$$D^{\pm}_{ij} = \begin{cases} d_i & i = j \\ 0 & \text{otherwise} \end{cases}, \tag{3.3}$$

respectively, and $d_i = \sum_j |w_{ij}|$ is known as the *total degree*, i.e., the sum of the absolute values of the wighted edges of vertex i. The signed adjacency matrix will have some negative entries and the signed combinatorial Laplacian has some positive off-diagonal entries. The incident factorization is extended to signed graphs such that the products of nonzero entries in each column of the incidence matrix, E, are equal to the negative of the sign of each edge weight. Let the e-th edge be $(i,j) \in \mathcal{E}$. Then e is oriented as follows for $i < j$:

$$E_{k,e} = \begin{cases} 1 & k = i \\ \pm 1 & k = j \text{ such that } E_{ie}E_{je} = -sign(w_{ij}) \\ 0 & otherwise \end{cases} \cdot \tag{3.4}$$

Let $W_{e,e} = |w_{ij}|$. Then the Laplacian is factored as in (2.6),

$$L^{\pm} = EWE^{t}. \tag{3.5}$$

Note that $L^{\pm}$ of an SU graph is a symmetric, diagonally dominant matrix.

The goal is to apply tools that work well for combinatorial Laplacians of UU graphs to solve $L^{\pm}\mathbf{x} = \mathbf{b}$ for SU graphs. For $\mathcal{G}^{\pm}$, we will show that, in special situations, $L^{\pm}$ is singular. In real-world datasets, $L^{\pm}$ is usually nonsingular. We address the singular case first. When $L^{\pm}$ is singular, the eigendecomposition is directly related to associated combinatorial Laplacian, $\tilde{L}^{\pm}$, that is generated by reversing the sign of all the negative edge weights. To formalize this, we require the concept of a *balanced* signed graph.

Definition 3.1.1. *A connected signed graph, $\mathcal{G}^{\pm}(\mathcal{V}, \mathcal{E}^{+} \cup \mathcal{E}^{-}, w)$, with nonzero edge weights is* ***balanced*** *if $\mathcal{V}$ can be partitioned into the two groups $\mathcal{U}$ and $\mathcal{W}$, such that $(i,j) \in \mathcal{E}^{+}$ implies either both vertices i, j are in $\mathcal{U}$, or both are in $\mathcal{W}$, and $(i,j) \in \mathcal{E}^{-}$ implies one of the two vertices is in $\mathcal{U}$ and the other is in $\mathcal{W}$.*

Equivalently, unbalanced graphs can be defined as graphs containing a cycle with an odd number of negative edges. If a graph is balanced, then the spectrum is easily relatable to the corresponding unsigned Laplacian, as shown in the following Theorem 1. A bipartition, $\mathbf{y}$, of the set of vertices, $\mathcal{V}$, is defined as $y_i = 1$ for $i \in \mathcal{U}$ and $y_j = -1$ for $j \in \mathcal{W}$, where $\mathcal{U} \cup \mathcal{W} = \mathcal{V}$ and $\mathcal{U} \cap \mathcal{W} = \emptyset$.

Theorem 1. *Let $L^{\pm}$ be a signed Laplacian matrix of the balanced connected graph, $\mathcal{G}^{\pm}$, with bipartition $\mathbf{y}$ and eigenvalue decomposition $L^{\pm} = U \Lambda U^t$. Let $\tilde{L}^{\pm} = diag(\mathbf{y}) L^{\pm} diag(\mathbf{y})$. Then $\tilde{L}^{\pm}$ is the corresponding Laplacian matrix of the unsigned graph Laplacian of $\mathcal{G}$. The eigenvalue decomposition of the Laplacian matrix, $\tilde{L}^{\pm}$, is similar to $L^{\pm}$, by $\tilde{L}^{\pm} = \tilde{U} \Lambda \tilde{U}^t$ where $\tilde{U} = diag(\mathbf{y}) U$. Moreover, $\mathbf{y}$ is the kernel of $L^{\pm}$.*

Proof. Order the vertices so that $\mathcal{U}$ is first and $\mathcal{W}$ is second and let block $L^{\pm}_{12}$ represent the edges between $i \in \mathcal{U}$ and $j \in \mathcal{W}$. A balanced graph implies that all positive off-diagonal entries in $L^{\pm}$ occur in $L^{\pm}_{12}$ and $L^{\pm}_{12} \geq 0$. The similarity transform $\tilde{L}^{\pm} = \mathrm{d}iag(\mathbf{y}) L^{\pm} \mathrm{d}iag(\mathbf{y})$ makes these entries negative (with their symmetric counterparts) and leaves all entries in the other blocks of $L^{\pm}$ alone. The rest of the results are derived easily via the similarity transform and the fact that $\mathbf{1}$ is the kernel of $\tilde{L}^{\pm}$. □

The partition of a balanced graph can be found by a depth-first traversal, assigning each vertex to a partition such that the balance property is fulfilled [19]. Additionally, the same process can be used to determine if the graph is unbalanced. If $\mathcal{G}^{\pm}$ is balanced, the spectrally equivalent Laplacian system $\tilde{L}^{\pm}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$ can be solved, where $\tilde{\mathbf{b}} = \text{diag}(\mathbf{y})\mathbf{b}$ and the solution to the original linear system is given by $\mathbf{x}$ =diag$(\mathbf{y})^{-1}\tilde{\mathbf{x}}$. Therefore, in what follows, we assume $\mathcal{G}$ is unbalanced. Moreover, it is unlikely that a real-world graph is balanced unless an application has special structure. The incidence factorization for $\mathcal{G}^{\pm}$, shown in (3.4), is used to show that $L^{\pm}$ is always at least positive semi-definite. From the factorization (3.5), we can derive the quadratic form

$$\mathbf{x}^t L^{\pm}\mathbf{x} = \sum_{(i,j)\in\mathcal{E}^+} |w_{ij}|(x_i - x_j)^2 + \sum_{(p,q)\in\mathcal{E}^-} |w_{pq}|(x_p + x_q)^2, \tag{3.6}$$

which is non-negative for any $\mathbf{x}$. For an unbalanced, connected $\mathcal{G}^{\pm}$, the quadratic form is strictly positive, implying matrix $L^{\pm}$ is strictly positive definite. If $\mathbf{x}^t L^{\pm}\mathbf{x} = 0$, then for each edge $(i, j) \in \mathcal{E}$ we have

$$|w_{ij}|(x_i - sign(w_{ij})x_j)^2 = 0 \tag{3.7}$$

$$x_i = sign(w_{ij})x_j. \tag{3.8}$$

The above equations show that the x_i's are equal, or they must alternate sign depending if they are connected by a positive edge or negative edge, respectively. Since the graph is connected, all $|x_i|$'s must be equal, and thus $\mathbf{x}$ gives a bipartition of the vertices that contradicts the definition of an unbalanced graph [19]. Thus, the signed Laplacian of an unbalanced $\mathcal{G}^{\pm}$ is nonsingular and $L^{\pm}\mathbf{x} = \mathbf{b}$ can be solved for any $\mathbf{b}$ without projection.

When a connected, signed graph is balanced $L^{\pm}$ is singular, as shown in Theorem 1, and thus the condition number of $L^{\pm}$ is defined as

$$\kappa(L^{\pm}) = \frac{\sigma_n}{\sigma_2}, \tag{3.9}$$

where σ_n is the largest eigenvalue and σ_2 is the smallest eigenvalue greater than zero of $L^{\pm}$. For a connected, unbalanced graph, $L^{\pm}$ is nonsingular, and the condition number of $L^{\pm}$ is defined as

$$\kappa(L^{\pm}) = \frac{\sigma_n}{\sigma_1}, \tag{3.10}$$

where σ_n and σ_1 are the largest and smallest eigenvalue of $L^{\pm}$, respectively. Thus, if the sign structure of a balanced graph changes slightly and becomes a *weakly unbalanced* graph, e.g. one edge changes sign, the condition number of the unbalanced version of the graph could potentially be much worse than the balanced version. Thus, SU graph Laplacians can be arbitrarily poorly conditioned. The condition number is dependent on the sign structure of the graph, thus the goal is to constuct a solver for SU graph Laplacian systems such that the performance is independent of the sign structure.

3.2 Gremban's Expansion

In [11], the author shows that any symmetric diagonally dominant (SDD) matrix can be expanded into a symmetric Z-matrix. The solution to the linear system involving the expanded matrix yields the solution of the linear system involving the original matrix. An important distinction to be made between signed and unsigned graph Laplacians is regarding the off-diagonal elements. Signed Laplacians have some positive off-diagonal elements while unsigned Laplacians have strictly nonpositive off-diagonal elements. The following defines a *diagonally dominant Z-matrix*, which is a generalization of an unsigned Laplacian:

Definition 3.2.1. *A matrix M is a **Diagonally Dominant Z-matrix** if it is diagonally* *dominant, $m_{ii} \geq \sum_{j=1}^{n} |m_{ij}|$, with positive diagonal, $m_{ii} > 0$ for every i, and has non-positive off-diagonal elements, $m_{ij} \leq 0$ for every $i \neq j$.*

With this definition, M can be strongly diagonally dominant, $m_{ii} > \sum_{j=1}^{n} |m_{ij}|$, for some i, but is not required to be symmetric. Note that diagonally dominant M-matrices, commonly used in the literature, are a subset of diagonally dominant Z-matrices. Any diagonally dominant matrix can be decomposed into the sum of a diagonally dominant Z-matrix and a non-negative matrix. Now that the required notation has been provided, we define the *Gremban expansion* matrix:

Definition 3.2.2. *Let S be diagonally dominant with positive diagonal. We decompose $S = M + P$ such that M is a diagonally dominant Z-matrix, and P has purely non-negative entries and zero diagonal. Define the* ***Gremban Expansion matrix*** *G as*

$$G = \begin{bmatrix} M & -P \\ -P & M \end{bmatrix}.$$

Note that G is a diagonally dominant Z-matrix. In the following definition we see that if S is a signed graph Laplacian, then G is an unsigned graph Laplacian.

Definition 3.2.3. *Let $L^{\pm}$ be the graph Laplacian of a signed graph, $\mathcal{G}^{\pm}(\mathcal{V}, \mathcal{E} = \mathcal{E}^{+} \cup \mathcal{E}^{-}, w)$. Then Gremban's Expansion matrix, G, of $L^{\pm}$ is an unsigned graph Laplacian of graph, $\tilde{\mathcal{G}}(\tilde{\mathcal{V}}, \tilde{\mathcal{E}}, \tilde{w})$, where $\tilde{\mathcal{V}} = \{1, \cdots n, n+1, \cdots, 2n\}$ and the edge set $\tilde{\mathcal{E}}$ is related to the edge sets $\mathcal{E}^{+}$ and $\mathcal{E}^{-}$ as follows. If $(i,j) \in \mathcal{E}^{+}$ then $(i,j), (i+n, j+n) \in \tilde{\mathcal{E}}$ with $\tilde{w}_{i,j} = \tilde{w}_{i+n,j+n} = w_{i,j}$ and if $(i,j) \in \mathcal{E}^{-}$ then $(i, j+n), (i+n, j) \in \tilde{\mathcal{E}}$ with $\tilde{w}_{i,j+n} = \tilde{w}_{i+n,j} = |w_{i,j}|$.*

By expressing the Gremban matrix, G, as graph, $\tilde{\mathcal{G}}$ by Definition 3.2.3, the structure of the matrix can be investigated. Figure 3.1 depicts $\tilde{\mathcal{G}}$ where the green ovals represent the two distinct communities, $\mathcal{V}_1$ and $\mathcal{V}_2$, that are each connected internally by the associated positive edges. The negative edges, represented in red, form the connection between the two

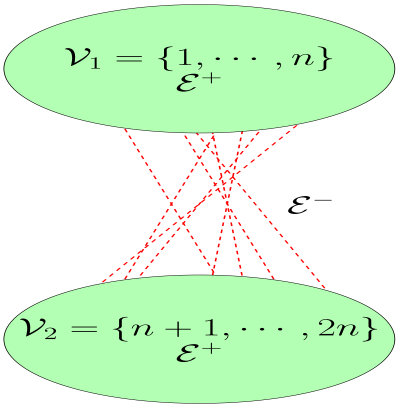

$\mathcal{V}_1 = \{1, \cdots, n\}$
$\mathcal{E}^+$
$\mathcal{E}^-$
$\mathcal{V}_2 = \{n+1, \cdots, 2n\}$
$\mathcal{E}^+$

Figure 3.1: Gremban expansion graph $\tilde{\mathcal{G}}$

communities, $\mathcal{V}_1$ and $\mathcal{V}_2$. In order to get from a vertex in $\mathcal{V}_1$ to a vertex in $\mathcal{V}_2$ an edge in $\mathcal{E}^-$ must be transversed. Random walks are used in many ranking algorithm and a similar study of random walks on Gremban's expansion matrix can give the user an understanding of the solution. It is well known that for an unsigned graph, the k^{th} power of the associated adjacency matrix defines the number of k-length walks connecting the vertices i and j. This very useful graph mining theorem associated with unsigned graphs can be generalized to the graph associated with the Gremban expansion matrix and provides interpretability of signed graphs using traditional graph analytics. Theorem 2 shows that the k powers of the associated binary adjacency matrix of the Gremban's graph, $\tilde{A}$, defines an even or odd number of edges in $\mathcal{E}^-$ that are in a walk of length k between node i and j. For a ranking application on a signed graph, a generalization of [10], the vertices are scored by the difference of two random-walk based rankings: one involving paths with an odd number of negative or "bad" edges and one with paths with even number of "bad" edges.

Theorem 2. *Let $\mathcal{G}^{\pm}(\mathcal{V}, \mathcal{E} = \mathcal{E}^+ \cup \mathcal{E}^-, w)$ be an SU graph and $\tilde{\mathcal{G}}(\tilde{\mathcal{V}}, \tilde{\mathcal{E}}, \tilde{w})$ be the associated Gremban's graph as defined in Definition 3.2.3. Let $\tilde{A}$ be the associated binary adjacency matrix of $\tilde{\mathcal{G}}$. If $i, j \in \{1, \cdots, n\}$, then $\left(\tilde{A}^k\right)_{i,j}$ is the number of walks of length k with an even number of negative edges and $\left(\tilde{A}^k\right)_{i,j+n}$ is the number of walks of length k with an odd number of negative edges.*

Proof. Let $A = A^+ - A^-$ be the binary adjacency matrix of graph $\mathcal{G}^{\pm}$ where A^+ and A^- represent the positive and negative edges, respectfully. Then,

$$\tilde{A} = \begin{bmatrix} A^+ & A^- \\ A^- & A^+ \end{bmatrix}$$

is the binary adjacency matrix associated with, $\tilde{\mathcal{G}}$. If $i, j \in \{1, \cdots, n\}$ and there exists a walk of length one between the vertices then,

$$\tilde{A}_{i,j} = \begin{bmatrix} \mathbf{e}_i^t & \mathbf{0} \end{bmatrix} \tilde{A} \begin{bmatrix} \mathbf{e}_j \\ \mathbf{0} \end{bmatrix} = \mathbf{e}_i^t A^+ \mathbf{e}_j = 1,$$

and zero negative edges exist in the walk. If $\hat{j} = j + n$ and there exists a walk of length one between the vertices then,

$$\tilde{A}_{i,\hat{j}} = \begin{bmatrix} \mathbf{e}_i^t & \mathbf{0} \end{bmatrix} \tilde{A} \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_{\hat{j}} \end{bmatrix} = \mathbf{e}_i^t A^- \mathbf{e}_{\hat{j}} = 1,$$

and one negative edge exists in the walk. Thus, for $k = 1$ the theorem holds. Assume for $i, j \in \{1, \cdots, n\}$ and there exists a path of length $k - 1$ between the vertices that $\left(\tilde{A}^{k-1}\right)_{i,j}$ is the number of walks of length $k - 1$ that has an even number of negative edges and $\left(\tilde{A}^{k-1}\right)_{i,j+n}$ is the number of walks of length $k - 1$ that has an odd number of negative edges. We will show that for k,

$$\begin{aligned}
\tilde{A}^k_{i,j} &= \begin{bmatrix} \mathbf{e}_i^t & \mathbf{0} \end{bmatrix} \tilde{A}^{k-1}\tilde{A} \begin{bmatrix} \mathbf{e}_j \\ \mathbf{0} \end{bmatrix} && (3.11)\\
&= \sum_{q\in\{1,\cdots,n\}} \begin{bmatrix} \mathbf{e}_i^t & \mathbf{0} \end{bmatrix} \tilde{A}^{k-1} \begin{bmatrix} \mathbf{e}_q \\ \mathbf{0} \end{bmatrix} \mathbf{e}_q^t A^+ \mathbf{e}_j && (3.12)\\
&\quad + \sum_{p=q+n,q\in\{1,\cdots,n\}} \begin{bmatrix} \mathbf{e}_i^t & \mathbf{0} \end{bmatrix} \tilde{A}^{k-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_p \end{bmatrix} \mathbf{e}_p^t A^- \mathbf{e}_j, && (3.13)\\
& && (3.14)
\end{aligned}$$

is a walk of length k with an even number of negative edges. In the left sum, $\tilde{A}^{k-1}_{i,q}$ is a walk of length $k-1$ from vertex i to q that has an even number of negative edges and only a positive edge was transversed to get from vertex q to j, thus, an even number of negative edges are in the walk. In the right sum, $\tilde{A}^{k-1}_{i,p}$ is a walk of length $k-1$ from vertex i to p that has an odd number of negative edges and one negative edge was transversed to get from vertex p to j, thus, again an even number of negative edges are in the walk. If $\hat{j} = j+n$ the $\tilde{A}^k_{i,\hat{j}}$ is a walk of length k with an odd number of negative edges as seen by,

$$\begin{aligned}
\tilde{A}^k_{i,\hat{j}} &= \begin{bmatrix} \mathbf{e}_i^t & \mathbf{0} \end{bmatrix} \tilde{A}^{k-1}\tilde{A} \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_{\hat{j}} \end{bmatrix} && (3.15)\\
&= \sum_{q\in\{1,\cdots,n\}} \begin{bmatrix} \mathbf{e}_i^t & \mathbf{0} \end{bmatrix} \tilde{A}^{k-1} \begin{bmatrix} \mathbf{e}_q \\ \mathbf{0} \end{bmatrix} \mathbf{e}_q^t A^- \mathbf{e}_{\hat{j}} && (3.16)\\
&\quad + \sum_{p=q+n,q\in\{1,\cdots,n\}} \begin{bmatrix} \mathbf{e}_i^t & \mathbf{0} \end{bmatrix} \tilde{A}^{k-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_p \end{bmatrix} \mathbf{e}_p^t A^+ \mathbf{e}_{\hat{j}}, && (3.17)\\
& && (3.18)
\end{aligned}$$

with similar logic as above.

□

A simple application of Theorem 2 on a real-world signed graph is a friend-enemy social network. When $k = 2$, Theorem 2 simplifies to the classic phrase "a enemy of my enemy is my friend." For ranking applications, as in [10], for an unsigned graph, one ranking method ranked a vertex j on how similar is to vertex i by $\mathbf{e}_j^t(L^{\pm})^{\dagger}\mathbf{e}_i$. The same process could be generalized for signed graph Laplacians. By Theorem 2, $\begin{bmatrix} \mathbf{e}_j^t & \mathbf{0} \end{bmatrix} G^{\dagger} \begin{bmatrix} \mathbf{e}_i \\ \mathbf{0} \end{bmatrix}$ is a ranking that scores vertex j by how close it is to vertex i along walks that cross zero or any even number of negative edges (a friendly ranking) and the $\begin{bmatrix} \mathbf{e}_j^t & \mathbf{0} \end{bmatrix} G^{\dagger} \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_i \end{bmatrix}$ is a ranking that scores vertex j by how close it is to vertex i along walks that cross an odd number of negative edges (an unfriendly ranking). Then

$$\mathbf{e}_i^t(L^{\pm})^{\dagger}\mathbf{e}_j = \begin{bmatrix} \mathbf{e}_j & \mathbf{0} \end{bmatrix} G^{\dagger} \begin{bmatrix} \mathbf{e}_i \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{e}_j & \mathbf{0} \end{bmatrix} G^{\dagger} \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_i \end{bmatrix},$$

is the difference of the friendly ranking and the unfriendly ranking. To generalize, Theorem 2 shows that the solution to $G\mathbf{w} = \mathbf{z}$, where $\mathbf{z}^t = [\mathbf{b}^t, -\mathbf{b}^t]$, is the difference of the solution with respect to the two communities, $\mathcal{V}_1$ and $\mathcal{V}_2$. The community $\mathcal{V}_1$ can be seen as the "good" community and $\mathcal{V}_2$ as the "bad" community, and the solution can be seen as a difference of a "good" score and a "bad" score as seen by

$$\mathbf{w} = G^{\dagger} \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} - G^{\dagger} \begin{bmatrix} \mathbf{0} \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} M & -P \\ -P & M \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} M & -P \\ -P & M \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{0} \\ \mathbf{b} \end{bmatrix}.$$

Theorem 2 gives us insight on the interpretability of the solution to various applications with respect to Gremban's expansion matrix. For ranking algorithms, we can extend Katz and personalized pagerank to signed graphs by involving slightly different graph-associated matrices, and one could easily derive versions of Gremban's expansion for these related calculations on signed graphs. In the next section we will investigate the numerical stability of the expansion matrix G.

3.2.1 Numerically Stability of Gremban's Expansion

The following demonstrates the relationships between the spectra of the expansion matrix and the original matrix. It is also shown that the expansion is numerically stable, meaning that a small residual for the expanded system implies a small residual for the original system. Although not the primary focus of this thesis, the expansion and some of the theory are relevant for the nonsymmetric matrices associated with SD graphs. Note that we are not assuming S is a signed Laplacian.

Theorem 3. *Let S be diagonally dominant with positive diagonal entries, and let G be a Gremban expansion of S. Employ the decomposition, $S = M + P$, with M a diagonally dominant Z-matrix, and $P \geq 0$ with zero diagonal. Let σ and ψ denote the eigenvalues and singular values respectively. Then, $\sigma(G) = \sigma(S) \cup \sigma(M-P)$ and $\psi(G) = \psi(S) \cup \psi(M-P)$.*

Proof. Let $(\lambda, \mathbf{p})$ be any eigenpair of S. Then, $(\lambda, [\mathbf{p}^t, -\mathbf{p}^t]^t)$ is an eigenpair for G as indicated by,

$$G \begin{bmatrix} \mathbf{p} \\ -\mathbf{p} \end{bmatrix} = \begin{bmatrix} M & -P \\ -P & M \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ -\mathbf{p} \end{bmatrix} = \begin{bmatrix} (M+P)\mathbf{p} \\ -(M+P)\mathbf{p} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{p} \\ -\mathbf{p} \end{bmatrix}. \tag{3.19}$$

Similarly, let $(\mu, \mathbf{u})$ be any eigenpair of $(M - P)$. Then, $(\mu, [\mathbf{u}^t, \mathbf{u}^t]^t)$ is an eigenpair for G as indicated by,

$$G \begin{bmatrix} \mathbf{u} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} M & -P \\ -P & M \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} (M-P)\mathbf{u} \\ (M-P)\mathbf{u} \end{bmatrix} = \mu \begin{bmatrix} \mathbf{u} \\ \mathbf{u} \end{bmatrix}. \tag{3.20}$$

This shows that $\sigma(G) \supset \sigma(S) \cup \sigma(M - P)$. To show, $\sigma(G) \subset \sigma(S) \cup \sigma(M - P)$, let

$\phi \in \sigma(G)$ with the corresponding eigenvector $\mathbf{v} = [\mathbf{v}_1^t, \mathbf{v}_2^t]^t$. Then

$$G \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} = \begin{bmatrix} M & -P \\ -P & M \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} \tag{3.21}$$

$$= \begin{bmatrix} M\mathbf{v}_1 - P\mathbf{v}_2 \\ -P\mathbf{v}_1 + M\mathbf{v}_2 \end{bmatrix} = \phi \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} = \begin{bmatrix} \phi\mathbf{v}_1 \\ \phi\mathbf{v}_2 \end{bmatrix}. \tag{3.22}$$

This implies, $M\mathbf{v}_1 - P\mathbf{v}_2 = \phi\mathbf{v}_1$ and $-P\mathbf{v}_1 + M\mathbf{v}_2 = \phi\mathbf{v}_2$. Adding these equations demonstrates that $(\phi, \mathbf{v}_1 + \mathbf{v}_2)$ is an eigenpair of $(M - P)$. Subtracting the second equation from the first demonstrates that $(\phi, \mathbf{v}_1 - \mathbf{v}_2)$ is an eigenpair of $S = M + P$. Therefore, $\sigma(G) \subset \sigma(S) \cup \sigma(M - P)$ and $\sigma(G) = \sigma(S) \cup \sigma(M - P)$.

We will now show that the singular values are related. Let $(\gamma, \mathbf{z}, \mathbf{y})$ be any singular value triplet of S such that, $S\mathbf{z} = \gamma\mathbf{y}$ and $\mathbf{y}^t S = \gamma\mathbf{z}^t$. Then, $(\gamma, [\mathbf{z}^t, -\mathbf{z}^t]^t, [\mathbf{y}^t, -\mathbf{y}^t]^t)$ is a singular value triplet of for G as indicated by,

$$G \begin{bmatrix} \mathbf{z} \\ -\mathbf{z} \end{bmatrix} = \begin{bmatrix} M & -P \\ -P & M \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ -\mathbf{z} \end{bmatrix} = \begin{bmatrix} (M+P)\mathbf{z} \\ -(M+P)\mathbf{z} \end{bmatrix} = \gamma \begin{bmatrix} \mathbf{y} \\ -\mathbf{y} \end{bmatrix} \tag{3.23}$$

and

$$\begin{bmatrix} \mathbf{y}^t & -\mathbf{y}^t \end{bmatrix} G = \begin{bmatrix} \mathbf{y}^t & -\mathbf{y}^t \end{bmatrix} \begin{bmatrix} M & -P \\ -P & M \end{bmatrix} \tag{3.24}$$

$$= \begin{bmatrix} \mathbf{y}^t(M+P) & -\mathbf{y}^t(M+P) \end{bmatrix} = \gamma \begin{bmatrix} \mathbf{z}^t & -\mathbf{z}^t \end{bmatrix}. \tag{3.25}$$

Similarly, let $(\beta, \mathbf{w}, \mathbf{q})$ be any singular value triplet of $(M - P)$. Thus, $(\beta, [\mathbf{w}^t, \mathbf{w}^t]^t, [\mathbf{q}^t, \mathbf{q}^t]^t)$

is a singular value triplet for G as indicated by,

$$G\begin{bmatrix}\mathbf{w}\\ \mathbf{w}\end{bmatrix}=\begin{bmatrix}M & -P\\ -P & M\end{bmatrix}\begin{bmatrix}\mathbf{w}\\ \mathbf{w}\end{bmatrix}=\begin{bmatrix}(M-P)\mathbf{w}\\ (M-P)\mathbf{w}\end{bmatrix}=\beta\begin{bmatrix}\mathbf{q}\\ \mathbf{q}\end{bmatrix} \tag{3.26}$$

and

$$\begin{bmatrix}\mathbf{q}^t & \mathbf{q}^t\end{bmatrix}G=\begin{bmatrix}\mathbf{q}^t & \mathbf{q}^t\end{bmatrix}\begin{bmatrix}M & -P\\ -P & M\end{bmatrix}=\begin{bmatrix}\mathbf{q}^t(M+P) & \mathbf{q}^t(M+P)\end{bmatrix}=\beta\begin{bmatrix}\mathbf{w}^t & \mathbf{w}^t\end{bmatrix}.$$

To show, $\sigma(G)\subset\sigma(S)\cup\sigma(M-P)$, let $\pi\in\sigma(G)$ with the corresponding right and left singular vectors $\mathbf{k}=[\mathbf{k}_1^t,\mathbf{k}_2^t]^t$ and $\mathbf{h}=[\mathbf{h}_1^t,\mathbf{h}_2^t]^t$. Then

$$G\begin{bmatrix}\mathbf{k}_1\\ \mathbf{k}_2\end{bmatrix}=\begin{bmatrix}M & -P\\ -P & M\end{bmatrix}\begin{bmatrix}\mathbf{k}_1\\ \mathbf{k}_2\end{bmatrix}=\begin{bmatrix}M\mathbf{k}_1-P\mathbf{k}_2\\ -P\mathbf{k}_1+M\mathbf{k}_2\end{bmatrix}=\pi\begin{bmatrix}\mathbf{h}_1\\ \mathbf{h}_2\end{bmatrix}=\begin{bmatrix}\pi\mathbf{h}_1\\ \pi\mathbf{h}_2\end{bmatrix}$$

and

$$\begin{aligned}\begin{bmatrix}\mathbf{h}_1^t & \mathbf{h}_2^t\end{bmatrix}G &= \begin{bmatrix}\mathbf{h}_1^t & \mathbf{h}_2^t\end{bmatrix}\begin{bmatrix}M & -P\\ -P & M\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{h}_1^tM-\mathbf{h}_2^tP & -\mathbf{h}_1^tP+\mathbf{h}_2^tM\end{bmatrix}=\pi\begin{bmatrix}\mathbf{k}_1^t & \mathbf{k}_2^t\end{bmatrix}=\begin{bmatrix}\pi\mathbf{k}_1^t & \pi\mathbf{k}_2^t\end{bmatrix}.\end{aligned}$$

This implies, $(\pi,\mathbf{k}_1+\mathbf{k}_2,\mathbf{h}_1+\mathbf{h}_2)$ is a singular triplet of $(M-P)$ as shown by,

$$(M-P)(\mathbf{k}_1+\mathbf{k}_2)=(M\mathbf{k}_1-P\mathbf{k}_2)+(M\mathbf{k}_2-P\mathbf{k}_1)=\lambda(\mathbf{h}_1+\mathbf{h}_2)$$

and

$$(\mathbf{h}_1+\mathbf{h}_2)^t(M-P)=(\mathbf{h}_1^tM-\mathbf{h}_2)+(\mathbf{h}_2^tM-\mathbf{h}_1^tP^t)=\lambda(\mathbf{v}_1+\mathbf{v}_2)^t,$$

and similarly $(\lambda, \mathbf{k}_1 - \mathbf{k}_2, \mathbf{h}_1 - \mathbf{h}_2)$ is also as a singular triplet S as shown by,

$$S(\mathbf{k}_1 - \mathbf{k}_2) = (M + P)(\mathbf{k}_1 - \mathbf{k}_2) = (M\mathbf{k}_1 - P\mathbf{k}_2) + (-M\mathbf{k}_2 + P\mathbf{k}_1) = \lambda(\mathbf{h}_1 - \mathbf{h}_2)$$

and

$$(\mathbf{h}_1 - \mathbf{h}_2)^t S = (\mathbf{h}_1 - \mathbf{h}_2)^t (M + P) = (\mathbf{h}_1^t M + \mathbf{h}_2^t P) + (-\mathbf{h}_2^t M + \mathbf{h}_1^t P) = \lambda(\mathbf{k}_1 - \mathbf{k}_2)^t.$$

Therefore, $\psi(G) \subset \psi(S) \cup \psi(M - P)$ and thus, $\psi(G) = \psi(S) \cup \psi(M - P)$.

For any S, let $S = Z\Gamma Y^t$ and $(M - P) = W\Delta Q^t$ be the singular value decompositions of the smaller matrices, then the singular value decomposition of G is

$$G = \frac{1}{\sqrt{2}} \begin{bmatrix} Z & W \\ -Z & W \end{bmatrix} \begin{bmatrix} \Gamma & \\ & \Delta \end{bmatrix} \left(\frac{1}{\sqrt{2}} \begin{bmatrix} -Y & Q \\ Y & Q \end{bmatrix} \right)^t.$$

For S symmetric, let $S = Y\Gamma Y^t$ and $(M - P) = Q\Delta Q^t$, then the singular value decomposition of G can be simplified

$$G = \frac{1}{\sqrt{2}} \begin{bmatrix} Y & Q \\ -Y & Q \end{bmatrix} \begin{bmatrix} \Gamma & \\ & \Delta \end{bmatrix} \left(\frac{1}{\sqrt{2}} \begin{bmatrix} Y & Q \\ -Y & Q \end{bmatrix} \right)^t.$$

□

It is often the case that G is singular even though S is nonsingular. If S is a signed graph Laplacian associated with an unbalanced $\mathcal{G}^{\pm}$, then it is nonsingular, as shown in the previous section. However, $M - P$ is singular, and thus G is singular with constant kernel. Note that nonsingular S does not imply an unbalanced sign structure of the off-diagonal elements. For example, consider a signed Laplacian of a balanced $\mathcal{G}^{\pm}$ shifted by a non-negative diagonal matrix. In such cases, the graph of G is not connected, but each of the two components are strictly diagonally dominant and G is nonsingular.

Lemma 3.2.1. *Assume S is nonsingular, strongly connected, and diagonally dominant. Let G be the Gremban expansion of S. Consider the systems $S\mathbf{x} = \mathbf{b}$ and $G\mathbf{w} = \mathbf{z}$ where $\mathbf{z} = [\mathbf{b}^t, -\mathbf{b}^t]^t$. If G is nonsingular then $\mathbf{w} = [\mathbf{x}^t, -\mathbf{x}^t]^t$. If G is singular then the null-space of G is dimension one and the solution is found via $\mathbf{w} = [\mathbf{x}^t, -\mathbf{x}^t]^t + \alpha[\mathbf{1}^t, -\mathbf{1}^t]^t$ for some $\alpha > 0$.*

Proof. If S is nonsingular and strongly connected, then by Theorem 3, G is only singular if $(M - P)$ is singular. By the Perron-Frobenius theorem, for $(M - P)$ to be singular, S needs to be only weakly diagonally dominant, and there exists unique right and left kernel components, $(M - P)\mathbf{v} = \mathbf{0}$ and $\mathbf{u}^t(M - P) = \mathbf{0}^t$. In this case, $[\mathbf{v}^t, \mathbf{v}^t]^t$ and $[\mathbf{u}^t, \mathbf{u}^t]^t$ is the unique right and left kernel components of G. If S is symmetric $\mathbf{u} = \mathbf{v} = \mathbf{1}$ and the solution is computed by the process described in (2.4) applied to $G\mathbf{w} = \mathbf{z}$ where $\mathbf{z} = [\mathbf{b}^t, -\mathbf{b}^t]^t$ is orthogonal to the kernel. For nonsymmetric S, the solution is found in a similar way and is left as an exercise for the reader. We see that

$$G\mathbf{w} = \begin{bmatrix} M & -P \\ -P & M \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ -\mathbf{x} \end{bmatrix} = \begin{bmatrix} (M+P)\mathbf{x} \\ -(M+P)\mathbf{x} \end{bmatrix} = \begin{bmatrix} S\mathbf{x} \\ -S\mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \end{bmatrix} = \mathbf{z}. \quad (3.27)$$

In the case of nonsingular G, $\mathbf{w} = [\mathbf{x}^t, -\mathbf{x}^t]^t$ is the unique solution. □

The results of Theorem 3 and Lemma 3.2.1 are extremely powerful. Given a robust linear solver for unsigned Laplacians, we can solve any diagonally dominant system, $S\mathbf{x} = \mathbf{b}$, by transforming it into the associated diagonally dominant Z-matrix system, $G\mathbf{w} = \mathbf{z} = [\mathbf{b}^t, -\mathbf{b}^t]^t$. Then $\mathbf{x}$ is easily found by solving for $\mathbf{w}$ and extracting $\mathbf{x}$. Note that the solvers use the expanded matrix G, however, the actual interest is for the solution to the smaller system involving S. The following theorem shows that if given an approximate solution to the Gremban expansion system, we can tightly bound the norm of the residual of the original system by the norm of the residual of the expansion.

Theorem 4. *Let S be any diagonally dominant matrix and G be the Gremban expansion of S. Let $\mathbf{y}$ be an approximate solution to $G\mathbf{w} = \mathbf{z} = [\mathbf{b}^t, -\mathbf{b}^t]^t$. Define $P_1 = [I \quad O]$ and $P_2 = [O \quad I]$. Let $\mathbf{v} = \alpha P_1\mathbf{y} + \beta P_2\mathbf{y}$, where α, β are chosen so that $\|S\mathbf{v} - \mathbf{b}\|$ is minimized, be the approximate solution to $S\mathbf{x} = \mathbf{b}$. Then,*

$$\|S\mathbf{v} - \mathbf{b}\| \le C_s \|G\mathbf{y} - \mathbf{z}\|$$

where $C_s = 1/\sqrt{2}$. This bound is tight.

Proof. Let $R = \frac{1}{2}[I, -I]$ be a restriction operator, then $R\mathbf{z} = \mathbf{b}$ and $R\mathbf{w} = \mathbf{x}$. We also have $R\mathbf{y} = \frac{1}{2}P_1\mathbf{y} - \frac{1}{2}P_2\mathbf{y}$. It follows that then

$$\begin{aligned} \|S\mathbf{v} - \mathbf{b}\| &\le \|SR\mathbf{y} - \mathbf{b}\| \\ &= \frac{\|SR(\mathbf{y} - \mathbf{w})\|}{\|G(\mathbf{y} - \mathbf{w})\|}\|G\mathbf{y} - \mathbf{z}\| \\ &= \frac{\|SR\mathbf{e}\|}{\|G\mathbf{e}\|}\|G\mathbf{y} - \mathbf{z}\| \\ &= C_s\|G\mathbf{y} - \mathbf{z}\|, \end{aligned}$$

where

$$C_s = \max_{\substack{\mathbf{e}\neq 0,\\ \mathbf{e}\perp\mathcal{N}(G)}} \sqrt{\frac{\langle R^tS^tSR\mathbf{e}, \mathbf{e}\rangle}{\langle G^tG\mathbf{e}, \mathbf{e}\rangle}},$$

which is equivalent to the maximum eigenvalue of the generalized eigenvalue problem $R^tS^tSR\mathbf{q} = \lambda G^tG\mathbf{q}$. Let $S = Z\Gamma Y^t$ and $(M - P) = W\Delta Q^t$ be the singular value decompositions of the smaller matrices. From Theorem 3 we have,

$$\begin{aligned}
R^tS^tSR\mathbf{q} &= \lambda G^tG\mathbf{q} \\
\frac{1}{2}\begin{bmatrix} I \\ -I \end{bmatrix} Y\Gamma^2Y^t\frac{1}{2}\begin{bmatrix} I & -I \end{bmatrix}\mathbf{q} &= \lambda\frac{1}{\sqrt{2}}\begin{bmatrix} -Y & Q \\ Y & Q \end{bmatrix}\begin{bmatrix} \Gamma^2 & \\ & \Delta^2 \end{bmatrix}\left(\frac{1}{\sqrt{2}}\begin{bmatrix} -Y & Q \\ Y & Q \end{bmatrix}\right)^t\mathbf{q}.
\end{aligned}$$

Let $\mathbf{p} = \left(\frac{1}{\sqrt{2}}\begin{bmatrix} -Y & Q \\ Y & Q \end{bmatrix}\right)^t\mathbf{q}$. Then,

$$\begin{aligned}
\frac{1}{4}\begin{bmatrix} Y \\ -Y \end{bmatrix}\Gamma^2\begin{bmatrix} Y^t & -Y^t \end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix} -Y & Q \\ Y & Q \end{bmatrix}\left(\frac{1}{\sqrt{2}}\begin{bmatrix} -Y & Q \\ Y & Q \end{bmatrix}\right)^t\mathbf{q} &= \lambda\frac{1}{\sqrt{2}}\begin{bmatrix} -Y & Q \\ Y & Q \end{bmatrix}\begin{bmatrix} \Gamma^2 & \\ & \Delta^2 \end{bmatrix}\mathbf{p}, \\
\left(\frac{1}{\sqrt{2}}\begin{bmatrix} -Y & Q \\ Y & Q \end{bmatrix}\right)^t\frac{1}{4\sqrt{2}}\begin{bmatrix} Y \\ -Y \end{bmatrix}\Gamma^2\begin{bmatrix} Y^t & -Y^t \end{bmatrix}\begin{bmatrix} -Y & Q \\ Y & Q \end{bmatrix}\mathbf{p} &= \lambda\begin{bmatrix} \Gamma^2 & \\ & \Delta^2 \end{bmatrix}\mathbf{p}, \\
\frac{1}{8}\begin{bmatrix} -2I \\ 0 \end{bmatrix}\Gamma^2\begin{bmatrix} -2I & 0 \end{bmatrix}\mathbf{p} &= \lambda\begin{bmatrix} \Gamma^2 & \\ & \Delta^2 \end{bmatrix}\mathbf{p}, \\
\frac{1}{2}\begin{bmatrix} \Gamma^2 & 0 \\ 0 & 0 \end{bmatrix}\mathbf{p} &= \lambda\begin{bmatrix} \Gamma^2 & \\ & \Delta^2 \end{bmatrix}\mathbf{p}, \\
\frac{1}{2}\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\mathbf{p} &= \lambda\begin{bmatrix} I & \\ & I \end{bmatrix}\mathbf{p}.
\end{aligned}$$

Thus, the eigenvalues of the generalized eigenvalue problem, $R^tS^tSR\mathbf{q} = \lambda G^tG\mathbf{q}$, are 1/2 and 0, and $C_s = \frac{1}{\sqrt{2}}$.

□

In Section 3.3 we validate this result by empirically demonstrating numerical stability of Gremban's expansion with LAMG, a known robust multilevel solver for symmetric, diagonally dominant Z-matrix systems.

3.3 Numerical Results

A Julia implementation of the LAMG algorithm was developed to serve as a platform for the investigations described in this section. There are several measurements that are used to assess the efficiency of a multigrid scheme: convergence factor, cycle complexity, effective convergence factor and total work. The **convergence factor**, ρ, measures the asymptotic reduction in the 2-norm of the residual for a single V-cycle. To estimate ρ, we calculate the convergence factor after the k-th cycle, ρ_k, and then report the geometric mean for a solve containing K multigrid cycles,

$$\rho_k = \frac{\|G\mathbf{y}_{k+1} - \mathbf{z}\|_2}{\|G\mathbf{y}_k - \mathbf{z}\|_2} \quad \text{and} \quad \rho \approx \left(\prod_{k=1}^{K} \rho_i\right)^{1/K}.$$

Define one **work unit** to be the cost of one fine grid matrix-vector multiplication, in other words one work unit is the number of nonzeros of the fine grid matrix. The **cycle complexity**, γ, is the number of work units required for a single multigrid cycle. The **effective convergence factor**, (ECF), is the average reduction in residual per work unit cost, ECF $= \rho^{1/\gamma}$. The total **work**, work $= \gamma * \nu$, where ν is the number of V-cycles, is the amount of total work units required to solve the linear system to a numerical tolerance using multigrid cycles.

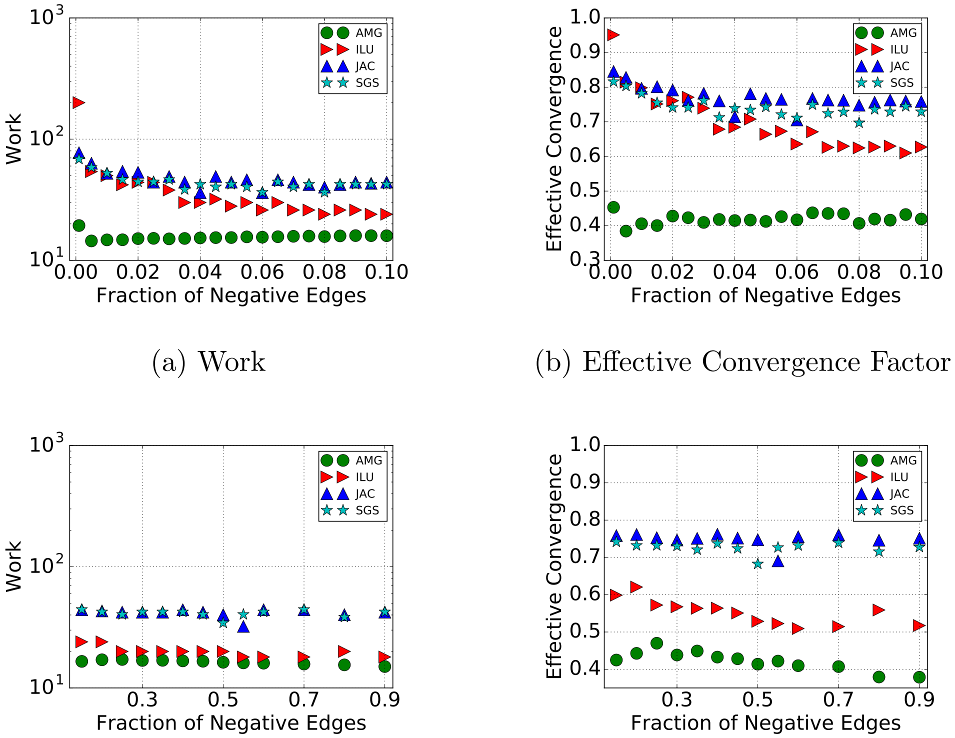
3.3.1 Comparison of Preconditioners on Generated Signed Graphs

In this section, we compare a variety of preconditioners with Preconditioned Conjugate Gradient (PCG), mainly weighted Jacobi, Incomplete LU ($ILU(0)$) and Symmetric Gauss-Seidel to LAMG on the larger Gremban's expansion system. Note that PCG is not used in conjunction with LAMG. LAMG is a non-stationary algorithm due to its use of a iterative recombination phase at every aggregation level [23]. For the graph *as-cadia* with $n = 26,475$

and $m = 53,381$, a percentage, $p \in [.001, .9]$, of the graphs edges were randomly chosen to be negative. For a random vector $\mathbf{b}$ the goal is to solve $L^{\pm}\mathbf{x} = \mathbf{b}$. The Gremban expansion, G, of $L^{\pm}$ was formed and we set $\mathbf{z} = [\mathbf{b}^t, -\mathbf{b}^t]^t$. LAMG then solved the system $G\mathbf{w} = \mathbf{z}$ with V-cycles with $(1, 2)$ pre- and post- Gauss-Seidel smoothing iterations, with a cut off of 100 nodes for the coarsest graph and a residual tolerance of 10^{-7}. Figure 3.2a and 3.2b depicts the of work (log-scale) and effective convergence factor required to solve the linear system $L^{\pm}\mathbf{x} = \mathbf{b}$ to a tolerance of 10^{-7} with same initial guess and right-hand side with the a small percentage, $p \in [.001, .1]$. In Figure 3.2a, it is clear that the work required for LAMG with the expansion is significantly smaller than any of the three preconditioner's even though the system has twice the number of degrees of freedom. As the number of negative edges increases we see that each method begins to level out to be a constant amount of work units despite the number of negative edges. We see a similar trend in Figure 3.2b for the effective convergence factor. When the percentage of negative edges increases to $p \in [.1, .9]$, as shown in Figure 3.2c and 3.2d, a similar trend continues. LAMG with Gremban's expansion results in less work and lower effective convergence factors. In both cases, LAMG with Gremban's expansion is robust as it is unaffected by the number of negatively signed edges. This is beneficial to the user since no analysis of the SU graph is needed. However, depending on the sign structure, PCG may have difficulties. The user needs to carefully select the preconditioner when using PCG. In the next section we analyze LAMG with Gremban's expansion and compare it to PCG with Gauss-Seidel preconditioning for a large suite of generated signed graphs.

3.3.2 Feasibility Testing on Generated Signed Graphs

For feasibility testing, we tested on a large set of graph topologies with random sign structure. A set of UU graphs were obtained from the SNAP graph database [20] and for each graph, a randomly chosen percentage, $p \in [0.1, 0.8]$, of the edges were turned into negatively signed edges. For a random vector $\mathbf{b}$ the goal is to solve $L^{\pm}\mathbf{x} = \mathbf{b}$. The Gremban



(a) Work

(b) Effective Convergence Factor

(c) Work

(d) Effective Convergence Factor

Figure 3.2: Percentage of Randomly Signed Edges

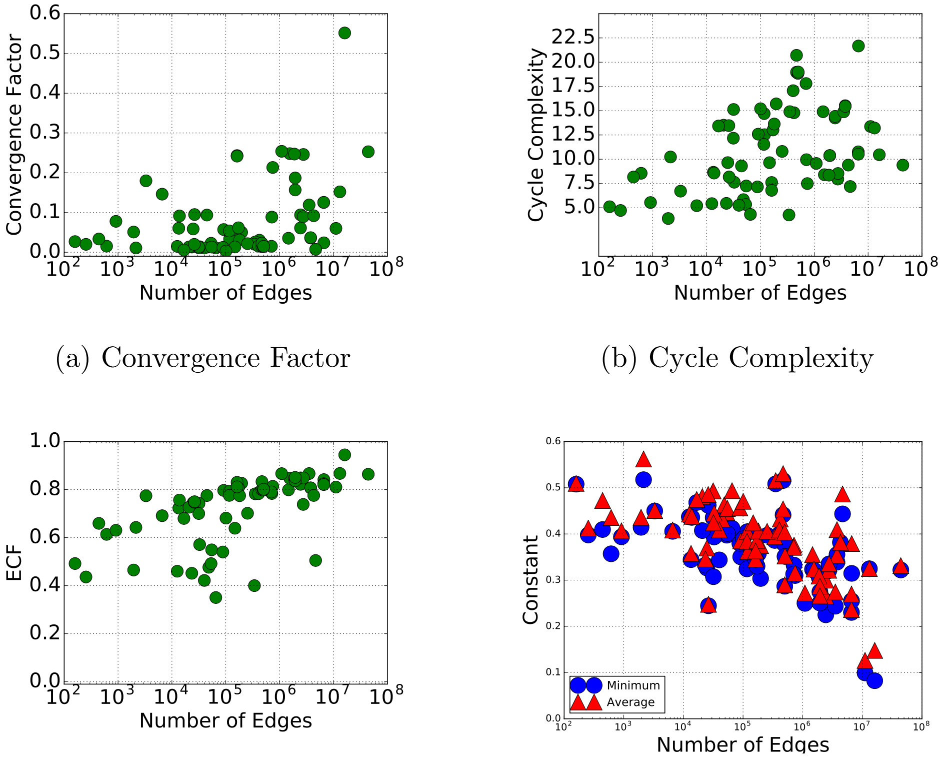
expansion, G, of $L^{\pm}$ was formed and we set $\mathbf{z} = [\mathbf{b}^t, -\mathbf{b}^t]^t$. LAMG then solved the system $G\mathbf{w} = \mathbf{z}$ with V-cycles with $(1, 2)$ pre- and post- Gauss-Seidel smoothing iterations, with a cut off of 100 nodes for the coarsest graph and a residual tolerance of 10^{-7}. Figures 3.3a, 3.3b, and 3.3c show the convergence factor, cycle complexity and effective convergence factor respectively. Note that, due to the Gremban expansion, the size of the problem is twice the size of the original problem. Table 3.1 displays the graphs that had an effective convergence factor greater than 0.85.

Graph	$2n$	$2m$	n_{level}	Solve Time(s)	Setup Time(s)	ρ	γ	ECF
eu-2005($p = .20$)	862664	16138468	6	348.1473	177.4885	0.5515	10.4608	0.9447
in-2004($p = .20$)	1353703	13126172	29	416.9488	58.4933	0.1519	13.2174	0.8671
amazon-2008($p = .30$)	735323	3523472	16	169.6135	29.0520	0.1189	14.8830	0.8667
web-NotreDame($p = .60$)	325729	1090108	21	44.0882	8.0447	0.2538	9.5660	0.8664
wb-edu($p = .80$)	8863287	44185251	33	1835.4726	214.4549	0.2528	9.3903	0.8637
web-Stanford($p = .60$)	255265	1941926	23	66.3343	12.7228	0.1868	10.3757	0.8507

Table 3.1: Constructed Signed Graphs with $ECF > 0.85$

From Figure 3.3a, one graph has a particularly high convergence factor compared to the rest of the test set, *eu-2005*, with 20% of the edges turned to negatively signed edges. The graph is a small subset of a large web graph. A negative edge in this context is not natural and this may be the cause of the high convergence factor compared to the other graphs tested. A convergence factor of ≈ 0.55 is not high for LAMG involving real-world graph structures.

After the system $G\mathbf{w} = \mathbf{z}$ was solved, an approximation to $\mathbf{x}$ was found as a weighted average, $\mathbf{x} = \alpha P_1\mathbf{w} + \beta P_2\mathbf{w}$, as determined by Theorem 4. The solution was found in two different ways: the true minimum of the residual, $\mathbf{x} = \min_{\alpha,\beta} ||L^{\pm}\mathbf{x} - \mathbf{b}||_2$, and a simple averaging, $\mathbf{x} = \frac{1}{2}P_1\mathbf{w} + \frac{1}{2}P_2\mathbf{w}$. Define the **ratio of relative residuals**, τ, as $\tau = ||L^{\pm}\mathbf{x} - \mathbf{b}||_2 \,/\, ||G\mathbf{w} - \mathbf{z}||_2$. Figure 3.3d shows the ratio of relative residuals for both the minimum constant using the optimal α and β for $\mathbf{x}$ and the averaging constant. The average solution still produce ratios of relative residuals that are less than $C_s = \frac{1}{\sqrt{2}}$. From a data-science perspective, the average solution will give a good approximation without the extra cost of



(a) Convergence Factor

(b) Cycle Complexity

(c) Effective Conference Factor

(d) Ratio of Relative Residuals

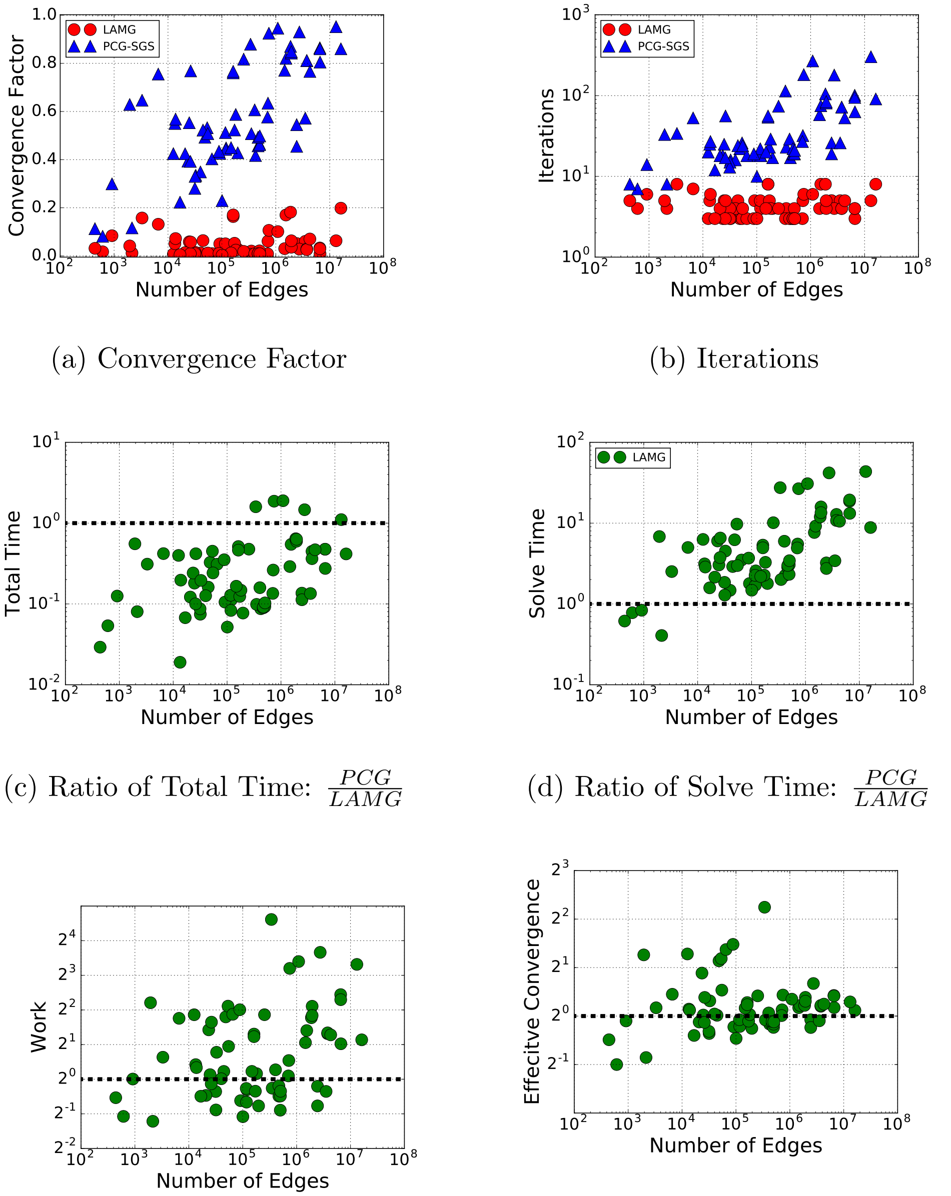
Figure 3.3: Performance of LAMG with Gauss-Seidel smoothing on Constructed Signed Graphs

finding the minimum.

We compare LAMG with Gauss-Seidel smoothing on the Gremban's expansion matrix with Symmetric Gauss-Seidel preconditioned CG (PCG-SGS). We ran each test to a tolerance of 10^{-7} and a maximum number of iterations of $1,000$. Figures 3.4a, 3.4b, 3.4c, 3.4d, 3.4e, and 3.4f compares the convergence factor, number of V-cycles verses number of iterations (log-scale), the ratio of total time to setup and solve (log-scale), ratio of only solve time (log-scale), ratio of total work (log-scale), and ratio of effective convergence factor (log-scale), for PCG with Symmetric Gauss-Seidel with respect to LAMG.

LAMG clearly provides better convergence factors and fewer iterations for most graphs. This finding is not surprising. Multigrid algorithms were created to improve the convergence of the basic iterative methods by incorporating coarse-grid correction. The amount of work and the effective convergence factor are better measures for comparing the two algorithms. Figures 3.4e and 3.4f depict the ratio of work (log-scale) and the ratio of effective convergence factor of PCG with respect to LAMG. LAMG with Gremban's expansion required less work for 63.2% of the graphs tested and lower effective convergence factors for 61.7% of the graphs tested. LAMG has both setup and solve phase. The solve phase clearly out performs PCG-SGS but the time and work taken to create the solver must be considered. LAMG resulted in faster total times than PCG for 7.4% of the graphs tested. For only solve time, LAMG was faster than PCG for 94.1% of graphs. Thus, if solving systems involving the same matrix with various right-hand sides, LAMG would be the better choice since the coarse-grids at each level could be saved. If only one right-hand side is needed for the application either solver would suffice

The above testing has shown that by using Gremban's expansion we are able to extend current UU graph Laplacian solvers to SU graph Laplacian systems without modifying the solvers. In some cases, depending on the sign structure, degree-distribution, size of a given SU graph, and the application, PCG in conjunction with a more advanced preconditioner may be an optimal solver. However, without the proper analysis of the graph and understanding



(a) Convergence Factor

(b) Iterations

(c) Ratio of Total Time: $\frac{PCG}{LAMG}$

(d) Ratio of Solve Time: $\frac{PCG}{LAMG}$

(e) Ratio of Work: $\frac{PCG}{LAMG}$

(f) Ratio of Effective Convergence: $\frac{PCG}{LAMG}$

Figure 3.4: LAMG with Gauss-Seidel smoothing verse PCG-SGS on Constructed SU Graphs

the requirements of the application, PCG may struggle and be unacceptably slow. The above tests were performed on manufactured signed graphs. The random switching of the edges may have led to a harder topology than real-world signed graphs.

3.3.3 Scalability Testing on Real-World Signed Graph

For further analysis on how the expansion works for real-world graphs, testing was performed on bipartite user-movie rating matrices, R, from the Grouplens research[1] group. The adjacency matrix of the data can be described as,

$$A = \begin{bmatrix} 0 & R \\ R^t & 0 \end{bmatrix}. \tag{3.28}$$

Grouplens provides four different size data sets, 100k, 1M, 10M and 20M. For each movie the user rated the movie from 1 to 5 with 5 being a high score. To transform the data to an SU graph we looked at four different scalings:

- **User Average:** If a user rates every movie high, then ratings are not proportional. The users average rating, μ_i, is subtracted from every movie that they have rated. The adjacency matrix is then:

$$A_{ij}^{(user)} = A_{ji}^{(user)} = R_{ij} - \mu_i. \tag{3.29}$$

- **Movie Average:** If a movie is rated highly by all users, then the rating is not proportional. The the movie average rating, μ_j, is subtracted from all users who have rated that movie. The adjacency matrix is given by:

$$A_{ij}^{(movie)} = A_{ji}^{(movie)} = R_{ij} - \mu_j. \tag{3.30}$$

[1] http://grouplens.org/datasets/movielens/

- **Full Average:** If a movie and/or user rating is high, again the rating is not proportional. The entries are scaled by both the user and movie average rating. The adjacency matrix is defined as:

$$A_{ij}^{(full)} = A_{ji}^{(full)} = R_{ij} - \frac{\mu_j + \mu_i}{2}. \tag{3.31}$$

- **Zero-Centered Shift:** The entries are shifted so that negative entries represent a low rating while a positive entry is a high rating. The entries of the adjacency matrix are mapped as follows:

$$[1, 2, 3, 4, 5] \rightarrow [-2, -1, 1, 2, 3]. \tag{3.32}$$

For the following numerical tests LAMG was performed with V-cycles with $(1, 2)$ pre- and post- weighted Jacobi smoothing iterations with weight 2/3, with a cut off of 100 nodes for the coarsest graph and a residual tolerance of 10^{-7}. Table 3.2 presents the convergence factor, cycle complexity, and effective convergence factor for all four shifts. In the last two columns, the ratio of relative residuals for the solution using the minimum α and β, and the average are presented.

LAMG performed well on all the graphs, with effective convergence factors less than 0.80. This indicates that the expansion, though it doubles the size of the problem, resulted in systems that were solvable. The shifted scaling produced the worse performance for LAMG. Table 3.3 compares the convergence factor as well as the solve time for LAMG with Gremban's expansion and PCG with Jacobi without Gremban's expansion. PCG without the expansion resulted in convergence factors that are much higher than LAMG with the expansion as expected. When solve time is considered, LAMG with the expansion outperformed PCG for all scalings. While different weighting schemes affect the preconditioners differently, LAMG's performance is less-correlated with the condition number than PCG.

Graph	m	Setup Time(s)	Solve Time(s)	ρ	γ	ECF	τ_{min}	$\tau_{average}$
ml-100k_average.txt	100000	1.5682	0.3986	0.0362	13.2118	0.7779	0.5479	0.5504
ml-100k_moive.txt	100000	1.3171	0.2949	0.0409	12.8255	0.7794	0.5528	0.5627
ml-100k_shift.txt	100000	1.5804	0.4097	0.0484	13.5393	0.7995	0.3890	0.4351
ml-100k_user.txt	100000	1.3673	0.3466	0.0376	13.0424	0.7777	0.4115	0.4162
ml-1m_average.txt	1000209	13.2945	2.2286	0.0253	13.7554	0.7653	0.5426	0.5426
ml-1m_moive.txt	1000209	12.4908	2.1539	0.0287	13.3893	0.7671	0.4868	0.4873
ml-1m_shift.txt	1000209	14.4251	3.1774	0.0429	13.5637	0.7928	0.4681	0.4681
ml-1m_user.txt	1000209	12.5042	2.2269	0.0311	13.5223	0.7736	0.4900	0.4902
ml-10m_average.txt	10000054	111.2039	21.6781	0.0279	13.2540	0.7634	0.4951	0.5040
ml-10m_moive.txt	10000054	110.1602	21.9756	0.0249	13.0803	0.7541	0.5073	0.5439
ml-10m_shift.txt	10000054	118.0933	31.1836	0.0510	12.9736	0.7950	0.3620	0.3629
ml-10m_user.txt	10000054	268.3608	63.9702	0.0273	13.0465	0.7588	0.5369	0.5528
ml-20m_average.txt	20000230	234.3954	46.8042	0.0244	13.3901	0.7578	0.4478	0.4491
ml-20m_moive.txt	20000230	238.6919	46.7372	0.0256	13.3183	0.7594	0.5161	0.5341
ml-20m_shift.txt	20000263	252.7742	70.7810	0.0434	12.7273	0.7815	0.4815	0.4842
ml-20m_user.txt	20000230	230.9860	47.2896	0.0287	12.9946	0.7609	0.5644	0.5645

Table 3.2: LAMG with w-Jacobi smoothing performance on Movielens Graphs

		PCG w/o	LAMG	PCG w/o Expansion
Graph	LAMG(ρ)	Expansion(ρ)	(SolveTime)	(SolveTime)
ml-100k_full	0.0362	0.8844	0.3986	0.8548
ml-100k_movie	0.0409	0.9087	0.3986	1.0951
ml-100k_shift	0.0484	0.7740	0.4097	0.3889
ml-100k_user	0.0376	0.8971	0.3466	0.8812
ml-1m_full	0.0253	0.8941	2.2286	9.2640
ml-1m_movie	0.0287	0.9005	2.1539	12.4039
ml-1m_shift	0.0429	0.7880	3.1774	4.9282
ml-1m_user	0.0311	0.9123	2.2269	11.3779
ml-10m_full	0.0279	0.9021	21.6781	301.9563
ml-10m_movie	0.0249	0.8973	21.9756	479.5299
ml-10m_shift	0.0510	0.7858	31.1836	213.9651
ml-10m_user	0.0273	0.8951	63.9702	357.2703
ml-20m_full	0.0244	0.8950	46.8042	370.1953
ml-20m_movie	0.0256	0.8963	46.7372	186.1037
ml-20m_shift	0.0434	0.7905	70.7810	109.1963
ml-20m_user	0.0287	0.9120	47.2896	175.2370

Table 3.3: LAMG with w-Jacobi smoothing and PCG-Jacobi performance on Movielens Graphs

For PCG, the zero-centered shift gave the best solve times while the same scaling was the slowest for LAMG. For the largest graph (20M) with the full average scaling, LAMG was eight times faster than PCG. For the shifted scaling, LAMG was only 1.5 times faster. For the problems tested, it can be concluded that Gremban's expansion is numerically stable when the right-hand side is restricted to $\mathbf{z} = [\mathbf{b}^t, -\mathbf{b}^t]^t$. It must be noted that the condition number of the expansion cannot be bounded by the condition number for the original graph Laplacian. Thus, when using the expansion, a robust UU solver is required to be confident about efficiency across a diverse set of graphs.

3.3.4 Real-world Motivation of Theorem 2

For the user-movie rating graphs, Theorem 2 can give us an interpretation of rankings involving signed graphs. Using the 100k user-movie rating matrix described above, we first discuss the changes in a personalized ranking [10] when the unsigned graph and the signed graph Laplacian are used. We will then discuss the impact of Theorem 2 on determining the ranking with respect to the signed graph Laplacian. Let A be the original, unsigned graph adjacency matrix of the 100k user-movie rating matrix, described in equation (3.28). We will compute the ranking, $\mathbf{r}$, for user i, by solving the system

$$L^{\pm}\mathbf{r} = \mathbf{e}_i,$$

where $L^{\pm}$ is a (unsigned or signed) graph Laplacian. The top ten ranked movies, for user $i = 1$, using the unsigned graph Laplacian are depicted in Table 3.4. All movies are either documentary, drama, or comedy. If instead the signed graph Laplacian is used (full-average shift represent by equation (3.31)) the ranking changes considerably. Table 3.5 list the top ten ranked movies using the signed Laplacian. One difference between the ranking are the genres that appear in the top ten. When the signed graph is used, the movies *Faster Pussycat! Kill! Kill!*, *Turbo: A Power Rangers Movie*, *Free Willy 2: The Adventure Home*, and *All Dogs Go to Heaven 2* appear in the top ten which are all either action, adventure, animation, or children's movie. These genres were not present in the unsigned graph Laplacians rankings. Another aspect to consider is that the signed graph rankings overlap with the unsigned ranking. The movies *Maya Lin: A Strong Clear Vision*, *Brother Minister: The Assassination of Malcolm X* , *The Horseman on the Roof*, *Nadja*, *Theodore Rex*, and *The Doom Generation* appear in both rankings but are ranked in a different order. The ranking involving the signed graph Laplacian have more variability and could enhance movie recommendation systems.

When the signed graph Laplacian ranking is found, we are inherently also solving the

Ranking	Movie	Genras
1	Maya Lin: A Strong Clear Vision (1994)	Documentary
2	Brother Minister: The Assassination of Malcolm X (1994)	Documentary
3	Unhook the Stars (1996)	Drama
4	White Balloon, The (1995)	Drama
5	Horseman on the Roof, The (Hussard sur le toit, Le) (1995)	Drama
6	unknown	unknown
7	Nadja (1994)	Drama
8	Theodore Rex (1995)	Comedy
9	Doom Generation, The (1995)	Comedy, Drama
10	Haunted World of Edward D. Wood Jr., The (1995)	Documentary

Table 3.4: Ranking with Unsigned Graph Laplacian

system with respect to the Gremban expansion system, i.e.

$$(L^{\pm})^{\dagger}\mathbf{e}_i = G^{\dagger}\begin{bmatrix} \mathbf{e}_i \\ \mathbf{0} \end{bmatrix} - G^{\dagger}\begin{bmatrix} \mathbf{0} \\ \mathbf{e}_i \end{bmatrix}.$$

The solution to the signed graph Laplacian system is the same as scoring vertices based on the difference of two random-walk based rankings. One random-walk involving paths with an odd number of negative edges and one walk with an even number of negative edges. By Theorem 2, the ranking of vertex j with respect to vertex i is a sum of a friendly score and an unfriendly score, since the value $\begin{bmatrix} \mathbf{e}_j & \mathbf{0} \end{bmatrix} G^{\dagger}\begin{bmatrix} \mathbf{e}_i \\ \mathbf{0} \end{bmatrix}$ scores vertex j by how close it is to vertex i along walks that cross zero or an even number of negative edges (a friendly ranking) and the value $\begin{bmatrix} \mathbf{e}_j & \mathbf{0} \end{bmatrix} G^{\dagger}\begin{bmatrix} \mathbf{0} \\ \mathbf{e}_i \end{bmatrix}$ scores vertex j by how close it is to vertex i along walks that cross an odd number of negative edges (a unfriendly ranking). Ranking is one application where Theorem 2 provides a logical interpretation of the ranking of a signed graph using a classical interpretation of the ranking of unsigned graphs. We hope to motivate data analysts to consider using a signed graph representation by providing a robust solver and presenting various relationships between the signed and unsigned variants.

Gremban's expansion matrix shows promise to be able to show information about the

Ranking	Movie	Genras
1	Faster Pussycat! Kill! Kill! (1965)	Action, Comedy, Drama
2	Theodore Rex (1995)	Comedy
3	Maya Lin: A Strong Clear Vision (1994)	Documentary
4	Turbo: A Power Rangers Movie (1997)	Action, Adventure, Children's
5	Horseman on the Roof, The (Hussard sur le toit, Le) (1995)	Drama
6	Free Willy 2: The Adventure Home (1995)	Adventure, Children's
7	Doom Generation, The (1995)	Comedy, Drama
8	All Dogs Go to Heaven 2 (1996)	Animation, Children's, Musical
9	Brother Minister: The Assassination of Malcolm X (1994)	Documentary
10	Nadja (1994)	Drama

Table 3.5: Ranking with Signed Graph Laplacian

signed graph from both the signed and signless perspective. Gremban's expansion matrix gives us a way to interpret a SU graph in terms of an UU and many applications could be applied to Gremban's expansion matrix that are similar to the UU graph applications, such as graph ranking and spectral clustering.

3.4 Conclusions

This chapter outlines an algorithm that relates signed graphs to unsigned graphs, which implicitly extends the solution space of current UU graphs Laplacian linear solvers to include SU graphs. We generalized Gremban's expansion to include any diagonally dominant matrix, thus Gremban's expansion can also be used to extend the UD graph Laplacian solver NS-LAMG, described in Chapter 4, to SD graph Laplacian systems. The Gremban expansion matrix, G, relates the signed graph to the respective unsigned graph in a meaningful way. Theorem 2 and 3 show that the Gremban's expansion matrix could be useful in understanding the underlaying graph as Theorem 2 generalizes a well known UU graph theorem to SU graphs and Theorem 3 relates spectrum and singular values of the expansion matrix to the spectrum and singular values of the signed and signless Laplacian. The Gremban's expansion matrix could be applied to a variety of applications, e.g. community detection with spectral clustering, an another analysis tool for understand signed graphs. In Theorem 4, we showed that we can tightly bound the norm of the residual of the original system by the norm of the residual of the expansion given $\mathbf{z} = [\mathbf{b}^t, -\mathbf{b}^t]^t$. Numerical tests indicated that Gremban's expansion with an adequate solver is numerically stable without prior knowledge of the sign structure. Though PCG with an adequate preconditioner may be faster for a single right-hand side, as LAMG has a setup phase, applications that require multiple right-hand sides would benefit from multigrid algorithms as the setup cost is amortized. Applications for graphs that require multiple right-hand sides include the centerpiece subgraph problem [45] and community detection with spectral clustering [48]. Thus, depending on the application the user must select an appropriate solver.

Chapter 4

Nonsymmetric Lean Algebraic Multigrid (NS-LAMG): An AMG Method for Directed Graph Laplacian Linear Systems

4.1 The Graph Laplacian for Unsigned, Directed Graphs

As mentioned in Chapter 1, the current methods used to solve linear systems for UD graphs are direct solvers such as SuperLU and QR Factorization, and iterative solvers such as Generalized Minimal Residual Method with Restart (GMRES(k)) and Generalized Least Squares QR Method (LSQR). Provided in this chapter are the details of NS-LAMG, an algebraic multigrid solver for UD graph Laplacian linear systems that was developed for UD graphs, including scale-free and mesh-like graphs. NS-LAMG is the only known multilevel iterative solver that has been adapted specifically for real-world UD graph Laplacians linear systems. For UD graphs, $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$, an edge $(i, j) \in \mathcal{E}$ is given a direction where information can flow from vertex i to vertex j, while the weight, $w_{ij} > 0$, defines the "importance" of an edge. The adjacency matrix A of a UD graph is defined by the same formula as a UU graph in (2.2). Since the edges are a not bi-directional, each vertex has an *out-degree* and an *in-degree*. The out-degree, d_i^{out}, of vertex i is the sum of the edge weights of edges directed out of the vertex, the columns of A, while the in-degree, d_i^{in}, is the sum of the edge weights of edges directed into the vertex, the rows of A. The diagonal degree matrix, D, associated with graph $\mathcal{G}$ has the out-degree, d_i^{out}, of each vertex along the diagonal. Similarly to UU graphs, a UD graph is scale-free if either the out-degree, in-degree, or total degree of the graph follows a power-law distribution. The difference between the adjacency and diagonal degree matrix form the combinatorial Laplacian, as before in (2.1). The combinatorial graph Laplacian of a UD graph is nonsymmetric with a column-sum of zero. It is not necessarily true that it will have a row-sum of zero.

It is assumed that the graph is *strongly connected*. We define strongly connected more concretely than in the pervious section. To define a strongly connected graph, we first must define a *directed path* in a graph.

Definition 4.1.1. *For vertices u and v in a UD graph, $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$, a directed walk in $\mathcal{G}$ is a finite sequence of vertices $\{u = v_0, v_1, \ldots, v_{k-1}, v_k = v\}$ beginning at u and ending at v, such that $(v_{i-1}, v_i) \in \mathcal{E}$ for $i = 1, \ldots, k$. A* **directed path** *is a directed walk such that no vertex is repeated [40].*

Definition 4.1.2. *A UD graph, $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$, is* **strongly connected** $\Leftrightarrow \exists$ *a directed path from v_i to v_j for any two distinct vertices $v_i, v_j \in \mathcal{V}$.*

The graph Laplacian of a strongly connected UD graph is inherently an irreducible matrix. We now define singular M-matrices and show that if a directed graph, $\mathcal{G}$, is strongly connected then the associated *normalized Laplacian*,

$$\mathcal{L} = I - AD^{-1}, \tag{4.1}$$

is an irreducible, singular M-matrix. For the remainder of this thesis, for UD graphs, the normalized Laplacian will be used.

Definition 4.1.3. *$H \in \mathbb{R}^{n\times n}$ is a singular M-matrix if and only if there exists $B \in \mathbb{R}^{n\times n}$, with $b_{ij} \geq 0$ for all i, j, such that $H = \rho(B)I - B$, where $\rho(B)$ is the spectral radius of B.*

The matrix decomposition shows that AD^{-1} is a column stochastic matrix with $\rho(AD^{-1}) = 1$ by the Gershgorin theorem applied to $(AD^{-1})^t$. With the definition of a singular M-matrix defined above, it is easy to see that $\mathcal{L}$ is an irreducible, singular M-matrix. The Perron-Frobenius theorem provides the following properties of singular M-matrices:

Theorem 5. *(Properties of singular M-matrices)[40]*

(1) Irreducible, singular M-matrices yield a unique solution to $A\mathbf{x} = \mathbf{0}$, up to a scaling, which can be chosen such that all components of $\mathbf{x}$ are strictly positive.

(2) Irreducible, singular M-matrices have nonpositive off-diagonal elements and strictly positive diagonal elements.

(3) If A has a strictly positive vector in its left or right null-space and its off-diagonal elements are nonpositive, then A is a singular M-matrix.

The constant vector $\mathbf{1}$ is a left null-space vector of $\mathcal{L}$. Since $\mathcal{L}$ is nonsymmetric, there exists a right null-space vector $\mathbf{v}$ such that $\mathcal{L}\mathbf{v} = \mathbf{0}$. From Theorem 5, the right null-space vector $\mathbf{v}$ is strictly positive and unique up to a scaling. The right null space vector can be found using linear solvers for Markov chain stationary probability systems since $\mathcal{L}\mathbf{v} = \mathbf{0}$ can be reformulated as

$$\mathcal{L}\mathbf{v} = \mathbf{0}, \tag{4.2}$$

$$(I - AD^{-1})\mathbf{v} = \mathbf{0}, \tag{4.3}$$

$$\mathbf{v} = AD^{-1}\mathbf{v}. \tag{4.4}$$

The matrix AD^{-1} for a normalized UD graph Laplacian can be seen as a Markov chain transition matrix and the eigenvector associated with an eigenvalue of one is referred to as the stationary probability vector. Many solvers exist for solving the above eigensystem (e.g. [41], [16], [44], [42]). The right null-space vector will be a key component in our NS-LAMG algorithm. Again, our goal is to find a robust solver for linear system

$$\mathcal{L}\mathbf{x} = \mathbf{b}. \tag{4.5}$$

For a solution to exist for the linear system, $\mathcal{L}\mathbf{x} = \mathbf{b}$, it must be the case that $\mathbf{b}$ is in the range of $\mathcal{L}$ and $\mathbf{1}^t\mathbf{b} = 0$. If $\mathbf{b}$ is not in the range of $\mathcal{L}$, one can project out the null-space of $\mathcal{L}^*$ from $\mathbf{b}$ by removing the constant vector.

4.2 NS-LAMG

In this section, we describe the building blocks of NS-LAMG, an algebraic multigrid algorithm for UD graph Laplacian linear systems, $\mathcal{L}\mathbf{x} = \mathbf{b}$. Like in LAMG, NS-LAMG uses low-degree elimination to reduce the problem size before forming the AMG coarse grid aggregation, which is especially useful for scale-free graphs. We now describe the AMG components of NS-LAMG. Let $\mathcal{L}$ be the normalized graph Laplacian of a strongly connected, UD graph, $\mathcal{G}$. Since $\mathcal{L}$ is nonsymmetric, the restriction and prolongation operators need not to be equal, i.e. $P \neq R$. First, the n_f fine level degrees of freedom of $\mathcal{L}$ are aggregated into n_c groups. This generates the *aggregation matrix*, $T \in \mathbb{R}^{n_f \times n_c}$, where $t_{ij} = 1$ if the fine level vertex i is a member of the coarse aggregate j, and $t_{ij} = 0$, otherwise. It is assumed that there is no overlap between the aggregates; thus, each row of the aggregation matrix, T, will contain only one nonzero term, i.e.

$$T = \begin{bmatrix} 1 & & \\ 1 & & \\ 1 & & \\ & 1 & \\ & 1 & \\ & & \ddots \end{bmatrix}. \tag{4.6}$$

In [37], P and R are scaled by the near null-space singular vectors. For a strongly connected, UD graph, the Laplacian is singular, and the kernel is of dimension one. We adapt the restriction and prolongation operators from [37] by defining the operators by the associated null-space vectors. Let $\mathbf{v}$ and $\mathbf{u}$ be the associated right and left null-space vector of $\mathcal{L}$. From

Section 2, the scaled adjacency matrix, AD^{-1}, of a UD graph is a column stochastic matrix, such that

$$\mathbf{1}^t AD^{-1} = \mathbf{1}^t. \tag{4.7}$$

The left null-space vector of $\mathcal{L}$ is then

$$\mathbf{1}^t \mathcal{L} = \mathbf{1}^t (I - AD^{-1}) = \mathbf{0}^t. \tag{4.8}$$

From Theorem 5, the right null-space vector, $\mathbf{v}$, is unique up to a scaling with strictly positive components. Define the interpolation operator as

$$P := diag(\mathbf{v})T, \tag{4.9}$$

and the restriction operator as

$$R := diag(\mathbf{u})T, \tag{4.10}$$

where $diag(\mathbf{v})$ and $diag(\mathbf{u})$ are diagonal matrices with the elements of the right and left null-space vectors along the diagonal. The restriction operator can be simplified to $R = T$ since $\mathbf{u} = \mathbf{1}$. An approximation to the right null-space vector, $\mathbf{v}$, is used to form P. The right null-space vector will be found using a semi-adaptive Markov chain AMG solver that will be discussed in Section 4.2.3. Assuming $\mathbf{v}$ is the exact right null-space vector, the coarse graph Laplacian is $\mathcal{L}_c = R^t \mathcal{L} P$, and the coarse graph correction is then given by

$$\mathbf{x} \leftarrow \mathbf{x} + P(R^t \mathcal{L} P)^\dagger R^t \mathbf{r}, \tag{4.11}$$

where $\mathbf{r} = \mathbf{b} - \mathcal{L}\mathbf{x}$ and $(R^t\mathcal{L}P)^\dagger$ is the Moore-Penrose pseudoinverse of the coarse graph Laplacian. The error propagation of this iteration is

$$\mathbf{e} \leftarrow (I - P(R^t\mathcal{L}P)^\dagger R^t\mathcal{L})\mathbf{e}. \tag{4.12}$$

Traditional smooth aggregation multigrid algorithms smooth the columns of R and P to enhance convergence. However, as in LAMG, R and P are kept unsmoothed since smoothing may create additional edges in the coarse graph representation. If the right null-space vector, $\mathbf{v}$, used to create P, is not scaled correctly, then this results in the coarse graph Laplacian to be scaled improperly as well. The decomposition of the coarse graph Laplacian can no longer be described as $I - AD^{-1}$, where AD^{-1} is a Markov chain transition matrix. To ensure that the coarse graph Laplacian remains a normalized graph Laplacian, the prolongation operator is redefined as

$$P := diag(\mathbf{v})T diag(T^t\mathbf{v})^{-1}. \tag{4.13}$$

This ensures that over each aggregate the columns of P sum to one. The coarse graph correction remains unchanged since the scaling is undone when the coarse graph correction is interpolated to the fine level, i.e. the current iterate is updated such that

$$\mathbf{x} \leftarrow \mathbf{x} + P diag(R^t\mathbf{v})^{-1}\mathbf{e}_c. \tag{4.14}$$

Notice, by definition, that $R^tP = I$, which implies that PR^t is a projection, i.e.

$$(PR^t)^2 = PR^t, \tag{4.15}$$

and $PR^t\mathbf{v} = \mathbf{v}$, where $\mathbf{v}$ is the right null-space vector of $\mathcal{L}$. It is easy to show that the right and left null-space vectors of $\mathcal{L}_c = R^t\mathcal{L}P$ are $\mathbf{1}_c$ and $R^t\mathbf{v}$, respectively,

$$\mathbf{1}_c^t\mathcal{L}_c = \mathbf{1}_c^tR^t\mathcal{L}P = \mathbf{1}_f^t\mathcal{L}P = \mathbf{0}_f^t, \tag{4.16}$$

and

$$\mathcal{L}_cR^t\mathbf{v} = R^t\mathcal{L}PR^t\mathbf{v} = R^t\mathcal{L}\mathbf{v} = \mathbf{0}_f. \tag{4.17}$$

Thus, $\mathcal{L}_c$ is a directed, normalized graph Laplacian because it can be decomposed as the identity and a column stochastic matrix:

$$\begin{aligned} \mathcal{L}_c &= R^t\mathcal{L}P, && (4.18)\\ &= T^t(I - AD^{-1})diag(\mathbf{v})Tdiag(T^t\mathbf{v})^{-1}, && (4.19)\\ &= I_c - T^tAD^{-1}diag(\mathbf{v})Tdiag(T^t\mathbf{v})^{-1}. && (4.20) \end{aligned}$$

NS-LAMG can be called recursively to form the multigrid hierarchy. As in LAMG, low-degree elimination is used to enhance the AMG algorithm by reducing the system before every aggregation level. The solution to the problem coarsened by low-degree elimination can be projected up to the finest level with zero error. The details of low-degree elimination will be discussed in the following section.

In NS-LAMG, it is assumed that the right null-space vector, $\mathbf{v}$, is given. However, it is unlikely that $\mathbf{v}$ is known. A good approximation of the right null-space vector is needed to build the coarse graph Laplacians in the NS-LAMG algorithm. To find the right null-space vector of $\mathcal{L}$, we need to solve the normalized Laplacian system, $\mathcal{L}\mathbf{v} = \mathbf{0}$, which can be rewritten as $AD^{-1}\mathbf{v} = \mathbf{v}$. This system can be treated as a Markov chain stationary distribution system. Thus, in order to solve $\mathcal{L}\mathbf{x} = \mathbf{b}$, we must first solve $\mathcal{L}\mathbf{v} = \mathbf{0}$. There exists a number of AMG algorithms for Markov chain stationary distribution systems (e.g. [4], [40], [41], [46], [47], [43]). However, the algorithms referenced are fully adaptive algorithms that update the multigrid hierarchy for every V-cycle, and thus are not easily parallelizable. We propose a semi-adaptive AMG solver that utilizes low-degree elimination. In order to use low-degree elimination effectively, the hierarchy must remain stationary.

Section 4.2.1 will discuss the details of low-degree elimination for directed graphs. Section 4.2.2 describes the coarse graph aggregation process. Section 4.2.3 describes solving for the right null-space vector using low-degree elimination and coarse graph aggregation.

4.2.1 Low-Degree Elimination

Due to the scale-free nature of most real-world graphs, many graphs have a large proportion of vertices that can be categorized as “low-degree”. In LAMG, low-degree elimination effectively removes the 1-D part of the graph and enhances the efficiency of the AMG coarse grid aggregation level. Low-degree elimination is shown to significantly reduce the problem size for real-world undirected graphs. A similar procedure can be done for directed graph Laplacians. For UU graphs, the degree is the number of symmetric edges incident to the vertex. For UD graphs, the degree is not as simple. Each vertex has edges directed towards the vertex and/or away from the vertex and an in-degree, d_i^{in}, and out-degree, d_i^{out}, respectively.

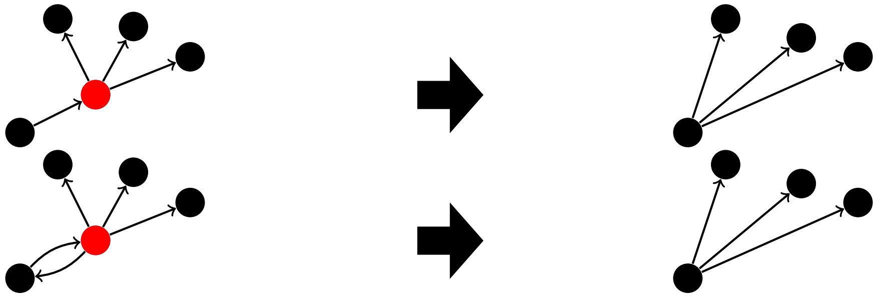


Figure 4.1: In-Degree One

For a vertex i, denote the *in-neighborhood* as the set of vertices that has an edge going toward i, i.e. $\mathcal{N}_i^{in} = \{j \in \mathcal{V} | (j,i) \in \mathcal{E}\}$, and the *out-neighborhood* of vertex i as $\mathcal{N}_i^{out} = \{j \in \mathcal{V} | (i,j) \in \mathcal{E}\}$. Define the unweighted in- and out-degree of a vertex as $\hat{d}_i^{in} = |\mathcal{N}_i^{in}|$ and $\hat{d}_i^{out} = |\mathcal{N}_i^{out}|$. If vertex i is removed from the linear system, then every vertex $j \in \mathcal{N}_i^{in}$ becomes attached to every vertex $k \in \mathcal{N}_i^{out}$. The goal is to eliminate vertices from the linear system without drastically increasing the nonzeros of the reduced problem matrix. The number of nonzeros added to the coarsened graph Laplacian, called τ_i, depends on the number of bi-directional edges vertex i has, i.e. $|\mathcal{N}_i^{in} \cap \mathcal{N}_i^{out}|$.

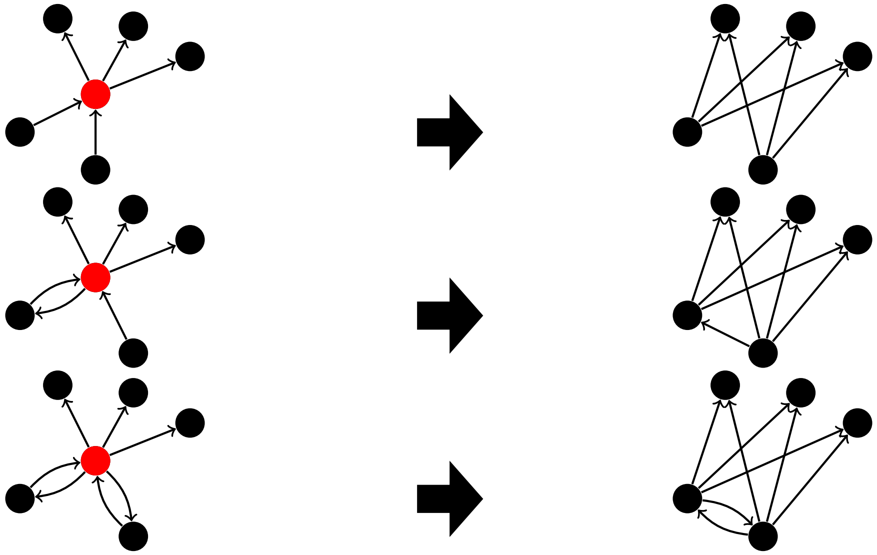


Figure 4.2: In-Degree Two

The reduced graph from low-degree elimination can be seen in Figure 4.1 and 4.2. In Figure 4.1, the left column depicts a vertex with one in-degree with the top having no bi-directional edges and the bottom having one bi-directional edge. The right column shows the coarsened graph when the center vertex is eliminated from the system. One can see that if a vertex with one in-degree is removed, at most $\hat{d}_i^{out}$ edges are added to the coarse graph representation that were not part of the original graph. If counting the nonzeros, $\hat{d}_i^{out} + \hat{d}_i^{in}$ edges were removed and at most $\hat{d}_i^{out}$ edges were added. Thus, the number of nonzeros in coarse graph Laplacian representation either remains the same or is reduced. The same occurs if the eliminated vertex has one out-degree. Figure 4.2 depicts the removal of a vertex with an in-degree of two. From the top, Figure 4.2 depicts the elimination with zero, one, and two bi-directional edges respectively. As the in-degree increases, the potential for additional nonzeros increases as well. Again, if the direction of the edges are reversed, the elimination of the vertex results in a similar formation. The number of nonzeros added, τ_i, can be classified as

$$\tau_i \leq (\hat{d}_i^{in} - 1)\hat{d}_i^{out} - (\hat{d}_i^{in} + |\mathcal{N}_i^{in} \cap \mathcal{N}_i^{out}|) = (\hat{d}_i^{out} - 1)\hat{d}_i^{in} - (\hat{d}_i^{out} + |\mathcal{N}_i^{in} \cap \mathcal{N}_i^{out}|). \quad (4.21)$$

The number of nonzeros, τ_i, may be less, as the additional edges may already be present in the matrix representation. The above relations are difficult to compute, as $|\mathcal{N}_i^{in} \cap \mathcal{N}_i^{out}|$ is hard to calculate quickly for each vertex. By removing vertices with a “total-degree,” sum of out-degree and in-degree (d_i^{total}), less than or equal to eight, we can guarantee τ_i to be less than four, which is equivalent to the undirected case. Vertices with an out-degree or in-degree of one will also be removed since the number of nonzeros of the problem matrix will always be reduced.

Let $\mathcal{V}_e$ be the set of vertices in $\mathcal{V}$ that are “low-degree”, i.e. the vertices will be eliminated from the linear system. Let $\mathcal{V}_n = \mathcal{V} \backslash \mathcal{V}_e$ and let $\mathcal{P}$ be a permutation matrix such that $\mathcal{P}^t \mathbf{x}$ lists all the $\mathcal{V}_e$ vertices values then all the $\mathcal{V}_n$ vertices values. Then the graph Laplacian and the associated linear system can be rewritten as

$$\mathcal{L} = \mathcal{P}^t \mathcal{L} \mathcal{P} = \begin{bmatrix} \mathcal{L}_{ee} & \mathcal{L}_{en} \\ \mathcal{L}_{ne} & \mathcal{L}_{nn} \end{bmatrix}, \quad \mathbf{b} = \mathcal{P}^t \mathbf{b} = \begin{bmatrix} \mathbf{b}_e \\ \mathbf{b}_n \end{bmatrix}, \quad \text{and} \quad \mathbf{x} = \mathcal{P}^t \mathbf{x} = \begin{bmatrix} \mathbf{x}_e \\ \mathbf{x}_n \end{bmatrix}. \quad (4.22)$$

Let

$$R_e = \begin{bmatrix} -\mathcal{L}_{ne} \mathcal{L}_{nn}^{-1} & I \end{bmatrix} \quad \text{and} \quad P_e = \begin{bmatrix} -\mathcal{L}_{ee}^{-1} \mathcal{L}_{en} \\ I \end{bmatrix}. \quad (4.23)$$

The linear system, $\mathcal{L}\mathbf{x} = \mathbf{b}$, reduces to the Schur complement system,

$$\mathcal{L}_c\mathbf{x}_c = \mathbf{b}_c, \quad \mathcal{L}_c = R_{_e}\mathcal{L}P_{_e}, \quad \mathbf{b}_c = R_{_e}\mathbf{b}, \tag{4.24}$$

and the solution is projected to the fine level by $\mathbf{x} = Q\mathbf{b}_c + P_{_e}\mathbf{x}_c$ where

$$Q = \begin{bmatrix} \mathcal{L}_{ee}^{-1} & 0 \end{bmatrix}^t. \tag{4.25}$$

Unlike in LAMG, $R_{_e}$ and $P_{_e}$ must be constructed separately since $\mathcal{L}$ is nonsymmetric. When a vertex i is identified as a vertex that will be removed, its neighbors, $j \in \mathcal{N} = \{\mathcal{N}_i^{out} \cup \mathcal{N}_i^{in}\}$, become ineligible to be eliminated. This guarantees that the eliminated vertices are independent of each other, which ensures $\mathcal{L}_{ee}$ is a diagonal matrix, and thus is invertible. Low-degree elimination for UD graphs modifies Algorithm 1 in [23] by choosing the fine vertices by looking at the total-degree and an in- or out-degree of one. Low-degree elimination decreases the problem size with minimal cost, and the solution to the coarse representation of the linear system is projected back to the fine space with no error.

4.2.2 AMG Aggregation

In order to effectively use low-degree elimination, the aggregates formed by traditional AMG methods must remain stationary; thus, the construction of an effective aggregate set is crucial. The NS-LAMG algorithm described here uses the affinity matrix described in LAMG as the strength of connection matrix, S, defined by equation (2.15). Once the affinity matrix has been determined, an approximation of the right singular vector, $\mathbf{v}_i$, is found by choosing the largest absolute value of each row of the test vector matrix, X, defined in equation (2.14), i.e

$$v_u := \max_j |X_{uj}|. \tag{4.26}$$

The approximate right null-space vector, $\mathbf{v}_i$, is then smoothed by applying θ relaxation sweeps to $\mathcal{L}\mathbf{v}_i = \mathbf{0}$. The following algorithm aggregates the vertices into aggregate sets, $S_1, \ldots, S_q$, encoded by a *status* vector, such that $status(i) := -1$ denotes an undecided vertex and $status(i) > 0$ indicates that vertex i will be aggregated with seed $status(i)$, such that $|S_i| \leq \eta$. The value η is an input in the algorithm and ensures that the aggregate size is not exceedingly large, since aggregate size affects the cycle complexity of NS-LAMG. The algorithm for forming the aggregates is defined as follows:

$$\text{For } u = sortperm(\mathbf{v}_i). \tag{4.27a}$$

$$\text{Let } \mathcal{N}_{in} \leftarrow \{v : s_{vu} \neq 0, \quad status(v) = -1\}, \tag{4.27b}$$

$$\mathcal{N}_{out} \leftarrow \{v : s_{uv} \neq 0, \quad status(v) = -1\}, \tag{4.27c}$$

$$\mathcal{N} = \mathcal{N}_{out} \cup \mathcal{N}_{in}. \tag{4.27d}$$

$$\text{If } status(u) = -1, \text{add } \eta \text{ strongest connected } j \in \mathcal{N} \text{ to aggregate } u. \tag{4.27e}$$

$$\tag{4.27f}$$

In the above algorithm, among the unassigned vertices, the vertex with the largest value in $\mathbf{v}_i$ is the seed vertex of a new aggregate. The sorting of $\mathbf{v}_i$ could be done with a partition sort or another type of sorting algorithm to increase speed and parallelism. From the unassigned vertices that are strongly connected to the seed vertex, η vertices are added to the aggregate. Vertices are chosen preferentially with respect to the strength of connection matrix. The process is repeated until every vertex has been assigned an aggregate. For the numerical results in Section 4.3, we set $K = 5$, $\eta = 4$, $\theta = 2$, and ν is a function of the current multigrid level, i.e. for the finest level ($l = 1$), $\nu = 2$ while on the coarser levels $\nu = 1 + l$. We limit $\nu \leq 10$ for all levels because, in our tests, more than ten test vectors does not improve performance.

4.2.3 Setup Phase

An adequate approximation to the right null-space vector $\mathbf{v}$, of $\mathcal{L}$ needs to be found. In this section, we will describe a semi-adaptive AMG algorithm that takes advantage of low-degree elimination for solving for the right null-space vector. To find the right null-space vector of $\mathcal{L}$, we need to solve the normalized Laplacian system, $\mathcal{L}\mathbf{v} = \mathbf{0}$, which can be rewritten as

$$\begin{aligned} \mathcal{L}\mathbf{v} &= \mathbf{0}, & (4.28) \\ (I - AD^{-1})\mathbf{v} &= \mathbf{0}, & (4.29) \\ AD^{-1}\mathbf{v} &= \mathbf{v}. & (4.30) \end{aligned}$$

This system can be treated as a Markov chain stationary distribution system since AD^{-1} is a column stochastic matrix. By using low-degree elimination and quality coarse graph aggregates, a semi-adaptive multigrid algorithm is used to find the right null-space vector. Recall that we can rewrite the exact solution, $\mathbf{v}$, in terms of the current approximation, $\mathbf{v}_i$, and its multiplicative error, $\mathbf{e}_i$, as $diag(\mathbf{v}_i)\mathbf{e}_i$, to obtain

$$\mathcal{L}diag(\mathbf{v}_i)\mathbf{e}_i = \mathbf{0}. \qquad (4.31)$$

It is assumed that the current approximation, $\mathbf{v}_i$, has nonzero elements, a property that the exact solution $\mathbf{v}$ has. Below is the two-level AMG algorithm for solving $\mathcal{L}\mathbf{v} = \mathbf{0}$ with multiplicative error correction:

$$\text{Do } \nu_1 \text{ smoothing steps on } \mathcal{L}\mathbf{v} = \mathbf{0}. \tag{4.32a}$$

$$\text{Compute residual } \mathcal{L}_c = T^t \mathcal{L} diag(\mathbf{v}) T^t diag(T^t \mathbf{v})^{-1} \tag{4.32b}$$

$$\text{Solve } \mathcal{L}_c \mathbf{v}_c = \mathbf{0} \tag{4.32c}$$

$$\text{Correct } \mathbf{v} \leftarrow diag(T diag(T^t \mathbf{v})^{-1} \mathbf{v}_c) \mathbf{v} \tag{4.32d}$$

$$\text{Do } \nu_2 \text{ smoothing steps on } \mathcal{L}\mathbf{v} = \mathbf{0}. \tag{4.32e}$$

Due to the singularity of $\mathcal{L}$, the approximate null-space vector, $\mathbf{v}_i$, must be normalized. To effectively use low-degree elimination, the aggregates must remain stationary (i.e. T at every level will remain the same), otherwise, low-degree elimination must be reconstructed for every cycle, which adds extra cost. If the aggregates are stationary, only the current iterate, $\mathbf{v}_i$, will be updated. The nonzero elements of $\mathcal{L}_c$ will change for every cycle, but the dimension and sparsity pattern will remain constant, and thus, a semi-adaptive algorithm is more parallelizable than traditional AMG methods for Markov chain stationary distributions systems.

4.2.4 Solve Phase

Once an adequate approximation to the right singular vector is found, the hierarchy of NS-LAMG remains stationary, i.e. the nonzero elements of the coarse graph Laplacians and transfer operators for both the AMG algorithm and low-degree elimination will remain the same. The solve phase uses k-cycles [31], also known as an iterate recombination phase in [23], at every aggregation level before the solution is projected to the fine graph. Details can be found in Section 3.4.5 in [23]. The next section will test the numerical validity of NS-LAMG described above.

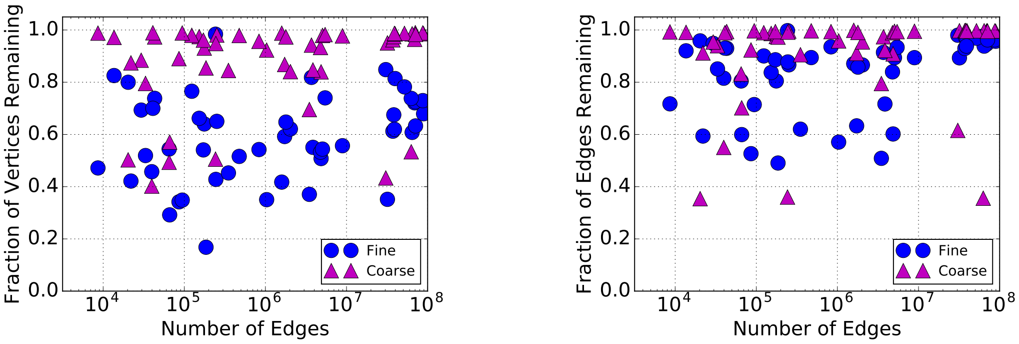
4.3 Numerical Results

A set of UD graphs was obtained from the Konect graph database[1] . For each graph, the largest strongly connected component was found and used to perform numerical testing. An implementation of the NS-LAMG algorithm, written in the language Julia was developed to serve as a platform for the investigations described in this section. Convergence factor, cycle complexity, effective convergence factor, and total work, as described in Section 3.3, are used to assess the performance of NS-LAMG. Unlike AMG methods for symmetric problems, the error propagation matrix for the coarse grid correction, $(I - \Pi) = (I - P(R^t \mathcal{L} P)^{-1} R^t \mathcal{L})$, derived in equation (4.11), is an oblique projection with respect to any reasonable inner product. Since $(I - \Pi)$ is an oblique projection, convergence is not guaranteed. A two-grid convergence proof for this approach for nonsymmetric problems is found in [37]. In the first set of numerical tests, we will look at the reduction of dimension and sparsity that results from low-degree elimination on a set of UD graphs.

4.3.1 Low-Degree Elimination

The graphs from the Konect graph database are pulled from real-world models. Thus, many of them are scale-free. Figure 4.3a shows the fraction of vertices remaining for each graph after low-degree vertices are eliminated on the finest level and the first coarse level of the algorithm. The graphs are plotted with respect to the number of edges. The circles represent the fraction of vertices remaining after low-degree elimination is performed on the original graph. The triangles represent the fraction of vertices remain after low-degree elimination is performed after the first elimination level and a coarse graph aggregation level. For the first low-degree elimination level, over 47% of graphs have less than 60% of the vertices remaining after low-degree elimination. On the second low-degree elimination, level after a low-degree elimination level and aggregation level, only 18% of graphs have less

[1] konect.uni-koblenz.de



(a) Fraction of Vertices

(b) Fraction of Edges

Figure 4.3: Low-Degree Elimination: Circles represent the fraction of vertices/edges remaining after low-degree elimination on the original graph. Triangles represent the fraction of vertices/edges remaining for each graph (plotted with respect to the number of vertices on the finest level) after low-degree elimination is performed after a coarse graph aggregation level.

that 80% of vertices remaining. There are a handful of graphs where low-degree elimination reduced the problem to less than 60% of the vertices remaining. Figure 4.3b depicts the fraction of edges remaining, which is the fraction of nonzeros in the coarse graph Laplacian after low-degree elimination compared with the number of nonzeros of fine graph Laplacian before low-degree elimination. A similar analysis shows that the number of nonzeros is reduced more aggressively on the finest level than on the coarse levels. It is apparent that low-degree elimination reduces the size of the graph significantly on the finest level, while also reducing the number of nonzeros in the coarse graph Laplacian, but may not be as effective on the coarser levels. However, once the vertices are labeled as low-degree, there is minimal cost in reducing the system, and thus, low-degree elimination is advantageous if it increases NS-LAMG's performance. In the first set of numerical tests, we will compare NS-LAMG's performance with and without low-degree elimination.

4.3.2 Performance with and without Low-Degree Elimination

The next test will compare the performance of NS-LAMG when using low-degree elimination with the performance of NS-LAMG without low-degree elimination. Using NS-LAMG method described above, with a "good" approximation to the right singular vector, we solved the system $\mathcal{L}\mathbf{x} = \mathbf{b}$ for each graph with V-cycles that employed two pre- and post- weighted Jacobi smoothing iterations, a cut off of 100 vertices for the coarsest graph, and a residual tolerance of 10^{-7} in the one-norm. Figure 4.4a(log-scale) plots the constant K such that

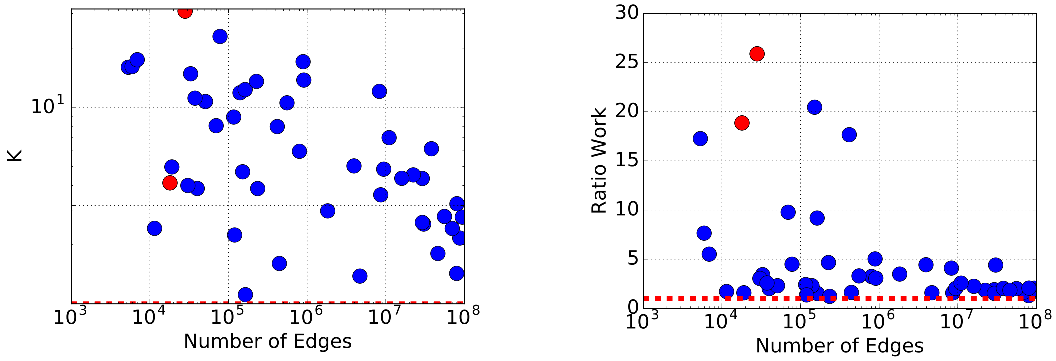
$$\rho_{elim} = \rho_{no\text{-}elim}^{K},$$

with respect to the convergence factors of NS-LAMG with and without elimination. Figure 4.4b depicts the ratio of the total work of NS-LAMG without and without low-degree elimination. For each measure, if the graph is above the red line, this indicates that low-degree elimination improved the measure. It is clear from the figures that for all graphs tested, low-degree elimination improved NS-LAMG's performance. The graphs depicted in red are the graphs for which NS-LAMG failed to converge without low-degree elimination. Thus, for the remainder of this section, all numerical testing will be performed with low-degree elimination.

4.3.3 Performance of NS-LAMG Semi-Adaptive Markov Chain Solver

In the following tests we will compare NS-LAMG's semi-adaptive Markov Chain stationary distribution system solver to a traditional full-adaptive AMG Markov chain solver describe in [41]. Using the same problem parameters as the previous section, we will solve $\mathcal{L}\mathbf{v} = \mathbf{0}$. Note that, for the full-adaptive solver the total work is with respect to the final cycle. Figure 4.6a(log-scale) plots the constant K such that

$$\rho_{semi-adaptive} = \rho_{full\text{-}adaptive}^{K},$$

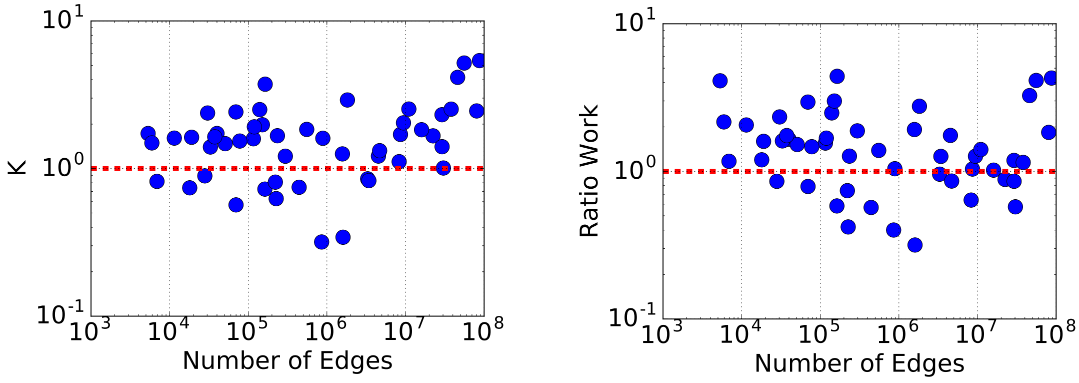


(a) Convergence Factor: $K = \frac{\ln(\text{Elimination})}{\ln(\text{No Elimination})}$

(b) Total Work: $\frac{\text{No Elimination}}{\text{Elimination}}$

Figure 4.4: Low-Degree Elimination

where $\rho_{semi-adaptive}$ and $\rho_{full-adaptive}$ are the convergence factors for the semi-adaptive and full-adaptive algorithms, respectively. Figure 4.6b(log-scale) depict the ratio of the total work of full-adaptive versus semi-adaptive Markov chain solver for each graph in the dataset. Again, for each figure, if the graph is above the red horizontal line, the semi-adaptive solver results in a better measurement than the full-adaptive solver. For 76% of the graphs tested the semi-adaptive solver resulted in lower convergence factors and for 72% of the graphs tested the semi-adaptive solver resulted in less total work. For the full-adaptive solver, the total work is only taking into account the work involved of applying the operators for the last V-cycle. However, the full-adaptive solver updates the hierarchy for every cycle which add cost. The semi-adaptive solver performs slightly better and is more parallelizable. Low-degree elimination is the reason the semi-adaptive solver performs well. If low-degree elimination was not used, the performance would drastically suffer. Due to the effectiveness of the semi-adaptive solver, in the next section we will test the semi-adaptive solver on traditional Markov chain stationary distribution systems.



(a) Convergence Factor: $K = \frac{\ln(\text{full-adaptive})}{\ln(\text{semi-adaptive})}$

(b) Total Work: $\frac{\text{full-adaptive}}{\text{semi-adaptive}}$

Figure 4.5: Performance of semi-adaptive verse full-adaptive Markov Chain solver

4.3.4 Tolerance of Null-Space Vector

The accuracy of the right null-space vector, $\mathbf{v}$, determines how well the NS-LAMG algorithm converges. The range of the interpolation operator should include $\mathbf{v}$ or at least contain a good approximation of it. However, a less accurate $\mathbf{v}$ results in a convergent NS-LAMG algorithm if there is a gap between the smallest and the second smallest singular values. Given a residual tolerance, $\beta = \|\mathcal{L}\mathbf{v}\|_1$, Table 4.1 depicts the percent of graphs that converged using the NS-LAMG algorithm. The two graphs that require a higher tolerance are $amazon0601$ and $web\text{-}NotreDame$. Table 4.2 depicts the performance measurements of the two graphs. An adaptive procedure could be implemented to increase the tolerance for $\mathbf{v}$ if the NS-LAMG algorithm begins to stall.

β	% of Graphs
10^{-3}	17.5%
10^{-4}	68.4%
10^{-5}	89.5%
10^{-6}	94.7%
10^{-7}	96.5%
10^{-8}	98.2%
10^{-10}	100%

Table 4.1: Tolerance of $\mathbf{v}$

$Graph:$	β	its	ρ	γ	ECF	Work
$web\text{-}NotreDame$	10^{-10}	74	0.7522	6.2681	0.9555	463.84
$amazon0601$	10^{-8}	55	0.6686	10.4439	0.8586	574.41

Table 4.2: Graph with a tolerance $\beta \geq 10^{-8}$

4.3.5 Performance of NS-LAMG and Comparison to GMRES

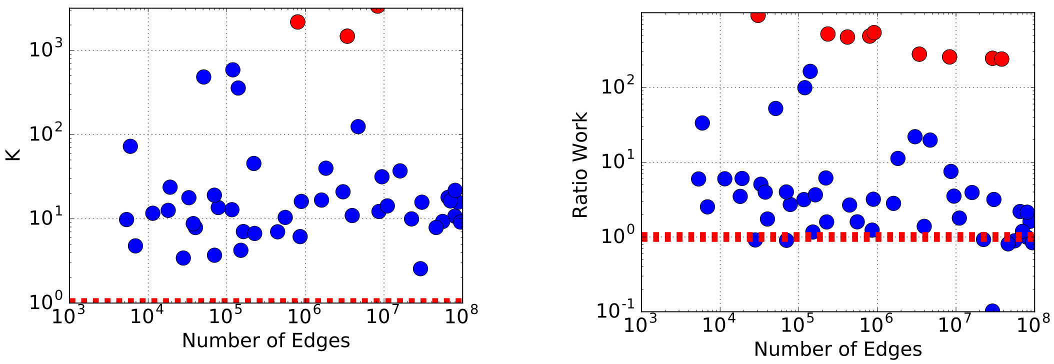
With the same parameters as the previous test, we compare NS-LAMG's performance to Generalized Minimal Residual Method with Restart (GMRES(20)) [34]. GMRES is a classical Krylov iterative solver for nonsymmetric problems. The average work unit per iteration for GMRES(20) for a matrix size n and number of nonzeros nnz is $WU = (20 + 3 + 1/20)\frac{n}{nnz} + 1$ and the total work is $WU * iterations$. Figure 4.6a(log-scale) plots the constant K such that

$$\rho_{NS\text{-}LAMG} = \rho_{GMRES}^{K},$$

where $\rho_{NS\text{-}LAMG}$ and ρ_{GMRES} are the convergence factors for NS-LAMG and GMRES, respectively for each graph. Figure 4.6b(log-scale) depict the ratio the total work of GMRES(20) versus NS-LAMG for each graph in the dataset. Above the red horizontal line, NS-LAMG results in a better measurement than GMRES(20). For the graphs that are depicted in red, GMRES(20) failed to converge in $20,000$ iterations (Graphs: $amazon0601$, $dbpedia\text{-}all$, $digg\text{-}friends$, $soc\text{-}LiveJournal1$, $trec\text{-}wt10g$, $web\text{-}Google$, $web\text{-}NotreDame$, $wikipedia\text{-}discussions\text{-}de$, $wiki\text{-}talk\text{-}en$, $wiki\text{-}talk\text{-}es$). Figure 4.6a shows that NS-LAMG produces much better convergence factors than GMRES(20). In Figure 4.6b, NS-LAMG requires less total work for 85.2% of the graphs tested. The most notable results where in Figure 4.6b. In some cases, GMRES(20) required over 10 or even 100 times more work than NS-LAMG.

4.4 Conclusions

A highly successful AMG algorithm, NS-LAMG, for UD graph Laplacian systems was developed. NS-LAMG utilized low-degree elimination and stationary aggregates with Petrov-Galerkin transfer operators. Low-degree elimination in [23] was generalized for UD graphs and was shown to enhance the performance of NS-LAMG algorithm as well as substantially reduces the problem size on the finest level. Numerical tests showed that NS-LAMG resulted in lower effective convergence factors and less work than GMRES(20). In the setup phase, we have developed a semi-adaptive AMG solver for Markov chain stationary distribution systems that is highly parallelizable. The semi-adaptive AMG solver for Markov chain systems was



(a) Convergence Factor: $K = \frac{\ln(NS\text{-}LAMG)}{\ln(GMRES)}$

(b) Total Work: $\frac{GMRES}{NS\text{-}LAMG}$

Figure 4.6: Performance of NS-LAMG verse GMRES

shown to be robust for solving for the right null-space vector for real-world graph Laplacian systems. These results suggest that for Markov chain stationary distribution systems low-degree elimination used in conjunction with an adequate coarse graph-aggregation process creates a highly parallelizable solver than traditional AMG Markov chain solvers for real-world, scale-free Markov chains. By providing a robust multilevel iterative solver, NS-LAMG, for UD graph Laplacian linear systems, we hope that new applications and research on UD graphs can be studied.

Chapter 5

Nonsymmetric Algebraic Multigrid Theory

5.1 NS-AMG and Generalizations of Approximation Properties

Even though there exist many nonsymmetric AMG algorithms that have been developed for Markov chains, PDEs involving convection-diffusion, and various other applications, there is a lack in the theoretical development for nonsymmetric AMG (NS-AMG). This chapter provides theoretical tools for NS-AMG with the hope to further the convergence theory. Let $H = U\Sigma V^* \in \mathbb{R}^{n\times n}$ be the SVD of a nonsingular, nonsymmetric matrix. Let $Q = VU^*$ and order the singular values such that $0 < \sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_n$ with the associated left and right QH-normalized singular vector pairs $(\mathbf{u}_i, \mathbf{v}_i)$. Let $R \in \mathbb{R}^{n\times n_c}$ and $P \in \mathbb{R}^{n\times n_c}$ be the restriction and prolongation operators, respectively. Now, consider applying symmetric AMG to the SPD linear system in (1.3) or (1.4). To satisfy any of the symmetric approximation properties, the minimal eigenvectors should be well represented by the coarse grids. The minimal eigenvectors of QH are the minimal right singular vectors of H, and the minimal eigenvectors of HQ are the minimal left singular vectors of H. If we have $R := P$, the two-grid error propagator for the linear systems (1.3) and (1.4) is given by

$$(I - \Pi_1) = I - P(P^*QHP)^\dagger P^*QH,$$

and

$$(I - \Pi_2) = (QR)((QR)^*QHQR)^\dagger(QR)^*QH,$$

respectfully. In the QH-norm ($\|\cdot\|_{QH} = \sqrt{\langle QH\cdot,\cdot\rangle}$), Π_1 and Π_2 are orthogonal projections. Though either projection would lead to a convergent AMG algorithm in the QH-norm, as the symmetric AMG applies, neither projection should be used in practice since QH results in a dense matrix. Instead, $(I - \Pi_1)$ and $(I - \Pi_2)$ are approximated by

$$(I - \Pi) = I - P(H_c)^{-1}R^*H, \tag{5.1}$$

where $H_c = R^*HP$ is the traditional coarse-grid AMG operator. In the QH-norm, Π results in an oblique projection with respect to any reasonable norm. See [6] for details. We will assume that H_c is nonsingular. Note that H_c maybe singular even if P and R are full rank and H is nonsingular. In Theorem 6, we show sufficient conditions to ensure the error propagation operator is stable, i.e. $\|\Pi\|_{QH} \leq C$ for some small constant $C > 1$, which ensures H_c is nonsingular given H is nonsingular. We first generalize the symmetric approximation properties to NS-AMG in the following definitions with respect to the interpolation operator and QH:

Definition 5.1.1 (Nonsymmetric Strong Approximation Property (NSAP) on P with respect to QH)**.** *An interpolation operator, P, satisfies the NSAP with constant K_P with respect to QH if, for any $\mathbf{v}$ on the fine grid, there exists an $\mathbf{e}_c$ on the coarse grid such that*

$$\|\mathbf{v} - P\mathbf{e}_c\|_{QH}^2 \leq \frac{K_P}{\|QH\|}\|QH\mathbf{v}\|^2. \tag{5.2}$$

Due to orthogonality in the QH-norm we have,

$$\|(I - \Pi_1)\mathbf{v}\|_{QH}^2 = \inf_{\mathbf{v}_c} \|\mathbf{v} - P\mathbf{v}_c\|_{QH}^2 \leq \frac{K_P}{\|QH\|}\|QH\mathbf{v}\|^2. \tag{5.3}$$

Definition 5.1.2 (Nonsymmetric Weak Approximation Property (NWAP) on P with respect to QH)**.** *An interpolation operator, P, satisfies the NWAP with constant K_w if, for any $\mathbf{v}$ on the fine grid, there exists an $\mathbf{e}_c$ on the coarse grid such that*

$$\|\mathbf{v} - P\mathbf{e}_c\|^2 \leq \frac{K_w}{\|QH\|}\|\mathbf{v}\|_{QH}^2. \tag{5.4}$$

Definition 5.1.3 (Nonsymmetric Super Strong Approximation Property (NSSAP) on P with respect to QH)**.** *An interpolation operator, P, satisfies the NSSAP with constant K_s if, for any $\mathbf{v}$ on the fine grid, there exists an $\mathbf{e}_c$ on the coarse grid such that*

$$\|\mathbf{v} - P\mathbf{e}_c\|^2 \leq \frac{K_s}{\|QH\|^2}\|QH\mathbf{v}\|^2. \tag{5.5}$$

When H is symmetric, $QH = V\Sigma V^* = H$ and NWAP, NSAP, and NSSAP are equivalent to their respective symmetric definitions. It is known, in the symmetric case, that the Super Strong Approximation Property implies Weak Approximation Property and is equivalent to Strong Approximation Property. Lemma 5.1.1 proves the same result for the nonsymmetric approximation properties for the interpolation operator with respect to QH.

Lemma 5.1.1 (Equivalence of approximation properties)**.**

(1) Assume P has a NSSAP with respect to QH. Then, P also has a NWAP with respect to QH, with constant $K_w = K_s$.

(2) Assume P has a NSSAP with respect to QH. Then, P also has a NSAP with respect to QH, with constant $K_P = K_s$.

(3) Assume P has a NSAP with respect to QH. Then, P also has a NWAP with respect to QH, with constant $K_w = K_P^2$.

(4) Assume P has a NSAP with respect to QH. Then, P also has a NSSAP with respect to QH, with constant $K_s = K_w K_P$

Proof.

(1) Assume P has a NSSAP with respect to QH. Let $\mathbf{e}_c$ be the vector such that Definition 5.1.3 is satisfied. Then for any $\mathbf{v}$ on the fine grid, we have

$$\begin{aligned}
\|\mathbf{v} - P\mathbf{e}_c\|^2 &\leq \frac{K_s}{\|QH\|^2}\|QH\mathbf{v}\|^2, && \text{(Definition 5.1.3)}\\
&= \frac{K_s}{\|QH\|^2}\|(QH)^{1/2}(QH)^{1/2}\mathbf{v}\|^2,\\
&\leq \frac{K_s}{\|QH\|^2}\|(QH)^{1/2}\|^2\|(QH)^{1/2}\mathbf{v}\|^2,\\
&= \frac{K_w}{\|QH\|}\|\mathbf{v}\|^2_{QH},
\end{aligned}$$

which satisfies the NWAP on P with respect to QH with $K_w = K_s$.

(2) Assume P has a NSSAP with respect to QH. Let $\mathbf{e}_c$ be the vector such that Definition 5.1.3 is satisfied. Then for any $\mathbf{v}$ on the fine grid, we have

$$\begin{aligned}
\|\mathbf{v} - P\mathbf{e}_c\|^2_{QH} &\leq \|QH\|\|\mathbf{v} - P\mathbf{e}_c\|^2,\\
&\leq \frac{K_P}{\|QH\|}\|QH\mathbf{v}\|^2, && \text{(Definition 5.1.3)}
\end{aligned}$$

which satisfies the NWAP with respect to P and QH with $K_P = K_s$.

(3) Assume P has a NSAP with respect to QH. Let $\mathbf{w}_c$ be the vector such that Definition 5.1.1 is satisfied for $(QH)^{-1}(\mathbf{v}-P\mathbf{e}_c)$. Let $\mathbf{e}_c$ be the vector such that $\inf_{\mathbf{e}_c}\|\mathbf{v}-P\mathbf{e}_c\| = \|(I-\Pi_1)\mathbf{v}\|$ is satisfied and, thus, $\mathbf{e}_c$ implicitly satisfies Definition 5.1.1 for $\mathbf{v}$. Thus,

$\mathbf{v} - P\mathbf{e}_c$ is QH-orthogonal to $\mathcal{R}(P)$ and we have

$$\begin{aligned}
\|\mathbf{v} - P\mathbf{e}_c\|^2 &= \langle QH(\mathbf{v} - P\mathbf{e}_c), (QH)^{-1}(\mathbf{v} - P\mathbf{e}_c)\rangle, \\
&= \langle QH(\mathbf{v} - P\mathbf{e}_c), (QH)^{-1}(\mathbf{v} - P\mathbf{e}_c) - P\mathbf{w}_c\rangle, \\
&\leq \|(\mathbf{v} - P\mathbf{e}_c)\|_{QH}\|(QH)^{-1}(\mathbf{v} - P\mathbf{e}_c) - P\mathbf{w}_c\|_{QH}, \\
&\leq \|(\mathbf{v} - P\mathbf{e}_c)\|_{QH}\sqrt{\frac{K_P}{\|QH\|}}\|QH(QH)^{-1}(\mathbf{v} - P\mathbf{e}_c)\|, \quad \text{(Definition 5.1.1)} \\
&= \|(\mathbf{v} - P\mathbf{e}_c)\|_{QH}\sqrt{\frac{K_P}{\|QH\|}}\|(\mathbf{v} - P\mathbf{e}_c)\|.
\end{aligned}$$

This implies

$$\begin{aligned}
\|\mathbf{v} - P\mathbf{v}_c\| &\leq \sqrt{\frac{K_P}{\|QH\|}}\|(\mathbf{v} - P\mathbf{v}_c)\|_{QH}, \\
&\leq \frac{K_P}{\|QH\|}\|QH\mathbf{v}\|. \qquad \text{(Definition 5.1.1)}
\end{aligned}$$

Squaring the results yields,

$$\begin{aligned}
\|\mathbf{v} - P\mathbf{v}_c\|^2 &\leq \frac{K_w}{\|QH\|^2}\|QH\mathbf{v}\|^2, \\
&\leq \frac{K_w}{\|QH\|}\|\mathbf{v}\|_{QH}^2,
\end{aligned}$$

which satisfies the NWAP on P with respect to QH with $K_w = K_P^2$.

(4) Assume P has a NSAP with respect to QH. From Part 3, the NSAP on P implies the NWAP on P with respect to QH. Let $\mathbf{x}_c$ be the vector that satisfies Definition 5.1.2 for $\mathbf{v} - P\mathbf{y}_c$ and let $\mathbf{y}_c$ be the vector that satisfies Definition 5.1.1 for $\mathbf{v}$. Then

we have

$$\begin{aligned}\|(\mathbf{v}-P\mathbf{y}_c)-P\mathbf{x}_c\|^2 &\leq \frac{K_w}{\|QH\|}\|(\mathbf{v}-P\mathbf{y}_c)\|^2_{QH}, && \text{(Definition 5.1.2)}\\ &\leq \frac{K_wK_P}{\|QH\|^2}\|QH\mathbf{v}\|^2. && \text{(Definition 5.1.1)}\end{aligned}$$

Let $K_s = K_wK_P$ and $\mathbf{e}_c = \mathbf{y}_c + \mathbf{x}_c$, which yields the result,

$$\|\mathbf{v}-P\mathbf{e}_c\|^2 \leq \frac{K_s}{\|QH\|^2}\|QH\mathbf{v}\|^2.$$

which satisfies the NSSAP on P with respect to QH with $K_s = K_wK_P$.

□

For nonsymmetric problems, it is often the case that the restriction and prolongation operators are chosen such that $P \neq R$. Thus, the approximation properties on R should also be considered. Definition 5.1.4 and 5.1.5 define the NSAP on QR and R with respect to QH and HQ, respectively. Definition 5.1.4 and 5.1.5 are interchangeable, which is shown in Lemma 5.1.2.

Definition 5.1.4 (Nonsymmetric Strong Approximation Property (NSAP) on QR with respect to QH)**.** *An restriction operator, QR, satisfies the NSAP with constant K_R with respect to QH if, for any $\mathbf{v}$ on the fine grid, there exists an $\mathbf{e}_c$ on the coarse grid such that*

$$\|\mathbf{v}-QR\mathbf{e}_c\|^2_{QH} \leq \frac{K_R}{\|QH\|}\|QH\mathbf{v}\|^2. \tag{5.6}$$

Due to orthogonality in the QH-norm we have,

$$\|(I-\Pi_2)\mathbf{v}\|^2_{QH} = \inf_{\mathbf{v}_c}\|\mathbf{v}-QR\mathbf{v}_c\|^2_{QH} \leq \frac{K_R}{\|QH\|}\|QH\mathbf{v}\|^2. \tag{5.7}$$

Definition 5.1.5 (Nonsymmetric Strong Approximation Property (NSAP) on R with respect to HQ)**.** *A restriction operator, R, satisfies the NSAP with constant K_R with respect to HQ if, for any $\mathbf{v}$ on the fine grid, there exists an $\mathbf{e}_c$ on the coarse grid such that*

$$\|\mathbf{v} - R\mathbf{e}_c\|_{HQ}^2 \leq \frac{K_R}{\|HQ\|}\|HQ\mathbf{v}\|^2. \tag{5.8}$$

Due to orthogonality in the HQ-norm we have,

$$\|(I - R(R^*HQR)^{-1}R^*H)\mathbf{v}\|_{HQ}^2 = \inf_{\mathbf{v}_c} \|\mathbf{v} - R\mathbf{v}_c\|_{HQ}^2 \leq \frac{K_R}{\|HQ\|}\|HQ\mathbf{v}\|^2. \tag{5.9}$$

Lemma 5.1.2. *The NSAP on QR with respect to QH holds if and only if the NSAP on R with respect to HQ holds.*

Proof. Let $\mathbf{u} = Q^*\mathbf{v}$. We then have the following equalities:

$$\begin{aligned}
\|(I - \Pi_2)\mathbf{v}\|_{QH}^2 &= \|(I - QR((QR)^*QHQR)^{-1})QR)^*H)\mathbf{v}\|_{QH}^2, & (5.10)\\
&= \|Q(I - R(R^*HQR)^{-1}R^*HQ)Q^*\mathbf{v}\|_{QH}^2, & (5.11)\\
&= \|(I - R(R^*HQR)^{-1}R^*H)\mathbf{u}\|_{HQ}^2, & (5.12)
\end{aligned}$$

$$\begin{aligned}
\|\mathbf{v} - QR\mathbf{v}_c\|_{QH}^2 &= \|Q(Q^*\mathbf{v} - R\mathbf{v}_c)\|_{QH}^2, & (5.13)\\
&= \|Q(\mathbf{u} - R\mathbf{v}_c)\|_{QH}^2, & (5.14)\\
&= \|\mathbf{u} - R\mathbf{v}_c\|_{HQ}^2, & (5.15)
\end{aligned}$$

and

$$\begin{aligned}
\|QH\mathbf{v}\|^2 &= \|QHQQ^*\mathbf{v}\|^2, & (5.16)\\
&= \|HQ\mathbf{u}\|^2. & (5.17)
\end{aligned}$$

Thus, NSAP on QR with respect to QH is equivalent to the NSAP on R with respect to HQ □

Remark 5.1.1. *The NSAP, NWAP, NSSAP on QR with respect to QH can be defined similarly as the approximation properties with respect to the interpolation operator P, i.e. Definition 5.1.1, 5.1.2, and 5.1.3. The same relations as in Lemma 5.1.1 can be shown for the restriction operator QR with respect to QH, by similar proof.*

In addition to the approximation properties, it is important that coarse-grid correction is *stable*, that is, coarse-grid correction can only increase the error by some small constant $C > 1$ (Definition 5.1.6). For SPD matrices, we have an orthogonal coarse-grid correction, which implicitly implies $C = 1$ and means that the error cannot be increased. In the nonsymmetric setting, we typically do not have an orthogonal coarse-grid correction and must ensure that this constant is bounded.

Definition 5.1.6 (Stability of Π)**.**

$$||\Pi||_{QH}^{2} \leq C \tag{5.18}$$

Thus far, in looking at the approximation properties, we have only assumed that there is *some* vector $\mathbf{v}$ that satisfies the given property. In fact, the best $\mathbf{v}$ will be given by an orthogonal projection onto the range of P in the appropriate norm. Again, for nonsymmetric problems, the error propagation operator of the coarse-grid correction is typically not an orthogonal projection. For an effective method, in conjunction with stability, we also need an approximation property on the error propagation operator used in practice, $(I-\Pi)$. In [6], sufficient conditions for an approximation property for $(I-\Pi)$ are shown to be stability and the NSAP on P with respect to QH ([6] Lemma 2.2). It was further shown in [6] that stability and the NSAP on P with respect to QH result in two-grid convergence for a sufficient number of smoothing iterations ([6], Theorem 2.3). Because a non-orthogonal coarse-grid correction can increase the total error in some cases, additional smoothing iterations are necessary to ensure that iterations are reducing the error; something which is not necessary in the symmetric setting. Building on the work from [6], we further explore sufficient conditions for two-grid convergence, along with the relation of stability of Π and approximation properties on P and R. Example 5.1.1 shows that no two imply the third.

Example 5.1.1.

(1) The NSAP on P with respect to QH and stability $\not\Rightarrow$ the NSAP on QR with respect to QH.

(2) The NSAP on QR with respect to QH and stability $\not\Rightarrow$ the NSAP on P with respect to QH.

(3) The NSAP on P and NSAP on QR with respect to QH $\not\Rightarrow$ stability of Π.

Proof.

(1) Let $H \in \mathbb{R}^{4\times 4}$ with the associated singular value triplets, $(\sigma_i, \mathbf{u}_i, \mathbf{v}_i)$, where the singular vectors are QH-normalized. Assume $\sigma_i \leq 1 \ \forall i$, $\sigma_i < 1/2$ for $i = 1, 2, 3$, and $\sigma_4 \geq 1/2$. Define P and R by the following columns:

$$P = \left[\mathbf{v}_1/\sqrt{\sigma_1}, \mathbf{v}_2/\sqrt{\sigma_2}, \mathbf{v}_3/\sqrt{\sigma_3}\right],$$

$$R = \left[\tfrac{\mathbf{u}_1+\mathbf{u}_4}{\sqrt{\sigma_1}}, \mathbf{u}_2/\sqrt{\sigma_2}, \mathbf{u}_3/\sqrt{\sigma_3},\right].$$

Then $P^*QHP = I$ and $(QR)^*QH(QR) = \begin{bmatrix} 2 & & \\ & 1 & \\ & & 1 \end{bmatrix}$. We can verify that Π is stable:

$$\begin{aligned}
\|\Pi\|_{QH}^2 &= \|P((QR)^*QHP)^{-1}(QR)^*QH\|_{QH}^2, \\
&= \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\langle QHP((QR)^*QHP)^{-1}(QR)^*QH\mathbf{x}, P((QR)^*QHP)^{-1}(QR)^*QH\mathbf{x}\rangle}{\|\mathbf{x}\|_{QH}^2}, \\
&= \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\langle ((QR)^*QHP)^{-1}(QR)^*QH\mathbf{x}, ((QR)^*QHP)^{-1}(QR)^*QH\mathbf{x}\rangle}{\|\mathbf{x}\|_{QH}^2}, \\
&\leq \|(QR)^*QHP)^{-1}\|^2 \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|(QR)^*QH\mathbf{x}\|^2}{\|\mathbf{x}\|_{QH}^2}, \\
&\leq \|(QR)^*QHP)^{-1}\|^2 \|(QR)^*(QH)^{1/2}\|^2 \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|(QH)^{1/2}\mathbf{x}\|^2}{\|\mathbf{x}\|_{QH}^2}, \\
&= \|(QR)^*(QH)^{1/2}\|^2, \qquad\qquad (\|A\|^2 = \|AA^*\|) \\
&= 2,
\end{aligned}$$

and the NSAP on P is easily satisfied with $K_P = 2$. However, for QR we have $\|(I - \Pi_2)\mathbf{v}_1\|_{QH}^2 = 1$ and $\|QH\mathbf{v}_1\|^2 = \sigma_1$. Thus, $K_R = \frac{C}{\sigma_1}$ which could be unbounded, as we could have $\sigma_1 << 1$.

(2) A similar construction to Part 2 shows the NSAP on R with respect to QH and stability do not imply the NSAP on P with respect to QH.

(3) Let $H \in \mathbb{R}^{4\times4}$ with the associated singular value triplets, $(\sigma_i, \mathbf{u}_i, \mathbf{v}_i)$, where the singular vectors are QH-normalized. Assume $\sigma_i \leq 1\ \forall i$, $\sigma_i < 1/2$ for $i = 1, 2$, and $\sigma_i \geq 1/2$ for $i = 3, 4$. Let $P = \left[\mathbf{v}_1/\sqrt{\sigma_1}, \mathbf{v}_2/\sqrt{\sigma_2}, \mathbf{v}_3/\sqrt{\sigma_3}\right]$ and $R = \left[\mathbf{u}_1/\sqrt{\sigma_1}, \mathbf{u}_2/\sqrt{\sigma_2}, \mathbf{u}_4/\sqrt{\sigma_4},\right]$, then $P^*QHP = (QR)^*QH(QR) = I$. The NSAP on P and on QR with respect to

QH are satisfied with $K_P = K_R = 2$. However, Π is not stable:

$$\begin{aligned}
\|\Pi\|_{QH}^2 &= \|P((QR)^*QHP)^{-1}(QR)^*QH\|_{QH}^2 \\
&= \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\langle QHP((QR)^*QHP)^{-1}(QR)^*QH\mathbf{x}, P((QR)^*QHP)^{-1}(QR)^*QH\mathbf{x}\rangle}{\|\mathbf{x}\|_{QH}^2} \\
&= \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\langle(((QR)^*QHP)^{-1}(QR)^*QH\mathbf{x}, ((QR)^*QHP)^{-1}(QR)^*QH\mathbf{x}\rangle}{\|\mathbf{x}\|_{QH}^2} \\
&\leq \|(QR)^*QHP)^{-1}\|^2 \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|(QR)^*QH\mathbf{x}\|^2}{\|\mathbf{x}\|_{QH}^2} \\
&\leq \|(QR)^*QHP)^{-1}\|^2 \|(QR)^*(QH)^{1/2}\|^2 \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|(QH)^{1/2}\mathbf{x}\|^2}{\|\mathbf{x}\|_{QH}^2} \\
&= \|(QR)^*QHP)^{-1}\|^2,
\end{aligned}$$

and $(QR)^*QHP = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix}$. Thus, $\|\Pi\|_{QH}^2$ is unbounded.

□

As we have shown, another condition must be satisfied to ensure stability besides the NSAP on P and QR with respect to QH. In the following section, we will explore a quality measure for R and P that ensures Π is stable.

5.2 Conditions for Stability

The crux of convergence theory for NS-AMG is in dealing with a non-orthogonal coarse-grid correction that can increase the error. In particular, if R^*HP ends up being a singular or near-singular matrix, then the resulting solver will likely diverge. In [6], this was handled by simply assuming that the oblique projection Π is stable, with the norm bounded by some small constant greater than one. Given that approximation properties alone are not sufficient for stability or two-grid convergence (Example 5.1.1), it is clear that some further conditions must be satisfied. We will introduce a quality measure on P and R, in conjunction with the approximation properties, that gives sufficient conditions for stability and thus, two-grid convergence. In particular, the basis matrices W and Z based on Π_1 and Π_2 are introduced, which correspond to the action of P and R, respectively. After proving several results on W and Z, a stability bound for $\|\Pi\|_{QH}$ is established.

Suppose we have scaled H such that $\sigma_n = 1$, and let $\{\mathbf{v}\}_{i=1}^n$ be the set of right singular vectors of H, normalized in the QH-norm, i.e. $\|\mathbf{v}_i\|_{QH} = 1$. For a right singular vector, $\mathbf{v}_i$, of H, the NSAP's on P and QR with respect to the QH-norm (Definition 5.1.1 and 5.1.4) simplify to:

$$\|(I - \Pi_1)\mathbf{v}_i\|_{QH}^2 \leq K_P \langle QH\mathbf{v}_i, QH\mathbf{v}_i\rangle \leq K_P\sigma_i\|\mathbf{v}_i\|_{QH}^2,$$
$$\|(I - \Pi_2)\mathbf{v}_i\|_{QH}^2 \leq K_R \langle QH\mathbf{v}_i, QH\mathbf{v}_i\rangle \leq K_R\sigma_i\|\mathbf{v}_i\|_{QH}^2,$$

for all i. Define $M := \max\{K_R, K_P\}$ and pick k such that $M\sigma_i < \theta \ll 1$ for all $i = 1, \ldots, k$, where θ will be chosen later. That is, we pick k such that the first k singular values are smaller than $\frac{\theta}{M} \ll \frac{1}{M}$. Denote $V_1 = [\mathbf{v}_1, ..., \mathbf{v}_k]$ as the an $n \times k$ matrix whose columns are the QH-normalized right singular vectors of H associated with the k smallest singular values.

Definition 5.2.1. *Consider two matrices $W, Z \in \mathbb{R}^{n\times n_c}$ in block form*

$$W = \Big[W_1, W_2\Big], \qquad Z = \Big[Z_1, Z_2\Big],$$

where W_1 and Z_1 make up the first k columns, and W_2 and Z_2 the remaining $(n_c - k)$ columns. Define W_1 and Z_1 as

$$W_1 = \Pi_1 V_1 \quad \textit{and} \quad Z_1 = \Pi_2 V_1.$$

Then, define W_2 as a QH-orthonormal basis for $\mathcal{R}(\Pi_1)\backslash\mathcal{R}(W_1)$ such that $W_1 \perp_{QH} W_2$ and $\mathcal{R}(W_1) \cup \mathcal{R}(W_2) = \mathcal{R}(\Pi_1)$, and likewise for Z_2 and Π_2.

From the definition above, we have the following useful properties:

$$\begin{aligned}
W_2^* QHW_2 &= Z_2^* QHZ_2 = I, \\
W_2^* QHW_1 &= Z_2^* QHZ_1 = \mathbf{0}, \\
\Pi_1 W &= W,
\end{aligned}$$

and

$$\Pi_2 Z = Z.$$

Lemmas 5.2.1 and 5.2.2 go on to show that W_2 and Z_2 are QH-orthogonal to V_1 and if the NSAP condition is satisfied for P and QR respectively, then the columns of W_1 and Z_1 are linearly independent with respect to QH.

Lemma 5.2.1. *The matrix V_1 is QH-orthogonal to W_2 and Z_2, that is, $V_1^* QHW_2 = V_1^* QHZ_2 = \mathbf{0}$.*

Proof. From the definition of W_1 and W_2, we have $W_2^* QHW_1 = \mathbf{0}$, which implies

$$\begin{aligned}
\mathbf{0} &= W_1^* QHW_2, \\
&= (\Pi_1 V_1)^* QHW_2, \\
&= (V_1 - (I - \Pi_1)V_1)^* QHW_2, \\
&= V_1^* QHW_2 - V_1^* (I - \Pi_1)^* QHW_2, \\
&= V_1^* QHW_2 - V_1^* QH(I - \Pi_1)W_2, \\
&= V_1^* QHW_2. && (\Pi_1 W_2 = W_2).
\end{aligned}$$

By a similar argument, we have $V_1^* QHZ_2 = \mathbf{0}$. □

Lemma 5.2.2. *If the NSAP on P with respect to the QH-norm holds, then the columns of*

W_1 form a linearly independent set in the QH-norm. Likewise, If the NSAP on QR with respect to the QH-norm holds, holds then the columns of Z_1 form a linearly independent set in the QH-norm.

Proof. Assume there exists $\mathbf{a} \in \mathbb{R}^n$ such that $\mathbf{a} \neq \mathbf{0}$ for which $\|W_1\mathbf{a}\|_{\mathbf{QH}} = \mathbf{0}$. Then,

$$\begin{aligned}
\|W_1\mathbf{a}\|^{\mathbf{2}}_{\mathbf{QH}} &= \|\Pi_1 V_1\mathbf{a}\|^{\mathbf{2}}_{\mathbf{QH}} \\
&= \|V_1\mathbf{a} - (I-\Pi_1)V_1\mathbf{a})\|^2_{QH} \\
&= \|V_1\mathbf{a}\|^2_{QH} - \|(I-\Pi_1)V_1\mathbf{a}\|^2_{QH} \\
&\geq \left\|\mathbf{a}\right\|^2_{QH} - M\sigma_k\left\|V_1\mathbf{a}\right\|^2_{QH} && \text{(Definition 5.1.1)} \\
&\geq (1-\theta)\left\|\mathbf{a}\right\|^2_{QH}
\end{aligned} \tag{5.19}$$

Since $(1-\theta) > 0$ and $\mathbf{a} \neq \mathbf{0}$, we have $\|W_1\mathbf{a}\|^{\mathbf{2}}_{\mathbf{QH}} > \mathbf{0}$. Thus, columns of W_1 form a linearly independent set. By similar proof, if Definition 5.1.4 holds then the columns of Z_1 are linearly independent. □

We will now bound $\|(W^*QHW)\|$, which is needed to bound $\|\Pi\|_{QH}$. Noting that

$$\begin{aligned}
W^*QHW &= \begin{bmatrix} W_1^* \\ W_2^* \end{bmatrix} QH \begin{bmatrix} W_1 & W_2^* \end{bmatrix} \\
&= \begin{bmatrix} W_1^*QHW_1 & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix}
\end{aligned}$$

we have

$$\|W^*QHW\| = \max\left\{\|W_1^*QHW_1\|, 1\right\}.$$

Similarly, one can show

$$\|Z^*QHZ\| = \max\Big\{\|Z_1^*QHZ_1\|, 1\Big\}.$$

Which, by Lemma 5.2.3, we have $\|W^*QHW\| \leq 1$ and $\|Z^*QHZ\| \leq 1$.

Lemma 5.2.3. *If Π_1 is an QH-orthogonal projection then $\|W_1^*QHW_1\| \leq 1$. Similarly, if Π_2 is an QH-orthogonal projection then $\|(Z_1^*QHZ_1)\| \leq 1$.*

Proof. From $\mathbf{w}_i = \Pi_1\mathbf{v}_i$ for $i = 1, \ldots, k$ we have,

$$\begin{aligned}
\|W_1^*QHW_1\| &= \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\langle W_1^*QHW_1\mathbf{x}, \mathbf{x}\rangle}{\|\mathbf{x}\|^2}, \\
&= \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\langle QH\Pi_1V_1\mathbf{x}, \Pi_1V_1\mathbf{x}\rangle}{\|\mathbf{x}\|^2}, \\
&= \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\langle QH(V_1 - (I-\Pi_1)V_1)\mathbf{x}, (V_1 - (I-\Pi_1)V_1)\mathbf{x}\rangle}{\|\mathbf{x}\|^2}, \\
&= \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|V_1\mathbf{x}\|_{QH}^2 - \|(I-\Pi_1)V_1\mathbf{x}\|_{QH}^2}{\|\mathbf{x}\|^2}, && (\perp\text{-projections}) \\
&= \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{x}\|^2 - \|(I-\Pi_1)V_1\mathbf{x}\|_{QH}^2}{\|\mathbf{x}\|^2}, \\
&\leq 1 && .
\end{aligned}$$

By a similar argument we have,

$$\|(Z_1^*QHZ_1)\| \leq 1.$$

□

As a consequence of Lemma 5.2.2, we have a bound on $\|(W_1^*QHW_1)^{-1}\|$ and $\|(Z_1^*QHZ_1)^{-1}\|$.

Lemma 5.2.4. *If Definition 5.1.1 holds then $\|(W_1^*QHW_1)^{-1}\| \leq \frac{1}{1-\theta}$. Similarly, if Definition 5.1.4 holds then $\|(Z_1^*QHZ_1)^{-1}\| \leq \frac{1}{1-\theta}$.*

Proof. If Definition 5.1.1 holds, from equation (**??**) we have

$$\|(W_1^* QHW_1)^{-1}\| \;\; \leq \;\; \frac{1}{(1-\theta)}. \tag{5.20}$$

By a similar argument, if Definition 5.1.4 holds then $\|(Z_1^* QHZ_1)^{-1}\| \leq \frac{1}{1-\theta}$. □

Note that Π is invariant of any change of basis for P and QR. To see this, let G_P and G_R be nonsingular $n_c \times n_c$ square matrices such that $P = WG_P$ and $QR = ZG_R$. We then have

$$\begin{aligned}
\Pi &= P(R^t HP)^{-1} R^t H && (5.21)\\
&= P((QR)^t QHP)^{-1} (QR)^t QH && (5.22)\\
&= WG_P((ZG_R)^t QHWG_P)^{-1} (ZG_R)^t QH && (5.23)\\
&= W(Z^t QHW)^{-1} Z^t QH. && (5.24)\\
& && (5.25)
\end{aligned}$$

If Π_1 and Π_2 are QH-orthogonal projections, we can bound the norm of $\|\Pi\|_{QH}$ by $\|(Z^t QHW)^{-1}\|$, as seen below.

Lemma 5.2.5. *If Π_1 and Π_2 are QH-orthogonal projections then*

$$\|\Pi\|_{QH}^2 \leq \|(Z^t QHW)^{-1}\|^2.$$

Proof. From Lemma 5.2.3, we have

$$\begin{aligned}
\|\Pi\|_{QH}^2 &= \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|W(Z^tQHW)^{-1}Z^tQH\mathbf{x}\|_{QH}^2}{\|\mathbf{x}\|_{QH}^2}, \\
&= \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\langle QHW(Z^tQHW)^{-1}Z^tQH\mathbf{x}, W(Z^tQHW)^{-1}Z^tQH\mathbf{x}\rangle}{\langle QH\mathbf{x},\mathbf{x}\rangle}, \\
&= \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\langle W^tQHW(Z^tQHW)^{-1}Z^tQH\mathbf{x}, (Z^tQHW)^{-1}Z^tQH\mathbf{x}\rangle}{\langle QH\mathbf{x},\mathbf{x}\rangle}, \\
&\leq \|W^tQHW\|\|(Z^tQHW)^{-1}\|^2 \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|Z^tQH\mathbf{x}\|^2}{\langle QH\mathbf{x},\mathbf{x}\rangle}, \\
&= \|W^tQHW\|\|(Z^tQHW)^{-1}\|^2 \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|Z^t(QH)^{1/2}(QH)^{1/2}\mathbf{x}\|^2}{\langle QH\mathbf{x},\mathbf{x}\rangle}, \\
&\leq \|W^tQHW\|\|(Z^tQHW)^{-1}\|^2\|Z^t(QH)^{1/2}\|^2 \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|(QH)^{1/2}\mathbf{x}\|^2}{\langle QH\mathbf{x},\mathbf{x}\rangle}, \\
&= \|W^tQHW\|\|(Z^tQHW)^{-1}\|^2\|Z^tQHZ\|, && (\|A\|^2 = \|AA^*\|) \\
&\leq \|(Z^tQHW)^{-1}\|^2.
\end{aligned}$$

□

Stability is dependent on how Z and W interact with each other. We will see that in addition to Definition 5.1.1 and 5.1.4, Z and W must perform similarly on the right singular vectors associated with the smallest singular values. We will study this phenomenon by bounding $\|(Z^*QHW)^{-1}\|$. In order to bound $\|(Z^*QHW)^{-1}\|$ we must bound $\|(Z_1^*QHW_1)^{-1}\|$, $\|Z_1^*QHW_2\|$, and $\|Z_2^*QHW_1\|$.

Lemma 5.2.6. *If Definition 5.1.1 and 5.1.4 hold and $\theta < \frac{1}{3}$ then*

$$\|(Z_1^*QHW_1)^{-1}\| \leq \frac{1}{1-3\theta}.$$

Proof. By expanding W_1 and Z_1 in terms of V_1 we have,

$$\begin{aligned}
\frac{1}{\|(Z_1^* QHW_1)^{-1}\|} &= \inf_{\mathbf{x}\neq\mathbf{0}} \sup_{\mathbf{y}\neq\mathbf{0}} \frac{\langle Z_1^* QHW_1\mathbf{x}, \mathbf{y}\rangle}{\|\mathbf{x}\|\|\mathbf{y}\|}, \\
&\geq \inf_{\mathbf{x}\neq\mathbf{0}} \frac{\langle QHW_1\mathbf{x}, Z_1\mathbf{x}\rangle}{\|\mathbf{x}\|^2}, \\
&= \inf_{\mathbf{x}\neq\mathbf{0}} \frac{\langle QH(V_1 - (I-\Pi_1)V_1)\mathbf{x}, (V_1 - (I-\Pi_2)V_1)\mathbf{x}\rangle}{\|\mathbf{x}\|^2}, \\
&\geq \inf_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{x}\|^2 - \|(I-\Pi_1)V_1\mathbf{x}\|_{QH}^2 - \|(I-\Pi_2)V_1\mathbf{x}\|_{QH}^2}{\|\mathbf{x}\|^2} \\
&\qquad - \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|(I-\Pi_1)V_1\mathbf{x}\|_{QH}\|(I-\Pi_2)V_1\mathbf{x}\|_{QH}}{\|\mathbf{x}\|^2}.
\end{aligned}$$

By Definition 5.1.1 and 5.1.4 we have,

$$\begin{aligned}
\frac{1}{\|(Z_1^* QHW_1)^{-1}\|} &\geq \inf_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{x}\|^2 - 2\theta\|\mathbf{x}\|^2 - (\theta^{1/2}\|\mathbf{x}\|)(\theta^{1/2}\|\mathbf{x}\|)}{\|\mathbf{x}\|^2}, \\
&= 1 - 3\theta.
\end{aligned}$$

Thus,

$$\|(Z_1^* QHW_1)^{-1}\| \leq \frac{1}{1-3\theta}.$$

□

Lemma 5.2.7. *If Definition 5.1.4 holds, we have*

$$\|Z_1^* QHW_2\| \leq \theta^{1/2}$$

and if Definition 5.1.1 holds, we have

$$\|Z_2^* QHW_1\| \leq \theta^{1/2}.$$

Proof. From the definition of W_2 we have,

$$\begin{aligned} Z_1^* QHW_2 &= (V_1 - (I - \Pi_2)V_1)^* QHW_2, \\ &= V_1^* QHW_2 - V_1^* QH(I - \Pi_2)W_2, \\ &= -V_1^* QH(I - \Pi_2)W_2. && (V_1^* QHW_2 = \mathbf{0}) \end{aligned}$$

Thus, if Definition 5.1.4 holds, we have

$$\begin{aligned} \|Z_1^* QHW_2\| &= \|V_1^* QH(I - \Pi_2)W_2\|, && (5.26) \\ &= \sup_{\mathbf{x} \neq \mathbf{0}} \sup_{\mathbf{y} \neq \mathbf{0}} \frac{\langle V_1^* QH(I - \Pi_2)W_2\mathbf{x}, \mathbf{y}\rangle}{\|\mathbf{x}\|\|\mathbf{y}\|}, && (5.27) \\ &= \sup_{\mathbf{x} \neq \mathbf{0}} \sup_{\mathbf{y} \neq \mathbf{0}} \frac{\langle QHW_2\mathbf{x}, (I - \Pi_2)V_1\mathbf{y}\rangle}{\|\mathbf{x}\|\|\mathbf{y}\|}, && (5.28) \\ &\leq \sup_{\mathbf{x} \neq \mathbf{0}} \sup_{\mathbf{y} \neq \mathbf{0}} \frac{\|W_2\mathbf{x}\|_{QH} \|(I - \Pi_2)V_1\mathbf{y}\|_{QH}}{\|\mathbf{x}\|\|\mathbf{y}\|}, && (5.29) \\ &\leq \sup_{\mathbf{x} \neq \mathbf{0}} \sup_{\mathbf{y} \neq \mathbf{0}} \frac{\|\mathbf{x}\|(\theta^{1/2}\|\mathbf{y}\|)}{\|\mathbf{x}\|\|\mathbf{y}\|}, && \text{(Definition 5.1.4)} \quad (5.30) \\ &= \theta^{1/2}. && (5.31) \end{aligned}$$

From the definition of Z_2, we have

$$\begin{aligned} Z_2^* QHW_1 &= Z_2^* QH(V_1 - (I - \Pi_1)V_1), \\ &= Z_2^* QHV_1 - Z_2^* QH(I - \Pi_1)V_1, \\ &= -Z_2^* QH(I - \Pi_1)V_1. && (Z_2^* QHV_1 = \mathbf{0}) \end{aligned}$$

By similar argument, if Definition 5.1.1 holds, we have

$$\|Z_2^* QHW_1\| \quad = \quad \theta^{1/2}. \tag{5.32}$$

□

We can then bound $\|\Pi\|_{QH}^2$ by bounding $\|(Z^* QHW)^{-1}\|$, shown in Theorem 6. To bound $\|(Z^* QHW)^{-1}\|$, first reformulate in block form,

$$\begin{aligned} Z^t QHW &= \begin{bmatrix} Z_1^* \\ Z_2^* \end{bmatrix} QH \begin{bmatrix} W_1 & W_2 \end{bmatrix}, \\ &= \begin{bmatrix} Z_1^* QHW_1 & Z_1^* QHW_2 \\ Z_2^* QHW_1 & Z_2^* QHW_2 \end{bmatrix}, \\ &= \begin{bmatrix} Z_1^* QHW_1 & 0 \\ 0 & Z_2^* QHW_2 \end{bmatrix} - \begin{bmatrix} 0 & -Z_1^* QHW_2 \\ -Z_2^* QHW_1 & 0 \end{bmatrix}. \end{aligned}$$

Theorem 6. *Assume Definition 5.1.1 and 5.1.4 hold. Assume k is chosen such that $1 - 3\theta - \theta^{1/2} > 0$ and $\gamma - \theta^{1/2} > 0$, where $\gamma = \sigma_1(Z_2^* QHW_2)$, then we have the following:*

$$\|(Z^t QHW)^{-1}\| \leq \frac{1}{\min\{1 - 3\theta - \theta^{1/2}, \gamma - \theta^{1/2}\}}$$

and

$$\|\Pi\|_{QH} \leq \frac{1}{\min\{1 - 3\theta - \theta^{1/2}, \gamma - \theta^{1/2}\}}.$$

Proof. From Lemma 5.2.7 we have,

$$
\begin{aligned}
\frac{1}{\|(Z^*QHW)^{-1}\|} &= \inf_{\mathbf{x}\neq\mathbf{0}} \frac{\|Z^*QHW\mathbf{x}\|}{\|\mathbf{x}\|}, \\
&\geq \inf_{\mathbf{x}\neq\mathbf{0}} \frac{\left\| \begin{bmatrix} Z_1^*QHW_1 & 0 \\ 0 & Z_2^*QHW_2 \end{bmatrix} \mathbf{x} \right\| - \left\| \begin{bmatrix} 0 & -Z_1^*QHW_2 \\ -Z_2^*QHW_1 & 0 \end{bmatrix} \mathbf{x} \right\|}{\|\mathbf{x}\|}, \\
&\geq \inf_{\mathbf{x}\neq\mathbf{0}} \frac{\left\| \begin{bmatrix} V_1^*QH\Pi_2\Pi_1V_1 & 0 \\ 0 & Z_2^*QHW_2 \end{bmatrix} \mathbf{x} \right\|}{\|\mathbf{x}\|}, \\
&\quad - \sup_{\mathbf{x}\neq 0} \frac{\left\| \begin{bmatrix} 0 & V_1^*QH(I-\Pi_2)W_2 \\ Z_2^*QH(I-\Pi_1)V_1 & 0 \end{bmatrix} \mathbf{x} \right\|}{\|\mathbf{x}\|}, \\
&= \min\{\sigma_1(Z_1^*QHW_1), \sigma_1(Z_2^*QHW_2)\} - \max\{\|Z_1^*QHW_2\|, \|Z_2^*QHW_1\|\}. \\
&\geq \min\{1-3\theta, \gamma\} - \theta^{1/2}, \\
&= \min\{1-3\theta-\theta^{1/2}, \gamma-\theta^{1/2}\}.
\end{aligned}
$$

Then,

$$\|\Pi\|_{QH} \leq \frac{1}{\min\{1-3\theta-\theta^{1/2}, \gamma-\theta^{1/2}\}}$$

follows from Lemma 5.2.5. □

We can choose θ to be smaller by choosing k to be smaller. But, γ also gets smaller as k gets smaller. Our assumption is that there exists some k for which both terms $1-3\theta-\theta^{1/2}$ and $\gamma-\theta^{1/2}$ are positive. Thus, if R and P satisfy their respective approximation property, we have found a quality measure that relates the action of P and R that ensures stability of the two-grid error propagation operator of the coarse-grid correction, and thus, two-grid convergence.

5.2.1 Conclusions

In this chapter, the standard AMG approximation properties are generalized to the nonsymmetric setting by introducing approximation properties on both the restriction and prolongation operators. Known approximation property relations for the SPD case, for example, the strong approximation property implies the weak approximation property, are then proved in the nonsymmetric setting. Counter-examples are presented showing that a number of natural assumptions on a NS-AMG solver are *not* sufficient for two-grid convergence. An investigation of stability is then discussed and a quality measure of the relation between the restriction and prolongation operators is introduced that ensures stability. As seen in [6], two-grid convergence is proved when stability of the error propagation operator is coupled with the strong approximation property on P with respect to the QH-norm.

Chapter 6

Conclusion and Future Work

This thesis encompasses two main extensions of AMG: extending UU graph Laplacian solvers to SU graph Laplacian systems using Gremban's expansion and developing an AMG solver for UD graph Laplacian systems. Each focus was to develop and extend linear solvers for graph Laplacian systems for a variety of modeling decisions. This ensures that information and attributes of a graph that data scientists wish to represent can stay intact and not be diluted due to the restrictiveness of current solvers. We conclude by summarizing what has been accomplished in this thesis and then discuss possible future work.

Chapter 3 generalized Gremban's expansion to be applicable to any diagonally dominant matrix, not just symmetric matrices as was in first introduced in [11]. We have provided various theorems for Gremban's expansion including a generalization of the classical UU graph walk theorem to SU graphs, the relationship of the spectrum and singular values of the Gremban expansion matrix of $S := M + P$ to the spectrum and singular values of S and $M - P$, and given an approximate solution to the Gremban expansion system with $\mathbf{z} = [\mathbf{b}^t, -\mathbf{b}^t]^t$, we can tightly bound the norm of the residual of the original system by the norm of the residual of the expansion. Without prior knowledge of the sign structure, LAMG, in conjunction with Gremban's expansion, provided a robust solver and it performed well on the constructed signed graphs as well as the movielens user-movie rating graphs. By applying Gremban's expansion we have expanded any UU graph Laplacian linear system solver's solution space to include SU graphs. Gremban's expansion provides a nice meaningful relationship between signed graphs and unsigned graphs.

Chapter 4 develops a highly successful AMG algorithm, NS-LAMG, for UD graph Laplacian systems. Low-degree elimination in [23] was generalized for UD graphs and was shown to enhance the performance of NS-LAMG as well as substantially reduces the problem size on the finest level, especially for scale-free graphs. A highly parallelizable semi-adaptive AMG solver for Markov chain stationary distribution systems was developed that uses low-degree elimination and stationary coarse-graph aggregation. In the setup phase of NS-LAMG, the semi-adaptive AMG solver for Markov chain stationary distribution systems was shown to be a robust solver for finding an adequate approximation to the right singular vector of a scale-free UD graph Laplacian. Numerical tests showed that NS-LAMG resulted in lower effective convergence factors and less work than GMRES(20), a traditional well-known iterative solver for nonsymmetric systems.

Chapter 5 fills in gaps in nonsymmetric AMG convergence theory, building off of the theory presented in [6]. The standard AMG approximation properties were generalized to the nonsymmetric case for both the prolongation and restriction operators. The well known relations between the approximation properties for the symmetric case were also shown to be true in the nonsymmetric case. It was shown that the approximation properties on the restriction and prolongation operators do not imply stability of the error propagation operator of the coarse-grid correction. In order to address this issue, we introduce a new quality measure on the action of the prolongation and restriction operators, thus ensuring stability of the coarse-grid correction.

6.0.1 Future Work

For SD graph Laplacian systems, a combination of Gremban's expansion in conjunction with NS-LAMG could be explored. The theory for using Gremban's expansion for nonsymmetric problems was developed in Chapter 3, but a robust solver is still needed as the condition number of the expanded system is not bounded by the condition number of the original matrix. NS-LAMG has been shown to be a robust, scalable solver for UD graph Laplacian systems and thus, is an adequate solver that can be used in conjunction with Gremban's expansion to solve SD graph Laplacian systems. Gremban's expansion matrix could provide insight into the structure of the underlying graph. Further research could be done to investigate the uses of Gremban's expansion matrix in applications involving signed graphs. Work should also include further research into NS-LAMG to enhance the scalability and robustness of the solver. Lastly, the semi-adaptive AMG solver for Markov chain stationary distribution systems could provide a parallel solver that has not be previously studied.

Solvers for large, real-world graph related systems are becoming highly important as the size and availability of data continues to grow. These two extensions of AMG provided in this thesis are useful in the graph community since it provides robust and scalable solvers for graph Laplacians linear systems that represent different relationships than just UU graphs. By providing adequate solvers for SU, UD, and SD graph Laplacian systems, we have allowed more modeling choices for graph representations, resulting in an increased opportunity in the graph community for UU graph applications to be adapted for more diverse modeling choices than the classical positively weighted, symmetric relationships between vertices.

Bibliography

[1] Michele Benzi, Ernesto Estrada, and Christine Klymko. Ranking hubs and authorities using matrix functions. Linear Algebra and its Applications, 438(5):2447 – 2474, 2013.

[2] Michele Benzi and Bora Uar. Product preconditioning for markov chain problems. In in Proceedings of the 2006 Markov Anniversary Meeting, pages 239–256.

[3] Jeff Bezanson, Stefan Karpinski, Viral B. Shah, and Alan Edelman. Julia: A fast dynamic language for technical computing. CoRR, abs/1209.5145, 2012.

[4] Matthias Bolten, Achi Brandt, James Brannick, Andreas Frommer, Karsten Kahl, and Ira Livshits. A bootstrap algebraic multilevel method for markov chains. SIAM Journal on Scientific Computing, 33(6):3425–3446, 2011.

[5] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge. Adaptive smoothed aggregation (alphasa) multigrid. SIAM Review, 47(2):317–346, 2005.

[6] M. Brezina, T. Manteuffel, S. MCormick, J. Ruge, and G. Sanders. Towards adaptive smoothed aggregation (alphasa) for nonsymmetric problems. SIAM Journal on Scientific Computing, 32(1):14–39, 2010.

[7] Paul Christiano, Jonathan A. Kelner, Aleksander Madry, Daniel A. Spielman, and Shang-Hua Teng. Electrical flows, laplacian systems, and faster approximation of maximum flow in undirected graphs. In Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing, STOC '11, pages 273–282, New York, NY, USA, 2011. ACM.

[8] T. Feng and H. Liu. The computer realization of the qr decomposition on matrices with full column rank. In 2009 International Conference on Computational Intelligence and Security, volume 2, pages 76–79, Dec 2009.

[9] Miroslav Fiedler. Algebraic connectivity of graphs. Czechoslovak mathematical journal, 23(2):298–305, 1973.

[10] Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Trans. on Knowl. and Data Eng., 19(3):355–369, March 2007.

[11] Keith Gremban. Combinatorial Preconditioners for Sparse, Symmetric, Diagonally Dominant Linear Systems. PhD thesis, Carnegie Mellon University, Pittsburgh, October 1996. CMU CS Tech Report CMU-CS-96-123.

[12] Graham Horton. On the multilevel solution algorithm for markov chains. Technical report, 1997.

[13] Andrew V Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. SIAM journal on scientific computing, 23(2):517–541, 2001.

[14] Tristan Konolige and Jed Brown. A parallel solver for graph laplacians, 2017.

[15] Ioannis Koutis, Gary L. Miller, and David Tolliver. Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing. Comput. Vis. Image Underst., 115(12):1638–1646, December 2011.

[16] U. R. Krieger, B. Muller-Clostermann, and M. Sczittnick. Modeling and analysis of communication systems based on computational methods for markov chains. IEEE Journal on Selected Areas in Communications, 8(9):1630–1648, Dec 1990.

[17] Udo R. Krieger. Numerical Solution of Large Finite Markov Chains by Algebraic Multigrid Techniques, pages 403–424. Springer US, Boston, MA, 1995.

[18] Udo R. Krieger. On a two-level multigrid solution method for finite markov chains. Linear Algebra and its Applications, 223:415 – 438, 1995. Honoring Miroslav Fiedler and Vlastimil Ptak.

[19] Jrme Kunegis, Stephan Schmidt, Andreas Lommatzsch, Jrgen Lerner, Ernesto W. De, and Luca Sahin Albayrak. Spectral analysis of signed graphs for clustering, prediction and visualization.

[20] Jure Leskovec and Rok Sosič. SNAP: A general purpose network analysis and graph mining library in C++. `http://snap.stanford.edu/snap`, June 2014.

[21] Xiaoye S Li. Sparse Gaussian elimination on high performance computers. PhD thesis, Citeseer, 1996.

[22] O. E. Livne. Coarsening by compatible relaxation. Numerical Linear Algebra with Applications, 11(2-3):205–227, 2004.

[23] O.E. Livne and A. Brandt. Lean algebraic multigrid (lamg): Fast graph laplacian linear solver. SIAM Journal of Scientific Computing, 2012.

[24] Ulrike Luxburg. A tutorial on spectral clustering. Statistics and Computing, 17(4):395–416, December 2007.

[25] Jan Mandel and Bohuslav Sekerka. A local convergence proof for the iterative aggregation method. Linear Algebra and its Applications, 51:163 – 172, 1983.

[26] C. Mense and R. Nabben. On algebraic multi-level methods for non-symmetric systems comparison results. Linear Algebra and its Applications, 429(10):2567 – 2588, 2008. Special Issue in honor of Richard S. Varga.

[27] Artem Napov and Yvan Notay. An efficient multigrid method for graph laplacian systems. Electronic Transactions on Numerical Analysis, 45:201–218, 2016.

[28] Artem Napov and Yvan Notay. An efficient multigrid method for graph laplacian systems ii: robust aggregation. SIAM journal on scientific computing, 2016.

[29] Y van Notay. A robust algebraic multilevel preconditioner for non-symmetric m-matrices. Numerical Linear Algebra with Applications, 7(5):243–267, 2000.

[30] Yvan Notay. Algebraic analysis of two-grid methods: The nonsymmetric case. Numerical Linear Algebra with Applications, 17(1):73–96, 2010.

[31] Yvan Notay and Panayot S. Vassilevski. Recursive krylov-based multigrid cycles. Numerical Linear Algebra with Applications, 15(5):473–487, 2008.

[32] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[33] Christopher C. Paige and Michael A. Saunders. Lsqr: An algorithm for sparse linear equations and sparse least squares. ACM Trans. Math. Softw., 8(1):43–71, March 1982.

[34] Youcef Saad and Martin H Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput., 7(3):856–869, July 1986.

[35] Marzio Sala and Raymond S. Tuminaro. A new petrovgalerkin smoothed aggregation preconditioner for nonsymmetric linear systems. SIAM Journal on Scientific Computing, 31(1):143–166, 2008.

[36] Geoff Sanders, Tom Manteuffel, Alyson Fox, and Nate Monning. An investigation into lamg, 2015.

[37] Geoffrey D. Sanders. Extensions to adaptive smooth aggregation (alphasa) multigrid: Eigensolver initialization and nonsymmetric problems, 2008. Copyright - Copyright ProQuest, UMI Dissertations Publishing 2008; Last updated - 2015-08-24; First page - n/a.

[38] B. Seibold. Performance of algebraic multigrid methods for non-symmetric matrices arising in particle methods. Numerical Linear Algebra with Applications, 17(2-3):433–451, 2010.

[39] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell., 22(8):888–905, August 2000.

[40] H. De Sterck, T. A. Manteuffel, S. F. McCormick, K. Miller, J. Pearson, J. Ruge, and G. Sanders. Smoothed aggregation multigrid for markov chains. SIAM Journal on Scientific Computing, 32(1):40–61, 2010.

[41] H. De Sterck, Thomas A. Manteuffel, Stephen F. McCormick, Quoc Nguyen, and John Ruge. Multilevel adaptive aggregation for markov chains, with application to web ranking. SIAM Journal on Scientific Computing, 30(5):2235–2262, 2008.

[42] Hans de Sterck, Van Emden Henson, Geoffrey Sanders, et al. Multilevel aggregation methods for small-world graphs with application to random-walk ranking. Computing and Informatics, 30(2):225–246, 2012.

[43] Hans De Sterck, Killian Miller, Eran Treister, and Irad Yavneh. Fast multilevel methods for markov chains. Numerical Linear Algebra with Applications, 18(6):961–980, 2011.

[44] William J. Stewart. Introduction to the numerical solution of Markov chains. Princeton Univ. Press, Princeton, NJ, 1994.

[45] Hanghang Tong and Christos Faloutsos. Center-piece subgraphs: Problem definition and fast solutions. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '06, pages 404–413, New York, NY, USA, 2006. ACM.

[46] Eran Treister and Irad Yavneh. Square and stretch multigrid for stochastic matrix eigenproblems. Numerical Linear Algebra with Applications, 17(2-3):229–251, 2010.

[47] Eran Treister and Irad Yavneh. On-the-fly adaptive smoothed aggregation multigrid for markov chains. SIAM Journal on Scientific Computing, 33(5):2927–2949, 2011.

[48] Scott White and Padhraic Smyth. A Spectral Clustering Approach To Finding Communities in Graphs, pages 274–285.

[49] T. A. Wiesner, R. S. Tuminaro, W. A. Wall, and M. W. Gee. Multigrid transfers for nonsymmetric systems based on schur complements and galerkin projections. Numerical Linear Algebra with Applications, 21(3):415–438, 2014.

[50] Leting Wu, Xiaowei Ying, Xintao Wu, Aidong Lu, and Zhi-Hua Zhou. Spectral analysis of k-balanced signed graphs. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 1–12. Springer, 2011.

[51] Irad Yavneh and Marion Weinzierl. Nonsymmetric black box multigrid with coarsening by three. Numerical Linear Algebra with Applications, 19(2):194–209, 2012.