Journal of Applied Mathematics

Preconditioning Techniques for Sparse Linear Systems

Guest Editors: Massimiliano Ferronato, Edmond Chow, and Kok-Kwang Phoon

Preconditioning Techniques for Sparse Linear Systems

Journal of Applied Mathematics

Preconditioning Techniques for Sparse Linear Systems

Guest Editors: Massimiliano Ferronato, Edmond Chow, and Kok-Kwang Phoon

Editorial Board

Contents

Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 518165, 3 pages
doi:10.1155/2012/518165

Editorial

Preconditioning Techniques for Sparse Linear Systems

Massimiliano Ferronato,[1] Edmond Chow,[2] and Kok-Kwang Phoon[3]

[1] *Department of Civil, Environmental and Architectural Engineering, University of Padova, 35131 Padova, Italy*
[2] *School of Computational Science and Engineering, College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA*
[3] *Department of Civil Engineering, National University of Singapore, Singapore 117576*

Correspondence should be addressed to Massimiliano Ferronato, ferronat@dmsa.unipd.it

Received 8 June 2012; Accepted 8 June 2012

The implementation of and the solution to large computational models is becoming quite a common effort in both engineering and applied sciences. The accurate and cost-effective numerical solution to the sequence of linearized algebraic systems of equations arising from these models often represents the main memory-demanding and time-consuming task. Direct methods for sparse linear systems often represent the de facto solver in several commercial codes on the basis of their robustness and reliability. However, these methods scale poorly with the matrix size, especially on three-dimensional problems. For such large systems, iterative methods based on Krylov subspaces can be a much more attractive option. A significant number of general-purpose Krylov subspace, or conjugate gradient-like, solvers have been developed during the 70s through the 90s. Interest in these solvers is growing in many areas of engineering and scientific computing. Nonetheless, to become really competitive with direct solvers, iterative methods typically need an appropriate preconditioning to achieve convergence in a reasonable number of iterations and time.

The term "preconditioning" refers to "*the art of transforming a problem that appears intractable into another whose solution can be approximated rapidly*" [L.N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997], while the "preconditioner" is the operator that is responsible for such a transformation. It is widely recognized that preconditioning is the key factor to increasing the robustness and the computational efficiency of iterative methods. Generally speaking, there are three basic requirements for a good preconditioner: (i) the preconditioned matrix should have a clustered eigenspectrum away from 0; (ii) the preconditioner should be as cheap to compute as possible;

(iii) its application to a vector should be cost-effective. It goes without saying that these are conflicting requirements, and an appropriate trade-off must be found for any specific problem at hand. Unfortunately, theoretical results are few, and frequently somewhat "empirical" algorithms may work surprisingly well despite the lack of a rigorous foundation. This is why finding a good preconditioner for solving a sparse linear system can be viewed rather as "*a combination of art and science*" [Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003] than a rigorous mathematical exercise.

Research on the construction of effective preconditioners has significantly grown over the last two decades. Currently, preconditioning appears to be a much more active and promising research field than either direct or iterative solution methods, particularly within the context of the fast evolution of the hardware technology. On one hand, this is due to the understanding that there are virtually no limits to the available options for obtaining a good preconditioner. On the other hand, it is also generally recognized that an optimal general-purpose preconditioner is unlikely to exist. There are undoubtedly fertile grounds for research in improving the solution efficiency of a specific problem within a specific computing environment. From this viewpoint, the knowledge of the governing physical processes, the structure of the resulting discrete system, and the prevailing computer technology are essential aspects that cannot be ignored when addressing the development of an appropriate preconditioning technique.

The present special issue of *Journal of Applied Mathematics* is intended to provide some insight on efficient preconditioning techniques in computational engineering and sciences. This special issue should not be viewed as a comprehensive survey or an exhaustive compilation of all the current trends in preconditioning research. However, the papers included in this volume cover different algorithms and applications, thus offering an overview of the recent advances achieved in this field and suggesting potential future research directions.

The special issue contains 11 articles addressing the numerical solution to sparse linear systems and eigenproblems arising from different applications, including the Helmholtz equation in electromagnetics and seismology, steady and unsteady Navier-Stokes equations for incompressible flow, soil consolidation with nonassociated plasticity, optimal control problems governed by coupled elliptic-type equations, graph approaches for electric power networks, industrial processes and traffic models, and the equilibrium of multibody discrete systems. The different linear systems and eigenproblems arising from these applications are addressed using (a) algebraic preconditioners, that is, general-purpose algorithms requiring the knowledge of the coefficient matrix only, and (b) problem-specific preconditioners, that is, specialized methods based on the peculiar structure of the mathematical problem at hand. In group (a), novel formulations of the algebraic multigrid (AMG) method are proposed and tested, using aggregation and parallel wavelet techniques as smoothers in a parallel environment, modified symmetric successive overrelaxation (MSSOR) iterations are introduced in both electromagnetics and soil consolidation, and factorized sparse approximate inverse- (FSAI-) based algorithms are developed for large symmetric positive definite parallel eigenanalyses. In group (b), an important role is played by saddle-point matrices, which can be encountered in several different applications. Such problems are typically addressed by specific algorithms, which can also make use of algebraic preconditioners as a kernel. Three papers are devoted to the development and testing of methods for saddle-point matrices, using the Hermitian/Skew-Hermitian (HSS) approach, the semi-implicit method for pressure linked equations (SIMPLE), and a relaxed splitting preconditioner. Finally, problem-specific approaches based on the mathematical structure of

the native application are considered for the solution to optimal control problems discretized with spectral elements, the equilibrium of multibody discrete systems, and the observability of physical processes through a graph approach.

Acknowlegments

We would like to thank the authors for their valuable contributions and the anonymous reviewers for their thorough and insightful comments that brought this special Issue to fruition.

Massimiliano Ferronato
Edmond Chow
Kok-Kwang Phoon

Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 305415, 26 pages
doi:10.1155/2012/305415

Research Article

A Graph Approach to Observability in Physical Sparse Linear Systems

Santiago Vazquez-Rodriguez, Jesús Á. Gomollón, Richard J. Duro, and Fernando López Peña

Grupo Integrado de Ingeniería, Universidade da Coruña, Mendizábal S/N, 15403 Ferrol, Spain

Correspondence should be addressed to Santiago Vazquez-Rodriguez, svr@udc.es

Received 28 November 2011; Revised 2 March 2012; Accepted 16 March 2012

Academic Editor: Massimiliano Ferronato

A sparse linear system constitutes a valid model for a broad range of physical systems, such as electric power networks, industrial processes, control systems or traffic models. The physical magnitudes in those systems may be directly measured by means of sensor networks that, in conjunction with data obtained from contextual and boundary constraints, allow the estimation of the state of the systems. The term observability refers to the capability of estimating the state variables of a system based on the available information. In the case of linear systems, diffierent graphical approaches were developed to address this issue. In this paper a new unified graph based technique is proposed in order to determine the observability of a sparse linear physical system or, at least, a system that can be linearized after a first order derivative, using a given sensor set. A network associated to a linear equation system is introduced, which allows addressing and solving three related problems: the characterization of those cases for which algebraic and topological observability analysis return contradictory results; the characterization of a necessary and sufficient condition for topological observability; the determination of the maximum observable subsystem in case of unobservability. Two examples illustrate the developed techniques.

1. Introduction

The state variables that characterize a physical system are estimated by means of the data available at any given time. This data can be generated from a sensor network spread out over an area or from contextual and boundary constraints. In general, the known system variables are said to be sensed or measured variables whether they are sensed with a real device or their magnitudes are obtained in a sort of virtual sensors. The remaining variables are considered nonsensed or unmeasured variables. In such a context, the observability issue

arises when we would like to know if the sensing system is enough to be able to determine the state of the system, that is, the system state variables.

This paper deals with a scenario where a well-known model describes the behavior of a physical system in terms of relationships between system variables and parameters. The system must be linear or linearized after a first-order derivative. In this context, a given sensing network is considered, and the system observability analysis is addressed.

The term observability was introduced in the realm of linear dynamical control systems [1]. It stems from the capability of estimating the state of a system based on the information available. Although observability is essentially a numerical and algebraic problem, some techniques based on topology and graph theory have been developed to provide solutions in this area.

Due to the fact that observability and the problems related to it were studied in different engineering disciplines, the technical terminology is not totally uniform. As a result, some terms are more widely used in some areas and not in others and, in a few cases, different terms describe the same thing in different fields.

Five examples are described below pursuing the following aims: on one hand, illustrating how observability and other related problems constitute research topics in different physical, engineering, and industrial areas, where a sensor network is designed in order to analyze a given system; on the other, showing the multiple points of view from which these issues can be addressed and, in particular, how topological and graph-based approaches were developed in some cases.

The term sensor network comprises a broad spectrum of engineering and physical systems and, in particular, the topic of wireless sensor networks has led to issues that, in one way or another, are related to observability. This is the case of coverage, optimal node placement, and the minimum number of nodes required to achieve connectivity. In [2], it is shown that a graph model can be used to describe those systems, and some graph approaches have been developed in order to provide an answer to the challenges posed.

Whithin the sphere of linear control systems, the *controllability* problem was addressed from a graph-theoretic approach. A graph associated to a system was defined in [3] and conclusions related to several system properties are derived from the analysis of such a graph. A survey of the techniques proposed in the literature for structured linear systems can be found in [4]. More recently, a graph approach to observability analysis is proposed in [5, 6].

High-voltage electric power networks constitute another field, where observability has been an important issue in system analysis for decades [7, 8]. It is worth mentioning that the approach to the problem in [9] where the authors characterize what they call topological observability through the existence of certain graphs that, defined in the electrical network, obey constraints derived from the sensing network. However, these graph techniques do not allow the inclusion of measurements that are currently being considered, such as current and phasor measures.

Observability has also been a motivation for research in traffic models in topics related to the origin/destination trip matrix estimation challenge. This is the case of [10], where the authors adapt topological techniques developed for electric power networks to this new context. Although this issue is more complex than the description made by the authors in their paper, it has been taken as an example to illustrate the techniques proposed in the present work as will be shown in a later section.

Material and energy balances that must take place in industrial processes are analyzed in [11]. There, its authors distinguish up to four categories of balancing equations, depending

on whether they consider or not materials, chemical reactions, energy, and entropy. They study the solvability of the resulting equations, for which a set of sensed variables is taken into account. The observability and redundancy of measurements as well as the errors in the measured values are included in the dissertation. Statistical techniques are used to estimate the state of the system by reconciliation. In the case of linear systems, a parallelism is established between system and sensing observability conditions and the existence of certain graphs defined from the process balancing flowsheet.

The common topic of the aforementioned scenarios, with regards to graph theory, is that certain graph techniques were developed in all the cases because of the existence of graphs or networks that characterized the systems with a given sensor set. Furthermore, the equations that describe the networks are linear or linearized. In this paper, a new graph technique is presented in order to characterize the observability of any linear physical system. The implementation of such a technique imposes constraints on the problem, summarized by the fact that the systems must be sparse and of large dimension. For any sparse and large dimensional physical system, an associated network will be defined based, exclusively, on structural considerations, that is, the topology of the equation system in its matrix form that relates the sensed variables with the state variables. It will be demonstrated that the system can be said to be topologically observable if there exists a certain graph within the associated network.

Krumpholz et al. developed in [9] a topological approach for the observability issue in the scope of electric power systems. Nevertheless, the problem related to the characterization of those cases for which algebraic and structural techniques return contradictory results is not studied. In this paper, the latter problem is solved, which has allowed carrying out a more general demonstration of the necessary and sufficient condition for topological observability than the one proposed by Krumpholz. Numerous techniques have been developed and widely and successfully tested for decades [12–15] in the scope of topological observability analysis in electric power systems. In this paper, a new graph approach is presented, which allows addressing the observability of any linear physical system or, at least, a system linearized after a first-order derivative, and not exclusively electric power systems. Boukhobza et al. had already developed a graph-theoretic technique in order to determine the state and input observability in structured linear systems [5]. Unlike that proposal, the approach presented in this paper makes it possible to exploit techniques like those mentioned above [12–15] to characterize concepts like parametric unobservability and to easily determine the maximum observable subsystem.

The rest of the paper is organized as follows. Starting from a mathematical model, some terms will be introduced concerning observability and sparse physical systems in the next section. Section 3 is devoted to the bases of graph theory and the concepts used throughout the paper. Once the theoretical assumptions have been described, an analogy between linear equation systems and graph theory is established by means of a network associated to the physical system and a given sensor set. Section 5 introduces the concept of topological observability, which is characterized through the existence of a constrained graph in the associated network. The following section is devoted to the cases where the system is not observable and how the search for the maximum observable subsystem is addressed by means of the same graph techniques. Section 7 includes two examples in order to illustrate the techniques proposed in this paper, and how they can be implemented in absolutely different real engineering scenarios. Finally, some conclusions are presented in Section 8.

2. Mathematical Model

In order to determine the state of a system, S, consider a set of m variables ω that are sensed. These variables can be expressed in terms of the n system state variables, φ:

$$\omega = \mathbf{h}(\varphi) + \varepsilon, \tag{2.1}$$

where ε represents a vector of errors due to the measurement acquisition process. In what follows, this error vector will be ignored because of its irrelevance regarding observability issues. Two different cases might be considered at this point, depending on the linearity of the above equations. On one hand, assume those equations are linear. Then, S is a linear system, and a matrix formulation can be proposed instead of (2.1):

$$\omega = \mathbf{H}\varphi, \tag{2.2}$$

where $\mathbf{H}$ is a $m \times n$ characterization matrix of the system. On the other hand, consider that S is a nonlinear system that can be linearized around a certain state φ_0 and let $\mathbf{J}(\varphi_0)$ be the $m \times n$ jacobian matrix, thus:

$$\Delta\omega = \mathbf{J}(\varphi_0)\Delta\varphi, \tag{2.3}$$

where $\Delta\omega = \omega - \mathbf{h}(\varphi_0)$ and $\Delta\varphi = \varphi - \varphi_0$. Summarizing, both cases resemble an equation system of the form:

$$\underset{m\times 1}{\mathbf{z}} = \underset{m\times n}{\mathbf{M}} \cdot \underset{n\times 1}{\mathbf{x}}, \tag{2.4}$$

where $\mathbf{z}$ is a constant term vector that results from the m magnitudes sensed throughout the system, $\mathbf{x}$ is the unknown vector that is directly related to the n state variables, and $\mathbf{M}$ is a coefficient matrix. In what follows, and in order to simplify the explanation, we will refer to $\mathbf{z}$ and $\mathbf{x}$ as the measurement and state variable column vectors, respectively. Also, z_k will denote a generic measured variable, and x_i will be a generic state variable. The observability issue arises when we would like to know if the m variables considered in the sensor set are enough to determine the state of the system. It depends not only on how large the number of measurements is but also on their nature, and how they are spread out over the system. From an algebraic point of view, a system S is said to be observable if the system given by (2.4) is solvable, that is, the equation system is consistent, and there exist at least n linear independent equations. As Krumpholz et al. define in [9], the system is said to be algebraically observable if and only if the rank of $\mathbf{M}$ is equal to n. A well-known problem comes up when the system is ill-conditioned [16] and (2.4) must be solved or matrix $\mathbf{M}$ is manipulated. For such cases, different numerical algorithms are proposed in the literature [17, 18]. In order to avoid this problem, other authors [19] take advantage of symbolic methods for sparse matrices [20]. What this paper is related to are the cases, where the observability of a system such as the one defined above can be addressed in terms of structural considerations, what is called topological observability [9]. In order to introduce this topic, let us define some concepts and hypotheses.

Let S be an n-dimensional physical system that is going to be the object of our study, and let a sensed variables set $\mathbf{z}$ be defined, where m magnitudes are measured over S. Furthermore, let $\mathbf{M}$ be the $m \times n$ matrix associated to S, as defined in (2.4). We will say that S is a sparse system if the behavior of S at any point can be justified exclusively by means of the knowledge of the variables in an area based on a certain neighborhood relationship. This is the case of a traffic model system where flow fluctuations in a certain region are strongly dependent on what happens in that area, whereas the events that take place in other parts show a weak dependence or absolute independence from them. One of the features that characterize a sparse system is that matrix $\mathbf{M}$ is a sparse matrix. Then, some conclusions can be established in terms of structural considerations of $\mathbf{M}$, when the matrix dimensions and the degree of sparsity are large enough. For this purpose, Bunch and Rose [21] define a graph associated to a matrix $\mathbf{M}$, where a nonzero element $m_{ij} \neq 0$ of $\mathbf{M}$ represents an edge that joins vertices i and j. Based on this, some properties can be studied in terms of graph theory because of the duality between sparse linear systems and graphs.

The obvious solution of calculating the rank of matrix $\mathbf{M}$ may present problems and may not be even possible in the case of ill-conditioned systems, as mentioned above. In these cases, a topological-based approach becomes a good choice that presents a series of additional advantages derived from the capability of graphs to answer questions related to observability analysis, including the identification of the maximum observable subsystem and optimal additional sensor placement. In short, in this paper we will introduce new topological analysis techniques by means of certain graphs associated with sparse systems in order to determine the topological observability of such systems.

3. Graph Theory

A graph is defined as a collection of nodes or vertices that are joined through the so-called edges or branches. For the sake of homogeneity here we will use the term branches both for general graphs and for the case of trees, which are basically graphs without loops. In the scope of this work we are interested in defining graphs within a given network, which is also a collection of nodes and branches. In other words, a network must be interpreted as the context where any given graph is declared, in such a way that nodes and branches belonging to a graph are also present in the network for which the graph is defined. Nevertheless, not all the nodes and branches of the network are always present in a graph.

Definition 3.1. Let $X = \{X^0, X^1\}$ be a network, where X^0 and X^1 are the sets of nodes and branches, respectively; a graph G of X is defined as a set of nodes, $G^0 \subseteq X^0$, and a set of branches, $G^1 \subseteq X^1$.

Thus, as in the case of networks, a graph can be denoted by a couple, as follows:

$$G = \left\{ G^0 \subseteq X^0, G^1 \subseteq X^1 \right\}. \tag{3.1}$$

In what follows, it is assumed that X is a connected network, that is, a network where there exists a path in X^1 between every pair of nodes of X^0. In the same way, a connected or unconnected graph G of X can be defined. If a graph G of X is not connected, each connected subgraph that makes it up is known as a connected component. When a connected graph contains no loops, it is called a tree of X.

Definition 3.2. A graph T of X is said to be a spanning tree if T contains no loops, and $T^0 = X^0$.

A directed graph results from the assignment of a direction to each branch in such a way that a node is known as the source, while another node is the target of a directed link.

The matrix representation of any graph G is the node-to-branch incidence matrix, $\mathbf{A}(G) = (a_{kj})$. This is a matrix with as many rows as nodes are in the graph, and where the number of columns is equal to the number of branches in the graph. The elements of $\mathbf{A}(G)$, in the case of directed graphs, are defined as follows:

$$a_{kj} = \begin{cases} 1 & \text{If node } k \text{ is the source of the directed branch } j, \\ -1 & \text{If node } k \text{ is the target of the directed branch } j, \\ 0 & \text{If node } k \text{ is not incident to branch } j. \end{cases} \tag{3.2}$$

The rank of a graph G of X is defined as:

$$\text{rank}\{G\} = \text{size}\left\{G^0\right\} - c, \tag{3.3}$$

where size $\{G^0\}$ denotes the number of nodes in G, while c indicates the number of connected components of G.

Definition 3.3. Let X be a connected network, a graph G of X is said to be of full rank if its rank equals the maximum possible value, $\text{rank}\{G\} = \text{size}\{X^0\} - 1$.

The rank of a graph G of X is, by definition, equal to the rank of its associated incidence matrix $\mathbf{A}(G)$. If G is of full rank, the rank of $\mathbf{A}(G)$ equals the number of rows minus one. In other words, one row of $\mathbf{A}(G)$ is linearly dependent on the others. That is the reason why a reduced node-to-branch incidence matrix $\mathbf{A}_r(G)$ is defined, resulting from the elimination of a row from $\mathbf{A}(G)$. The following expression summarizes all of the above:

$$\text{rank}\{G\} = \text{rank}\{\mathbf{A}_r(G)\}. \tag{3.4}$$

The selection of one node among others for which the associated row is erased is arbitrary. In what follows, this node is going to be known as the reference node.

Definition 3.4. The closure [22] of a connected graph G in X is defined as a graph $\overline{G}$, where $\overline{G}^0 = G^0$ and $\overline{G}^1$ is composed by all the branches in X^1 that join pairs of nodes in G^0.

4. Network Flow Analogy

Consider a set of linear independent variables, $\{x_1, \ldots, x_n\}$, that determine the state of a system S. Let z be a system variable whose magnitude may be expressed as a linear relationship between the state variables, as follows (*notation*: in what follows, subscripts i and j, $1 \leq i, j \leq n$ are used to refer to generic state variables and network nodes; subscript s, $1 \leq s \leq n$ refers to a node that is known as source node; subscript k, $1 \leq k \leq m$ denotes measurements and equations; subscript a, $1 \leq a \leq r$ refers to generic network branches;

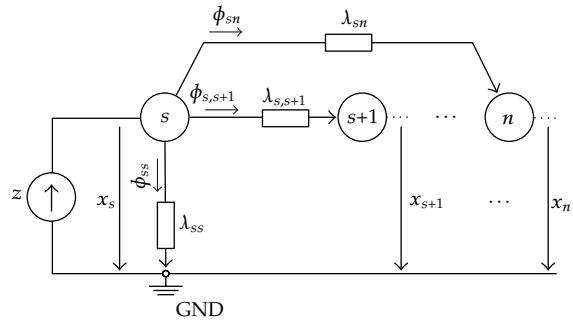


Figure 1: Elementary network.

subscript b under arrays or vectors denotes that those structures contain exclusively branch parameters or variables):

$$z = \alpha_1 x_1 + \cdots + \alpha_n x_n = \sum_{i=1}^{n} \alpha_i x_i, \tag{4.1}$$

where there is at least one value of i for which $\alpha_i \neq 0$. Consider that α_s, where $1 \leq s \leq n$, is the first nonzero coefficient in the above expression. In other words, $\alpha_i = 0$ for all $1 \leq i < s$. Then, the expression can be rewritten as:

$$z = x_s \sum_{i=s}^{n} \alpha_i - \sum_{i=s+1}^{n} (x_s - x_i)\alpha_i, \tag{4.2}$$

which is consistent with the analogy to a flow network as shown in Figure 1. In it, a current z is injected into the network through node s and flows to the remaining nodes, $s + 1$ to n, according to the admittance values and potential differences of the branches connecting them. Therefore, the following equality must hold:

$$z = \sum_{i=s}^{n} \phi_{si} = x_s \lambda_{ss} + \sum_{i=s+1}^{n} (x_s - x_i)\lambda_{si}, \tag{4.3}$$

where, for a generic node i, x_i represents the potential level of the node with respect to a zero potential reference node, *GND* in the figure; $\lambda_{si} = -\alpha_i$ represents the admittance that characterizes a branch connecting node s to node i, so that $\phi_{si} = (x_s - x_i)\lambda_{si}$ is the current that flows from node s to node i due to the potential difference $(x_s - x_i)$ observed from node s to node i; similarly, $\lambda_{ss} = \sum_{i=s}^{n} \alpha_i$ is the admittance between node s and *GND*; hence, a current $\phi_{ss} = x_s \lambda_{ss}$ flows from node s to the reference node. The network in Figure 1 is defined as the elementary network associated to the linear (4.3), which is known as the network nodal equation at node s. The elementary network is a tree, and s is defined as the source node of that tree, while the remaining nodes are considered target nodes.

Note that, on one hand, the elementary network in Figure 1 is characterized by the nodal (4.3), where the flow $\phi_s = \sum_{i=s}^{n} \phi_{si}$ injected in node s equals the z magnitude; on

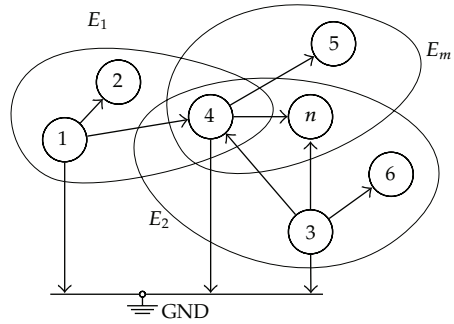


Figure 2: Associated network X resulting from the superposition of m elementary networks.

the other hand, a solution $\{x_s, \ldots, x_n\}$ to the elementary network in Figure 1 is consistent with (4.1).

Let S be a system, where $\{z_1, \ldots, z_m\}$ is a set of m variables whose magnitudes may be expressed as linear relationships of the form:

$$z_k = \alpha_{k1}x_1 + \cdots + \alpha_{kn}x_n = \sum_{i=1}^{n} \alpha_{ki}x_i \quad \forall 1 \leq k \leq m. \tag{4.4}$$

A network associated to a linear equation system such as the one shown above is defined as the result of the superposition of the elementary networks associated to each z_k for all $1 \leq k \leq m$. Then, the solvability of the linear equation system (4.4) is equivalent to that of its associated network, since a particular solution $\{x_1, \ldots, x_n\}$ to the equation system is consistent with the associated network. Figure 2 shows an example of an associated network X as a result of considering all the elementary networks in their entirety, denoted by $E_1, E_2, \ldots, E_m$. Let us take a look at a generic node in the figure, such as node number 4. It is easy to see how the incident branches to node 4 are due to elementary networks associated to variables for which 4 is the source node, such as E_m, and those elementary networks including 4 as a flow target node, such as E_1 and E_2.

Let E_k be the elementary network associated to a generic variable z_k defined in S as shown in (4.4), where s is the source node. A branch admittance matrix of E_k is defined as a diagonal matrix as follows:

$$\mathbf{Y}_b(E_k) = \begin{pmatrix} \sum_{i=s}^{n} \alpha_{ki} & & & 0 \\ & -\alpha_{k,s+1} & & \\ & & \ddots & \\ 0 & & & -\alpha_{kn} \end{pmatrix}. \tag{4.5}$$

In what follows, it is assumed that all coefficients α_{ki} considered in the construction of a matrix $\mathbf{Y}_b(E_k)$, such as the one defined above, are nonzero. In other words, null coefficients, α_{ki}, are removed from (4.4). Note that this constraint does not guarantee that all diagonal elements in $\mathbf{Y}_b(E_k)$ are nonzero because there might exist a case in which, for a certain k,

the sum $\sum_{i=1}^{n} \alpha_{ki}$ equals zero. Those cases are related to the concept of parametric unobservability, and it will be introduced later.

Taking into account the contribution of all the variables $\{z_1, \ldots, z_m\}$ in S to the whole associated network X, a branch admittance matrix of X is defined as a block diagonal matrix:

$$\mathbf{Y}_b(X) = \begin{pmatrix} \mathbf{Y}_b(E_1) & & 0 \\ & \ddots & \\ 0 & & \mathbf{Y}_b(E_m) \end{pmatrix}. \tag{4.6}$$

Then, the following equality is satisfied:

$$\begin{aligned} \mathbf{\Phi}_b(X) &= \mathbf{Y}_{\mathbf{A}}(X) \cdot \mathbf{x}, \\ \mathbf{Y}_{\mathbf{A}}(X) &= \mathbf{Y}_b(X) \cdot \mathbf{A}_r^{\top}(X), \end{aligned} \tag{4.7}$$

where $\mathbf{A}_r(X)$ is the reduced node to branch incidence matrix of X; $\mathbf{x} = (x_i)$ is the $n \times 1$ nodal potential vector, that is, the system state variable column vector; if r is the number of branches in X, and they are numbered from 1 to r, $\mathbf{\Phi}_b(X) = (\phi_a)$ is the $r \times 1$ branch flow vector, that is, a column vector of magnitudes that flow through branches in X; $\mathbf{Y}_{\mathbf{A}}(X)$ is a $r \times n$ matrix that relates potentials x_i at nodes in X with branch flows ϕ_a.

Equations (4.4) can be expressed in matrix form as follows:

$$\mathbf{z} = \mathbf{M} \cdot \mathbf{x}, \tag{4.8}$$

where $\mathbf{M} = (\alpha_{ki})$ is defined as a $m \times n$ coefficient matrix, and where $\mathbf{z} = (z_k)$ and $\mathbf{x} = (x_i)$ are column vectors. Note that each row k of $\mathbf{M}$, that is, each variable z_k considered in the system, will result in an elementary network of X that is a tree because of the lack of loops. Therefore, as any branch in X arises from the existence of a nonzero element in $\mathbf{M}$, a $m \times r$ equation to branch incidence matrix, $\mathbf{B}(X) = (b_{ka})$, associated to X can also be defined as follows:

$$b_{ka} = \begin{cases} 1 & \text{If branch } a \text{ of } X \text{ arises from row } k \text{ of } \mathbf{M}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.9}$$

The following equality holds:

$$\mathbf{M} = \mathbf{B}(X) \cdot \mathbf{Y}_{\mathbf{A}}(X) = \mathbf{B}(X) \cdot \mathbf{Y}_b(X) \cdot \mathbf{A}_r^{\top}(X). \tag{4.10}$$

Equation (4.7) characterizes network X as well as (4.8) characterizes system S from a set of variables $\{z_1, \ldots, z_m\}$ and, therefore, from equalities (4.7) and (4.10) it can be concluded that the study of the determinism of S is equivalent to the observability of X under constraints related to the variables z_k taken into account.

5. Topological Observability

Krumpholz et al. introduced in [9] the term parametric unobservability as a vague notion needed to justify the concept of topological observability in electric power networks under

certain assumptions. In this section, we present a formal description that allows defining and characterizing parametric unobservability and demonstrates how topological observability can be addressed by means of the existence of certain graphs under constraints.

Let S be a large n-dimensional sparse physical system, where a sensing system $\mathbf{z}$ is defined by means of m measured variables, $m \geq n$. Let $\mathbf{M}$ be the coefficient matrix, as defined in (4.8), associated to S and the sensing system, and let X be the associated network. It is important to note that $\mathbf{M}$ characterizes only those parts of the system related to measurements, but not the whole physical system. In particular, it shows the relationship between the sensor set considered and the state variables. Therefore, $\mathbf{M}$ might be a diagonal or block diagonal matrix, without implying either the existence or nonexistence of decoupled subsystems in S. Obviously, the observability analysis of decoupled subsystems, if they exist, can be carried out independently.

The necessary and sufficient condition for algebraic observability of a system S and a sensing configuration $\mathbf{z}$, as proposed above, is

$$\operatorname{rank}\{\mathbf{M}\} = n. \tag{5.1}$$

Let us consider an algebraically observable system S with respect to a sensor set $\mathbf{z}$. As $\mathbf{M}$ is an $m \times n$ matrix and $m \geq n$, from (5.1), it follows that a collection of n linearly independent rows of $\mathbf{M}$ can be found. Let $\mathbf{z}^n$ be the subset of $\mathbf{z}$ corresponding to those linearly independent rows of $\mathbf{M}$. Therefore, an equation subsystem might be defined in S with respect to $\mathbf{z}^n$, that should be characterized using an $n \times n$ coefficient matrix $\mathbf{M}^n$ and its associated network X^n in such a way that:

$$\mathbf{z}^n = \mathbf{M}^n \cdot \mathbf{x}, \tag{5.2}$$

where $\mathbf{z}^n \subseteq \mathbf{z}$, $X^n \subseteq X$, and the determinant $|\mathbf{M}^n| \neq 0$. $\mathbf{z}^n$ is known as a critical sensing configuration in the sense that the loss of any measurement in $\mathbf{z}^n$ should derive in the loss of the observability condition with respect to $\mathbf{z}^n$. For the same reason, system S is said to be critically observable with respect to $\mathbf{z}^n$. The determinant $|\mathbf{M}^n|$ is calculated as a sum of products, each coming from n elements in $\mathbf{M}^n$, and no two coming from the same row or column. Since $\mathbf{M}^n$ is a nonsingular matrix, at least one of these products must be nonzero. Thus, without loss of generality, in what follows let a permutation of rows be considered such that all the factors of the aforementioned nonnull product lie on the principal diagonal of $\mathbf{M}^n$. Note that any row permutation in $\mathbf{M}^n$ does not alter the associated network X^n.

It is clear that the first entry in $\mathbf{M}^n$ in the first row is nonnull and, therefore, there exists in X^n a branch joining node 1 and the reference node. In the second row, there are two possible cases: on one hand, if the diagonal element is the first nonzero element in that row, there exists a branch in X^n joining node 2 and the reference node and, indirectly, the first node too; on the other hand, if the diagonal element is not the first nonzero one, there is a link in X^n between nodes 2 and 1. This argument can be repeated for the next row and up to the last one. Eventually, a spanning tree of full-rank T of X^n is completed because of the lack of loops and the inclusion of the totality of the nodes in the network. Furthermore, the previous analysis leads exclusively to one branch in T from each row in $\mathbf{M}^n$. In other words, the n branches in T are derived from n different measurements in $\mathbf{z}$. Since X^n results from the superposition of n elementary networks, one for each sensed value, each branch in T belongs to a different elementary network.

In order to demonstrate that the existence of such a spanning tree is sufficient, under certain conditions, for the observability of a system with respect to a sensing configuration, a reverse path is considered in which branches are added recursively to a starting spanning tree until the entire network is encompassed.

Consider a spanning tree T of X^n, where each one of the n branches of T belongs to a different elementary network out of the n that form X^n. That is, each elementary network in X^n has a branch and only one that belongs to T. From (4.10), it follows that a matrix $\mathbf{M}(T)$ can be defined as:

$$\mathbf{M}(T) = \mathbf{B}(T) \cdot \mathbf{Y}_{\mathbf{A}}(T), \tag{5.3}$$

where $\mathbf{B}(T)$ is a selection of columns from $\mathbf{B}(X^n)$, while $\mathbf{Y}_{\mathbf{A}}(T)$ is a selection of rows from $\mathbf{Y}_{\mathbf{A}}(X^n)$ corresponding to the n branches of T. Thus, $\mathbf{B}(T)$ is the $n \times n$ identity matrix because the n branches of T belong to n different elementary networks and it follows that:

$$\mathbf{M}(T) = \mathbf{Y}_{\mathbf{A}}(T). \tag{5.4}$$

Note that as $\mathbf{Y}_{\mathbf{A}}(T)$ has the same sparse pattern as $\mathbf{A}_r^{\top}(T)$, and T is a spanning tree of X^n of full rank, $\operatorname{rank}\{\mathbf{Y}_{\mathbf{A}}(T)\} = n$. In other words, $|\mathbf{M}(T)| \neq 0$ because $\mathbf{M}(T)$ is nonsingular. Let k be a generic row of $\mathbf{M}(T)$. The first nonzero entry in row k is in the same column, generically represented by s, as the first nonzero element in row k of $\mathbf{M}^n$. At this point, two cases might take place: one in which column s is the only nonzero entry in row k, and another for which there exists a second nonzero element in column l of row k in $\mathbf{M}(T)$. Since the determinant of a square matrix can be calculated, according to Laplace's formula, as a weighed sum of cofactors or adjuncts along a row or a column, it follows that:

$$|\mathbf{M}(T)| = m_{ks} \cdot C_{ks} + m_{kl} \cdot C_{kl} \neq 0, \tag{5.5}$$

where m_{ks} and m_{kl} are the elements of $\mathbf{M}(T)$ in row k and columns s and l, respectively, and C_{ks} and C_{kl} are their cofactors. Taking into account the same notation as used in (4.4), if m_{ks} is the only nonzero entry in row k, $m_{ks} = \sum_{i=s}^{n} \alpha_{ki}$; otherwise, $m_{ks} = -m_{kl} = \alpha_{kl}$. In both cases, the determinant must be different from zero.

Let T^{+1} be a graph of X^n that results from the union of T and one-branch a of X^n not in T. Consider that the additional branch belongs to an elementary network E_k that corresponds to row k of $\mathbf{M}(T)$ and whose source node is denoted by s. If matrix $\mathbf{M}(T^{+1}) = (m_{ki}^{+1})$ is defined from T^{+1} in the same way as matrix $\mathbf{M}(T)$ was from T, then, two different cases may follow

(1) the additional branch a joins node s and the reference one. Therefore, as the admittance of this branch is equal to $\lambda_a = \sum_{i=s}^{n} \alpha_{ki}$, the only entry that makes matrices $\mathbf{M}(T)$ and $\mathbf{M}(T^{+1})$ different is:

$$m_{ks}^{+1} = m_{ks} + \sum_{i=s}^{n} \alpha_{ki}, \tag{5.6}$$

and, from (5.5), the determinant:

$$\left|\mathbf{M}\left(T^{+1}\right)\right| = |\mathbf{M}(T)| + \sum_{i=s}^{n} \alpha_{ki} \cdot C_{ks}, \tag{5.7}$$

that is equal to zero when:

$$\lambda_a = \sum_{i=s}^{n} \alpha_{ki} = \frac{|\mathbf{M}(T)|}{-C_{ks}}; \tag{5.8}$$

(2) the additional branch a joins node s and a node j, where $s < j \leq n$. In this case, the branch admittance equals $\lambda_a = -\alpha_{kj}$ and both matrices $\mathbf{M}(T)$, and $\mathbf{M}(T^{+1})$ are equal but for two entries in row k:

$$\begin{aligned} m_{ks}^{+1} &= m_{ks} - \alpha_{kj}, \\ m_{kj}^{+1} &= \alpha_{kj}, \end{aligned} \tag{5.9}$$

and, again, the determinant:

$$\left|\mathbf{M}\left(T^{+1}\right)\right| = |\mathbf{M}(T)| + \alpha_{kj} \cdot \left(C_{kj} - C_{ks}\right), \tag{5.10}$$

that vanishes when:

$$\lambda_a = -\alpha_{kj} = \frac{|\mathbf{M}(T)|}{C_{kj} - C_{ks}}. \tag{5.11}$$

Consider T^{+r} to be a graph of X^n that results from the addition to T of a number r of branches of X^n not in T, and let the matrix $\mathbf{M}(T^{+r})$ be defined such that $|\mathbf{M}(T^{+r})| \neq 0$. Let T^{+r+1} be a graph of X^n formed after the inclusion in T^{+r} of a branch a of X^n not in T^{+r}, and consider that the additional branch belongs to an elementary network E_k that corresponds to row k of $\mathbf{M}(T^{+r})$ for which the source node is denoted by s. One of the next two cases will follow:

(1) the additional branch a joins nodes s and the reference one. The admittance of branch a is equal to $\lambda_a = \sum_{i=s}^{n} \alpha_{ki}$ and the determinant of $\mathbf{M}(T^{+r+1})$ is estimated by:

$$\left|\mathbf{M}\left(T^{+r+1}\right)\right| = |\mathbf{M}(T^{+r})| + \sum_{i=s}^{n} \alpha_{ki} \cdot C_{ks}^{+r}, \tag{5.12}$$

where C_{ks}^{+r} is the cofactor of m_{ks}^{+r}. The above determinant becomes null when:

$$\lambda_a = \sum_{i=s}^{n} \alpha_{ki} = \frac{|\mathbf{M}(T^{+r})|}{-C_{ks}^{+r}}; \tag{5.13}$$

(2) the additional branch a joins node s and a node j, where $s < j \leq n$. Then, the branch admittance is equal to $\lambda_a = -\alpha_{kj}$, and the determinant of $\mathbf{M}(T^{+r+1})$ is given by:

$$\left|\mathbf{M}\left(T^{+r+1}\right)\right| = |\mathbf{M}(T^{+r})| + \alpha_{kj} \cdot \left(C_{kj}^{+r} - C_{ks}^{+r}\right), \tag{5.14}$$

where C_{kj}^{+r} and C_{ks}^{+r} are the cofactors of m_{kj}^{+r} and m_{ks}^{+r}, respectively. The determinant will be null if:

$$\lambda_a = -\alpha_{kj} = \frac{|\mathbf{M}(T^{+r})|}{C_{kj}^{+r} - C_{ks}^{+r}}. \tag{5.15}$$

Note that (5.13) and (5.15) allow identifying a set of values of coefficients α_{ki} for which the determinant $|\mathbf{M}(T^{+r+1})|$ might be canceled.

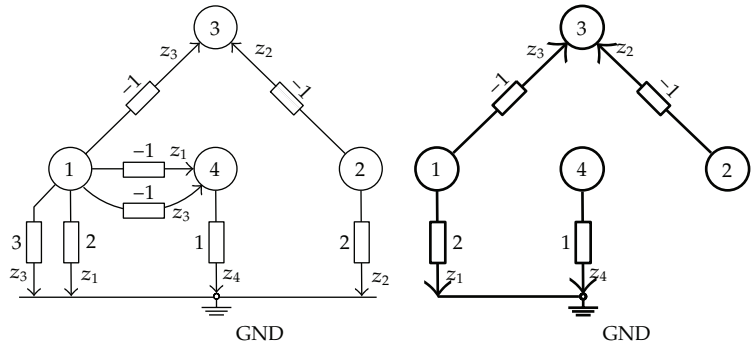
New branches can be added to the given graph, one by one, until the entire network X^n is completed, after the inclusion of all the branches in X^n. Therefore, it is concluded by induction that the determinant of matrix $\mathbf{M}(X^n)$ is nonzero if its entries α_{ki}, $1 \leq k, i \leq n$, do not meet any equality such as (5.8) and (5.13) for branches that join the reference node and (5.11) and (5.15) otherwise.

Consider an example in which a collection of four sensed magnitudes $\mathbf{z} = (z_1, \ldots, z_4)^\top$ are acquired from a four-dimensional physical system. As a result, an equal number of linear equations that relate $\mathbf{z}$ and the state variables $\mathbf{x} = (x_1, \ldots, x_4)^\top$ are established, and a matrix of coefficients $\mathbf{M}$ is given by:

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\substack{\mathbf{M} \\ 4\times 4}} = \underbrace{\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{\substack{\mathbf{B}(X) \\ 4\times 8}} \cdot \underbrace{\begin{pmatrix} 2 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 3 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\substack{\mathbf{Y_A}(X) \\ 8\times 4}}. \tag{5.16}$$

Figure 3(a) shows the resulting associated network, where the branch admittance values are indicated as well as the sensed variables to which each branch is associated. Figure 3(b) shows a spanning tree T of full rank, in which it can be noted that the four branches that conform the tree are associated to four different measured variables. Then, it follows that:

$$\underbrace{\begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{M}(T)} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{B}(T)} \underbrace{\begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{Y_A}(T)}, \tag{5.17}$$



(a) associated network X (b) spanning tree T

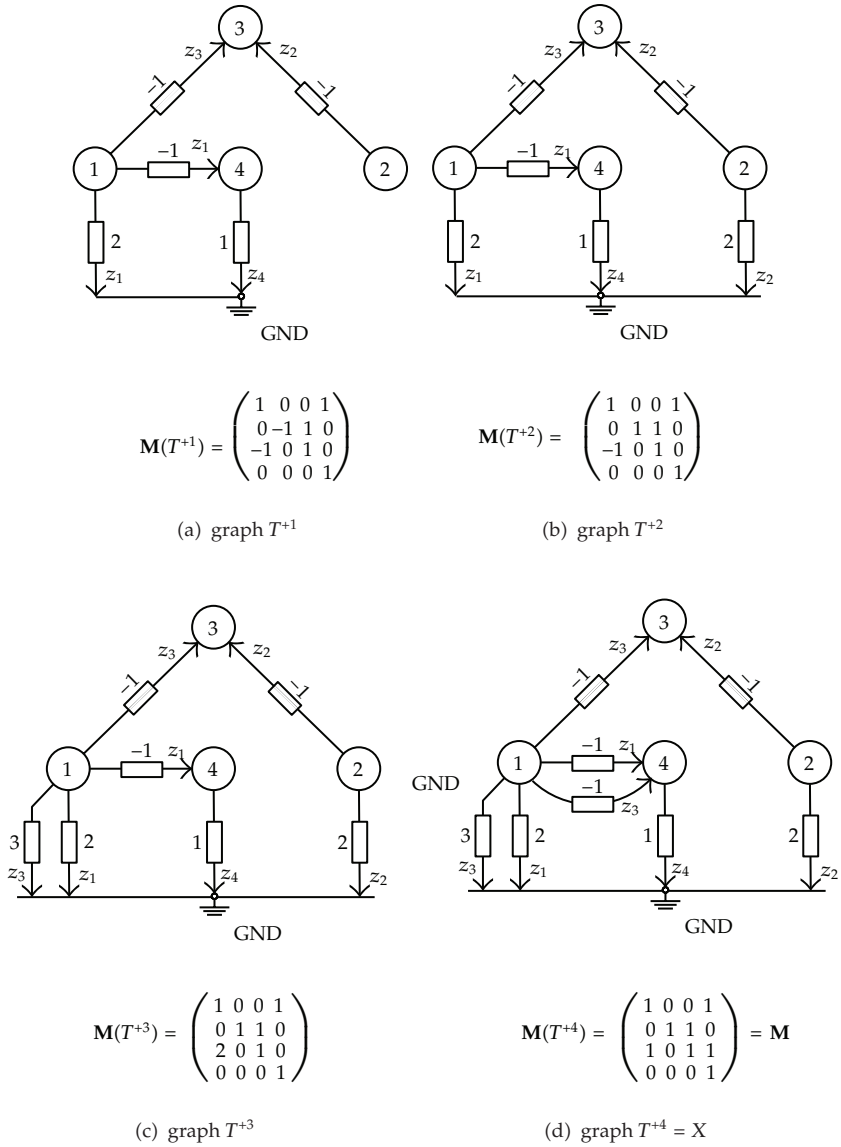
Figure 3: Four-node network example.

where $|\mathbf{M}(T)| \neq 0$. The graphs in Figure 4 show how the entire network X can be reached from T by the addition of each branch of X not belonging to T, and how, at each step $r + 1$, a new $\mathbf{M}(T^{+r+1})$ is defined from the previous one $\mathbf{M}(T^{+r})$ after modifying one or two matrix entries, depending on the case. It can be seen that it always follows that $|\mathbf{M}(T^{+r+1})| \neq 0$ except for the exception cases defined in (5.8), (5.11), (5.13), and (5.15).

Note that this result was reached from the consideration of, on one hand, the network topology and the number, nature, and location of sensors in the network and, on the other, the network parameters. To deal with these two approaches, the concept of parametric unobservability is introduced.

Definition 5.1. A large dimensional and sparse physical system S, for which a sensing system $\mathbf{z}$ is defined, is said to be parametrically unobservable with respect to $\mathbf{z}$ if, in spite of the fact that the ranks of matrices $\mathbf{B}(X)$ and $\mathbf{Y}_{\mathbf{A}}(X)$ are equal to n, the rank of $\mathbf{M}$ is less than n due to the value of one or more coefficients α_{ki} of $\mathbf{M}$.

The relevance of this concept lies in the fact that, in large dimensional sparse physical systems where the parameters are roughly estimated from empirical data or are subject to environmental distortion, it is unlikely for parametric unobservability to occur [9]. In other fields, such as structured linear systems, it is often necessary to work under the assumption of a lack of knowledge of system parameters [4]. In these scenarios, the parametrically unobservability should be associated with a particular set of parameter values. Thus, the observability of a system is said to be true when it is so for almost all parameter values, that is, for all of them except for a set of particular cases in the parameter space. Even though not all physical systems may meet this requirements, there exist evidences that are true for real cases. For example, electric power network analysis involves hundreds or even thousands of state variables that are usually related to the voltage at network nodes. The system state [7] can be estimated by means of the measurement of the power that flows into and through the electric network and which is influenced only by neighboring node states. Thus, the resulting system is clearly sparse, and circuit parameters are affected by environmental conditions such as temperature and humidity as well as by the unreliability of parameter estimation.



(a) graph T^{+1} (b) graph T^{+2}

(c) graph T^{+3} (d) graph $T^{+4} = X$

Figure 4: Four-node network example.

Another example is the case of traffic model analysis [10]. As explained later in the example in this paper, vehicles usually move along a geographical area according to a set of established origin/destination pairs. Traffic flows are sensed at routes in the network in order to estimate the state of the system, that is, origin/destination pair traffic flows. As the network grows, the sparsity becomes more plausible. Additionally, system coefficients are estimated, among other factors, from probabilistic considerations related to the ability of people to opt for one route or another. In brief, parametric unobservability is, in these two cases, highly improbable despite being mathematically possible.

On the basis of the large dimension, sparsity and parameterization uncertainty of such systems, in order to address the observability issue a new strategy is proposed involving exclusively structural and not numerical considerations. For this, a new observability definition should be provided.

Definition 5.2. Let S be a large dimensional sparse physical system, where a sensor network $\mathbf{z}$ is considered; S with $\mathbf{z}$ is said to be topologically observable if S is algebraically observable or parametrically unobservable with respect to $\mathbf{z}$.

Summarizing, it has been demonstrated that the existence of a spanning tree of full-rank T of X where the n branches of T belong to n different elementary networks of X constitutes a necessary and sufficient condition for topological observability. In what follows, any graph G of X with a number r_G of branches that belong to r_G different elementary networks, that is, are associated to r_G different measurements z_k of $\mathbf{z}$, is known as a measured graph.

Theorem 5.3. *Let S be a linear and large n-dimensional sparse physical system, where a sensing system $\mathbf{z}$ is defined by means of a number of m measured variables, $m \geq n$; S is said to be topologically observable with respect to $\mathbf{z}$ if and only if there exists a measured spanning tree T of X.*

The analysis of the observability of a large dimensional sparse physical system S with respect to a sensing system $\mathbf{z}$ from a topological point of view involves searching for a measured spanning tree T of full rank among all possible graphs G of X constructed in such a way that each elementary network that forms X contributes with and only with one branch to G. If the number of sensed values m considered is larger than the dimension n of system S, T will be included, if it exists, as part of a measured spanning graph G of X. In what follows, it is assumed that any graph G of X is a measured graph.

There could be different ways to construct a spanning tree, and any one of them would be valid [12–15]. However what is important here is the fact that the existence of a measured spanning tree is a sufficient condition for the topological observability of a linear system.

Summarizing, taking as a basis the experience in observability analysis in electric power systems, a generalization of the topological approach was developed to address this issue in the scope of other linear, or linearized after a first order derivative, real engineering physical systems. A necessary and sufficient condition for topological observability was established by means of a graph theoretic approach. Finally, thanks to this approach, the characterization of the cases where algebraic strategies do not lead to the same results as those derived from structural analysis was carried out.

6. Maximum Observable Subsystem and Observability Islands

If the observability system test fails for a sensing configuration, it is said that the system is not observable or unobservable. In such cases, the knowledge that might be acquired about one or more parts of the system by all the measures considered should not be underestimated. If a system is not observable, it may be possible to identify a subsystem for which the state can be estimated, it is said that the subsystem is observable. A nondivisible observable subsystem

is known as an observability island. The number of observability islands may vary and depends on the associated network topology, the sensors considered and their location in the network.

Consider an n-dimensional sparse physical system S and a sensing configuration $\mathbf{z}$ for which an associated network X is defined. Let O be an observable island of S and $\mathbf{z}$, and let $X^O \subseteq X$ be its associated subnetwork; X^O is known as an observable subnetwork.

A node belonging to X^O is said to be an observable node, and a branch belonging to X^O is an observable branch. A measured spanning graph G of X^O is known as an observable graph. Nodes and branches that do not belong to any observable subnetwork are said to be unobservable.

Let Y be a measured graph of X; a measure z_k associated to a branch of Y is said to be wholly contained [22] in Y if the elementary network associated to z_k is contained in the closure of Y in X. By extension, a measure is said to be wholly contained in a subnetwork X^O if its associated elementary network is included in the closure of X^O in X.

Any measurement z_k considered in O is wholly contained in X^O. Hence, the state of an observability island may be estimated by means of a wholly contained sensor set.

The union of all the observability islands in a system S derives in a maximum observable subsystem while the union of all their associated observable subnetworks in X results in the maximum observable subnetwork. This subnetwork is maximum because it comprises the largest possible number of observable nodes and, if it exists, it is unique [22].

Consider a system S that is not observable for a sensor set $\mathbf{z}$. Then, no measured spanning tree will be found, as derived from Theorem 5.3. Instead, consider a measured graph G of X as one of the largest connected graphs that can be formed according to the sensing system and the constraints described earlier. Then, G is known as a maximum measured graph of X but not spanning. The next theorem was extracted from [22], where relevant properties concerning maximum measured graphs and observable subsystems are described.

Theorem 6.1. *Let S be a system and X the associated network from a given sensor set. If G is any maximum measured graph of X, the maximum observable subnetwork is contained in the closure of G in X.*

Therefore, based on one of the maximum measured graphs, an iterative process can take place by which the not wholly contained measurements, and their elementary networks are removed from the system until the maximum observable subnetwork is obtained. Additionally, other strategies concerning the search for the maximum observable subsystem can be found in [13, 14] in the scope of electric power networks.

7. Examples

Two examples are presented in this section in order to illustrate the techniques developed in this paper, focusing the attention on the fact that these techniques are valid for different real engineering problems, where a collection of linear equations or equations linearized after

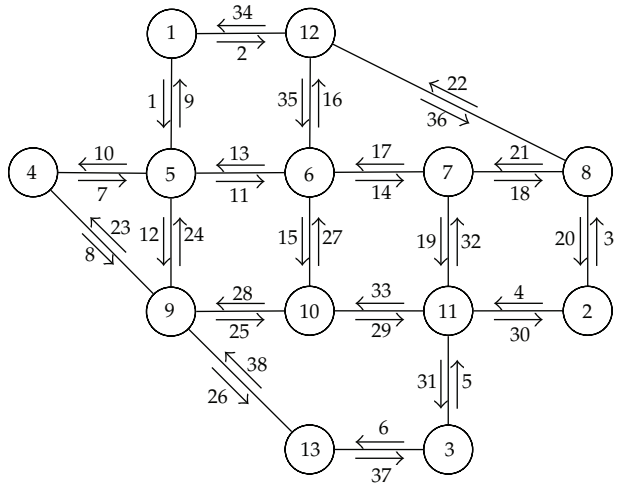


Figure 5: Nguyen-Dupuis traffic network example.

a first order derivative describe the behavior of the system from a measurement acquisition system viewpoint.

7.1. Traffic Model Example

One of the fundamental problems in traffic models concerns the estimation of the origin/destination (OD) trip matrix. Traffic flows are measured by means of sensors spread out at different locations in a study area. These data, in conjunction with other available information, are used to estimate the target matrix, that is, the traffic derived from any OD movement. For each OD pair there exist, in general, more than one alternative to complete the trip that are usually expressed in terms of percentages or probabilities based on contextual factors. In addition, the flow magnitudes at a link in a traffic network can be broken down into percentages of vehicles moving along different OD trips. Thus, linear relationships can be established between OD-pair and link flows. Let $\mathbf{t} = (t_{\mathrm{od}})$ and $\mathbf{v} = (v_b)$ be OD-pair and link flow vectors, respectively; their linear relationships can be described by a matrix $\mathbf{F}$ as follows:

$$\mathbf{v} = \mathbf{F} \cdot \mathbf{t}. \tag{7.1}$$

Figure 5 shows a benchmark case, known as the Nguyen-Dupuis network [23] in the literature, consisting of 13 plausible origin/destination places that are interconnected by 19 bidirectional links. In that scenario, vehicles can move from one place to another through suitable routes. Figure 5 shows indices assigned to links along with their directions. Therefore, for an OD-pair, any possible path is defined as a series of oriented link indices; for example, the sequence $\{2, 36, 20\}$ denotes an alternative for a displacement from 1 (origin) to 2 (destination).

In what follows, it is assumed that matrix **F** and the OD-pairs are known. Below are all the OD pairs considered in this example and their potential paths as well as matrix **F**. They are the same as those tested in [10]:

OD-pair 1-2 : {2, 36, 20}

OD-pair 1-3 : {1, 11, 14, 19, 31}, {1, 11, 15, 29, 31}
{1, 12, 25, 29, 31}, {1, 12, 26, 37}
{2, 35, 14, 19, 31}, {2, 35, 15, 29, 31}

OD-pair 2-1 : {3, 21, 17, 13, 9}, {3, 21, 17, 16, 34}
{3, 22, 34}, {4, 32, 17, 13, 9}
{4, 32, 17, 16, 34}, {4, 33, 27, 13, 9}
{4, 33, 27, 16, 34}, {4, 33, 28, 24, 9}

OD-pair 2-4 : {3, 21, 17, 13, 10}, {4, 32, 17, 13, 10}
{4, 33, 27, 13, 10}, {4, 33, 28, 23}
{4, 33, 28, 24, 10}

OD-pair 3-1 : {5, 32, 17, 13, 9}, {5, 32, 17, 16, 34}
{5, 33, 27, 13, 9}, {5, 33, 27, 16, 34}
{5, 33, 28, 24, 9}, {6, 38, 24, 9}

OD-pair 3-4 : {5, 32, 17, 13, 10}, {5, 33, 27, 13, 10}
{5, 33, 28, 23}, {5, 33, 28, 24, 10}
{6, 38, 23}, {6, 38, 24, 10}

OD-pair 4-2 : {7, 11, 14, 18, 20}, {7, 11, 14, 19, 30}
{7, 11, 15, 29, 30}, {7, 12, 25, 29, 30}
{8, 25, 29, 30}

OD-pair 4-3 : {7, 11, 14, 19, 31}, {7, 11, 15, 29, 31}
{7, 12, 25, 29, 31}, {7, 12, 26, 37}
{8, 25, 29, 31}, {8, 26, 37}

$$\mathbf{F} = \begin{pmatrix}
\cdot & 4 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
1 & 2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & 3 & 1 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & 5 & 4 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & 5 & 4 & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & 1 & 2 & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 4 & 4 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & 2 \\
\cdot & \cdot & -4 & \cdot & -4 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & -4 & \cdot & -4 & \cdot & \cdot \\
\cdot & 2 & \cdot & \cdot & \cdot & \cdot & 3 & 2 \\
\cdot & 2 & \cdot & \cdot & \cdot & \cdot & 1 & 2 \\
\cdot & \cdot & -3 & -3 & -2 & -2 & \cdot & \cdot \\
\cdot & 2 & \cdot & \cdot & \cdot & \cdot & 2 & 1 \\
\cdot & 2 & \cdot & \cdot & \cdot & \cdot & 1 & 1 \\
\cdot & \cdot & 3 & \cdot & 2 & \cdot & \cdot & \cdot \\
\cdot & \cdot & -4 & -2 & -2 & -1 & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\
\cdot & 2 & \cdot & \cdot & \cdot & \cdot & 1 & 1 \\
-1 & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot \\
\cdot & \cdot & -2 & -1 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & -1 & \cdot & -2 & \cdot & \cdot \\
\cdot & \cdot & -1 & -1 & -2 & -2 & \cdot & \cdot \\
\cdot & 1 & \cdot & \cdot & \cdot & \cdot & 2 & 2 \\
\cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & 2 \\
\cdot & \cdot & -2 & -1 & -2 & -1 & \cdot & \cdot \\
\cdot & \cdot & -1 & -2 & -1 & -2 & \cdot & \cdot \\
\cdot & 3 & \cdot & \cdot & \cdot & \cdot & 3 & 3 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -4 & \cdot \\
\cdot & -5 & \cdot & \cdot & \cdot & \cdot & \cdot & -4 \\
\cdot & \cdot & -2 & -1 & -2 & -1 & \cdot & \cdot \\
\cdot & \cdot & -3 & -3 & -3 & -3 & \cdot & \cdot \\
\cdot & \cdot & -4 & \cdot & -2 & \cdot & \cdot & \cdot \\
\cdot & -2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
-1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & -2 \\
\cdot & \cdot & \cdot & \cdot & -1 & -2 & \cdot & \cdot
\end{pmatrix}. \tag{7.2}$$

Note that **F** characterizes the physics of the whole traffic network because it relates the defined OD-pair flows with all the 38 possible oriented traffic link flows:

$$(v_1, \ldots, v_{38})^{\top} = \mathbf{F} \cdot (t_{1\text{-}2}, t_{1\text{-}3}, t_{2\text{-}1}, t_{2\text{-}4}, t_{3\text{-}1}, t_{3\text{-}4}, t_{4\text{-}2}, t_{4\text{-}3})^{\top}. \tag{7.3}$$

A question arises when we want to know if a given sensor network allows to estimate the state of the traffic system or where sensors should be placed in order to complete an observable sensed system. Two cases are going to be taken into account concerning these issues.

7.1.1. Case 1: Observable Configuration

Consider a sensor network consisting of 8 traffic flow meters that result in a measured variable vector **z** whose magnitudes might be estimated by means of a submatrix of **F** and the system state variables **x**, that is, OD-pair traffic flows, as follows:

$$\underbrace{\begin{pmatrix} v_2 \\ v_4 \\ v_5 \\ v_7 \\ v_{12} \\ v_{15} \\ v_{27} \\ v_{38} \end{pmatrix}}_{\substack{\mathbf{z} \\ 8\times 1}} = \underbrace{\begin{pmatrix} 1 & 2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 5 & 4 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 5 & 4 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 4 & 4 \\ \cdot & 2 & \cdot & \cdot & \cdot & \cdot & 1 & 2 \\ \cdot & 2 & \cdot & \cdot & \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & -2 & -1 & -2 & -1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & -1 & -2 & \cdot & \cdot \end{pmatrix}}_{\substack{\mathbf{M} \\ 8\times 8}} \cdot \underbrace{\begin{pmatrix} t_{1\text{-}2} \\ t_{1\text{-}3} \\ t_{2\text{-}1} \\ t_{2\text{-}4} \\ t_{3\text{-}1} \\ t_{3\text{-}4} \\ t_{4\text{-}2} \\ t_{4\text{-}3} \end{pmatrix}}_{\substack{\mathbf{x} \\ 8\times 1}}. \tag{7.4}$$

The question arises as to whether OD-pair traffic flows **x** can be estimated from this sensor set **z** among the aforementioned oriented link flows.

In Figure 6, the elementary networks derived from the coefficient matrix **M** of (7.4) are shown. Note how OD-pairs play the role of network nodes, while OD-pair traffic flows are the network node potential levels. In the figures, branch admittance values are indicated; indices were assigned to the branches and are shown in the figures by smaller numbers next to the arrows.

Figure 7 shows the entire associated network and how a measured spanning tree, highlighted using thick line, was found among other possibilities. Note that each elementary network is related to one and only one branch in the resulting measured spanning tree. This tree is not unique but the existence of, at least one, guarantees the topological observability of the system for the sensor set defined in (7.4).

7.1.2. Case 2: Not Observable Configuration

In a second case, a total of 6 traffic flow meters are considered. The question arises as to whether system observability can be achieved by incorporating additional sensors. And if it is not possible, which is the maximum observable subsystem?

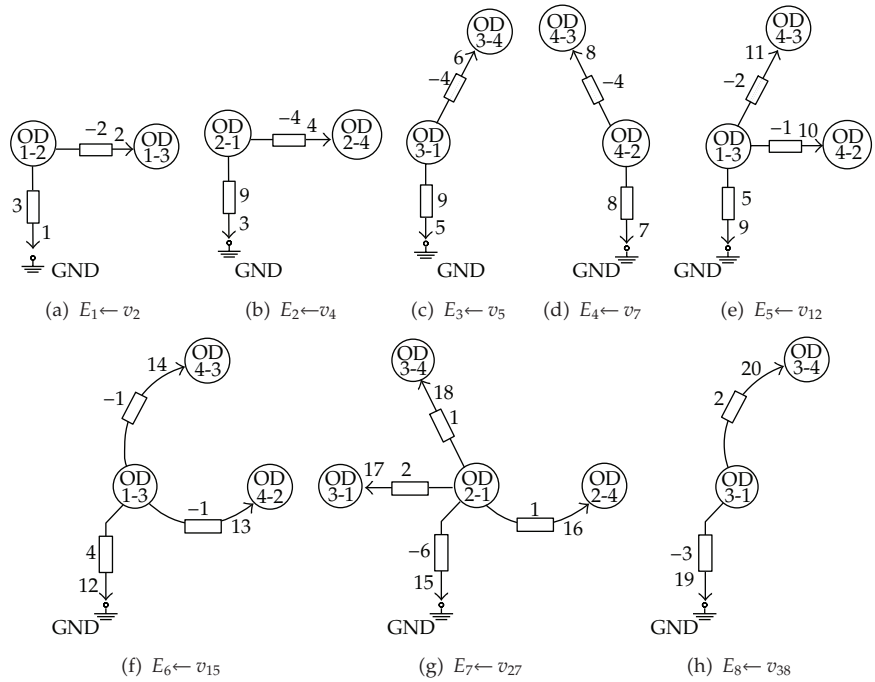


Figure 6: Nguyen-Dupuis elementary networks.

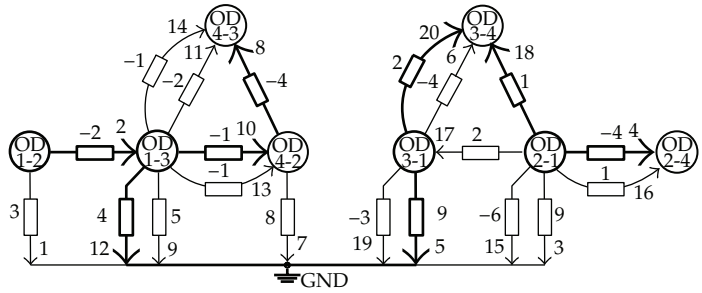


Figure 7: Nguyen-Dupuis case 1: entire associated network and measured spanning tree.

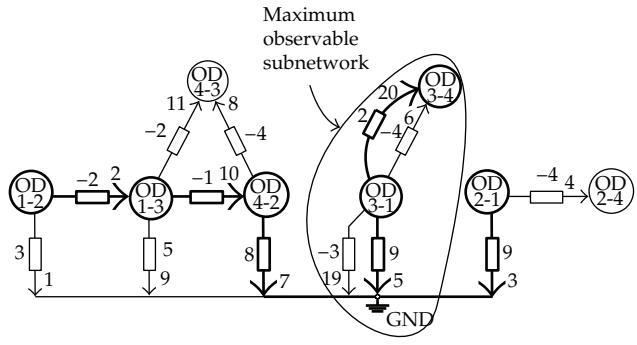


Figure 8: Nguyen-Dupuis case 2: associated network, maximum measured graph, and maximum observable subnetwork.

Let $\{v_2, v_4, v_5, v_7, v_{12}, v_{38}\}$ be the initial sensor set. This is a subset of the observable configuration discussed earlier. Therefore, the linear equations that characterize this case are given by:

$$\underbrace{\begin{pmatrix} v_2 \\ v_4 \\ v_5 \\ v_7 \\ v_{12} \\ v_{38} \end{pmatrix}}_{\substack{\mathbf{z} \\ 6\times 1}} = \underbrace{\begin{pmatrix} 1 & 2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 5 & 4 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 5 & 4 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 4 & 4 \\ \cdot & 2 & \cdot & \cdot & \cdot & \cdot & 1 & 2 \\ \cdot & \cdot & \cdot & \cdot & -1 & -2 & \cdot & \cdot \end{pmatrix}}_{\substack{\mathbf{M} \\ 6\times 8}} \cdot \underbrace{\begin{pmatrix} t_{1\text{-}2} \\ t_{1\text{-}3} \\ t_{2\text{-}1} \\ t_{2\text{-}4} \\ t_{3\text{-}1} \\ t_{3\text{-}4} \\ t_{4\text{-}2} \\ t_{4\text{-}3} \end{pmatrix}}_{\substack{\mathbf{x} \\ 8\times 1}}. \tag{7.5}$$

Figure 8 shows the resulting associated network, X, and one of the possible maximum measured graphs, G (thick lines). Note that OD-pairs 2-4 and 4-3 are clearly not observable, that is, their traffic flows cannot be estimated by means of the available measurements. A more detailed analysis leads to the conclusion that measurements v_4, v_7, and v_{12} are not wholly contained in G and, therefore, their associated elementary networks E_2, E_4, and E_5, respectively, should be removed from the network in order to search for the maximum observable subnetwork. This argument should be repeated until the resulting subnetwork is made up exclusively of elementary networks associated to wholly contained measurements. That is the case after removing E_1, the elementary network associated to measure v_2. From there, the maximum observable traffic subsystem is immediate and is given by OD-pairs 3-1 and 3-4 and oriented traffic link flow sensed values $\{v_5, v_{38}\}$.

To achieve a totally observable system, it is necessary to add two new traffic flow meters that allow to join the maximum measured graph in Figure 8 and the isolated nodes given by OD-pairs 2-4 and 4-3. Each row in matrix $\mathbf{F}$ of (7.2) corresponds to an oriented traffic link flow and, in particular, those rows with nonzero coefficients in columns related to isolated OD-pairs are plausible candidates to improve the system observability. Thus, the inclusion of one of the sensed values from:

$$\{v_3, v_{10}, v_{13}, v_{17}, v_{21}, v_{23}, v_{24}, v_{27}, v_{28}, v_{32}, v_{33}\}, \tag{7.6}$$

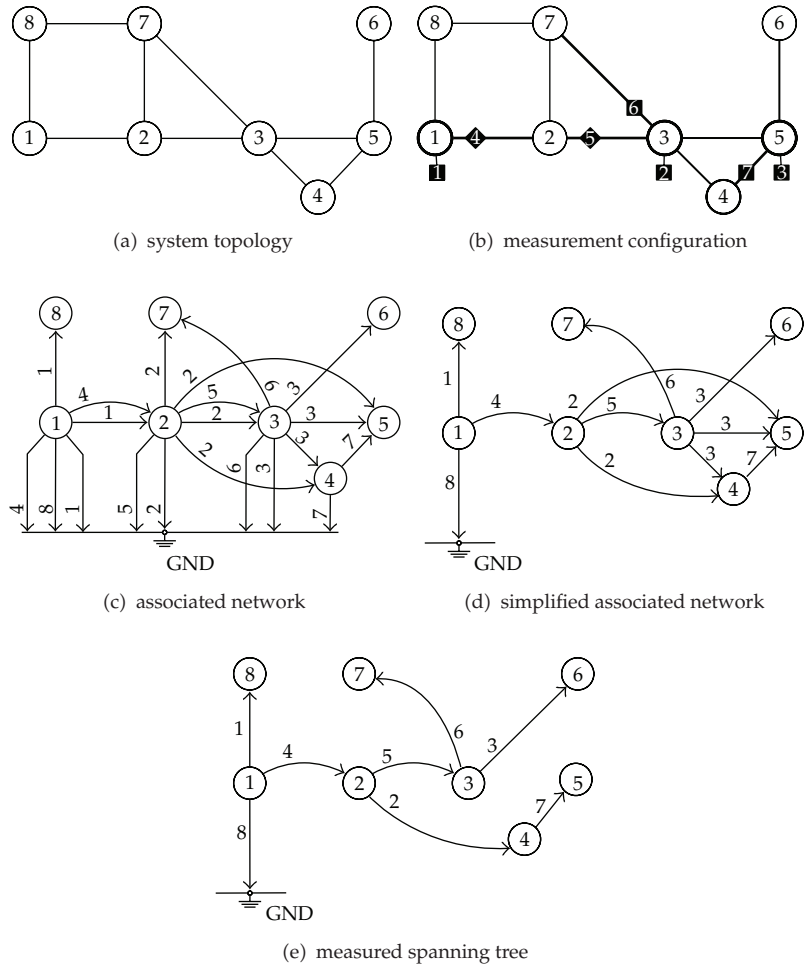
that allow joining the OD-pair 2-4 node, in conjunction with one of the following:

$$\{v_8, v_{11}, v_{14}, v_{15}, v_{19}, v_{25}, v_{26}, v_{29}, v_{31}, v_{37}\}, \tag{7.7}$$

that allow joining OD-pair 4-3 node, would permit observing the whole traffic system.

7.2. Electric Power System Example

As it was mentioned in the introduction, observability analysis in electric power systems has been an important research topic for decades. In particular, this issue can be addressed by means of topological methods, when the set of measured variables are made up exclusively



(a) system topology (b) measurement configuration

(c) associated network (d) simplified associated network

(e) measured spanning tree

Figure 9: Electric power system example.

of bus voltages and active and reactive powers that are injected into or flow through the system [9]. In those cases the system can be considered as a decoupled system [7], that is, a pair of two independent subsystems: one of which can be analyzed by means of active power measurements, and is known as *P-δ* subsystem; the other one, the *Q-U* subsystem, can be studied exclusively from bus voltages and reactive powers measured in the system. Only the *Q-U* subsystem is going to be analyzed in this example. Such a subsystem is observable [7] when a sufficient number of well-placed reactive powers are measured, and, at least, one node voltage is known at any node.

An electric power system is commonly represented as a mesh where the edges denotes the lines in charge of transporting the electric energy, and where the nodes are the places where the lines are incident, that is, the places where electricity is generated, consumed, or transformed. Figure 9(a) shows the topology of an example of an electric power system with 8 nodes. The places where reactive powers are acquired in the system are shown in Figure 9(b), where two kinds of measurements may be distinguished:

(i) node measurements, numbered as 1, 2, and 3 in Figure 9(b), corresponding to reactive powers injected into the system through a node. These derive in equations of the form:

$$\begin{aligned} z_1 &= Q_1 - c_1 = \alpha_{11}U_1 + \alpha_{12}U_2 + \alpha_{18}U_8, \\ z_2 &= Q_3 - c_2 = \alpha_{22}U_2 + \alpha_{23}U_3 + \alpha_{24}U_4 + \alpha_{25}U_5 + \alpha_{27}U_7, \\ z_3 &= Q_5 - c_3 = \alpha_{33}U_3 + \alpha_{34}U_4 + \alpha_{35}U_5 + \alpha_{36}U_6, \end{aligned} \tag{7.8}$$

where Q_i denotes the i-th node reactive power, U_i represents the voltage at node i, α_{ki} is a coefficient related to measurement k and node i, and c_k denotes a constant term related to measurement k;

(ii) branch measurements, numbered as 4, 5, 6, and 7 in Figure 9(b), corresponding to reactive powers that flow through the lines. These derive in equations of the form:

$$\begin{aligned} z_4 &= Q_{12} - c_4 = \alpha_{41}U_1 + \alpha_{42}U_2, \\ z_5 &= Q_{23} - c_5 = \alpha_{52}U_2 + \alpha_{53}U_3, \\ z_6 &= Q_{37} - c_6 = \alpha_{63}U_3 + \alpha_{67}U_7, \\ z_7 &= Q_{45} - c_7 = \alpha_{74}U_4 + \alpha_{75}U_5, \end{aligned} \tag{7.9}$$

where Q_{ij} denotes a branch reactive power that is acquired in a line that joins nodes i and j.

Finally, a voltage measure at node 1 is also considered, resulting in an equation as follows:

$$z_8 = U_1. \tag{7.10}$$

Summarizing, the linear equations that characterize the Q-U subsystem and the given measurement configuration are as follows:

$$\underbrace{\begin{pmatrix} Q_1 - c_1 \\ Q_3 - c_2 \\ Q_5 - c_3 \\ Q_{12} - c_4 \\ Q_{23} - c_5 \\ Q_{37} - c_6 \\ Q_{45} - c_7 \\ U_1 \end{pmatrix}}_{\substack{\mathbf{z} \\ 8\times 1}} = \underbrace{\begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdot & \cdot & \cdot & \cdot & \cdot & \alpha_{18} \\ \cdot & \alpha_{22} & \alpha_{23} & \alpha_{24} & \alpha_{25} & \cdot & \alpha_{27} & \cdot \\ \cdot & \cdot & \alpha_{33} & \alpha_{34} & \alpha_{35} & \alpha_{36} & \cdot & \cdot \\ \alpha_{41} & \alpha_{42} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \alpha_{52} & \alpha_{53} & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \alpha_{63} & \cdot & \cdot & \cdot & \alpha_{67} & \cdot \\ \cdot & \cdot & \cdot & \alpha_{74} & \alpha_{75} & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}}_{\substack{\mathbf{M} \\ 8\times 8}} \underbrace{\begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \\ U_7 \\ U_8 \end{pmatrix}}_{\substack{\mathbf{x} \\ 8\times 1}}. \tag{7.11}$$

Figure 9(c) shows the associated network derived from (7.11), where branch admittances were suppressed in order to clarify the drawing. The numbers close to the oriented edges of the graph denote the order of the measurement from which the edge is derived, that is, the order of the elementary network in which it is defined. Note that the only branch associated

to measurement z_8 and, in general, to any node voltage measure, is the one that joins the node where the voltage is acquired and the reference node. As a result, the reference node is implicitly connected to the rest of the nodes due to the inclusion of just one node voltage measurement, and a simplified associated network may be taken into account as shown in Figure 9(d), where thicker lines represent the edges that are present in the entire individuals of the search space of measured graphs. Note that those edges are the ones due to the node voltage and branch reactive power measurements.

Finally, one of the possible measured spanning trees is shown in Figure 9(e), after the assignment of each of the eight measurements considered to one of the edges in the associated network. The existence of such a tree permits concluding that the electric power *Q-U* subsystem is topologically observable for the given sensing system.

8. Conclusions

In this paper, a new topological approach to the determination of the observability of a physical system where a sensor network is defined has been presented. The techniques developed in this paper were inspired by the contributions of researchers in the scope of electric power systems and generalized to other physical sparse linear systems. The terms parametric unobservability and topological observability have been introduced and justified in a formal way, which allows characterizing those parameter dependent cases where an algebraic approach to the observability issue led to different results than the topological one. A sensing system has been considered for any linear physical system or, at least, linearized after a first order derivative. From there, an associated network has been defined, and it has been demonstrated that the existence of certain constrained graphs, known as measured graphs, in the scope of the associated network permits characterizing the topological observability of the system. From this graph approach, the determination of the maximum observable subsystem can be carried out in case of unobservability. The technique has been illustrated with the help of two examples in the scope of traffic sensing structures and electric power systems.

Acknowledgments

This work was partially funded by the Xunta de Galicia, the MICINN of Spain and European Regional Development Funds through projects 09DPI012166PR, 10DPI005CT, and TIN2011-28753-C02-01.

References

[1] R. E. Kalman, "Mathematical description of linear dynamical systems," *SIAM Journal on Control and Optimization*, vol. 1, pp. 152–192, 1963.

[2] J. Li, L. L. H. Andrew, C. H. Foh, M. Zukerman, and H. H. Chen, "Connectivity, coverage and placement in wireless sensor networks," *Sensors*, vol. 9, no. 10, pp. 7664–7693, 2009.

[3] C. T. Lin, "Structural controllability," *IEEE Transactions on Automatic Control*, vol. 19, pp. 201–208, 1974.

[4] J.-M. Dion, C. Commault, and J. van der Woude, "Generic properties and control of linear structured systems: a survey," *Automatica*, vol. 39, no. 7, pp. 1125–1144, 2003.

[5] T. Boukhobza, F. Hamelin, and S. Martinez-Martinez, "State and input observability for structured linear systems: a graph-theoretic approach," *Automatica*, vol. 43, no. 7, pp. 1204–1210, 2007.

[6] T. Boukhobza and F. Hamelin, "Observability analysis for structured bilinear systems: a graph-theoretic approach," *Automatica*, vol. 43, no. 11, pp. 1968–1974, 2007.
[7] F. C. Schweppe, "Power system static-state estimation, parts I,II and III," *IEEE Transactions on Power Apparatus and Systems*, vol. 89, no. 1, pp. 120–135, 1970.
[8] A. Monticelli and F. F. Wu, "Network observability: theory," *IEEE Transactions on Power Apparatus and Systems*, vol. 104, no. 5, pp. 1042–1048, 1985.
[9] G. R. Krumpholz, K. A. Clements, and P. W. David, "Power system observability: a practical algorithm using network topology," *IEEE Transactions on Power Apparatus and Systems*, vol. 99, no. 4, pp. 1534–1542, 1980.
[10] E. Castillo, P. Jiménez, J. M. Menéndez, and A. J. Conejo, "The observability problem in traffic models: Algebraic and topological methods," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 2, pp. 275–287, 2008.
[11] V. Veverka and F. Madron, *Material and Energy Balancing in the Process Industries: from Microscopic Balances to Large Plants*, Computeraided Chemical Engineering, Elsevier Science, 1997.
[12] V. H. Quintana, A. Simoes-Costa, and A. Mandel, "Power system topological observability using a direct graph-theoretic approach," *IEEE Transactions on Power Apparatus and Systems*, vol. 101, no. 3, pp. 617–626, 1982.
[13] R. R. Nucera and M. L. Gilles, "Observability analysis: a new topological algorithm," *IEEE Transactions on Power Systems*, vol. 6, no. 2, pp. 466–475, 1991.
[14] G. N. Korres and P. J. Katsikas, "A hybrid method for observability analysis using a reduced network graph theory," *IEEE Transactions on Power Systems*, vol. 18, no. 1, pp. 295–304, 2003.
[15] S. Vazquez-Rodriguez, A. Faina, and B. Neira-Duenas, "An evolutionary technique with fast convergence for power system topological observability analysis," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 3086–3090, 2006.
[16] J. R. Rice, *Matrix Computations and Mathematical Software*, McGraw-Hill Computer Science Series, McGraw-Hill, New York, NY, USA, 1983.
[17] H. J. Kim, "A new algorithm for solving ill-conditioned linear systems," *IEEE Transactions on Magnetics*, vol. 32, no. 3, pp. 1373–1376, 1996.
[18] S. K. Kurtzl, "A direct algorithm for solving ill-conditioned linear algebraic systems," *Advances*, vol. 42, pp. 629–633, 2000.
[19] M. Cosnard and L. Grigori, "Using postordering and static symbolic factorization for parallel sparse lu," in *Proceedings of the 14th International Parallel and Distributed Processing Symposium (IPDPS '00)*, pp. 807–812, 2000.
[20] W. F. Tinney, V. Brandwajn, and S. M. Chan, "Sparse vector methods," *IEEE Transactions on Power Apparatus and Systems*, vol. 104, no. 2, pp. 295–301, 1985.
[21] J. R. Bunch and D. J. Rose, "Partitioning, tearing and modification of sparse linear systems," *Journal of Mathematical Analysis and Applications*, vol. 48, pp. 574–593, 1974.
[22] K. A. Clements, G. R. Krumpholz, and P. W. Davis, "Power system state estimation with measurement deficiency: an algorithm that determines the maximal observable subnetwork," *IEEE Transactions on Power Apparatus and Systems*, vol. 101, no. 9, pp. 3044–3052, 1982.
[23] S. Nguyen and C. Dupuis, "An efficient method for computing traffic equilibria in networks with asymmetric transportation costs," *Transportation Science*, vol. 18, no. 2, pp. 185–202, 1984.

Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 402490, 12 pages
doi:10.1155/2012/402490

Research Article

A Relaxed Splitting Preconditioner for the Incompressible Navier-Stokes Equations

Ning-Bo Tan, Ting-Zhu Huang, and Ze-Jun Hu

School of Mathematical Sciences, University of Electronic Science and Technology of China, Sichuan, Chengdu 611731, China

Correspondence should be addressed to Ting-Zhu Huang, tingzhuhuang@126.com

Received 8 December 2011; Revised 2 April 2012; Accepted 19 April 2012

Academic Editor: Massimiliano Ferronato

A relaxed splitting preconditioner based on matrix splitting is introduced in this paper for linear systems of saddle point problem arising from numerical solution of the incompressible Navier-Stokes equations. Spectral analysis of the preconditioned matrix is presented, and numerical experiments are carried out to illustrate the convergence behavior of the preconditioner for solving both steady and unsteady incompressible flow problems.

1. Introduction

We consider systems of linear equations arising from the finite-element discretization of the incompressible Navier-Stokes equations governing the flow of viscous Newtonian fluids. The primitive variables formulation of the Navier-Stokes equations is

$$\frac{\partial \mathbf{u}}{\partial t} - \upsilon \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p = \mathbf{f} \quad \text{on } \Omega \times (0, T], \tag{1.1}$$

$$\operatorname{div} \mathbf{u} = 0 \quad \text{on } \Omega \times [0, T], \tag{1.2}$$

$$\mathbf{u} = \mathbf{g} \quad \text{on } \partial\Omega \times [0, T], \tag{1.3}$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \quad \text{on } \Omega, \tag{1.4}$$

where $\Omega \subset \mathbb{R}^2$ is an open bounded domain with sufficiently smooth boundary $\partial\Omega$, $[0, T]$ is an time interval of interest, $\mathbf{u}(\mathbf{x}, t)$ and $p(\mathbf{x}, t)$ are unknown velocity and pressure fields, υ is the kinematic viscosity, Δ is the vector Laplacian, ∇ is the gradient, div is the divergence, and

$\mathbf{f}$, $\mathbf{g}$, and $\mathbf{u}_0$ are given functions. The Stokes problem is obtained by dropping the nonlinearity $(\mathbf{u}\cdot\nabla)\mathbf{u}$ from the momentum equation (1.1). Refer to [1] for an introduction to the numerical solution of the Navier-Stokes equations. Implicit time discretization and linearization of the Navier-Stokes equations by Picard or Newton fixed iteration result in a sequence of (generalized) Oseen problems. The Oseen problems by spatial discretization with LBB-stable finite elements (see [1, 2]) are reduced to a series of large sparse systems of linear equations with a saddle point matrix structure as follows:

$$\widetilde{\mathbf{A}}\mathbf{x} = \mathbf{b}, \tag{1.5}$$

with

$$\widetilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ -\mathbf{B} & 0 \end{bmatrix}, \qquad \mathbf{x} = \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} \mathbf{f} \\ -g \end{pmatrix}, \tag{1.6}$$

where $\mathbf{u}$ and p represent the discrete velocity and pressure, respectively. In two-dimensional cases, $\mathbf{A} = \operatorname{diag}(A_1, A_2)$ denotes the discretization of the reaction diffusion, and each diagonal submatrix A_i is a scalar discrete convection-diffusion operator represented as

$$A_i = \sigma V + \upsilon L + N_i \quad (i = 1, 2), \tag{1.7}$$

where V denotes the velocity mass matrix, L the discrete (negative) Laplacian, and N_i the convective terms. The matrix $\mathbf{A}$ is positive definite in the sense that $\mathbf{A}^T + \mathbf{A}$ is symmetric positive definite. Matrix $\mathbf{B}^T = (B_1^T, B_2^T)$ denotes the discrete gradient with B_1^T, B_2^T being discretizations of the partial derivatives $\partial/\partial x$, $\partial/\partial y$, respectively. $\mathbf{f} = (f_1, f_2)^T$ and g contain the forcing and boundary terms.

In the past few years, a considerable amount of work has been spent in developing efficient solvers for systems of linear equations in the form of (1.5); see [3] for a comprehensive survey. Here we consider preconditioned Krylov subspace methods, in particular preconditioned GMRES [4] in this paper. The convergence performance of this method is mainly determined by the underlying preconditioner employed. An important class of preconditioners is based on the block LU factorization of the coefficient matrix, including a variety of block diagonal and triangular preconditioners. A crucial ingredient in all these preconditioners is an approximation to the Schur complement $\mathbf{S} = \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$. This class of preconditioners includes the pressure convection diffusion (PCD) preconditioner, the least-squares commutator (LSC) preconditioner, and their variants [5–7]. Somewhat related to this class of preconditioners are those based on the augmented Lagrangian (AL) reformulation of the saddle point problem; see [8–11]. Other types of preconditioners for the saddle point problems include those based on the Hermitian and skew-Hermitian splitting (HSS) [12–15] and the dimensional splitting (DS) [16] of the coefficient matrix $\widetilde{\mathbf{A}}$. In [17], a relaxed dimensional factorization preconditioner is introduced.

The remainder of the paper is organized as follows. In Section 2, we present a relaxed splitting preconditioner based on matrix splitting and prove that the preconditioned matrix has eigenvalue 1 of algebraic multiplicity at least n (recall that n is the number of velocity degrees of freedom). In Section 3, we show the results of a series of numerical experiments indicating the convergence behavior of the relaxed splitting preconditioner. In the final section, we draw our conclusions.

2. A Relaxed Splitting Preconditioner

2.1. A Splitting of the Matrix

In this paper, we limit to 2D case. The system matrix $\widetilde{\mathbf{A}}$ admits the following splitting:

$$\widetilde{\mathbf{A}} = \begin{bmatrix} A_1 & 0 & B_1^T \\ 0 & A_2 & B_2^T \\ -B_1 & -B_2 & 0 \end{bmatrix} = \begin{bmatrix} A_1 & 0 & 0 \\ 0 & 0 & 0 \\ -B_1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & B_1^T \\ 0 & A_2 & B_2^T \\ 0 & -B_2 & 0 \end{bmatrix} = H + S, \tag{2.1}$$

where $A_1 \in \mathbb{R}^{n_1 \times n_1}$, $A_2 \in \mathbb{R}^{n_2 \times n_2}$, $B_1 \in \mathbb{R}^{m \times n_1}$, and $B_2 \in \mathbb{R}^{m \times n_2}$. Thus, $\widetilde{\mathbf{A}} \in \mathbb{R}^{(n+m)\times(n+m)}$ is of dimension $n = n_1 + n_2$. Let $\alpha > 0$ be a parameter and denote by I the identity matrix of order $n_1 + n_2 + m$. Then, $H+\alpha I$ and $S+\alpha I$ are both nonsingular, nonsymmetric, and positive definite. Consider the two splittings of $\widetilde{\mathbf{A}}$:

$$\widetilde{\mathbf{A}} = (H + \alpha I) - (\alpha I - S), \qquad \widetilde{\mathbf{A}} = (S + \alpha I) - (\alpha I - H). \tag{2.2}$$

Associated to these splittings is the alternating iteration, $k = 0, 1, \ldots,$

$$\begin{aligned} (H + \alpha I)\mathbf{x}^{k+1/2} &= (\alpha I - S)\mathbf{x}^k + \mathbf{b}, \\ (S + \alpha I)\mathbf{x}^{k+1} &= (\alpha I - H)\mathbf{x}^{k+1/2} + \mathbf{b}. \end{aligned} \tag{2.3}$$

Eliminating $\mathbf{x}^{k+1/2}$ from these, we can rewrite (2.3) as the stationary scheme:

$$\mathbf{x}^{k+1} = T_\alpha \mathbf{x}^k + \mathbf{c}, \quad k = 0, 1, \ldots, \tag{2.4}$$

where

$$T_\alpha = (S + \alpha I)^{-1}(\alpha I - H)(H + \alpha I)^{-1}(\alpha I - S) \tag{2.5}$$

is the iteration matrix and $\mathbf{c} = 2\alpha(S+\alpha I)^{-1}(H+\alpha I)^{-1}$. The iteration matrix T_α can be rewritten as follows:

$$\begin{aligned} T_\alpha &= (S + \alpha I)^{-1}(H + \alpha I)^{-1}(\alpha I - H)(\alpha I - S) \\ &= (S + \alpha I)^{-1}(H + \alpha I)^{-1}\left[(\alpha I + H)(\alpha I + S) - 2\alpha\widetilde{\mathbf{A}}\right] \\ &= I - \left[\frac{1}{2\alpha}(H + \alpha I)(S + \alpha I)\right]^{-1}\widetilde{\mathbf{A}} \\ &= I - P_\alpha^{-1}\widetilde{\mathbf{A}}, \end{aligned} \tag{2.6}$$

where $P_\alpha = (1/2\alpha)(H + \alpha I)(S + \alpha I)$.

Obviously, P_α is nonsingular and $\mathbf{c} = P_\alpha^{-1}\mathbf{b}$. As in [18], one can show there is a unique splitting $\widetilde{\mathbf{A}} = P_\alpha - Q_\alpha$ such that the iteration T_α is the matrix induced by that splitting, that is, $T_\alpha = P_\alpha^{-1}Q_\alpha = I - P_\alpha^{-1}\widetilde{\mathbf{A}}$. Matrix Q_α is given by $Q_\alpha = (1/2\alpha)(\alpha I - H)(\alpha I - S)$.

2.2. A Relaxed Splitting Preconditioner

The relaxed splitting preconditioner is defined as follows:

$$\mathbf{M} = \begin{bmatrix} A_1 & 0 & \dfrac{1}{\alpha}A_1B_1^T \\ 0 & A_2 & B_2^T \\ -B_1 & -B_2 & \alpha I - \dfrac{1}{\alpha}B_1B_1^T \end{bmatrix}. \tag{2.7}$$

It is important to note that the preconditioner $\mathbf{M}$ can be written in a factorized form as

$$\begin{aligned} \mathbf{M} &= \frac{1}{\alpha}\begin{bmatrix} A_1 & 0 & 0 \\ 0 & \alpha I & 0 \\ -B_1 & 0 & \alpha I \end{bmatrix}\begin{bmatrix} \alpha I & 0 & B_1^T \\ 0 & A_2 & B_2^T \\ 0 & -B_2 & \alpha I \end{bmatrix} = \begin{bmatrix} A_1 & 0 & 0 \\ 0 & I & 0 \\ -B_1 & 0 & I \end{bmatrix}\begin{bmatrix} I & 0 & \dfrac{1}{\alpha}B_1^T \\ 0 & A_2 & B_2^T \\ 0 & -B_2 & \alpha I \end{bmatrix} \\ &= \begin{bmatrix} A_1 & 0 & 0 \\ 0 & I & 0 \\ -B_1 & 0 & I \end{bmatrix}\begin{bmatrix} I & 0 & \dfrac{1}{\alpha^2}B_1^T \\ 0 & I & \dfrac{1}{\alpha}B_2^T \\ 0 & 0 & I \end{bmatrix}\begin{bmatrix} I & \dfrac{1}{\alpha^2}B_1^TB_2 & 0 \\ 0 & \widehat{A}_2 & 0 \\ 0 & 0 & \alpha I \end{bmatrix}\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & -\dfrac{1}{\alpha}B_2 & I \end{bmatrix}, \end{aligned} \tag{2.8}$$

where $\widehat{A}_2 = A_2 + (1/\alpha)B_2^TB_2$. Note that both factors on the right-hand side are invertible provided that A_1 have $\widehat{A}_2$ have positive definite symmetric parts. Hence, the new preconditioner is nonsingular. This condition is satisfied for both Stokes and Oseen problems. We can see from (2.1) and (2.7) that the difference between $\mathbf{M}$ and $\widetilde{\mathbf{A}}$ is given by

$$\mathbf{R} = \mathbf{M} - \widetilde{\mathbf{A}} = \begin{bmatrix} 0 & 0 & \dfrac{1}{\alpha}A_1B_1^T - B_1^T \\ 0 & 0 & 0 \\ 0 & 0 & \alpha I - \dfrac{1}{\alpha}B_1B_1^T \end{bmatrix}. \tag{2.9}$$

This observation suggests that $\mathbf{M}$ could be a good preconditioner, since the appropriate values for the parameters involved in the new preconditioners are estimated. Furthermore, the structure of (2.9) somewhat facilitates the analysis of the eigenvalue distribution of the preconditioned matrix. In the following, we analyze the spectral properties of the preconditioned matrix $\mathbf{T} = \widetilde{\mathbf{A}}\mathbf{M}^{-1}$.

Theorem 2.1. *The preconditioned matrix $\mathbf{T} = \widetilde{\mathbf{A}}\mathbf{M}^{-1}$ has an eigenvalue 1 with multiplicity at least n, and the remaining eigenvalues are λ_i, where λ_i are the eigenvalues of an $m \times m$ matrix $Z_\alpha := (1/\alpha)(S_1 + S_2) - (1/\alpha^2)S_2S_1$ with $S_1 = B_1A_1^{-1}B_1^T$ and $S_2 = B_2\widehat{A}_2^{-1}B_2^T$.*

Proof. First of all, from $\widehat{\mathbf{T}} := \mathbf{M}^{-1}(\widetilde{\mathbf{A}}\mathbf{M}^{-1})\mathbf{M} = \mathbf{M}^{-1}\widetilde{\mathbf{A}}$ we see that the right-preconditioned matrix $\mathbf{T}$ is similar to the left-preconditioned one $\widehat{\mathbf{T}}$, then $\mathbf{T}$ and $\widehat{\mathbf{T}}$ have the same eigenvalues. Furthermore, we have

$$\begin{aligned}
\widehat{\mathbf{T}} &= \mathbf{I} - \mathbf{M}^{-1}\mathbf{R} \\
&= \mathbf{I} - \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & \frac{1}{\alpha}B_2 & I \end{bmatrix} \begin{bmatrix} I & -\frac{1}{\alpha^2}B_1^T B_2 \widehat{A}_2^{-1} & 0 \\ 0 & \widehat{A}_2^{-1} & 0 \\ 0 & 0 & \frac{1}{\alpha}I \end{bmatrix} \begin{bmatrix} I & 0 & -\frac{1}{\alpha^2}B_1^T \\ 0 & I & -\frac{1}{\alpha}B_2^T \\ 0 & 0 & I \end{bmatrix} \\
&\quad \times \begin{bmatrix} A_1^{-1} & 0 & 0 \\ 0 & I & 0 \\ B_1 A_1^{-1} & 0 & I \end{bmatrix} \begin{bmatrix} 0 & 0 & \frac{1}{\alpha}A_1 B_1^T - B_1^T \\ 0 & 0 & 0 \\ 0 & 0 & \alpha I - \frac{1}{\alpha}B_1 B_1^T \end{bmatrix} \\
&= \begin{bmatrix} I & 0 & * \\ 0 & I & * \\ 0 & 0 & \frac{1}{\alpha}(S_1 + S_2) - \frac{1}{\alpha^2}S_2 S_1 \end{bmatrix}.
\end{aligned} \tag{2.10}$$

Therefore, from (2.10) we can see that the eigenvalues of $\mathbf{T}$ are given by 1 (with multiplicity at least $n = n_1 + n_2$) and by the λ_i's. □

Lemma 2.2. *Let* $A_\alpha = \begin{pmatrix} A_1 - (1/\alpha)B_1^T B_1 & -(1/\alpha)B_1^T B_2 \\ 0 & A_2 \end{pmatrix} \in \mathbb{R}^{n\times n}$, $\alpha > \sigma_{\max}(B_1^T B_1)/\sigma_{\min}(A_1)$, *and* A_1, *and* A_2 *be positive definite. Then* A_α *is positive definite.*

Lemma 2.3. *Let* $A_\alpha \in \mathbb{R}^{n\times n}$ *and* $B \in \mathbb{R}^{m\times n}$ $(m \le n)$. *Let* $\alpha \in \mathbb{R}$, *and assume that matrices* A_α, $A_\alpha + (1/\alpha)B^T B$, $BA_\alpha^{-1}B^T$ *and* $B(A_\alpha + (1/\alpha)B^T B)^{-1}B^T$ *are all invertible. Then*

$$\left[B\left(A_\alpha + \frac{1}{\alpha}B^T B\right)^{-1} B^T\right]^{-1} = \left(BA_\alpha^{-1}B^T\right)^{-1} + \frac{1}{\alpha}I. \tag{2.11}$$

Theorem 2.4. *Let* $\alpha > \sigma_{\max}(B_1^T B_1)/\sigma_{\min}(A_1)$. *The remaining eigenvalues* λ_i *of* Z_α *are of the form:*

$$\lambda_i = \frac{\mu_i}{\alpha + \mu_i}, \tag{2.12}$$

where the μ_i*'s satisfy the eigenvalue problem:* $BA_\alpha^{-1}B^T\phi_i = \mu_i\phi_i$.

Proof. We note

$$\begin{aligned}
Z_\alpha &= \frac{1}{\alpha}(S_1 + S_2) - \frac{1}{\alpha^2}S_2 S_1 = \frac{1}{\alpha}(B_1 B_2)\begin{pmatrix} A_1^{-1} & 0 \\ -\frac{1}{\alpha}\widehat{A}_2^{-1}B_2^T B_1 A_1^{-1} & \widehat{A}_2^{-1} \end{pmatrix}\begin{pmatrix} B_1^T \\ B_2^T \end{pmatrix} \\
&= \frac{1}{\alpha}(B_1 B_2)\begin{pmatrix} A_1 & 0 \\ \frac{1}{\alpha}B_2^T B_1 & \widehat{A}_2 \end{pmatrix}^{-1}\begin{pmatrix} B_1^T \\ B_2^T \end{pmatrix}
\end{aligned}$$

$$= \frac{1}{\alpha}(B_1 B_2)\left(\begin{pmatrix} A_1 - \frac{1}{\alpha}B_1^T B_1 & -\frac{1}{\alpha}B_1^T B_2 \\ 0 & A_2 \end{pmatrix} + \begin{pmatrix} \frac{1}{\alpha}B_1^T B_1 & \frac{1}{\alpha}B_1^T B_2 \\ \frac{1}{\alpha}B_2^T B_1 & \frac{1}{\alpha}B_2^T B_2 \end{pmatrix}\right)^{-1}\begin{pmatrix} B_1^T \\ B_2^T \end{pmatrix}$$

$$= \frac{1}{\alpha}B\left(A_\alpha + \frac{1}{\alpha}B^T B\right)^{-1}B^T. \tag{2.13}$$

Thus, the remaining eigenvalues are the solutions of the eigenproblem:

$$\frac{1}{\alpha}B\left(A_\alpha + \frac{1}{\alpha}B^T B\right)^{-1}B^T\phi_i = \lambda_i\phi_i. \tag{2.14}$$

By Lemma 2.3, we obtain

$$\frac{1}{\alpha}\phi_i = \lambda_i\left(B\left(A_\alpha + \frac{1}{\alpha}B^T B\right)^{-1}B^T\right)^{-1}\phi_i = \lambda_i\left(BA_\alpha^{-1}B^T\right)^{-1}\phi_i + \frac{\lambda_i}{\alpha}\phi_i. \tag{2.15}$$

Hence, $\lambda_i = \mu_i/(\alpha + \mu_i)$, where μ_i's satisfy the eigenvalue problem $BA_\alpha^{-1}B^T\phi_i = \mu_i\phi_i$. □

In addition, we obtain easily that the remaining eigenvalues $\lambda_i \to 0$ as $\alpha \to \infty$. Figures 1 and 2 show this behavior, that is, the nonunity eigenvalues of the preconditioned matrix are increasingly clustered at the origin as the parameters become larger.

2.3. Practical Implementation of the Relaxed Splitting Preconditioner

In this subsection, we outline the practical implementation of the relaxed splitting preconditioner in a subspace iterative method. The main step is applying the preconditioner, that is, solving linear systems with the coefficient matrix $\mathbf{M}$. From (2.8), we can see that the relaxed splitting preconditioner can be factorized as follows:

$$\mathbf{M} = \begin{bmatrix} A_1 & 0 & 0 \\ 0 & I & 0 \\ -B_1 & 0 & I \end{bmatrix}\begin{bmatrix} I & 0 & \frac{1}{\alpha^2}B_1^T \\ 0 & I & \frac{1}{\alpha}B_2^T \\ 0 & 0 & I \end{bmatrix}\begin{bmatrix} I & \frac{1}{\alpha^2}B_1^T B_2 & 0 \\ 0 & \widehat{A}_2 & 0 \\ 0 & 0 & \alpha I \end{bmatrix}\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & -\frac{1}{\alpha}B_2 & I \end{bmatrix}, \tag{2.16}$$

showing that the preconditioner requires solving two linear systems at each step, with coefficient matrices A_1 and $\widehat{A}_2 = A_2 + (1/\alpha)B_2^T B_2$. Several different approaches are available for solving linear systems involving A_1 and $\widehat{A}_2$. We defer the discussion of these to Section 3.

We conclude this section with a discussion of diagonal scaling. We found that scaling can be beneficial for the relaxed splitting preconditioner. Unless otherwise specified, we perform a preliminary symmetric scaling of the linear systems $\tilde{\mathbf{A}}\mathbf{x} = \mathbf{b}$ in the form $\mathbf{D}^{-1/2}\tilde{\mathbf{A}}\mathbf{D}^{-1/2}\mathbf{y} = \mathbf{D}^{-1/2}\mathbf{b}$ with $\mathbf{y} = \mathbf{D}^{1/2}\mathbf{x}$, and $\mathbf{D} = \mathrm{diag}(D_1, D_2, I)$, where $\mathrm{diag}(D_1, D_2)$ is the main diagonal of the velocity submatrix $\mathbf{A}$. Incidentally, it is noted that diagonal scaling is very beneficial for the HSS preconditioner (see [13]) and the DS preconditioner (see [16, 17]).

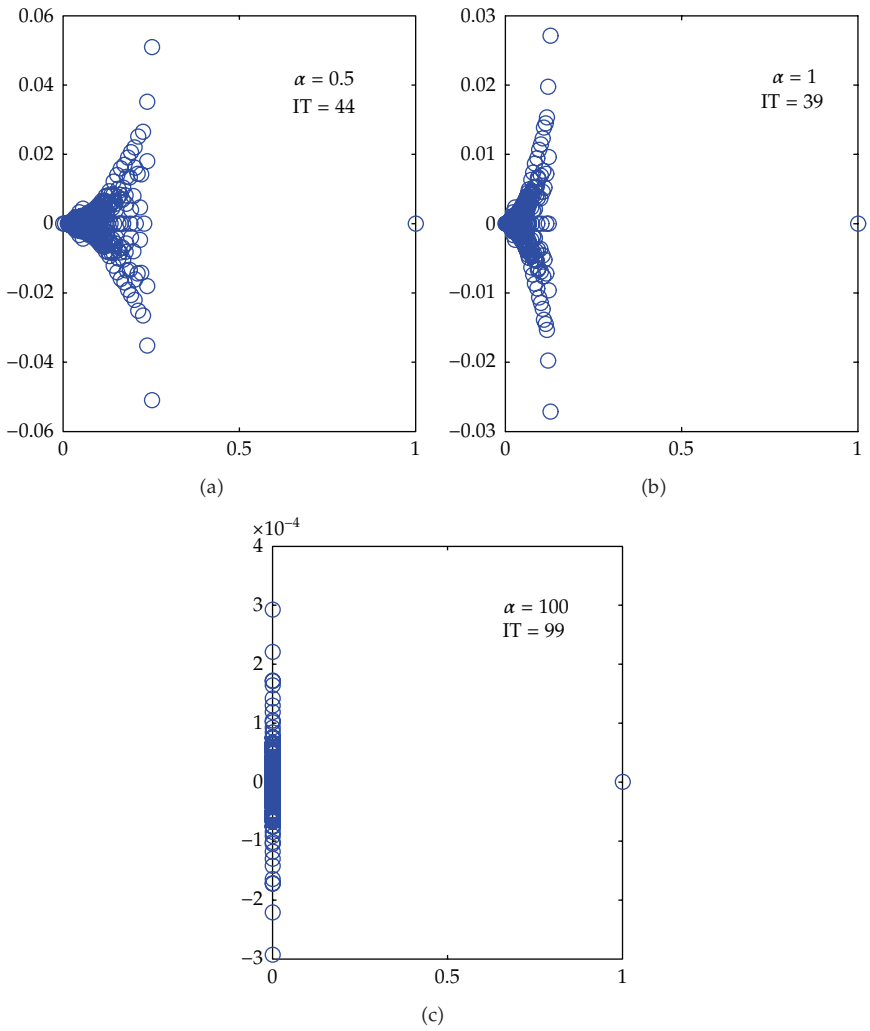


Figure 1: Spectrum of preconditioned steady Oseen matrix, 32 × 32 grid with $\upsilon = 0.1$.

3. Numerical Experiments

In this section, numerical experiments are carried out for solving the linear system coming from the finite-element discretization of the two-dimensional linearized Stokes and Oseen models of incompressible flow in order to verify the performance of our preconditioner. The test problem is the leaky lid-driven cavity problem generated by the IFISS software package [19]. We used a zero initial guess and stopped the iteration when $||\mathbf{r}_k||_2/||\mathbf{b}||_2 \leq 10^{-6}$, where $\mathbf{r}_k$ is the residual vector. The relaxed splitting preconditioner is combined with restarted GMRES(m). We set $m = 30$.

We consider the 2D leaky lid-driven cavity problem discretized by the finite-element method on uniform grids [1]. The subproblems arising from the application of the relaxed splitting preconditioner are solved by direct methods. We use AMD reordering technique [20, 21] for the degrees of freedom that makes the application of the Cholesky (for Stokes) or

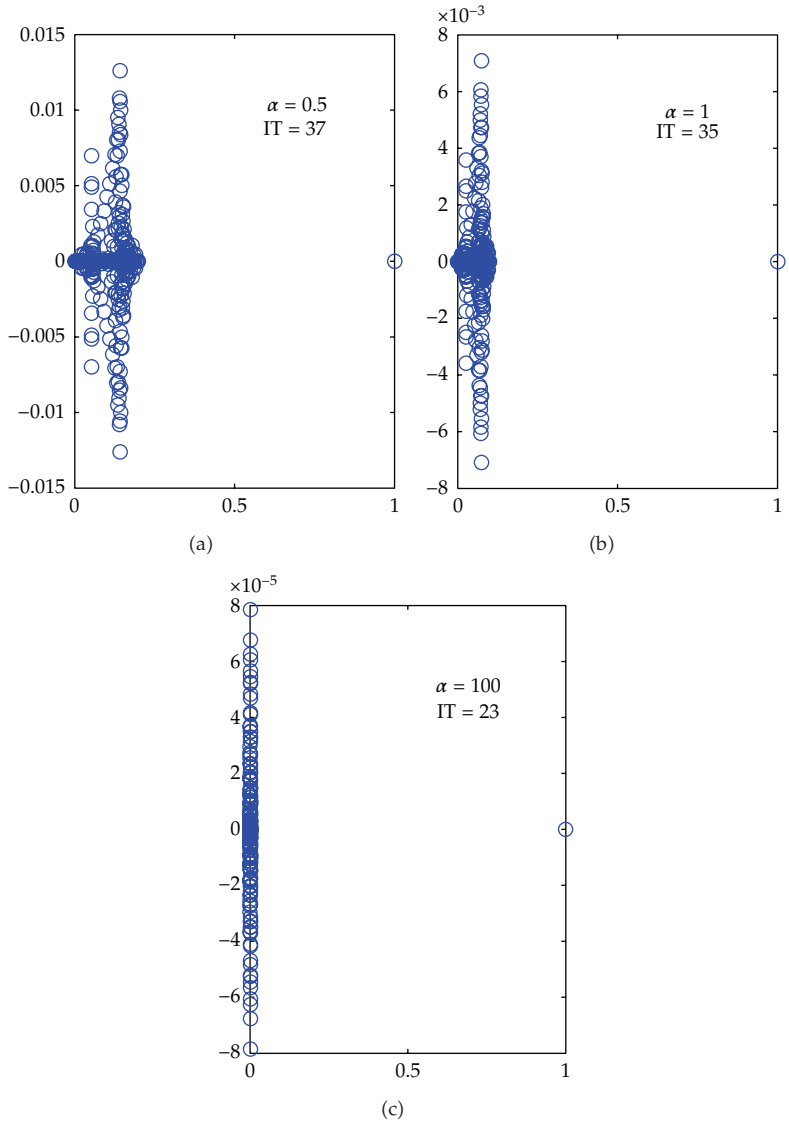


Figure 2: Spectrum of preconditioned generalized steady Oseen matrix, 32 × 32 grid with $\upsilon = 0.001$.

Table 1: Iterations of preconditioned GMRES(30) for steady Stokes problem.

Grid	Q2-Q1	Q2-P1
16 × 16	25	15
32 × 32	26	12
64 × 64	23	10
128 × 128	19	11
256 × 256	16	10

Table 2: Iterations of preconditioned GMRES(30) for steady Oseen problems (Picard).

Grid	$\upsilon = 1$		$\upsilon = 0.1$		$\upsilon = 0.02$	
	Q2-Q1	Q2-P1	Q2-Q1	Q2-P1	Q2-Q1	Q2-P1
16 × 16	27	14	29	16	48	27
32 × 32	27	12	29	14	54	26
64 × 64	23	10	26	12	49	24
128 × 128	19	11	21	10	37	20
256 × 256	16	10	10	8	17	14

Table 3: Iterations of preconditioned GMRES(30) for steady Oseen problems (Newton).

Grid	$\upsilon = 1$		$\upsilon = 0.1$		$\upsilon = 0.02$	
	Q2-Q1	Q2-P1	Q2-Q1	Q2-P1	Q2-Q1	Q2-P1
16 × 16	27	15	29	16	46	26
32 × 32	27	12	29	14	53	24
64 × 64	23	10	26	12	46	23
128 × 128	19	11	21	10	34	18
256 × 256	16	10	10	8	15	13

LU (for Oseen) factorization of A_1 and $\widehat{A}_2$ relatively fast. For simplicity, we use $\alpha = 100$ for all numerical experiments.

In Table 1, we show iteration counts (referred to as "its") for the relaxed splitting preconditioned GMRES(30) when solving the steady Stokes problem on a sequence of uniform grids. We see that the iteration count is independent of mesh size involved in the Q2-Q1 and the Q2-P1 finite-element scheme. The Q2-P1 finite-element scheme has much better profile than the Q2-Q1 finite-element scheme.

In Tables 2 and 3, we show iteration counts for the steady Oseen problem on a sequence of uniform grids and for different values of υ, using Picard and Newton linearization of generalized Oseen problems, respectively. We found that the relaxed splitting preconditioner has difficulties dealing with low-viscosity, that is, the number of iterations increases with the decrease in the kinematic viscosity. In this case, it appears that the Q2-P1 finite-element scheme gives faster convergence results than the Q2-Q1 finite-element scheme.

Next, we report on analogous experiments involving the generalized Stokes problem and the generalized Oseen problem. As we can see from Table 4, for the generalized Stokes problem, the results are virtually the same as those obtained in the steady case. Indeed, we can see from the results in Table 1 that the rate of convergence for the relaxed splitting preconditioned GMRES (30) is essentially independent of mesh size involved in the Q2-Q1 and the Q2-P1 finite-element schemes.

In Tables 5 and 6, for generalized Oseen problems, we compare our preconditioner with the RDF preconditioner in [17]. The RDF preconditioner can be factorized as follows:

$$P = \begin{bmatrix} I & 0 & \dfrac{B_1^T}{\alpha} \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \widehat{A}_1 & 0 & 0 \\ 0 & I & 0 \\ -B_1 & 0 & I \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & \widehat{A}_2 & B_2^T \\ 0 & 0 & \alpha I \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & \dfrac{-B_2}{\alpha} & I \end{bmatrix}, \tag{3.1}$$

Table 4: Iterations of preconditioned GMRES(30) for generalized Stokes problems.

Grid	Q2-Q1	Q2-P1
16 × 16	25	13
32 × 32	26	12
64 × 64	23	10
128 × 128	19	11
256 × 256	16	10

Table 5: Iterations of preconditioned GMRES(30) for generalized Oseen problem (Picard, Q2-Q1, 128 × 128 uniform grids).

Viscosity	RDF		New preconditioner	
	its	CPU	its	CPU
0.1	12	22.75374	13	15.90265
0.01	7	20.93871	9	14.30008
0.001	5	20.31654	6	13.36891

Table 6: Iterations of preconditioned GMRES(30) for generalized Oseen problem (Newton, Q2-Q1, 128 × 128 uniform grids).

Viscosity	RDF		New preconditioner	
	its	CPU	its	CPU
0.1	12	22.86654	13	16.12583
0.01	7	21.02928	8	14.34972
0.001	12	22.81674	16	17.49554

where $\widehat{A}_1 = A_1 + (1/\alpha)B_1^T B_1$ and $\widehat{A}_2 = A_2 + (1/\alpha)B_2^T B_2$. It shows that RDF preconditioner requires solving two linear systems at each step. The new preconditioner requires solving linear systems with A_1 and $\widehat{A}_2$ at each step. We can see that the linear system with A_1 is easier to solve than that with $\widehat{A}_1$. From Tables 5 and 6, we can see for 128 × 128 grid with different viscosities that the RDF preconditioner leads to slightly less iteration counts than the new preconditioner, but the new preconditioner is slightly faster in terms of elapsed CPU time.

From Figures 3 and 4, we found that for the relaxed splitting preconditioner the intervals containing values of parameter α are very wide. Those imply that the relaxed splitting preconditioner is not sensitive to the value of parameter. Noting that the optimal parameters of the relaxed splitting preconditioner are always larger than 50, we can always take $\alpha = 100$ to obtain essentially optimal results.

4. Conclusions

In this paper, we have described a relaxed splitting preconditioner for the linear systems arising from discretizations of the Navier-Stokes equations and analyzed the spectral properties of the preconditioned matrix. The numerical experiments show good performance on a wide range of cases. We use direct methods for the solution of inner linear systems, but it is not a good idea to solve larger 2D or 3D problems at the constraint of memory and

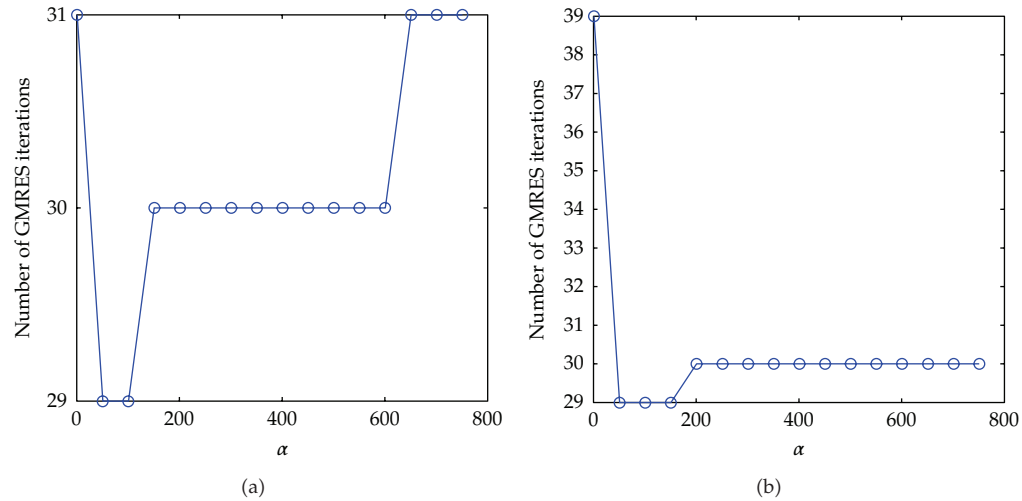


Figure 3: Iteration number versus parameter, steady Oseen problem, with $\upsilon = 0.1$. (a) 16×16 grid, (b) 32×32 grid.

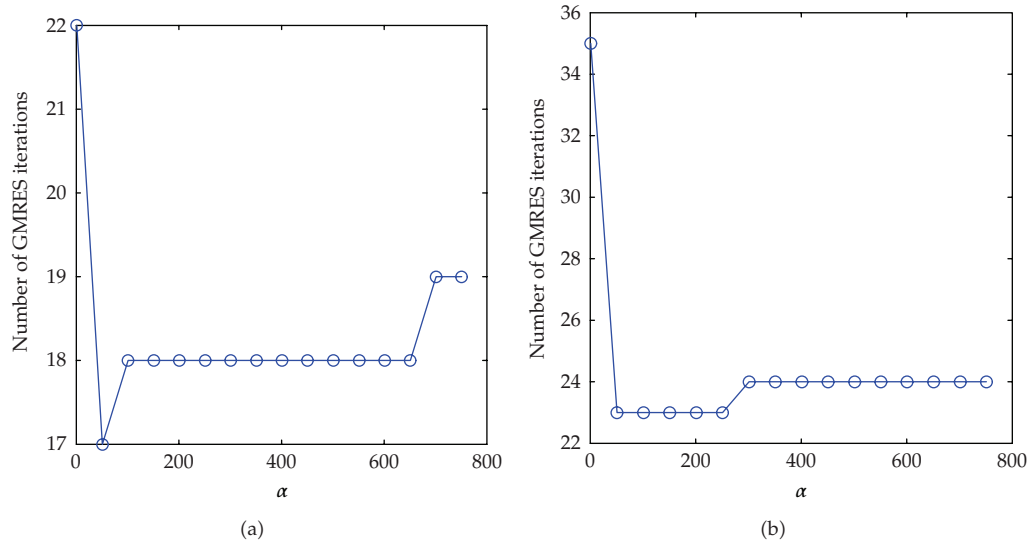


Figure 4: Iteration number versus parameter, generalized Oseen problem, with $\upsilon = 0.001$. (a) 16×16 grid, (b) 32×32 grid.

time requirement. In this case, exact solve can be replaced with inexact solve, which requires further research in the future.

Acknowledgments

This research is supported by NSFC (60973015 and 61170311), Chinese Universities Specialized Research Fund for the Doctoral Program (20110185110020), and Sichuan Province Sci. & Tech. Research Project (12ZC1802).

References

[1] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, NY, USA, 2005.

[2] R. Glowinski, "Finite element methods for incompressible viscous flow," in *Handbook of Numerical Analysis*, P. G. Ciarlet and J. L. Lions, Eds., vol. 9, North-Holland, Amsterdam, The Netherlands, 2003, Numerical Methods for Fluids (part3).

[3] M. Benzi, G. H. Golub, and J. Liesen, "Numerical solution of saddle point problems," *Acta Numerica*, vol. 14, pp. 1–137, 2005.

[4] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2nd edition, 2003.

[5] H. C. Elman, V. E. Howle, J. Shadid, D. Silvester, and R. Tuminaro, "Least squares preconditioners for stabilized discretizations of the Navier-Stokes equations," *SIAM Journal on Scientific Computing*, vol. 30, no. 1, pp. 290–311, 2007.

[6] H. C. Elman and R. S. Tuminaro, "Boundary conditions in approximate commutator preconditioners for the Navier-Stokes equations," *Electronic Transactions on Numerical Analysis*, vol. 35, pp. 257–280, 2009.

[7] M. A. Olshanskii and Y. V. Vassilevski, "Pressure schur complement preconditioners for the discrete Oseen problem," *SIAM Journal on Scientific Computing*, vol. 29, no. 6, pp. 2686–2704, 2007.

[8] M. Benzi and M. A. Olshanskii, "An augmented Lagrangian-based approach to the oseen problem," *SIAM Journal on Scientific Computing*, vol. 28, no. 6, pp. 2095–2113, 2006.

[9] M. Benzi, M. A. Olshanskii, and Z. Wang, "Modified augmented Lagrangian preconditioners for the incompressible Navier-Stokes equations," *International Journal for Numerical Methods in Fluids*, vol. 66, no. 4, pp. 486–508, 2011.

[10] S. P. Hamilton, M. Benzi, and E. Haber, "New multigrid smoothers for the Oseen problem," *Numerical Linear Algebra with Applications*, vol. 17, no. 2-3, pp. 557–576, 2010.

[11] M. Benzi and Z. Wang, "Analysis of augmented Lagrangian-based preconditioners for the steady incompressible Navier-Stokes equations," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2761–2784, 2011.

[12] Z. Z. Bai, G. H. Golub, and M. K. Ng, "Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems," *SIAM Journal on Matrix Analysis and Applications*, vol. 24, no. 3, pp. 603–626, 2003.

[13] M. Benzi and G. H. Golub, "A preconditioner for generalized saddle point problems," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 1, pp. 20–41, 2004.

[14] M. Benzi and J. Liu, "An efficient solver for the incompressible Navier-Stokes equations in rotation form," *SIAM Journal on Scientific Computing*, vol. 29, no. 5, pp. 1959–1981, 2007.

[15] L. Chan, M. K. Ng, and N. K. Tsing, "Spectral analysis for HSS preconditioners," *Numerical Mathematics*, vol. 1, no. 1, pp. 57–77, 2008.

[16] M. Benzi and X.-P. Guo, "A dimensional split preconditioner for stokes and linearized Navier-Stokes equations," *Applied Numerical Mathematics*, vol. 61, no. 1, pp. 66–76, 2011.

[17] M. Benzi, M. Ng, Q. Niu, and Z. Wang, "A relaxed dimensional factorization preconditioner for the incompressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 230, no. 16, pp. 6185–6202, 2011.

[18] M. Benzi and D. B. Szyld, "Existence and uniqueness of splittings for stationary iterative methods with applications to alternating methods," *Numerische Mathematik*, vol. 76, no. 3, pp. 309–321, 1997.

[19] H. C. Elman, A. Ramage, and D. J. Silvester, "Algorithm 886: IFISS, a Matlab toolbox for modelling incompressible flow," *Association for Computing Machinery*, vol. 33, no. 2, article 14, 2007.

[20] P. R. Amestoy, T. A. Davis, and I. S. Duff, "An approximate minimum degree ordering algorithm," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 4, pp. 886–905, 1996.

[21] T. A. Davis, *Direct Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2006.

Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 894074, 15 pages
doi:10.1155/2012/894074

Research Article

A Parallel Wavelet-Based Algebraic Multigrid Black-Box Solver and Preconditioner

Fabio Henrique Pereira[1] and Sílvio Ikuyo Nabeta[2]

[1] *Industrial Engineering Post Graduation Program, Nove de Julho University (PMEP/UNINOVE), Francisco Matarazzo Avenue, 612, 05001100 São Paulo, SP, Brazil*
[2] *Electrical Machine and Drives Lab, São Paulo University (GMAcq/EP/USP), Luciano Gualberto Avenue, 380, 05508-010 São Paulo, SP, Brazil*

Correspondence should be addressed to Fabio Henrique Pereira, fabiohp@uninove.br

Received 1 November 2011; Revised 26 January 2012; Accepted 8 February 2012

Academic Editor: Massimiliano Ferronato

This work introduces a new parallel wavelet-based algorithm for algebraic multigrid method (PWAMG) using a variation of the standard parallel implementation of discrete wavelet transforms. This new approach eliminates the grid coarsening process in traditional algebraic multigrid setup phase simplifying its implementation on distributed memory machines. The PWAMG method is used as a parallel black-box solver and as a preconditioner in some linear equations systems resulting from circuit simulations and 3D finite elements electromagnetic problems. The numerical results evaluate the efficiency of the new approach as a standalone solver and as preconditioner for the biconjugate gradient stabilized iterative method.

1. Introduction

The algebraic multigrid (AMG) method is one of the most efficient algorithms for solving large sparse linear systems. Especially in the context of large-scale problems and massively parallel computing, the most desirable property of AMG is its potential for algorithmic scalability: in the ideal case, for a matrix problem with n unknowns, the number of iterative V-cycles required for convergence is independent of the problem size n and the work in the setup phase and in each V-cycle is linearly proportional to the problem size n [1, 2]. For all this, the need to solve linear systems arising from problems posed on extremely large, unstructured grids has been generating great interest in parallelizing AMG.

However, there are two major problems: first, the core of the AMG setup phase includes the grid coarsening process, which is inherently sequential in nature [1–3]. This coarsening scheme, for traditional AMG, can lead to computational complexity growth as

the problem size increases, resulting in an elevated memory use and execution time and in a reduced scalability [4, 5]. Second, most parallel AMG algorithms are based on domain decomposition ideas, which have been proved to be very efficient but require a hierarchy of meshes that eliminates the algebraic characteristic of AMG and precludes its use as a black-box method.

Due to those difficulties and the importance of the development of efficient parallel preconditioners for large, sparse systems of linear equations, the investigation of new parallel approaches has been the main subject of many researchers [6–12]. In this context, a great amount of work has been aimed to extract some parallelism from serial preconditioners such as factorization-based methods, which provide effective preconditioning on sequential architectures [8–10]. However, scalable parallel implementation of incomplete factorization preconditioners presents many limitations and challenges, and although some interesting approaches have presented good performance for certain classes of problems, quite scalable parallel algorithms for this kind of preconditioners seem to have not been available [8].

Also has received much attention in the last years the development of preconditioning approaches that have inherently parallel characteristics. In particular, approximate inverse preconditioners have proven extremely promising and effective for the solution of general sparse linear systems of equations [6, 7, 10–12]. Unfortunately, this kind of preconditioner also has some drawbacks: in general, as the discretization is refined the amount of work per grid point grows with problem size, and it is inherently difficult to approximate the inverse of very ill-conditioned linear systems with a sparse matrix [13].

In this work we introduce a new parallel algorithm for wavelet-based AMG (PWAMG) using a variation of the parallel implementation of discrete wavelet transforms. This new approach eliminates the grid coarsening process present at standard AMG setup phase, simplifying significantly the implementation on distributed memory machines and allowing the use of PWAMG as a parallel black-box solver and preconditioner. The parallel algorithm uses the message passing interface (MPI) that provides a standard for message passing for parallel computers and workstation clusters.

A sequential version of WAMG was introduced recently [14], and it has revealed to be a very efficient and promising method for several problems related to the computation of electromagnetic fields [15, 16]. Here, the method is used as a parallel black-box solver and as a preconditioner in some linear equations systems resulting from circuit simulations and 3D finite elements electromagnetic problems. The numerical results evaluate the efficiency of the new approach as a standalone solver and as preconditioner for the biconjugate gradient stabilized iterative method.

2. The Discrete Wavelet Transform

The discrete wavelet transform (DWT) corresponds to the application of low-pass and high-pass filters, followed by the elimination of one out of two samples (decimation or subsampling). The discrete signal, which in one dimension is represented by a vector of values, is filtered by a set of digital filters that are associated to the wavelet adopted in the analysis.

Starting from a vector $y(N)$ at level 0, two sets of coefficients are generated in each level l of the process: a set d_l of wavelets coefficients (detail coefficients) and a set c_l of approximation coefficients. This procedure can be applied again, now using c_l as an input vector to create new coefficients c_{l+1} and d_{l+1}, successively.

Very important classes of filters are those of finite impulse response (FIR). The main characteristic of these filters is the convenient time-localization properties. These filters are originated from compact support wavelets, and they are calculated analytically. An example of FIR filters is the length-2 scaling filter with Haar or Daubechies-2 coefficients, which are given by (2.1):

$$h_{D2} = [h_0, h_1] = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right]. \tag{2.1}$$

For more details about compact FIR filters see [17].

For a given vector $y = (y_1, y_2, \ldots, y_N)$, the Haar wavelet transform creates an approximation vector c and a detail vector d according to (2.2) and (2.3), respectively:

$$c = (c_1, c_2, \ldots, c_{N/2}), \quad \text{with } c_i = \frac{(y_{2i-1} + y_{2i})}{\sqrt{2}}, \tag{2.2}$$

$$d = (d_1, d_2, \ldots, d_{N/2}), \quad \text{with } d_i = \frac{(y_{2i-1} - y_{2i})}{\sqrt{2}}. \tag{2.3}$$

It is not difficult to see that this procedure will work only if N is even.

In the 2D case, in which the discrete signal is represented by a matrix, the DWT is obtained through the application of successive steps of 1D transform into the rows and columns of the matrix. This process generates a matrix formed by four types of coefficients: the approximation coefficients and the detail coefficients (horizontal, vertical, and diagonal), as illustrated in Figure 1. Blocks H and G represent, respectively, the low-pass and high-pass filters.

In both cases, the approximation coefficients keep the most important information of the discrete signal, whereas the detail coefficients possess very small values, next to zero. These approximation coefficients will contain low-pass information, which is essentially a low-resolution version of the signal and represent a coarse version of the original data.

3. A Wavelet-Based Algebraic Multigrid

The approximation property of wavelet is explored by wavelet-based algebraic multigrid for creating the hierarchy of matrices. The method considers the use of a modified discrete wavelet transform in the construction of the transfer operators and the hierarchy of matrices in the multigrid approach.

A two-dimensional modified DWT is applied to produce an approximation of the matrix in each level of the wavelets multiresolution decomposition process. An operator formed only by low-pass filters is created, which is applied to the rows and columns of the matrix. This same operator is used for the intergrid transfer in the AMG. If a length-2 (first order) scaling filter is used, for example, which means the application of the operation

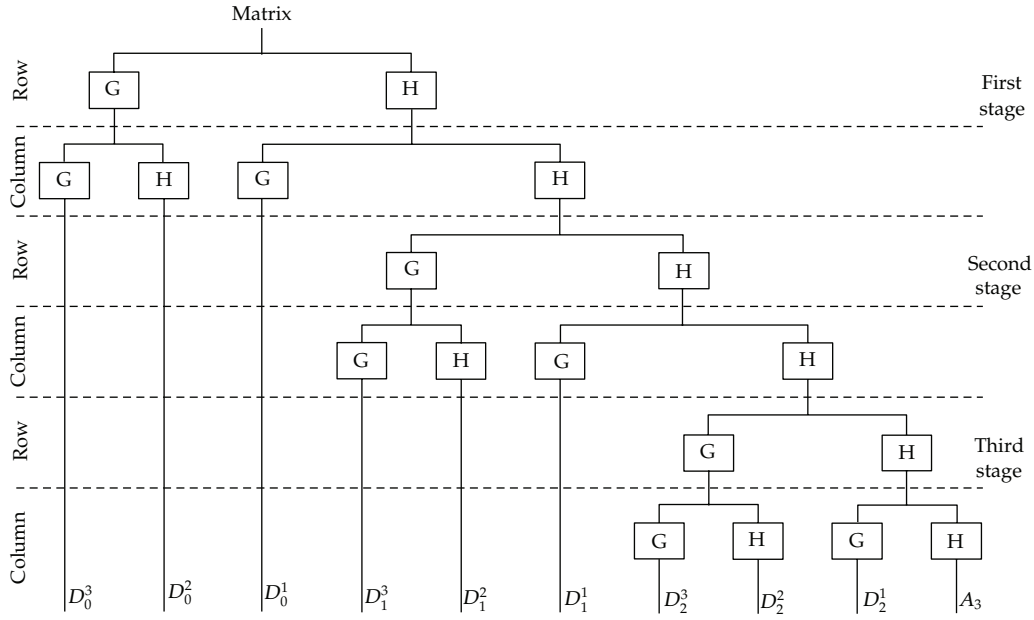


Figure 1: The two-dimensional DWT: one-dimensional transform in the rows and columns of the matrix.

defined in (2.2) in the rows and columns of the matrix, the operation is matricially defined as (3.1):

$$P_k^{k+1} = \begin{pmatrix} h_1 & h_0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & h_1 & h_0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & & \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & \cdots & \cdots & 0 & h_1 & h_0 \end{pmatrix}_{(N/2)\times N}. \tag{3.1}$$

The prolongation operator is defined in the usual form,

$$P_{k+1}^k = \left(P_k^{k+1}\right)^T, \tag{3.2}$$

and the matrix in the corresponding level k with the Galerkin condition:

$$A^{k+1} = P_k^{k+1} A^k P_{k+1}^k, \tag{3.3}$$

reminding that $A^0 = A$ is the matrix of the original system.

Once the intergrid transfer operators and the hierarchy of matrices are created, the multigrid method (V-cycle) can be defined as usual, using a recursive call of the following two-level algorithm.

Algorithm 3.1 (Two-level multigrid). The following holds.

Input: the right hand side vector b, the original matrix A
and the coarse grid matrix PAP^T

Output: approximation $\tilde{x}$

(1) Choose an initial guess x and apply ν_1 smoothing steps in $Ax = b$
(2) Compute $r = b - Ax$
(3) $e = (\mathrm{PAP}^T)^{-1} Pr$
(4) $\tilde{x} = x + P^{\mathrm{T}} e$
(5) Apply ν_2 smoothing steps in $A\tilde{x} = b$
(6) Return $\tilde{x}$.

In this paper, a V-cycle multigrid with $\nu_1 = \nu_2 = 1$ is applied iteratively as a solver and also as a preconditioner inside the biconjugate gradient stabilized (BiCGStab) iterations.

3.1. The Choice of the Wavelet Filters

A common problem on the choice of the filters is to decide between the fill-in control and the wavelet properties. As the WAMG often deals with sparse matrices, the control of the nonzero number is a very important task. In this case, if the matrix A^k is sparse, then the number of nonzero elements in the next level matrix $A^{k+1} = P_k^{k+1} A^k P_{k+1}^k$ will depend on the order of the filter used in the restriction and prolongation operators. In fact, the longer the filter used the larger the number of nonzero entries in the next computed matrix. Consequently, most of the wavelet-based multigrid methods use shorter filters such as Haar or Daubechies-2 coefficients in its approaches [18–23]. This is also the case in this paper.

4. A Parallel Wavelet-Based Algebraic Multigrid

One advantage of the proposed approach is the elimination of the coarsening scheme from the setup phase. The application of the operations defined in (2.3) in the rows and columns of a matrix allows creating an approximation without any information about meshes or the adjacency graph of the matrix. Thus the implementation on distributed memory machines becomes simpler allowing the use of the method as a parallel black-box solver and preconditioner. Our approach, based on the parallel implementation of discrete wavelet transform presented in [24], avoids any communication among processors in the setup phase if first-order filters are used.

The parallelization strategy starts dividing equally the number N of rows of the matrix between the processors, in such a way that the 1-D transform defined in (2.3) could be applied entirely locally in the rows. As each matrix column is equally divided, the transformation in this direction is accomplished partially in each processing unit, considering the size of the column in its local memory. For np processors, for example, each processor will apply the wavelet transform in the $\lceil N/np \rceil$ elements of the column in its memory, where $\lceil x \rceil$ means the largest integer even less than x. If the largest integer less than N/np is odd, then the last element of the column in the local memory is unchanged, as illustrated in Figure 2 for $N = 10$ and $np = 2$. In the figure, $c_{i,j}^l$ means the ith local element of the level l vector c in the processor j.

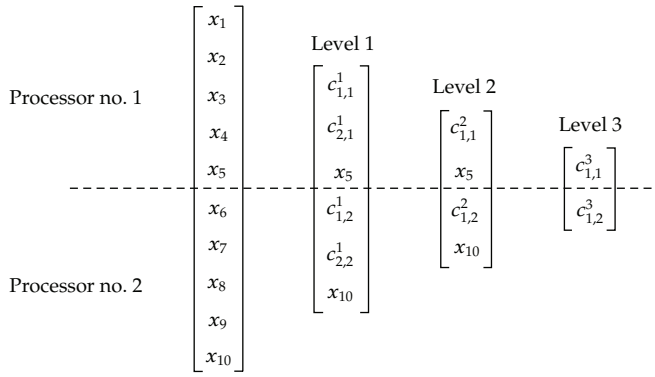


Figure 2: Illustration of the 1D parallel wavelet transform with 2 processors.

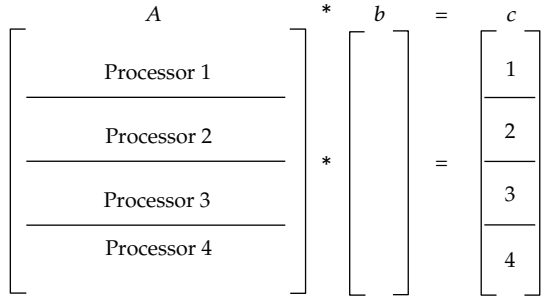


Figure 3: Matrix-vector product of a matrix distributed by rows.

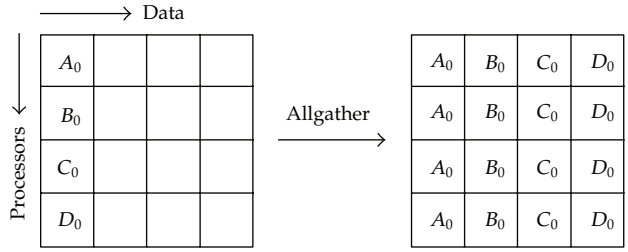


Figure 4: MPI collective communication function.

Thus the resulting coarse level matrix is calculated entirely locally.

In the solver phase the interprocessors communication is necessary only for operations involving matrices. More specifically, it is necessary to update the vector always after the smoothing step and after the matrix-vector product in the residual calculation (lines 1, 2, and 5 of the algorithm). In a matrix-vector product $A * b = c$, for example, the matrix A is distributed in rows, the vector b is shared by all processors, and vector c is calculated in parallel as illustrated in Figure 3, for 4 processors. Then the resulting vector c is updated by the processors through the message passing interface library (MPI). This task is accomplished by using the MPI collective communication function `MPI_Allgather` [25]. The `MPI_Allgather` function effect is shown in Figure 4. It is important to highlight that only the resulting vector should be updated. It means each processor communicates to the other only a few elements

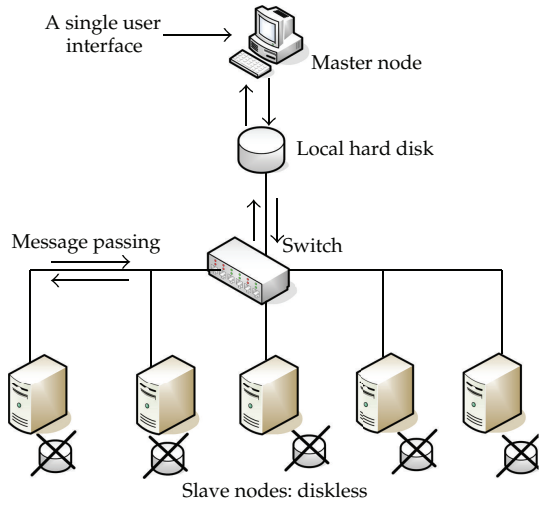


Figure 5: The Beowulf Linux cluster architecture.

Table 1: Electromagnetic matrices properties.

Matrix properties	*2cubes_sphere*	*Offshore*	*dielFilterV2real*	*Circuit5M_dc*
Number of rows	101,492	259,789	1,157,456	3,523,317
Number of columns	101,492	259,789	1,157,456	3,523,317
Nonzeros	1,647,264	4,242,673	48,538,952	14,865,409
Explicit zero entries	0	0	0	4,328,784
Type	Real	Real	Real	Real
Structure	Symmetric	Symmetric	Symmetric	Unsymmetric
Positive definite?	Yes	Yes	No	No

Table 2: Results for sequential tests (not enough memory).

Matrix	ILU + BiCGStab			WAMG			WAMG + BiCGStab		
	Setup	Solver	n	Setup	Solver	n	Setup	Solver	n
Circuit5M_dc	6.96	29.98	3	47.4	14.37	3	61.30	45.49	2
2cubes_sphere	1.98	2.15	3	5.29	2.02	3	5.30	1.55	4
Offshore	5.36	21.55	10	16.67	18.42	6	16.69	21.17	10
dielFilter-V2real	—	—	—	—	—	—	—	—	—

that it stores. However, in order for the process to continue, the whole vector must be updated on each processor and some kind of collective communication should take place.

The PWAMG method in this work uses a hybrid Jacobi-Gauss method as smoother and the V-cycle for the resolution scheme. The Gauss-Seidel smoothing is applied inside each processor and the Jacobi method applied interprocessors.

5. The Numerical Test Problems

The parallel algorithm uses the version one of the MPI that provides a standard for message passing for parallel computers and workstation clusters. The method has been implemented

Table 3: Parallel results for *2cubes_sphere* matrix.

	np	Nrows	Nonzeros	PWAMG solver			PWAMG BiCGStab		
				Setup time CPU (Wtime)	Solver time CPU (Wtime)	n	Setup time CPU (Wtime)	Solver time CPU (Wtime)	n
2cubes_sphere	1	101492	1647264	5.29	2.02	3	5.30	1.55	4
	2	50746	828332	2.43 (2.43)	1.84 (2.99)	3	2.40 (2.40)	6.81 (9.83)	7
		50746	818932	2.20 (2.20)	2.22 (2.99)		2.17 (2.17)	5.74 (9.83)	
	3	33830	552260	1.61 (1.62)	1.41 (3.46)	3	1.62 (1.62)	5.50 (12.09)	7
		33830	550206	1.50 (1.50)	1.15 (3.47)		1.49 (1.49)	3.63 (12.09)	
		33832	544798	1.42 (1.42)	1.74 (3.47)		1.41 (1.41)	4.36 (12.08)	
	4	25373	415150	1.19 (1.20)	1.12 (3.26)	3	1.19 (1.19)	4.18 (11.48)	7
		25373	412416	1.12 (1.12)	0.96 (3.22)		1.12 (1.12)	3.83 (11.48)	
		25373	406516	1.07 (1.07)	1.01 (3.23)		1.06 (1.06)	3.27 (11.46)	
		25373	413182	1.06 (1.06)	1.18 (3.26)		1.06 (1.06)	3.22 (11.46)	
	5	20298	333205	0.95 (0.95)	1.00 (3.29)	3	0.96 (0.94)	4.27 (11.87)	7
		20300	330383	0.88 (0.88)	0.77 (3.29)		0.88 (0.88)	3.35 (11.85)	
		20298	330868	0.88 (0.88)	0.78 (3.30)		0.87 (0.87)	2.70 (11.85)	
		20298	328976	0.84 (0.83)	0.75 (3.31)		0.85 (0.84)	2.68 (11.87)	
		20298	323832	0.85 (0.85)	1.08 (3.32)		0.84 (0.84)	2.59 (11.85)	
	6	16915	277472	0.81 (0.82)	0.87 (3.27)	3	0.81 (0.81)	4.28 (11.95)	7
		16915	276059	0.74 (0.74)	0.69 (3.24)		0.74 (0.74)	3.15 (11.94)	
		16915	274147	0.73 (0.74)	0.71 (3.24)		0.74 (0.74)	2.40 (11.93)	
		16915	269896	0.73 (0.73)	0.72 (3.24)		0.74 (0.74)	2.32 (11.94)	
		16915	274788	0.69 (0.68)	0.68 (3.24)		0.72 (0.72)	2.32 (11.94)	
		16917	274902	0.72 (0.72)	1.03 (3.26)		0.68 (0.68)	2.32 (11.93)	

using C++ and tested in a homogeneous Beowulf cluster with 6 machine nodes (Core 2 Duo, 1 GB RAM) connected to a switched fast Ethernet network, as illustrated in Figure 5.

The multigrid V-cycle approach was applied as a resolution scheme to solve some linear equations systems resulting from circuit simulations and finite elements electromagnetic problems. The parallel method was also applied as a preconditioner for the iterative biconjugate gradient stabilized (BiCGStab) method, which has been implemented using the same vector updating approach.

The three first matrices are related to 3D finite element electromagnetic analysis. The matrices *2cubes_sphere* and *offshore* are from a study of edge-based finite-element time domain solvers for 3D electromagnetic diffusion equations. The *dielFilterV2real* is a real symmetric matrix, which comes from a high-order vector finite element analysis of a 4th-pole dielectric resonator [26]. The last real unsymmetric matrix is from circuit simulation problems from Tim Davis sparse matrix collection [27]. The matrices properties are presented in Table 1.

6. Results

The problems were solved firstly using the sequential version of the proposed method. For comparison, the BiCGStab method preconditioned by the incomplete LU was used.

Table 4: Parallel results for *offshore* matrix.

	np	Nrows	Nonzeros	PWAMG solver			PWAMG BiCGStab		
				Setup time CPU (Wtime)	Solver time CPU (Wtime)	n	Setup time CPU (Wtime)	Solver time CPU (Wtime)	n
Offshore	1	259789	4242673	16.67	18.42	6	16.69	21.17	10
	2	129894	2120163	8.13 (8.12)	4.65 (7.71)	3	8.08 (8.08)	31.94 (46.27)	12
		129895	2122510	7.38 (7.38)	5.80 (7.71)		7.37 (7.37)	26.93 (46.28)	
	3	86596	1424747	5.13 (5.15)	3.46 (9.45)	3	5.07 (5.07)	31.0 (64.24)	14
		86596	1410644	4.93 (4.93)	3.25 (9.50)		4.97 (4.97)	22.6 (64.19)	
		86597	1407282	5.01 (5.00)	4.97 (9.50)		4.90 (4.90)	20.6 (64.24)	
	4	64947	1065971	3.60 (3.61)	2.96 (8.74)	3	4.13 (4.12)	19.47 (60.21)	14
		64947	1060994	4.15 (4.14)	2.67 (8.64)		3.74 (3.73)	19.31 (60.18)	
		64947	1059169	3.42 (3.42)	2.73 (8.68)		3.54 (3.55)	16.98 (60.09)	
		64948	1056539	3.80 (3.80)	3.19 (8.74)		3.39 (3.38)	16.53 (60.13)	
	5	51957	858175	2.77 (2.76)	2.50 (8.76)	3	3.28 (3.28)	21.64 (60.97)	14
		51957	847435	3.31 (3.30)	2.31 (8.74)		2.94 (2.94)	16.83 (60.94)	
		51957	846747	2.96 (2.95)	2.23 (8.77)		2.88 (2.87)	14.76 (60.90)	
		51957	845466	2.87 (2.87)	1.99 (8.80)		2.86 (2.85)	14.28 (60.94)	
		51961	844850	2.93 (2.93)	3.13 (8.81)		2.75 (2.75)	13.83 (60.97)	
	6	43298	715228	2.22 (2.21)	2.21 (8.87)	3	2.61 (2.61)	22.69 (61.66)	13
		43298	709519	2.62 (2.61)	2.07 (8.90)		2.60 (2.60)	14.01 (61.60)	
		43298	707023	2.62 (2.61)	2.09 (8.90)		2.52 (2.52)	13.63 (61.65)	
		43298	705998	2.20 (2.20)	1.87 (8.89)		2.51 (2.51)	13.59 (61.63)	
		43298	703621	2.54 (2.53)	1.98 (8.90)		2.20 (2.20)	12.52 (61.66)	
		43299	701284	2.29 (2.29)	3.09 (8.87)		2.18 (2.19)	12.29 (61.63)	

The results are presented in Table 2. In all cases, the convergence is defined by $\|r^n\|/\|b\| < 10^{-6}$, where r^n is the residual vector at the nth iteration and the right hand side vector b is chosen so that the solution is a unitary vector. The setup and solver times are in seconds, and n is the number of iterations.

The parallel results for the circuit simulation and electromagnetics problems are presented in Tables 3, 4, 5, and 6 for the PWAMG solver and preconditioner. The setup and solver times are in seconds and have been presented in the form $t1$ ($t2$), where $t1$ is the processing time returned by the C clock() function and $t2$ is the total time spent in the corresponding phase, which includes the MPI communication time and is measured using the MPI function MPI_Wtime().

In some cases the CPU and wall-clock times are nearly equal, which means that the processes that are waiting for synchronization are still consuming full CPU time. Moreover, the CPU time and the wall-clock time are practically the same in setup phase indicating that there are no interprocessor communications during the setup. It is a great advantage of the proposed approach since this phase is the most time-consuming task in multilevel approaches, as can be seen in Table 2. For single processor algorithms the times $t1$ and $t2$ are the same.

Table 5: Parallel results for *circuit5M_dc* matrix.

	np	*N*rows	Nonzeros	PWAMG solver			PWAMG BiCGStab		
				Setup time CPU (Wtime)	Solver time CPU (Wtime)	n	Setup time CPU (Wtime)	Solver time CPU (Wtime)	n
circuit5M_dc	1	3523317	14865409	47.4	14.37	3	61.30	45.49	2
	2	1761659	8279908	27.73 (27.89)	5.86 (21.51)	3	27.76 (27.90)	14.37 (37.37)	2
		1761658	6585501	11.94 (11.94)	10.56 (21.24)		11.96 (11.94)	6.63 (40.58)	
	3	1174439	5531627	19.83 (19.84)	4.88 (27.07)	3	19.83 (19.82)	5.24 (27.43)	2
		1174439	4956074	10.95 (10.95)	3.79 (27.05)		10.96 (10.95)	4.8 (26.59)	
		1174439	4377708	8.03 (8.03)	4.07 (26.25)		8.03 (8.02)	4.1 (27.39)	
	4	880829	4249964	15.27 (15.27)	4.38 (26.34)	3	15.24 (15.31)	6.31 (28.30)	2
		880829	4029944	11.09 (11.09)	3.61 (26.90)		10.99 (10.99)	4.66 (27.14)	
		880829	3302046	6.09 (6.09)	3.20 (26.90)		6.23 (6.22)	3.99 (27.70)	
		880830	3283455	6.27 (6.27)	6.08 (26.29)		6.09 (6.09)	3.67 (28.30)	
	5	704663	3136583	12.24 (12.25)	4.04 (26.60)	2	12.24 (12.24)	10.08 (27.01)	2
		704663	2907582	9.88 (9.88)	3.48 (26.59)		9.87 (9.87)	4.40 (26.95)	
		704663	2607145	6.22 (6.21)	3.15 (26.22)		6.14 (6.14)	3.83 (26.89)	
		704663	3577997	4.97 (4.96)	3.05 (26.69)		4.94 (4.94)	3.51 (26.54)	
	6	704665	2636102	4.92 (9.92)	9.67 (26.69)	2	4.94 (4.94)	3.31 (27.02)	2
		587219	3002895	10.11 (10.12)	3.81 (27.08)		10.07 (10.08)	9.23 (30.16)	
		587219	2748283	8.40 (8.39)	3.26 (27.04)		8.39 (8.39)	4.20 (30.26)	
		587219	2528726	6.81 (6.80)	3.18 (27.44)		6.77 (6.77)	3.79 (30.19)	
		587219	2207789	4.23 (4.24)	2.95 (27.83)		4.26 (4.25)	3.35 (30.19)	
		587219	2182690	4.14 (4.15)	2.89 (28.23)		4.26 (4.25)	3.50 (28.99)	
		587222	2195026	4.26 (4.26)	5.57 (28.23)		4.14 (4.14)	3.27 (29.39)	

Three important aspects related to the parallel results should be highlighted.

(1) The proposed method uses a hybrid Jacobi Gauss-Seidel method as a smoother and a coarse system solver. This method applies the Gauss-Seidel method inside each processor and the Jacobi method between processors. So, as the number of processors increases, both smoother and coarse solvers become different. Also in the sequential version of the method this approach has not been reproduced. This can help us to explain the difference in the number of iterations in the parallel and sequential versions.

(2) The way in which the matrices are split between the processors (the same number of rows) may cause load unbalance in some cases that can affect the overall performance of the method. For the matrix *circuit5M_dc*, for example, the number of nonzero elements in the processor one is 25% larger than in the processor two, when 2 processors are used. A new way to divide the matrix among the processors should be investigated.

(3) Despite the use of a homogeneous Beowulf cluster, the type of network connection based on a switched fast Ethernet network shows to be an important bottleneck. So, the effects of the fast Ethernet connection on the results should be evaluated separately. In order to do so, the $time(p)$ required by the solver phase execution on p processors, in which there is interprocessors communication, was evaluated

considering a relative time in relation to the MPI π calculation example [25]. It means the $time(p)$ is defined as (6.1)

$$time(p) = \frac{wtime(p)}{\pi\text{-}time(p)}, \tag{6.1}$$

in which $wtime(p)$ is the total time spent in the setup phase, which includes the MPI communication time and is measured using the MPI function MPI_Wtime(), and π-$time(p)$ is the time spent by the MPI π example, both with p processors. As an example, Table 7 presents the values of π-$time(p)$, $wtime(p)$, $time(p)$ and $speedup(p)$, $p = 1, \ldots, 6$, for the matrix *2cubes_sphere*, which were used to create the illustration in Figure 6(a). The values of π-$time(p)$ were obtained as the mean of 5 runs. All the other results in Figure 6 were derived in a similar way.

As usual, the absolute speedup S_p is used for analyzing the parallel performance, and it is defined as (6.2)

$$S_p = speedup(p) = \frac{time(1)}{time(p)}, \tag{6.2}$$

in which $time(1)$ is the time spent by the best sequential algorithm and $time(p)$ as defined in (6.1).

As the MPI π example uses only point-to-point communication functions, the relative time approach can help to clarify if the apparent poor scalability is due to a high collective communication cost or mainly due to the type of network connection used.

The solver and preconditioner speedups are illustrated in Figure 6 for the matrices *circuit5M_dc*, *2cubes_sphere*, and *offshore*.

7. Conclusions

The PWAMG method, proposed in this work, has been applied as a black-box solver and preconditioner in some circuit simulations and finite element electromagnetic problems with good results.

An important characteristic of the proposed approach is its small demand for interprocessors communication. Actually, no communication is required in the setup phase if first-order filters are used. This characteristic is confirmed by the results for setup time presented in Tables 3–6, observing that the times measured by MPI_Wtime() and C clock() functions are practically the same. It is a great advantage of the proposed approach since this phase is the most time-consuming one.

In the solver phase, in which there is interprocessors communication, the numerical results seem to show a poor performance with more than 2 processors, even when the number of iterations does not increase by using more processors. These results can be caused in part due to the use of a collective communication function to update the vector after the matrix operations, which is motivated by the manner the matrix is divided between the processors.

In order to update the vector each processor should send and receive the part of the vector to all of the other processors. Of course, when the number of processors increases, this work also becomes larger.

Table 6: Parallel results for *dielFilterV2real* matrix (not enough memory).

	np	Nrows	Nonzeros	PWAMG solver		
				Setup time CPU (Wtime)	Solver time CPU (Wtime)	n
dielFilterV2real	1	1157456	48538952	—	—	—
	2	578728	26710406	60.00 (182.0)	176.21 (1763.4)	4
		578728	21828546	34.46 (39.34)	86.20 (1759.74)	
	3	385818	19476388	36.92 (78.71)	98.12 (192.62)	4
		385818	14529204	22.49 (22.54)	43.69 (192.13)	
		385820	14533360	22.18 (22.19)	54.49 (192.14)	
	4	289364	15807646	28.18 (28.39)	75.34 (108.75)	4
		289364	10902962	16.61 (16.59)	32.17 (108.35)	
		289364	10902760	16.79 (16.79)	33.26 (108.55)	
		289364	10925584	16.56 (16.58)	34.80 (108.75)	
	5	231491	13644000	24.42 (24.47)	64.90 (101.42)	4
		231491	8751322	13.28 (13.28)	27.28 (101.41)	
		231491	8727696	13.30 (13.29)	28.07 (101.57)	
		231491	8710580	13.35 (13.34)	26.93 (101.72)	
		231492	8705354	13.25 (13.25)	35.10 (101.72)	
	6	192909	11858182	22.81 (38.14)	107.36 (176.56)	7
		192909	7618206	11.76 (11.76)	47.08 (176.63)	
		192909	7295218	10.99 (10.99)	42.92 (176.76)	
		192909	7255410	11.18 (11.18)	42.60 (176.76)	
		192909	7233986	11.07 (11.07)	42.53 (176.50)	
		192911	7277950	11.01 (11.01)	58.37 (176.50)	

Table 7: Solver phase speedup using a relative $time(p)$: example for matrix *2cubes_sphere.*

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$
(A) $wtime(p)$	2.02	2.99	3.47	3.26	3.32	3.27
(B) π-$time(p)$	0.0000885	0.0002926	0.0003222	0.000997	0.001011	0.0009938
(C) $time(p)$ = A/B	22824.86	10218.73	10769.71	3269.81	3283.88	3290.40
(D) $speedup(p)$	1.00	2.23	2.12	6.98	6.95	6.94

An alternative to overcome this problem may be to apply some graph partitioning method and develop an approach in which each processor should communicate only with its neighbors. Such approach is under development, and it is out of the scope of this paper.

However, the type of network connection based on a switched fast Ethernet network shows to be an important bottleneck, and its effects should be evaluated separately. As the MPI π example uses only point-to-point communication functions, the relative time approach can help to clarify if the apparent poor scalability is due to a high collective communication cost or mainly due to the type of network connection used. The speedup results based on the relative times show that the proposed approach is promising and that the kind of network connection that has been used is maybe the most important drawback.

Moreover, in spite of the speedup being less than the linear in some cases, it is important to mention an important aspect of this application: in the context of large sparse linear system of equations, where this paper is inserted, the problems have large memory

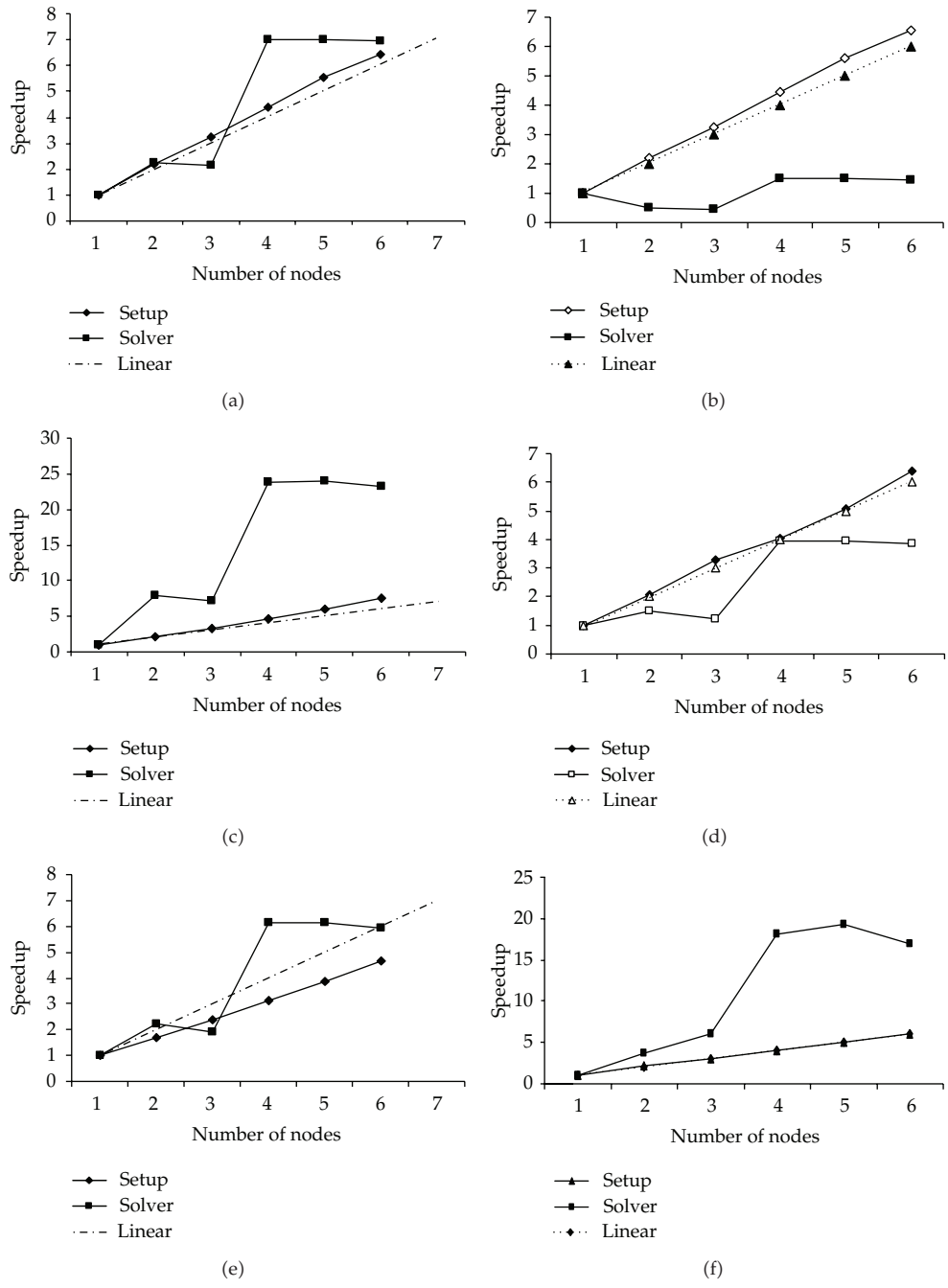


Figure 6: Speedup for *2cubes_sphere* solver (a) and preconditioner (b), *offshore* solver (c) and preconditioner (d), and *circuit5M_dc* solver (e) and preconditioner (f).

requirements. In these cases, as presented in [28], the speedup necessary to be cost effective can be much less than linear. The parallel program does not need p times memory of the unit processor, since parallelizing a job rarely multiplies its memory requirements by p.

Finally, the number of iteration of the proposed method seems to be independent of the number of processors. However, it is necessary to carry out more tests with a larger number of processors in order to draw more definitive conclusions. Nowadays, the authors are looking for new resources and/or partnerships to enable the continuation of this work.

Acknowledgments

Authors would like to thank the São Paulo Research Foundation (FAPESP) (06/59547-5 and 2011/06725-1) and Nove de Julho University (UNINOVE) for the financial support.

References

[1] G. Haase, M. Kuhn, and S. Reitzinger, "Parallel algebraic multigrid methods on distributed memory computers," *SIAM Journal on Scientific Computing*, vol. 24, no. 2, pp. 410–427, 2002.

[2] V. E. Henson and U. M. Yang, "BoomerAMG: a parallel algebraic multigrid solver and preconditioner," *Applied Numerical Mathematics*, vol. 41, no. 1, pp. 155–177, 2002.

[3] H. de Sterck, U. M. Yang, and J. J. Heys, "Reducing complexity in parallel algebraic multigrid preconditioners," *SIAM Journal on Matrix Analysis and Applications*, vol. 27, no. 4, pp. 1019–1039, 2006.

[4] M. Griebel, B. Metsch, D. Oeltz, and M. A. Schweitzer, "Coarse grid classification: a parallel coarsening scheme for algebraic multigrid methods," *Numerical Linear Algebra with Applications*, vol. 13, no. 2-3, pp. 193–214, 2006.

[5] A. J. Cleary, R. D. Falgout, V. E. Henson, and J. E. Jones, *Coarse-Grid Selection for Parallel Algebraic Multigrid*, Lawrence Livermore National Laboratory, Livermore, Calif, USA, 2000.

[6] L. Yu. Kolotilina and A. Yu. Yeremin, "Factorized sparse approximate inverse preconditionings. I. Theory," *SIAM Journal on Matrix Analysis and Applications*, vol. 14, no. 1, pp. 45–58, 1993.

[7] M. J. Grote and T. Huckle, "Parallel preconditioning with sparse approximate inverses," *SIAM Journal on Scientific Computing*, vol. 18, no. 3, pp. 838–853, 1997.

[8] D. Hysom and A. Pothen, "A scalable parallel algorithm for incomplete factor preconditioning," *SIAM Journal on Scientific Computing*, vol. 22, no. 6, pp. 2194–2215, 2000.

[9] P. Raghavan, K. Teranishi, and E. G. Ng, "A latency tolerant hybrid sparse solver using incomplete Cholesky factorization," *Numerical Linear Algebra with Applications*, vol. 10, no. 5-6, pp. 541–560, 2003.

[10] P. Raghavan and K. Teranishi, "Parallel hybrid preconditioning: incomplete factorization with selective sparse approximate inversion," *SIAM Journal on Scientific Computing*, vol. 32, no. 3, pp. 1323–1345, 2010.

[11] C. Janna, M. Ferronato, and G. Gambolati, "A block Fsai-Ilu parallel preconditioner for symmetric positive definite linear systems," *SIAM Journal on Scientific Computing*, vol. 32, no. 5, pp. 2468–2484, 2010.

[12] C. Janna and M. Ferronato, "Adaptive pattern research for block FSAI preconditioning," *SIAM Journal on Scientific Computing*, vol. 33, no. 6, pp. 3357–3380, 2011.

[13] M. Benzi and M. Tuma, "A comparative study of sparse approximate inverse preconditioners," *Applied Numerical Mathematics*, vol. 30, no. 2-3, pp. 305–340, 1999.

[14] F. H. Pereira, S. L. L. Verardi, and S. I. Nabeta, "A wavelet-based algebraic multigrid preconditioner for sparse linear systems," *Applied Mathematics and Computation*, vol. 182, no. 2, pp. 1098–1107, 2006.

[15] F. H. Pereira, M. F. Palin, S. L. L. Verardi, V. C. Silva, J. R. Cardoso, and S. I. Nabeta, "A wavelet-based algebraic multigrid preconditioning for iterative solvers in finite-element analysis," *IEEE Transactions on Magnetics*, vol. 43, no. 4, pp. 1553–1556, 2007.

[16] F. H. Pereira, M. M. Afonso, J. A. De Vasconcelos, and S. I. Nabeta, "An efficient two-level preconditioner based on lifting for FEM-BEM equations," *Journal of Microwaves and Optoelectronics*, vol. 9, no. 2, pp. 78–88, 2010.

[17] T. K. Sarkar, M. Salazar-Palma, and C. W. Michael, *Wavelet Applications in Engineering Electromagnetics*, Artech House, Boston, Mass, USA, 2002.

[18] V. M. Garcıa, L. Acevedo, and A. M. Vidal, "Variants of algebraic wavelet-based multigrid methods: application to shifted linear systems," *Applied Mathematics and Computation*, vol. 202, no. 1, pp. 287–299, 2008.

[19] A. Avudainayagam and C. Vani, "Wavelet based multigrid methods for linear and nonlinear elliptic partial differential equations," *Applied Mathematics and Computation*, vol. 148, no. 2, pp. 307–320, 2004.
[20] D. de Leon, "Wavelet techniques applied to multigrid methods," CAM Report 00-42, Department of Mathematics UCLA, Los Angeles, Calif, USA, 2000.
[21] G. Wang, R. W. Dutton, and J. Hou, "A fast wavelet multigrid algorithm for solution of electromagnetic integral equations," *Microwave and Optical Technology Letters*, vol. 24, no. 2, pp. 86–91, 2000.
[22] R. S. Chen, D. G. Fang, K. F. Tsang, and E. K. N. Yung, "Analysis of millimeter wave scattering by an electrically large metallic grating using wavelet-based algebratic multigrid preconditioned CG method," *International Journal of Infrared and Millimeter Waves*, vol. 21, no. 9, pp. 1541–1560, 2000.
[23] F. H. Pereira and S. I. Nabeta, "Wavelet-based algebraic multigrid method using the lifting technique," *Journal of Microwaves and Optoelectronics*, vol. 9, no. 1, pp. 1–9, 2010.
[24] J. M. Ford, K. Chen, and N. J. Ford, "Parallel implementation of fast wavelet transforms," Numerical Analysis 39, Manchester University, Manchester, UK, 2001.
[25] H. J. Sips and H. X. Lin, *High Performance Computing Course MPI Tutorial*, Delft University of Technology Information Technology and Systems, Delft, The Netherlands, 2002.
[26] F. Alessandri, M. Chiodetti, A. Giugliarelli et al., "The electric-field integral-equation method for the analysis and design of a class of rectangular cavity filters loaded by dielectric and metallic cylindrical pucks," *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, no. 8, pp. 1790–1797, 2004.
[27] T. A. Davis, "Algorithm 849: a concise sparse Cholesky factorization package," *ACM Transactions on Mathematical Software*, vol. 31, no. 4, pp. 587–591, 2005.
[28] D. A. Wood and M. D. Hill, "Cost-effective parallel computing," *Computer*, vol. 28, no. 2, pp. 69–72, 1995.

Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 872901, 14 pages
doi:10.1155/2012/872901

Research Article

Parallel Rayleigh Quotient Optimization with FSAI-Based Preconditioning

Luca Bergamaschi, Angeles Martínez, and Giorgio Pini

Department of Mathematical Methods and Models for Scientific Applications, University of Padova, Via Trieste 63, 35121 Padova, Italy

Correspondence should be addressed to Luca Bergamaschi, luca.bergamaschi@unipd.it

Received 2 November 2011; Revised 1 February 2012; Accepted 3 February 2012

Academic Editor: Massimiliano Ferronato

The present paper describes a parallel preconditioned algorithm for the solution of partial eigenvalue problems for large sparse symmetric matrices, on parallel computers. Namely, we consider the Deflation-Accelerated Conjugate Gradient (DACG) algorithm accelerated by factorized-sparse-approximate-inverse- (FSAI-) type preconditioners. We present an enhanced parallel implementation of the FSAI preconditioner and make use of the recently developed Block FSAI-IC preconditioner, which combines the FSAI and the Block Jacobi-IC preconditioners. Results onto matrices of large size arising from finite element discretization of geomechanical models reveal that DACG accelerated by these type of preconditioners is competitive with respect to the available public parallel *hypre* package, especially in the computation of a few of the leftmost eigenpairs. The parallel DACG code accelerated by FSAI is written in MPI-Fortran 90 language and exhibits good scalability up to one thousand processors.

1. Introduction

The computation by iterative methods of the s partial eigenspectrum of the generalized eigenproblem:

$$A\mathbf{u} = \lambda B\mathbf{u}, \tag{1.1}$$

where $A, B \in \mathbb{R}^{n\times n}$ are large sparse symmetric positive definite (SPD) matrices, is an important and difficult task in many applications. It has become increasingly widespread owing to the development in the last twenty years of robust and computationally efficient schemes and corresponding software packages. Among the most well-known approaches for the important class of symmetric positive definite (SPD) matrices are the implicitly restarted Arnoldi method (equivalent to the Lanczos technique for this type of matrices) [1, 2],

the Jacobi-Davidson (JD) algorithm [3], and schemes based on preconditioned conjugate gradient minimization of the Rayleigh quotient [4, 5].

The basic idea of the latter is to minimize the Rayleigh Quotient

$$q(x) = \frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T B\mathbf{x}}. \tag{1.2}$$

in a subspace which is orthogonal to the previously computed eigenvectors via a preconditioned CG-like procedure. Among the different variants of this technique we chose to use the Deflation-Accelerated Conjugate Gradient (DACG) scheme [4, 6] which has been shown to be competitive with the Jacobi Davidson method and with the PARPACK package [7]. As in any other approach, for our DACG method, the choice of the preconditioning technique is a key factor to accelerate and, in some cases even to allow for, convergence. To accelerate DACG in a parallel environment we selected the Factorized Sparse Approximate inverse (FSAI) preconditioner introduced in [8]. We have developed a parallel implementation of this algorithm which has displayed excellent performances on both the setup phase and the application phase within a Krylov subspace solver [9–11]. The effectiveness of the FSAI preconditioner in the acceleration of DACG is compared to that of the Block FSAI-IC preconditioner, recently developed in [12], which combines the FSAI and the Block Jacobi-IC preconditioners obtaining good results on a small number of processors for the solution of SPD linear systems and for the solution of large eigenproblems [13]. We used the resulting parallel codes to compute a few of the leftmost eigenpairs of a set of test matrices of large size arising from Finite Element discretization of geomechanical models. The reported results show that DACG preconditioned with either FSAI or BFSAI is a scalable and robust algorithm for the partial solution of SPD eigenproblems. The parallel performance of DACG is also compared to that of the publicly available parallel package *hypre* [14] which implements a number of preconditioners which can be used in combination with the Locally Optimal Block PCG (LOBPCG) iterative eigensolver [15]. The results presented in this paper show that the parallel DACG code accelerated by FSAI exhibits good scalability up to one thousand processors and displays comparable performance with respect to *hypre*, specially when a low number of eigenpairs is sought.

The outline of the paper is as follows: in Section 2 we describe the DACG Algorithm; in Sections 3 and 4 we recall the definition and properties of the FSAI and BFSAI preconditioners, respectively. Section 5 contains the numerical results obtained with the proposed algorithm in the eigensolution of very large SPD matrices of size up to almost 7 million unknowns and 3×10^8 nonzeros. A comparison with the *hypre* eigensolver code is also included. Section 6 ends the paper with some conclusions.

2. The DACG Iterative Eigensolver and Implementation

The DACG algorithm sequentially computes the eigenpairs, starting from the leftmost one $(\lambda_1, \mathbf{u}_1)$. To evaluate the jth eigenpair, $j > 1$, DACG minimizes the Rayleigh Quotient (RQ) in a subspace orthogonal to the $j-1$ eigenvectors previously computed. More precisely, DACG minimizes the Rayleigh Quotient:

$$q(\mathbf{z}) = \frac{\mathbf{z}^T A\mathbf{z}}{\mathbf{z}^T\mathbf{z}}, \tag{2.1}$$

Choose tolerance ε, set $U = 0$.
DO $j = 1, s$
(1) Choose $\mathbf{x}_0$ such that $U^T\mathbf{x}_0 = 0$; set $k = 0$, $\beta_0 = 0$;
(2) $\mathbf{x}_0^A = A\mathbf{x}_0$, $\gamma = \mathbf{x}_0^T\mathbf{x}_0^A$, $\eta = \mathbf{x}_0^T\mathbf{x}_0$, $q_0 \equiv q(\mathbf{x}_0) = \gamma/\eta$, $\mathbf{r}_0 = \mathbf{x}_0^A - q_0\mathbf{x}_0$;
(3) REPEAT
(3.1) $\mathbf{g}_k \equiv \nabla q(\mathbf{x}_k) = (2/\eta)\mathbf{r}_k$;
(3.2) $\mathbf{g}_k^M = M\mathbf{g}_k$;
(3.3) IF $k > 0$ THEN $\beta_k = \mathbf{g}_k^T(\mathbf{g}_k^M - \mathbf{g}_{k-1}^M)/\mathbf{g}_{k-1}^T\mathbf{g}_{k-1}^M$;
(3.4) $\tilde{\mathbf{p}}_k = \mathbf{g}_k^M + \beta_k\mathbf{p}_{k-1}$;
(3.5) $\mathbf{p}_k = \tilde{\mathbf{p}}_k - U(U^T\tilde{\mathbf{p}}_k)$
(3.6) $\mathbf{p}_k^A = A\mathbf{p}_k$,
(3.7) $\alpha_k = \operatorname{argmin}_t\{q(\mathbf{x}_k + t\mathbf{p}_k)\} = (\eta d - \gamma b + \sqrt{\Delta})/2(bc - ad)$, with
$a = \mathbf{p}_k^T\mathbf{x}_k^A$, $b = \mathbf{p}_k^T\mathbf{p}_k^A$, $c = \mathbf{p}_k^T\mathbf{x}_k$, $d = \mathbf{p}_k^T\mathbf{p}_k$,
$\Delta = (\eta d - \gamma b)^2 - 4(bc - ad)(\gamma a - \eta c)$;
(3.8) $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{p}_k$, $\mathbf{x}_{k+1}^A = \mathbf{x}_k^A + \alpha_k\mathbf{p}_k^A$;
(3.9) $\gamma = \gamma + 2a\alpha_k + b\alpha_k^2$, $\eta = \eta + 2c\alpha_k + d\alpha_k^2$;
(3.10) $q_{k+1} \equiv q(\mathbf{x}_{k+1}) = \gamma/\eta$;
(3.11) $k = k + 1$;
(3.12) $\mathbf{r}_k = \mathbf{x}_k^A - q_k\mathbf{x}_k$
UNTIL $(q_{k+1} - q_k)/q_{k+1} < tol$;
(4) $\lambda_j = q_k$, $\mathbf{u}_j = \mathbf{x}_k/\sqrt{\eta}$, $U = [U, \mathbf{u}_j]$.
END DO

Algorithm 1: DACG Algorithm.

where

$$\mathbf{z} = \mathbf{x} - U_j\left(U_j^T\mathbf{x}\right), \quad U_j = \left[\mathbf{u}_1, \ldots, \mathbf{u}_{j-1}\right], \quad \mathbf{x} \in R^n. \tag{2.2}$$

The first eigenpair $(\lambda_1, \mathbf{u}_1)$ is obtained by minimization of (2.1) with $\mathbf{z} = \mathbf{x}(U_1 = \emptyset)$. Indicating with M the preconditioning matrix, that is, $M \approx A^{-1}$, the s leftmost eigenpairs are computed by the conjugate gradient procedure [6] described in Algorithm 1.

The schemes relying on the Rayleigh quotient optimization are quite attractive for parallel computations; however preconditioning is an essential feature to ensure practical convergence. When seeking for an eigenpair $(\lambda_j, \mathbf{u}_j)$ it can be proved that the number of iterations is proportional to the square root of the condition number $\xi_j = \kappa(H_j)$ of the Hessian of the Rayleigh quotient in the stationary point $\mathbf{u}_j$ [4]. It turns out that H_j is similar to $(A - \lambda_j I)M$ which is not SPD. However, H_j operates on the orthogonal space spanned by the previous eigenvectors, so that the only important eigenvalues are the positive ones. In the non-preconditioned case (i.e., $M = I$) we would have

$$\kappa(H_j) \approx \frac{\lambda_N}{\lambda_{j+1} - \lambda_j}. \tag{2.3}$$

where in the ideal case $M \equiv A^{-1}$, we have

$$\kappa(H_j) \approx \frac{\lambda_j}{\lambda_{j+1} - \lambda_j} = \xi_j \ll \frac{\lambda_N}{\lambda_{j+1} - \lambda_j}. \tag{2.4}$$

Therefore, even though A^{-1} is not the optimal preconditioner for $A - \lambda_j I$, however, if M is a good preconditioner of A then the condition number $\kappa(H_j)$ will approach ξ_j.

3. The FSAI Preconditioner

The FSAI preconditioner, initially proposed in [8, 16], has been later developed and implemented in parallel by Bergamaschi and Martínez in [9]. Here, we only shortly recall the main features of this preconditioner. Given an SPD matrix A the FSAI preconditioner approximately factorizes its inverse as a product of two sparse triangular matrices as

$$A^{-1} \approx W^T W. \tag{3.1}$$

The choice of nonzeros in W is based on a sparsity pattern which in our work may be the same as $\tilde{A}^d$ where $\tilde{A}$ is the result of *prefiltration* [10] of A, that is, dropping of all elements below of a threshold parameter δ. The entries of W are computed by minimizing the Frobenius norm of $I - WL$, where L is the exact Cholesky factor of A, without forming explicitly the matrix L. The computed W is then sparsified by dropping all the elements which are below a second tolerance parameter (ε). The final FSAI preconditioner is therefore related to the following three parameters: δ, prefiltration threshold; d, power of A generating the sparsity pattern (we allow $d \in \{1, 2, 4\}$ in our experiments); ε, postfiltration threshold.

3.1. Parallel Implementation of FSAI-DACG

We have developed a parallel code written in FORTRAN 90 and which exploits the MPI library for exchanging data among the processors. We used a block row distribution of all matrices (A, W, and W^T), that is, with complete rows assigned to different processors. All these matrices are stored in static data structures in CSR format.

Regarding the preconditioner computation, we stress that any row i of matrix W of FSAI preconditioner is computed independently of each other, by solving a small SPD dense linear system of size n_i equal to the number of nonzeros allowed in row i of W. Some of the rows which contribute to form this linear system may be nonlocal to processor i and should be received from other processors. To this aim we implemented a routine called *get_extra_rows* which carries out all the row exchanges among the processors, before starting the computation of W, which proceed afterwards entirely in parallel. Since the number of nonlocal rows needed by each processor is relatively small we chose to temporarily replicate these rows on auxiliary data structures. Once W is obtained a parallel transposition routine provides every processor with its part of W^T.

The DACG iterative solver is essentially based on scalar and matrix-vector products. We made use of an optimized parallel matrix-vector product which has been developed in [17] showing its effectiveness up to 1024 processors.

4. Block FSAI-IC Preconditioning

The Block FSAI-IC preconditioner, BFSAI-IC in the following, is a recent development for the parallel solution to Symmetric Positive Definite (SPD) linear systems. Assume that D is an arbitrary nonsingular block diagonal matrix consisting of n_b equal size blocks.

Let $\mathcal{S}_L$ and $\mathcal{S}_{\mathrm{BD}}$ be a sparse lower triangular and a dense block diagonal nonzero pattern, respectively, for an $n \times n$ matrix. Even though not strictly necessary, for the sake of simplicity assume that $\mathcal{S}_{\mathrm{BD}}$ consists of n_b diagonal blocks with equal size $m = n/n_b$ and let $D \in \mathbb{R}^{n\times n}$ be an arbitrary full-rank matrix with nonzero pattern $\mathcal{S}_{\mathrm{BD}}$.

Consider the set of lower block triangular matrices F with a prescribed nonzero pattern S_{BL} and minimize over F the Frobenius norm:

$$\|D - FL\|_F, \tag{4.1}$$

where L is the exact lower Cholesky factor of an SPD matrix A. A matrix F satisfying the minimality condition (4.1) for a given D is the lower block triangular factor of BFSAI-IC. Recalling the definition of the classical FSAI preconditioner, it can be noticed that BFSAI-IC is a block generalization of the FSAI concept.

The differentiation of (4.1) with respect to the unknown entries $[F]_{ij}$, $(i, j) \in S_{\mathrm{BL}}$, yields the solution to n independent dense subsystems which, in the standard FSAI case, do not require the explicit knowledge of L. The effect of applying F to A is to concentrate the largest entries of the preconditioned matrix FAF^T into n_b diagonal blocks. However, as D is arbitrary, it is still not ensured that FAF^T is better than A in an iterative method, so it is necessary to precondition FAF^T again. As FAF^T resembles a block diagonal matrix, an efficient technique relies on using a block diagonal matrix which collects an approximation of the inverse of each diagonal block B_{i_b} of FAF^T.

It is easy to show that F is guaranteed to exist with SPD matrices and B_{i_b} is SPD, too [12]. Using an IC decomposition with partial fill-in for each block B_{i_b} and collecting in J the lower IC factors, the resulting preconditioned matrix reads

$$J^{-1}FAF^TJ^{-T} = WAW^T \tag{4.2}$$

with the final preconditioner

$$M = W^TW = F^TJ^{-T}J^{-1}F. \tag{4.3}$$

M in (4.3) is the BFSAI-IC preconditioner of A.

For its computation BFSAI-IC needs the selection of n_b and $\mathcal{S}_{\overline{L}}$. The basic requirement for the number of blocks n_b is to be larger than or equal to the number of computing cores p. From a practical viewpoint, however, the most efficient choice in terms of both wall clock time and iteration count is to keep the blocks as large as possible, thus implying $n_b = p$. Hence, n_b is by default set equal to p. By distinction, the choice of $\mathcal{S}_{\overline{L}}$ is theoretically more challenging and still not completely clear. A widely accepted option for other approximate inverses, such as FSAI or SPAI, is to select the nonzero pattern of A^d for small values of d on the basis of the Neumann series expansion of A^{-1}. Using a similar approach, in the BFSAI construction we select $\mathcal{S}_{\overline{L}}$ as the lower block triangular pattern of A^d. As the nonzeros located in the diagonal blocks are not used for the computation of F a larger value of d, say 3 or 4, can still be used.

Though theoretically not necessary, three additional user-specified parameters are worth introducing in order to better control the memory occupation and the BFSAI-IC density:

(1) ε is a postfiltration parameter that allows for dropping the smallest entries of F. In particular, $[F]_{ij}$ is neglected if $[F]_{ij} < \varepsilon \|\mathbf{f}_i\|_2$, where $\mathbf{f}_i$ is the ith row of F;

(2) ρ_B is a parameter that controls the fill-in of B_{i_b} and determines the maximum allowable number of nonzeros for each row of B_{i_b} in addition to the corresponding entries of A. Quite obviously, the largest ρ_B entries only are retained;

(3) ρ_L is a parameter that controls the fill-in of each IC factor $\widetilde{L}_{i_b}$ denoting the maximum allowable number of nonzeros for each row of $\widetilde{L}_{i_b}$ in addition to the corresponding entries of B_{i_b}.

An OpenMP implementation of the algorithms above is available in [18].

5. Numerical Results

In this section we examine the performance of the parallel DACG preconditioned by both FSAI and BFSAI in the partial solution of four large-size sparse eigenproblems. The test cases, which we briefly describe below, are taken from different real engineering mechanical applications. In detail, they are as follows.

(i) FAULT-639 is obtained from a structural problem discretizing a faulted gas reservoir with tetrahedral finite elements and triangular interface elements [19]. The interface elements are used with a penalty formulation to simulate the faults behavior. The problem arises from a 3D discretization with three displacement unknowns associated to each node of the grid.

(ii) PO-878 arises in the simulation of the consolidation of a real gas reservoir of the Po Valley, Italy, used for underground gas storage purposes (for details, see [20]).

(iii) GEO-1438 is obtained from a geomechanical problem discretizing a region of the earth crust subject to underground deformation. The computational domain is a box with an areal extent of 50×50 km and 10 km deep consisting of regularly shaped tetrahedral finite elements. The problem arises from a 3D discretization with three displacement unknowns associated to each node of the grid [21].

(iv) CUBE-6091 arises from the equilibrium of a concrete cube discretized by a regular unstructured tetrahedral grid.

Matrices FAULT-639 and GEO-1438 are publicly available in the University of Florida Sparse Matrix Collection at http://www.cise.ufl.edu/research/sparse/matrices/.

In Table 1 we report sizes and nonzeros of the four matrices together with three of the most significant eigenvalues for each problem.

The computational performance of FSAI is compared to the one obtained by using BFSAI as implemented in [12]. The comparison is done evaluating the number of iterations n_{iter} to converge at the same tolerance, the wall clock time in seconds T_{prec} and T_{iter} for the preconditioner computation, and the eigensolver to converge, respectively, with the total time $T_{\text{tot}} = T_{\text{prec}} + T_{\text{iter}}$. All tests are performed on the IBM SP6/5376 cluster at the CINECA Centre for High Performance Computing, equipped with IBM Power6 processors at 4.7 GHz

Table 1: Size, number of nonzeros, and three representative eigenvalues of the test matrices.

	Size	Nonzeros	λ_1	λ_{10}	λ_N
FAULT-639	638,802	28,614,564	$6.99 \cdot 10^6$	$1.73 \cdot 10^7$	$2.52 \cdot 10^{16}$
PO-878	878,355	38,847,915	$1.46 \cdot 10^6$	$4.45 \cdot 10^6$	$5.42 \cdot 10^{15}$
GEO-1438	1,437,960	63,156,690	$7.81 \cdot 10^5$	$1.32 \cdot 10^6$	$1.11 \cdot 10^{13}$
CUBE-6091	6,091,008	270,800,586	$1.82 \cdot 10^1$	$3.84 \cdot 10^2$	$1.05 \cdot 10^{07}$

with 168 nodes, 5376 computing cores, and 21 Tbyte of internal network RAM. The FSAI-DACG code is written in Fortran 90 and compiled with `-O4 -q64 -qarch=pwr6 -qtune=pwr6 -qnoipa -qstrict -bmaxdata:0x70000000` options. For the BFSAI-IC code only an OpenMP implementation is presently available.

To study parallel performance we will use a strong scaling measure to see how the CPU times vary with the number of processors for a fixed total problem size. Denote with T_p the total CPU elapsed times expressed in seconds on p processors. We introduce a relative measure of the parallel efficiency achieved by the code, $S_p^{(\overline{p})}$, which is the pseudo speedup computed with respect to the smallest number of processors ($\overline{p}$) used to solve a given problem. Accordingly, we will denote by $E_p^{(\overline{p})}$ the corresponding efficiency:

$$S_p^{(\overline{p})} = \frac{T_{\overline{p}}\overline{p}}{T_p}, \qquad E_p^{(\overline{p})} = \frac{S_p^{(\overline{p})}}{p} = \frac{T_{\overline{p}}\overline{p}}{T_p p}. \tag{5.1}$$

5.1. FSAI-DACG Results

In this section we report the results of our FSAI-DACG implementation in the computation of the 10 leftmost eigenpairs of the 4 test problems. We used the exit test described in the DACG algorithm (see Algorithm 1) with tol $= 10^{-10}$. The results are summarized in Table 2. As the FSAI parameters, we choose $\delta = 0.1$, $d = 4$, and $\varepsilon = 0.1$ for all the test matrices. This combination of parameters produces, on the average, the best (or close to the best) performance of the iterative procedure. Note that the number of iterations does not change with the number of processors, for a fixed problem. The scalability of the code is very satisfactory in both the setup stage (preconditioner computation) and the iterative phase.

5.2. BFSAI-IC-DACG Results

We present in this section the results of DACG accelerated by the BFSAI-IC preconditioner for the approximation of the $s = 10$ leftmost eigenpairs of the matrices described above.

Table 3 provides iteration count and total CPU time for BFSAI-DACG with different combinations of the parameters needed to construct the BFSAI-IC preconditioner for matrix PO-878 and using from 2 to 8 processors. It can be seen from Table 3 that the assessment of the optimal parameters, ε, ρ_B, and ρ_L, is not an easy task, since the number of iterations may highly vary depending on the number of processors. We chose in this case the combination of parameters producing the second smallest total time with $p = 2, 4, 8$ processors. After

Table 2: Number of iterations, timings, and scalability indices for FSAI-DACG in the computation of the 10 leftmost eigenpairs of the four test problems.

	p	iter	T_{prec}	T_{iter}	T_{tot}	$S_p^{(4)}$	$E_p^{(4)}$
FAULT-639	4	4448	25.9	261.4	287.3		
	8	4448	13.2	139.0	152.2	7.6	0.94
	16	4448	6.6	69.4	76.0	15.1	0.95
	32	4448	4.0	28.2	32.2	35.7	1.11
	64	4448	1.9	15.5	17.4	66.1	1.03
	128	4448	1.1	9.4	10.5	109.0	0.85
PO-878	4	5876	48.1	722.5	770.6		
	8	5876	25.2	399.8	425.0	7.3	0.91
	16	5876	11.4	130.2	141.6	21.8	1.36
	32	5876	6.8	65.8	72.5	42.5	1.33
	64	5876	4.1	30.1	34.1	90.3	1.41
	128	5876	1.9	19.1	21.0	146.8	1.15
GEO-1437	4	6216	90.3	901.5	991.7		
	8	6216	47.5	478.9	526.4	7.5	0.94
	16	6216	24.7	239.4	264.1	15.0	0.94
	32	6216	13.6	121.0	134.6	29.5	0.92
	64	6216	8.2	60.9	69.1	57.4	0.90
	128	6216	4.2	29.5	33.8	117.5	0.92
	256	6216	2.3	19.1	21.4	185.4	0.72
	p	iter	T_{prec}	T_{iter}	T_{tot}	$S_p^{(16)}$	$E_p^{(16)}$
CUBE-6091	16	15796	121.5	2624.8	2746.2		
	32	15796	62.2	1343.8	1406.0	31.3	0.98
	64	15796	32.5	737.0	769.5	57.1	0.89
	128	15796	17.3	388.4	405.7	108.3	0.85
	256	15796	9.1	183.9	192.9	227.8	0.89
	512	15796	5.7	106.0	111.7	393.5	0.77
	1024	15796	3.8	76.6	80.4	546.6	0.53

intensive testing for all the test problems, we selected similarly the "optimal" values which are used in the numerical experiments reported in Table 4:

(i) FAULT-639: $d = 3$, $\varepsilon = 0.05$, $\rho_B = 10$, $\rho_L = 60$,

(ii) PO-878 $d = 3$, $\varepsilon = 0.05$, $\rho_B = 10$, $\rho_L = 10$,

(iii) GEO-1438: $d = 3$, $\varepsilon = 0.05$, $\rho_B = 10$, $\rho_L = 50$,

(iv) CUBE-6091: $d = 2$, $\varepsilon = 0.01$, $\rho_B = 0$, $\rho_L = 10$.

The user-specified parameters for BFSAI-IC given above provide evidence that it is important to build a dense preconditioner based on the lower nonzero pattern of A^3 (except for CUBE-6091, which is built on a regular discretization) with the aim at decreasing the number of DACG iterations. Anyway, the cost for computing such a dense preconditioner appears to be almost negligible with respect to the wall clock time needed to iterate to convergence.

We recall that, presently, the code BFSAI-IC is implemented in OpenMP, and the results in terms of CPU time are significant only for $p \leq 8$. For this reason the number of iterations

Table 3: Performance of BFSAI-DACG for matrix PO-878 with 2 to 8 processors and different parameter values.

d	ρ_B	ε	ρ_L	$p = 2$		$p = 4$		$p = 8$	
				iter	T_{tot}	iter	T_{tot}	iter	T_{tot}
2	10	0.01	10	2333	385.76	2877	286.24	3753	273.77
2	10	0.05	10	2345	415.81	2803	245.42	3815	**142.93**
2	10	0.05	20	2186	**370.86**	2921	276.41	3445	257.18
2	10	0.00	10	2328	445.16	2880	241.23	3392	269.41
2	20	0.05	10	2340	418.20	2918	**224.32**	3720	253.98
3	10	0.05	10	2122	375.17	2638	228.39	3366	149.59
3	10	0.05	20	1946	433.04	2560	304.43	3254	263.51
3	10	0.05	30	1822	411.00	2481	321.30	3176	179.67
3	10	0.05	40	1729	439.47	2528	346.82	3019	188.13
4	10	0.05	10	2035	499.45	2469	350.03	3057	280.31

Table 4: Number of iterations for BFSAI-DACG in the computations of the 10 leftmost eigenpairs.

Matrix	n_b									
	2	4	8	16	32	64	128	256	512	1024
FAULT-639	1357	1434	1594	2002	3053	3336	3553			
PO-878	2122	2638	3366	4157	4828	5154	5373			
GEO-1438	1458	1797	2113	2778	3947	4647	4850	4996		
CUBE-6091			5857	6557	7746	8608	9443	9996	10189	9965

reported in Table 4 is obtained with increasing number of blocks n_b and with $p = 8$ processors. This iteration number accounts for a potential implementation of BFSAI-DACG under the MPI (or hybrid OpenMP-MPI) environment as the number of iterations depends only on the number of blocks, irrespective of the number of processors.

The only meaningful comparison between FSAI-DACG and BFSAI-DACG can be carried out in terms of iteration numbers which are smaller for BFSAI-DACG for a small number of processors. The gap between FSAI and BFSAI iterations reduces when the number of processors increases.

5.3. Comparison with the LOBPCG Eigensolver Provided by hypre

In order to validate the effectiveness of our preconditioning in the proposed DACG algorithm with respect to already available public parallel eigensolvers, the results given in Tables 2 and 4 are compared with those obtained by the schemes implemented in the *hypre* software package [14]. The Locally Optimal Block Preconditioned Conjugate Gradient method (LOBPCG) [15] is experimented with, using the different preconditioners developed in the *hypre* project, that is, algebraic multigrid (AMG), diagonal scaling (DS), approximate inverse (ParaSails), additive Schwarz (Schwarz), and incomplete LU (Euclid). The *hypre* preconditioned CG is used for the inner iterations within LOBPCG. For details on the implementation of the LOBPCG algorithm, see, for instance, [22]. The selected preconditioner, ParaSails, is on its turn based on the FSAI preconditioner, so that the different

Table 5: Iterations and CPU time for the iterative solver of LOBPCG-*hypre* preconditioned by Parasails with different values of bl and $p = 16$ processors.

Matrix	bl = 10		bl = 11		bl = 12		bl = 15	
	iter	T_{iter}	iter	T_{iter}	iter	T_{iter}	iter	T_{iter}
FAULT-639	156	79.5	157	85.3	157	96.1	160	128.1
PO-878	45	117.0	41	131.6	38	151.3	35	192.6
GEO-1438	23	123.7	72	173.7	30	152.5	121	291.1
CUBE-6091	101	1670.5	143	2414.0	38	1536.7	35	1680.9

FSAI-DACG and ParaSails-LOBPCG performances should be ascribed mainly to the different eigensolvers rather than to the preconditioners.

We first carried out a preliminary set of runs with the aim of assessing the optimal value of the block size bl parameter, that is, the size of the subspace where to seek for the eigenvectors. Obviously it must be $\texttt{bl} \geq s = 10$. We fixed to 16 the number of processors and obtained the results summarized in Table 5 with different values of $\texttt{bl} \in [10, 15]$. We found that, only in problem CUBE-6091, a value of bl larger than 10, namely, $\texttt{bl} = 12$, yields an improvement in the CPU time. Note that we also made this comparison with different number of processors, and we obtained analogous results.

Table 6 presents the number of iterations and timings using the LOBPCG algorithm in the *hypre* package. The LOBPCG wall clock time is obtained with the preconditioner allowing for the best performance in the specific problem at hand, that is, ParaSails for all the problems. Using AMG as the preconditioner did not allow for convergence in three cases out of four, with the only exception of the FAULT-639 problem, in which the CPU timings were however very much larger than using ParaSails.

All matrices have to be preliminarily scaled by their maximum coefficient in order to allow for convergence. To make the comparison meaningful, the outer iterations of the different methods are stopped when the average relative error measure of the computed leftmost eigenpairs gets smaller than 10^{-10}, in order to obtain a comparable accuracy as in the other codes. We also report in Table 6 the number of inner preconditioned CG iterations (pcgitr).

To better compare our FSAI DACG with the LOBPCG method, we depict in Figure 1 the total CPU time versus the number of processor for the two codes. FSAI-DACG and LOBPCG provide very similar scalability, being the latter code a little bit more performing on the average. On the FAULT-639 problem, DACG reveals faster than LOBPCG, irrespective of the number of processors employed.

Finally, we have carried out a comparison of the two eigensolvers in the computation of only the leftmost eigenpair. Differently from LOBPCG, which performs a simultaneous approximation of all the selected eigenpairs, DACG proceeds in the computation of the selected eigenpairs in a sequential way. For this reason, DACG should be the better choice, at least in principle, when just one eigenpair is sought. We investigate this feature, and the results are summarized in Table 7. We include the total CPU time and iteration count needed by LOBPCG and FSAI-DACG to compute the leftmost eigenpair with 16 processors. For the LOBPCG code we report only the number of outer iterations.

The parameters used to construct the FSAI preconditioner for these experiments are as follows:

(1) FAULT-639. $\delta = 0.1$, $d = 2$, $\varepsilon = 0.05$,

(2) PO-878. $\delta = 0.2$, $d = 4$, $\varepsilon = 0.1$,

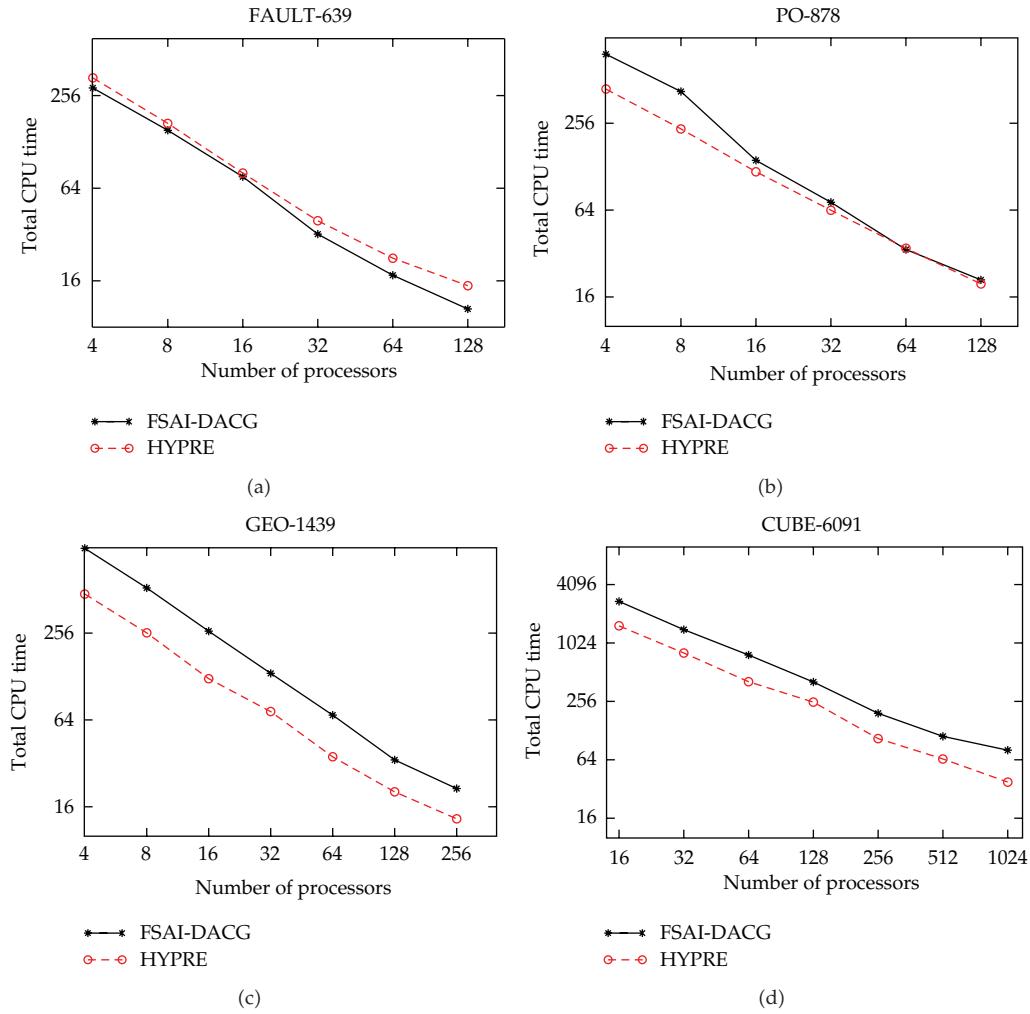


Figure 1: Comparison between FSAI-DACG and LOBPCG-*hypre* in terms of total CPU time for different number of processors.

(3) GEO-1438. $\delta = 0.1$, $d = 2$, $\varepsilon = 0.1$,

(4) CUBE-6091. $\delta = 0.0$, $d = 1$, $\varepsilon = 0.05$.

These parameters differ from those employed to compute the FSAI preconditioner in the assessment of the 10 leftmost eigenpairs and have been selected in order to produce a preconditioner relatively cheap to compute. This is so because otherwise the setup time would prevail over the iteration time. Similarly, to compute just one eigenpair with LOBPCG we need to setup a different value for pcgitr, the number of inner iterations. As it can be seen from Table 7, in the majority of the test cases, LOBPCG takes less time to compute 2 eigenpairs than just only 1. FSAI-DACG reveals more efficient than the best LOBPCG on problems PO-878 and GEO-1438. On the remaining two problems the slow convergence exhibited by DACG is probably due to the small relative separation ξ_1 between λ_1 and λ_2.

Table 6: Number of iterations, timings, and scalability of LOBPCG-*hypre* preconditioned by Parasails.

	p	iter	T_{prec}	T_{iter}	T_{tot}	$S_p^{(4)}$	$E_p^{(4)}$
FAULT-639 pcgitr = 5	4	155	2.5	331.2	333.7		
	8	156	1.3	167.6	168.9	7.9	0.99
	16	156	0.8	79.5	80.3	16.6	1.04
	32	150	0.5	38.8	39.3	34.0	1.06
	64	145	0.3	22.2	22.5	59.4	0.93
	128	157	0.1	14.8	14.9	89.7	0.70
PO-878 pcgitr = 30	4	45	3.3	438.4	441.7		
	8	50	1.3	232.3	234.0	7.6	0.94
	16	45	1.0	117.0	118.0	15.0	0.94
	32	45	0.7	63.2	63.9	27.6	0.86
	64	47	0.4	34.4	34.8	50.8	0.79
	128	41	0.3	19.44	19.74	89.5	0.70
GEO-1438 pcgitr = 30	4	26	7.7	478.0	485.7		
	8	22	4.0	256.8	260.8	7.5	0.93
	16	23	2.1	123.7	125.8	15.4	0.96
	32	28	1.2	73.1	74.3	26.2	0.82
	64	23	0.8	35.5	36.3	53.5	0.84
	128	25	0.5	20.3	20.8	93.2	0.73
	256	26	0.3	12.9	13.2	147.2	0.57
	p	iter	T_{prec}	T_{iter}	T_{tot}	$S_p^{(16)}$	$E_p^{(16)}$
CUBE-6091	16	38	9.2	1536.7	1545.9		
	32	36	4.7	807.5	812.2	30.5	0.95
	64	38	3.2	408.2	411.4	60.1	0.94
	128	41	1.6	251.4	253.0	97.8	0.76
	256	35	0.9	105.9	106.8	231.6	0.90
	512	39	0.6	65.3	65.9	375.3	0.73
	1024	37	0.3	37.7	38.0	650.9	0.64

Table 7: Performance of LOBPCG-*hypre* with Parasails and 16 processors in the computation of the smallest eigenvalue using `bl` = 1 and `bl` = 2 and FSAI-DACG.

	LOBPCG, `bl` = 1		LOBPCG, `bl` = 2		FSAI-DACG	
	iter	T_{tot}	iter	T_{tot}	iter	T_{tot}
FAULT-639	144	10.1	132	17.5	1030	15.4
PO-878	99	43.2	34	29.1	993	20.4
GEO-1493	55	40.2	26	37.3	754	27.0
CUBE-6091	144	5218.8	58	522.4	3257	561.1

6. Conclusions

We have presented the parallel DACG algorithm for the partial eigensolution of large and sparse SPD matrices. The scalability of DACG, accelerated with FSAI-type preconditioners,

has been studied on a set of test matrices of very large size arising from real engineering mechanical applications. Our FSAI-DACG code has shown comparable performances with the LOBPCG eigensolver within the well-known public domain package, *hypre*. Numerical results reveal that not only the scalability achieved by our code is roughly identical to that of *hypre* but also, in some instances, FSAI-DACG proves more efficient in terms of absolute CPU time. In particular, for the computation of the leftmost eigenpair, FSAI-DACG is more convenient in 2 problems out of 4.

Acknowledgment

The authors acknowledge the CINECA Iscra Award SCALPREC (2011) for the availability of HPC resources and support.

References

[1] P. Arbenz, U. Hetmaniuk, R. Lehoucq, and R. Tuminaro, "A comparison of eigensolvers for large-scale 3D modal analysis using AMG-preconditioned iterative methods," *International Journal for Numerical Methods in Engineering*, vol. 64, no. 2, pp. 204–236, 2005.

[2] R. B. Lehoucq and D. C. Sorensen, "Deflation techniques for an implicitly restarted Arnoldi iteration," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 4, pp. 789–821, 1996.

[3] G. L. G. Sleijpen and H. A. Van der Vorst, "A Jacobi-Davidson iteration method for linear eigenvalue problems," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 2, pp. 401–425, 1996.

[4] L. Bergamaschi, G. Gambolati, and G. Pini, "Asymptotic convergence of conjugate gradient methods for the partial symmetric eigenproblem," *Numerical Linear Algebra with Applications*, vol. 4, no. 2, pp. 69–84, 1997.

[5] A. V. Knyazev and A. L. Skorokhodov, "Preconditioned gradient-type iterative methods in a subspace for partial generalized symmetric eigenvalue problems," *SIAM Journal on Numerical Analysis*, vol. 31, no. 4, pp. 1226–1239, 1994.

[6] L. Bergamaschi, G. Pini, and F. Sartoretto, "Approximate inverse preconditioning in the parallel solution of sparse eigenproblems," *Numerical Linear Algebra with Applications*, vol. 7, no. 3, pp. 99–116, 2000.

[7] L. Bergamaschi and M. Putti, "Numerical comparison of iterative eigensolvers for large sparse symmetric positive definite matrices," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 45, pp. 5233–5247, 2002.

[8] L. Yu. Kolotilina and A. Yu. Yeremin, "Factorized sparse approximate inverse preconditionings. I. Theory," *SIAM Journal on Matrix Analysis and Applications*, vol. 14, no. 1, pp. 45–58, 1993.

[9] L. Bergamaschi and A. Martínez, "Parallel acceleration of Krylov solvers by factorized approximate inverse preconditioners," in *VECPAR 2004*, M. Dayd et al., Ed., vol. 3402 of *Lecture Notes in Computer Sciences*, pp. 623–636, Springer, Heidelberg, Germany, 2005.

[10] L. Bergamaschi, A. Martínez, and G. Pini, "An efficient parallel MLPG method for poroelastic models," *Computer Modeling in Engineering & Sciences*, vol. 49, no. 3, pp. 191–215, 2009.

[11] L. Bergamaschi and A. Martínez, "Parallel inexact constraint preconditioners for saddle point problems," in *Proceedings of the 17th International Conference on Parallel Processing (Euro-Par'11)*, R. N. E. Jeannot and J. Roman, Eds., vol. 6853, part 2, pp. 78–89, Springer, Bordeaux, France, 2011, Lecture Notes in Computer Sciences.

[12] C. Janna, M. Ferronato, and G. Gambolati, "A block FSAI-ILU parallel preconditioner for symmetric positive definite linear systems," *SIAM Journal on Scientific Computing*, vol. 32, no. 5, pp. 2468–2484, 2010.

[13] M. Ferronato, C. Janna, and G. Pini, "Efficient parallel solution to large-size sparse eigenproblems with block FSAI preconditioning," *Numerical Linear Algebra with Applications*. In press.

[14] Lawrence Livermore National Laboratory, hypre user manual, software version 1.6.0. Center for Applied Scientific Computing (CASC), University of California, 2001.

[15] A. V. Knyazev, "Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method," *SIAM Journal on Scientific Computing*, vol. 23, no. 2, pp. 517–541, 2001.

[16] L. Yu. Kolotilina, A. A. Nikishin, and A. Yu. Yeremin, "Factorized sparse approximate inverse preconditionings. IV. Simple approaches to rising efficiency," *Numerical Linear Algebra with Applications*, vol. 6, no. 7, pp. 515–531, 1999.

[17] A. Martínez, L. Bergamaschi, M. Caliari, and M. Vianello, "A massively parallel exponential integrator for advection-diffusion models," *Journal of Computational and Applied Mathematics*, vol. 231, no. 1, pp. 82–91, 2009.

[18] C. Janna, M. Ferronato, and N. Castelletto, "BFSAI-IC OpenMP implementation," Release V1.0, January 2011, http://www.dmsa.unipd.it/~ferronat/software.html.

[19] M. Ferronato, C. Janna, and G. Gambolati, "Mixed constraint preconditioning in computational contact mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 45–48, pp. 3922–3931, 2008.

[20] N. Castelletto, M. Ferronato, G. Gambolati et al., "3D geomechanics in UGS projects: a comprehensive study in northern Italy," in *Proceedings of the 44th US Rock Mechanics Symposium*, Salt Lake City, Utah, USA, 2010.

[21] P. Teatini, M. Ferronato, G. Gambolati, D. Bau, and M. Putti, "An-thropogenic Venice uplift by seawater pumping into a heterogeneous aquifer system," *Water Resources Research*, vol. 46, Article ID W11547, 16 pages, 2010.

[22] A. V. Knyazev and M. E. Argentati, "Implementation of a preconditioned eigensolver using hypre," 2005, http://math.ucdenver.edu/~rargenta/index_files/rep220.pdf.

Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 367909, 12 pages
doi:10.1155/2012/367909

Research Article

Comparison of Algebraic Multigrid Preconditioners for Solving Helmholtz Equations

Dandan Chen, Ting-Zhu Huang, and Liang Li

School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China

Correspondence should be addressed to Ting-Zhu Huang, tingzhuhuang@126.com

Received 6 December 2011; Revised 6 February 2012; Accepted 6 February 2012

Academic Editor: Edmond Chow

An algebraic multigrid (AMG) with aggregation technique to coarsen is applied to construct a better preconditioner for solving Helmholtz equations in this paper. The solution process consists of constructing the preconditioner by AMG and solving the preconditioned Helmholtz problems by Krylov subspace methods. In the setup process of AMG, we employ the double pairwise aggregation (DPA) scheme firstly proposed by Y. Notay (2006) as the coarsening method. We compare it with the smoothed aggregation algebraic multigrid and meanwhile show shifted Laplacian preconditioners. According to numerical results, we find that DPA algorithm is a good choice in AMG for Helmholtz equations in reducing time and memory. Spectral estimation of system preconditioned by the three methods and the influence of second-order and fourth-order accurate discretizations on the three techniques are also considered.

1. Introduction

In this paper, the time-harmonic wave equation in 2D homogeneous media is solved numerically. The essential equation is Helmholtz equation, which governs wave scattering and propagation phenomena arising in acoustic problems in many areas, such as geophysics, aeronautics, and optical problems. In particular, we are in search of solutions of the Helmholtz equation discretized by the finite difference method. The discrete problem becomes extremely large for very high wavenumber, because the number of gridpoints per wavelength should be sufficiently large in order to result in accepted solutions. In this case, direct methods are difficult to solve, and iterative methods are the interesting alternative. However, Krylov subspace iterative methods are not competitive without a good preconditioner. In this paper, we consider an algebraic multigrid with aggregation scheme as preconditioning to improve the convergence of Krylov subspace iterative methods.

In [1], Bayliss et al. proposed a preconditioner based on the Laplace operator for solving the discrete Helmholtz equation efficiently with CGNR. In [2], Laird and Giles proposed a preconditioner where an extra term is added to the Laplace operator for solving the discrete Helmholtz equation. Subsequently, in [3], Erlangga et al. generalized the above two kinds of preconditioners and obtained a new class of preconditioners, the so-called "shifted Laplacican" preconditioner of the form $-\Delta\phi - \alpha k^2\phi$ with $\alpha = -(\beta_1 - \beta_2 i) \in \mathbb{C}$, where $i = \sqrt{-1}$ is the imaginary unit. In 2006, Erlangga et al. [4] compared multigrid with incomplete LU used to approximate the shifted Laplacian preconditioner to construct the final preconditioner for the inhomogeneous Helmholtz equation, and concluded that multigrid applied to approximate the shifted Laplacian preconditioner with Bi-CGSTAB resulted in a fast and robust method. In 2006, Erlangga et al. [5] proposed a multigrid $V(1,1)$-cycle with de Zeeuw's prolongation operator, FW (full weighted) restriction, and Jacobi smoothing with the relaxation parameter $\omega = 0.5$. The smallest size of the parameter β_2 in front of the imaginary Helmholtz term in the preconditioner, for which the multigrid method could be successfully employed, has been determined to $\beta_2 = 0.5$ and meanwhile $\beta_1 = 1$. In 2007, Van Gijzen et al. [6] analyzed the spectral of the discrete Helmholtz operator preconditioned with a shifted Laplacian, and proposed an optimal value for the shift by combining the results of the spectral analysis with an upper bound on the GMRES residual norm. In 2009, Umetani et al. [7] proposed a multigrid $V(0,1)$-cycle with AMG's prolongation operator, FW restriction, and incomplete LU postsmoothing for a fourth-order Helmholtz discretization based on [5]. The fourth-order accurate shifted Laplacian preconditioner could be easily approximated by one $V(0,1)$-cycle of multigrid and enables us to choose a somewhat smaller imaginary shift parameter ($\beta_2 = 0.4$) in the shifted Laplacian preconditioner, which improves the solvers convergence (especially for high wavenumbers on fine meshes). In 2007, Airaksinen et al. [8] proposed a preconditioner based on an algebraic multigrid approximate of the inverse of a shifted Laplacian for the Helmholtz equation. This is a generalization of the preconditioner proposed by Erlangga et al. in [5]. In 2010, Olson and Schroder [9] proposed a smoothed aggregation algebraic multigrid method for 1D and 2D scalar Helmholtz problems with exterior radiation boundary conditions discretized by 1D finite difference and 2D discontinues Galerkin. In the same year, Notay [10] proposed an aggregation-based algebraic multigrid (AGMG) method, in which double pairwise aggregation scheme firstly proposed by Notay in [11] is used.

We will consider using the double pairwise aggregation algorithm to coarsen in the setup process of the general algebraic multigrid method in this paper in order to improve the solution effects of Helmholtz problems.

This paper is organized as follows. In Section 2, we discuss the Helmholtz equation, and the discrete finite difference formulations of second order and fourth order. The iterative solution methods with the three different kinds of preconditionings are presented in Section 3. Numerical results are reported in Section 4, and conclusions are given in Section 5.

2. The Helmholtz Equation and Discretizations

2.1. Mathematical Problem Definition

We solve wave propagations in a two dimensional medium in a unit domain governed by the Helmholtz equation

$$-\Delta\phi - k^2(x,y)\phi = g, \qquad \Omega = [0,1]^2, \tag{2.1}$$

Table 1: Discretization steps related to wavenumbers.

k	10	20	30	40	50	60	70	80	90
h	1/16	1/32	1/48	1/64	1/80	1/96	1/112	1/128	1/144

where $\Delta \equiv \partial^2/\partial x^2 + \partial^2/\partial y^2$ is the Laplace operator, $\phi(x, y)$ represents the pressure field in the frequency domain, the source term is denoted by g, and $k(x, y)$ is the wavenumber, which is a constant in the homogeneous domain. Otherwise, the wavenumber $k = \omega/c(x, y)$ is space dependent because of a spatially dependent speed of sound $c(x, y)$ in a heterogeneous medium. With angular frequency $\omega = 2\pi f$ (f is the frequency in Hertz), wavelength ℓ is defined by $\ell = c/f$. So the wavenumber $k = 2\pi/\ell$. The number of wavelengths in a domain of size L equals L/ℓ. n_ω, the number of points per wavelength, is typically chosen to be 10–12 points. A dimensionless discretization step reads $h = \ell/(n_\omega L)$, and therefore $kh = 2\pi/(n_\omega L)$. With domain size $L = 1$, an accuracy requirement for second-order discretizations is that $kh \leq \pi/5$ (≈ 0.63) for $n_\omega = 10$ points per wavelength, and $kh \leq 0.53$ with $n_\omega = 12$ points per wavelength. We select combinations of wavenumber k and the discretization step h with $kh = 0.625$, which is displayed in Table 1, in the latter numerical experiments.

The boundary condition can be the Dirichlet boundary condition or the first-order radiation boundary condition, and they are, respectively, as follows:

$$\phi = 0, \quad \text{on } \Gamma = \partial\Omega, \quad \text{(Dirichlet boundary condition)} \tag{2.2}$$

$$\frac{\partial \phi}{\partial n} - ik\phi = 0, \quad \text{on } \Gamma = \partial\Omega, \quad \text{(Sommerfeld condition)}, \tag{2.3}$$

where n is an outward direction normal to the boundary.

2.2. Finite Difference Discretizations

The Helmholtz equations are discretized either by a second-order or by a fourth-order finite difference scheme, resulting in the linear system

$$A_h \phi_h = b_h, \tag{2.4}$$

where ϕ_h and b_h represent the discrete frequency-domain pressure field and the source, respectively.

The well-known 5-point stencil related to a second-order ($o(h^2)$) accurate discretization for Helmholtz equation (2.1) reads

$$A_h \triangleq \frac{1}{h^2} \begin{bmatrix} & -1 & \\ -1 & 4 - (kh)^2 & -1 \\ & -1 & \end{bmatrix}. \tag{2.5}$$

The fourth-order ($o(h^4)$) accurate discretization named by HO discretization [12] for Helmholtz equation (2.1) is given with stencil

$$A_h^{\mathrm{HO}} \triangleq \frac{1}{h^2}\begin{bmatrix} -\frac{1}{6} & -\frac{2}{3}-\frac{(kh)^2}{12} & -\frac{1}{6} \\ -\frac{2}{3}-\frac{(kh)^2}{12} & -\frac{10}{3}-\frac{2(kh)^2}{3} & -\frac{2}{3}-\frac{(kh)^2}{12} \\ -\frac{1}{6} & -\frac{2}{3}-\frac{(kh)^2}{12} & -\frac{1}{6} \end{bmatrix}, \tag{2.6}$$

Compared with a second-order discretization method, a fourth-order discretization method could make the number of grid points per wavelength be smaller and discrete the equations to smaller matrices for the same level of accuracy. It is possible to lead to an algorithm that is more efficient in terms of the accuracy of the solution versus the computational cost. As in [7], the multigrid-based shifted Laplacian preconditioner with the fourth-order discretization is preferable.

The boundary discretization is also considered, and we could employ the first-order forward or backward scheme to approximate the first-order derivative in (2.3).

Next, we can obtain a linear system through substituting

$$A_h x = b_h, \quad A_h \in \mathbb{C}^{N\times N}, \tag{2.7}$$

where A_h is a large, spare symmetric matrix and N is the number of grid points. The matrix A_h remains positive definite as long as k^2 is smaller than the minimum eigenvalue of the discrete Laplacian, is indefinite with both positive and negative eigenvalues for large values of k, and is complex-valued because of absorbing boundary conditions. However, it is non-Hermitian. Apparently, the size of the linear system (2.7) gets very large for the high frequency.

3. Iterative Solution Methods

3.1. Preconditioned Krylov Subspace Iterative Methods

Iterative methods for linear system (2.7) within the class of Krylov subspace method are based on the construction of iterants in the subspace

$$\mathbf{K}^j(A, r_0) = \mathbf{span}\left\{r_0, Ar_0, A^2r_0, \ldots, A^{j-1}r_0\right\}, \tag{3.1}$$

where $\mathbf{K}^j(A, r_0)$ is the jth Krylov subspace associated with A and r_0, $r_0 = b - Ax_0$, is the initial residual.

The Bi-CGSTAB algorithm [13] is one of the better known Krylov subspace algorithms for non-Hermitian problems, which has been used for Helmholtz problems, for example, in [5, 8]. One of the advantages of Bi-CGSTAB, compared to full GMRES, is its limited memory requirements. Bi-CGSTAB is based on the idea of computing two mutually biorthogonal bases for the Krylov subspaces based on matrix A_h and its conjugate transpose A_h^H and is easy to implement.

Without a preconditioner, Krylov subspace methods converge very slowly, or not at all, for the problem of interest in [4]. So a preconditioner should be incorporated to improve the convergence of Krylov subspace methods. By left preconditioning, one solves a linear system premultiplied by a preconditioning matrix M_h^{-1},

$$M_h^{-1}A_h = M_h^{-1}b_h. \tag{3.2}$$

The challenge is to find a form of matrix M_h, whose inverse matrix M_h^{-1} can be efficiently approximated, such that $M_h^{-1}A_h$ has a spectrum that is favorable for Krylov subspace iterative solution methods.

In this paper, we choose Bi-CGSTAB and GMRES with preconditionings to solve discrete Helmholtz equations. In [3], GMRES is used to solve the Helmholtz equation with the shifted Laplacian preconditioner and is compared with Bi-CGSTAB. As a result, Bi-CGSTAB is preferable since the convergence for heterogeneous high wavenumber Helmholtz problems is typically faster than that of GMRES.

3.2. Shifted Laplacian (SL) Preconditioners

In [5], a shifted Laplacian operator was proposed as a preconditioner for the Helmholtz equation, with M_h defined as a discretization of

$$M_{\mathrm{SL}} = -\partial xx - \partial yy + \alpha k^2(x,y), \quad \alpha \in \mathbb{C},\ \alpha = -(\beta_1 - \beta_2 i), \tag{3.3}$$

and boundary conditions were set identically to those for the original Helmholtz equation. The readers could refer to the related contents in [4, 7, 8, 14].

There are different choices of the parameter pair (β_1, β_2) such as $(\beta_1, \beta_2) = (0,0)$ (Laplace preconditioner in [1]), $(\beta_1, \beta_2) = (-1,0)$ (Laird and Giles preconditioner in [2]), $(\beta_1, \beta_2) = (0,1)$ (Erlangga et al. preconditioner in [3]), $(\beta_1, \beta_2) = (1,1)$ (basic parameter pair choice). In [5], Erlangga et al. evaluated the influence of parameters β_1 and β_2 to the multigrid method which was applied to approximate the shifted Laplacian operator to construct the final preconditioner, and the proposed optimal parameter pair for the solver was that $(\beta_1, \beta_2) = (1, 0.5)$. Based on [5], Umetani et al. [7] considered using the fourth-order accurate finite difference discretization with the result that the parameter pair $(\beta_1, \beta_2) = (1, 0.4)$ is the best choice. In this paper, we merely consider the shifted Laplacian preconditioners, without the multigrid or incomplete LU method to approximate, which is not the same as that in [5, 7]. So we will also choose that $(\beta_1, \beta_2) = (1, 0.5)$ and $(\beta_1, \beta_2) = (1, 0.3)$ for comparison with the previously mentioned parameter pair (β_1, β_2) in the numerical experiments.

3.3. Smoothed Aggregation (SA) Preconditioning

In this section, we will present an overview of the smoothed aggregation setup algorithm. The function of the SA setup phase is to construct coarse operators A_k through interpolation operators

$$P_k : \Re^{n_{k+1}} \longrightarrow \Re^{n_k}, \tag{3.4}$$

(1) $t_\ell = S_\ell^{\nu_1}\ (\text{ones}(n,1),0)$
(2) While $\ell \le L$
(3) [neighbour, hoods] = aggregate (n_ℓ, A_ℓ);
(4) $[P, t_{\ell+1}]$ = qr$(n_\ell, t_\ell, \text{hoods})$;
(5) S_ℓ = smoother $(n_\ell, A_\ell, \omega, \text{neighbour})$;
(6) $I_{\ell+1}^{\ell} = S_\ell P$;
(7) $A_{\ell+1} = (I_{\ell+1}^{\ell})^T A_\ell I_{\ell+1}^{\ell}$;
(8) $\ell = \ell + 1$;
(9) Endwhile

Algorithm 1: SA_setup (n_ℓ, A_ℓ, ℓ).

where n_k and n_{k+1} are sizes of two successively coarser grids. The initial matrix A_0 associated with the finest level equals A. The inputs need the matrix A and one near null-space candidate t that is intended to form the interpolation basis. t is an algebraically smooth mode that is slow to relax on the fine mesh. The details are described in Algorithm 1.

When the current coarse level $\ell \le L$(L is the coarsest level), we gain two sets respectively named by "neighbour" and "hoods." The "neighbor $[i]$" includes the nodes which are strongly connective to the node i, and the "hoods $[i]$" contains the nodes which are in the ith aggregation set. Next, we reset t_ℓ according to "hoods," and QR factorization is applied to it in order to get the initial interpolation operator P and the next $t_{\ell+1}$. Third, we obtain the smooth operator S_ℓ according to the relative algorithm in [15]. Fourth, we get the final interpolation operator $I_{\ell+1}^{\ell}$, through the smooth operator S_ℓ is acted on the initial interpolation operator P. Finally, we can easily obtain the coarse operator $A_{\ell+1}$. The interested readers could consult the related contents in [9]. We will employ the SA setup algorithm in AMG to approximate the original system in order to obtain the preconditioner and we name this process by SA preconditioning.

3.4. Double Pairwise Aggregation (DPA) Preconditioning

An algebraic coarsening algorithm sets up an interpolation operator P only making use of the information available in the fine grid matrix A. The interpolation operator P is a $n \times n_c$ matrix, where n_c ($n_c \le n$) is the number of coarse variables. With acted on by the interpolation operator, a vector defined on the coarse variable set $[1, n_c]$ could be transferred on the fine grid. The coarsen grid matrix A_c depends not only on the interpolation matrix P, but also on the restriction matrix R and the fine grid matrix A. It is usual to take the restriction operator R equals to the transpose of the interpolation operator P. Therefore, the coarsen grid matrix A_c is computed from the Galerkin formula

$$A_c = RAP = P^T AP. \tag{3.5}$$

Coarsening by aggregation needs to define aggregates G_i, which are disjoint subsets of the variable set. The number of these aggregations is the number of coarse variables n_c, and the interpolation operator P is given by

$$P_{i,j} = \begin{cases} 1, & \text{if } i \in G_j, \\ 0, & \text{otherwise} \end{cases} \quad (1 \le i \le n,\ 1 \le j \le n_c). \tag{3.6}$$

Note that there is no need to explicitly form interpolation operator P, and the coarse grid matrix is practically computed by

$$(A_c)_{i,j} = \sum_{k\in G_i} \sum_{\ell\in G_j} a_{k\ell} \quad (1 \le i, j \le n_c). \tag{3.7}$$

The double pairwise aggregation algorithm was first proposed by Notay in [11], which employed two passes of the pairwise aggregation algorithm, with the result of decreasing the number of variables by a factor slightly less than four in most cases. Subsequently, we will take the double pairwise aggregation scheme in the setup process of AMG, in order to construct a better preconditioner with Bi-CGSTAB for discrete Helmholtz equations. We also name this process by DPA preconditioning. The details of aggregation algorithms could be consulted in [10, 11].

4. Experiments

We will consider the following problem and use a uniform mesh with constant mesh size h in all directions. In the experiments, we will compare the three preconditionings combined with Bi-CGSTAB or GMRES for solving discrete Helmholtz equations. The shifted Laplacian preconditioner (SLP), the smoothed aggregation preconditioning (SAP), and the double pairwise aggregation preconditioning (DPAP) are, respectively, introduced in Section 3. We let that the iterative method is terminated at the kth step if $\|b - Ax_k\|_2/\|b\|_2 \le 10^{-6}$. All the numerical results are from running relative Matlab programs in the computer with the CPU of AMD Pentium Dual Core 4000+.

4.1. The Close-off Problem and Numerical Results

We consider a problem in a rectangular homogeneous medium governed by

$$-\Delta\phi - k^2\phi = \sin(\pi x)\sin(\pi y)\sin\left(\sqrt{2}\pi x\right)\sin\left(\sqrt{3}\pi y\right), \quad x = [0,1],\ y = [0,1], \tag{4.1}$$

$\phi = 0$, at the boundaries.

Different grid resolutions are used to solve this problem with various wavenumbers in Table 1.

Firstly, we will observe the distribution of eigenvalues of the systems preconditioned by the three different preconditionings in order to estimate which preconditioning is best combined with Krylov subspace iterative methods. Since the spectral analysis of shifted Laplacian preconditioners $(M_{\mathrm{SL}})_h$ with different parameter pairs (β_1, β_2) was considered in many papers [5–7, 14] and $(\beta_1, \beta_2) = (1, 0.5)$ is an almost better choice, we will only present the distribution of eigenvalues of the system preconditioned by the shifted Laplacian preconditioner $(M_{\mathrm{SL}})_h$ with $(\beta_1, \beta_2) = (1, 0.5)$ with second-order and fourth-order accurate discretizations. We employ two-gird process to approximate V-cycle of AMG, and so $M_{\mathrm{AMG}}^{-1}A = I - S(I - PA_{\mathrm{C}}^{-1}P^TA)S$, where I is unit matrix and S is smoothing matrix. Second-order and fourth-order accurate discretizations are taken into account. Next, we note preconditioners produced by the smoothed aggregation preconditioning and by the double pairwise aggregation preconditioning separately as $(M_{\mathrm{SA}})_h$ and $(M_{\mathrm{DPA}})_h$. In Figures 1(a),

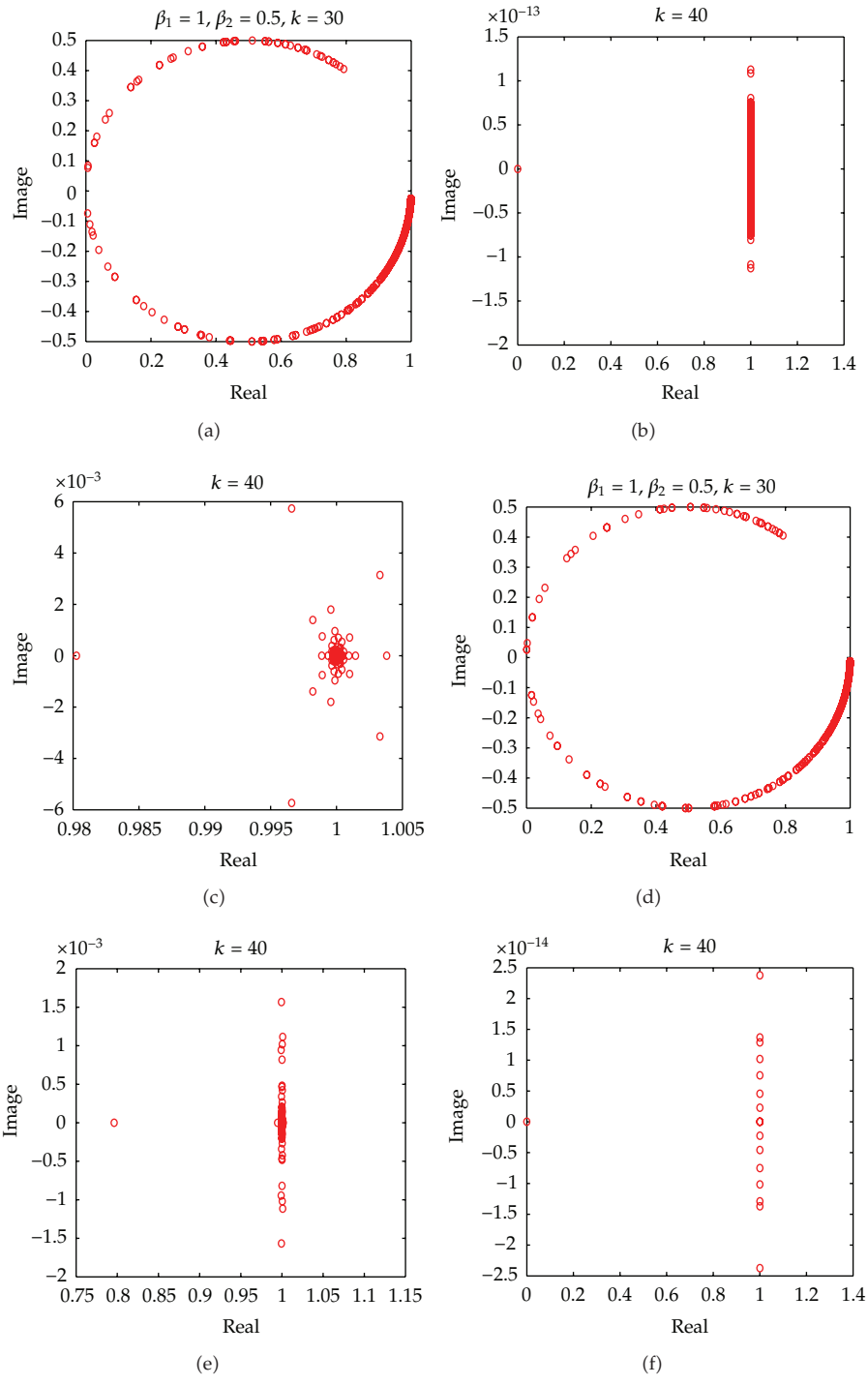


Figure 1: Spectral pictures.

Table 2: Computational performance of GMRES with SLPs.

k	$(\beta_1,\beta_2)=(0,0)$		$(\beta_1,\beta_2)=(-1,0)$		$(\beta_1,\beta_2)=(0,1)$		$(\beta_1,\beta_2)=(1,0.5)$		$(\beta_1,\beta_2)=(1,0.3)$	
	Iter	T_solve (s)	Iter	T_solve (s)	Iter	T_solve (s)	Iter	T_solve (s)	Iter	T_solve (s)
10	15	0.593	19	0.282	18	0.625	13	0.39	11	3.422
20	50	1.218	64	1.406	56	98.032	38	2.046	29	6.328
30	99	4.547	125	5.203	113	185.468	69	170.484	—	—

Table 3: Computational performance of Bi-CGSTAB with SAP.

k	SAP (T_O)			SAP (F_O)		
	Iter	T_setup (s)	T_solve (s)	Iter	T_setup (s)	T_solve (s)
10	5	4.484	0.328	5	4.391	0.265
20	7	72.718	0.500	5	72.516	0.391
30	4	417.172	0.672	4	419.953	0.703
40	6	3114.88	14.578	6	1586.28	6.375

Note: presmooth = 0, postsmooth = 1, smoother = ILU, setup_Alg = sagg, V_cycle.

1(b), and 1(c) are, respectively, spectral pictures of $(M_{\mathrm{SL}})_h^{-1}A_h$, $(M_{\mathrm{SA}})_h^{-1}A_h$ and $(M_{\mathrm{DPA}})_h^{-1}A_h$ with second-order accurate discretization, and Figures 1(d), 1(e), and 1(f) are respectively spectral pictures of $(M_{\mathrm{SL}})_h^{-1}A_h$, $(M_{\mathrm{SA}})_h^{-1}A_h$, and $(M_{\mathrm{DPA}})_h^{-1}A_h$ with fourth-order accurate discretization. Meanwhile, we compute the condition number of the preconditioned systems, and $\mathrm{cond}_2((M_{\mathrm{SL}})_h^{-1}A_h) = 211.1400$, $\mathrm{cond}_2((M_{\mathrm{SA}})_h^{-1}A_h) = 1.3062$, and $\mathrm{cond}_2((M_{\mathrm{DPA}})_h^{-1}A_h) = 1.3078$ separately correspond to Figures 1(a), 1(b), and 1(c).

Next, we will present numerical results in Tables 2, 3, and 4 and explain the meaning of signs in them. Tables 2, 3, and 4, respectively, show the computational performance of GMRES with SLPs, Bi-CGSTAB with SAP, and Bi-CGSTAB with DPAP for solving the closed-off problem with finite difference discretizations, in terms of the number of iterations and computational time, to reach the specified convergence. We also test Bi-CGSTAB with SLPs, but it leads to more iterations and solve time. So we do not present the corresponding results. Meanwhile, we find that Bi-CGSTAB with SLPs when $\beta_1 = 1$ and β_2 in the interval $(0,1)$ is better than GMRES, and the fourth-order accurate discretization could make the iterative solution rate more quick. Nevertheless, we make the focus only on the second-order accurate discretization for shifted Laplacian preconditioners, so the second-order accurate discretization is used in Table 2. "Iter" denotes the number of iterations, and "—" means that the corresponding experimental result is not given. "T_setup (s)," and "T_solve (s)" respectively, denote the setup time of preconditioner and the time of iterative solution by the second, and "T_O" and "F_O," respectively, mean the second-order and fourth-order finite difference discretizations shown in Section 2. The values of "presmooth" and "postsmooth" respectively, denote the number of pre- and postsmoothing iterations in the solve process of AMG, "smoother" denotes the chosen smoothing method, and "setup_Alg" denotes the coarsening algorithm used in the setup process of AMG.

4.2. Analysis of Numerical Results

We will analyze the numerical results from the following aspects.

Firstly, from Figure 1, by comparing pictures (a), (b), (c), we find that the eigenvalues of (b) and (c) are more farther from zero point and more close with one, but the eigenvalues of

Table 4: Computational performance of Bi-CGSTAB with DPAP.

k	DPAP(T_O)			DPAP(F_O)		
	Iter	T_setup(s)	T_solve(s)	Iter	T_setup(s)	T_solve(s)
10	3	0.968	0.547	3	0.922	0.453
20	3	2.938	0.531	2	3.313	0.421
30	2	9.031	0.578	3	10.844	0.812
40	3	22.828	1.265	4	29.969	1.703
50	4	57.391	2.859	3	70.594	2.125
60	3	113.17	3.344	11	144.844	12.032
70	4	202.578	7.000	6	273.75	14.375
80	3	339.531	7.547	5	453.469	12.281
90	4	561.031	13.766	8	723.343	40.765

Note: presmooth = 1, postsmooth = 1, smoother = ILU, setup_Alg = dpagg, V_cycle.

(b) contain zero point. So we could boldly deduce that DPAP and SAP are more beneficial for Krylov subspace iterative method. Perhaps DPAP is better than SAP since the eigenvalues of (b) contain zero point. Next, we contrast (a), (b), (c), respectively, with (d), (e), (f). We notice that the higher accurate discretization is possible to improve the solution efficiency of SLP, since the fourth-order discretization makes eigenvalues more close to one in (d), is probably good for SAP for avoiding the zero point eigenvalue, and is likely bad for DPAP because of appearing the zero point eigenvalue. As we all know, the larger the condition number of matrix is, the worse the numerical stability of solution of the corresponding matrix equation is. So the system preconditioned by the shifted Laplacian operator has worse numerical stability, which means that the small change of the right hand will bring large change of solution, since $\mathrm{cond}_2((M_{\mathrm{SL}})_h^{-1}A_h) = 211.1400$ is much larger. It is perhaps the reason for using multigrid or LU to approximate the shifted Laplacian operator to obtain final preconditioner, such as [4, 5, 7, 8].

Secondly, from Table 2, we could easily see that the number and time of iterative solution are both going up largely with the increasing wavenumber. The shifted Laplacian preconditioner degenerates into the Laplace preconditioner when $(\beta_1, \beta_2) = (0, 0)$. Next, we find out that the number of iterations is smallest when $(\beta_1, \beta_2) = (1, 0.3)$, but it is not best in terms of the time of iterative solution. In general, the best choice is that $(\beta_1, \beta_2) = (1, 0.5)$ by considering both aspects of the number and time of iterative solution.

Thirdly, from Table 3, we find that the fourth-order accurate discretization only affects the iterative solution a little, and it is little better than the second-order accurate discretization when the wavenumber $k = 10, 20, 40$. However, only the low wavenumbers k are computed in Table 3 because of the too long time of iterative solution or the inadequate memory for higher wavenumbers. From Table 4, we discover that the fourth-order accurate discretization does not evidently improve the efficiency of iterative solution and that the second-order accurate discretization looks more stable in the aspect of the number of iterations. The fourth-order accurate discretization improves the efficiency of iterative solution both on the sides of setup-time and solve time when the wavenumber $k = 10, 20$ but costs more time and iterations for the higher wavenumber $k = 30, \ldots, 90$. So we could conclude that the second-order accurate discretization combined with DPAP is preferable.

Finally, to compare results of Tables 2, 3, and 4, according to the number of iterations, we can easily find that Bi-CGSTAB with DPAP and second-order accurate discretization is preferable. However, presmooth = 0 is chosen in Table 3, and, when we take presmooth

= 1 for Bi-CGSTAB with SAP, the iterative steps will be smaller but still not better than Bi-CGSTAB with DPAP, with leading to the longer solve time for the added smoothing process. Following, we contrast solve time and setup time with second-order accurate discretization in Tables 2, 3, and 4. Then, we see that GMRES with SLP with $(\beta_1, \beta_2) = (-1, 0)$ need the least solve-time for wavenumber $k = 10$, Bi-CGSTAB with SAP has the least time of iterative solution for wavenumber $k = 20$, and Bi-CGSTAB with DPAP has the quickest iterative rate for wavenumber $k = 30, 40$. SAP has much more setup time than DPAP for the same wavenumber, and setup time of SAP increases more rapidly than that of DPAP. Though setup time will lead to the whole solution time longer, if there are some matrix equations with the same left matrix and the different right hands, DPAP will excel for higher wavenumber because of needing the setup process once. The results of preconditioning for wavenumber $k = 50, \ldots, 90$ are only shown in Table 4, and the reason is that the larger the wavenumber k is, the longer the solution time is for higher wavenumber. So we do not present the results for the higher wavenumber in Tables 2 and 3 due to the limited experiment condition. Next, we observe that, with the increasing wavenumber k, the number of iterative solution changes between linearly and squarely in Table 2, but it fluctuates between 4 and 7 in Table 3 and between 2 and 4 in Table 4. In other words, the number of iterative solution is not related to the wavenumber k when Bi-CGSTAB with DPAP and SAP is employed. In contrast with Bi-CGSTAB with SAP in Table 3, Bi-CGSTAB with DPAP in Table 4 makes the time of iterative solution change more slowly while the wavenumber k is increasing.

5. Conclusions

As is known to all, Helmholtz equations are difficult to solve, especially with high wavenumbers. The reason is that the high wavenumber is able to make the corresponding matrices of Helmholtz equations highly indefinite. The original Krylov subspace iterative method is not efficient to solve these problems. Preconditioning is needed to apply into Krylov subspace iterative methods in order to improve the efficiency of solution. We consider using the algebraic multigrid method to construct the preconditioner and look for a concrete AMG method which is fit for Helmholtz problems. According to numerical results and analysis in Section 4, the double pairwise aggregation is applied in the setup process of AMG and finally better results are gained. Meanwhile, the number of iterative solution is independent on the wavenumber, with the result that Helmholtz equations with high wavenumbers can be solved more efficiently in terms of time and memory. The methods that are Bi-CGSTAB with DPAP and with SAP get fewer number of iterative solution however, SAP prolongs the setup time in contrast with DPAP. The two kinds of preconditionings are obviously more efficient than the shifted Laplacian preconditioners. In view of the above, we can draw the conclusion that the double pair aggregation algorithm in the setup process of AMG is a good choice to solve Helmholtz equations especially with high wavenumbers. Though only the 2 D close-off problem with Dirichlet boundary condition is experimented in Section 4, this conclusion is also possible to generalize to Helmholtz equations with other boundary conditions.

Acknowledgments

This research is supported by NSFC (60973015, 61170311), Sichuan Province Sci. and Tech. Research Project (2011JY0002, 12ZC1802), the tianyuan foundation. Chinese Universities Specialized Research Fund for the Doctoral Program (20110185110020).

References

[1] A. Bayliss, C. I. Goldstein, and E. Turkel, "An iterative method for the Helmholtz equation," *Journal of Computational Physics*, vol. 49, no. 3, pp. 443–457, 1983.

[2] A. L. Laird and M. B. Giles, *Preconditioned Iterative solution of the 2D Helmholtz equation, report 02/12*, Oxford Computer Laboratory, Oxford, UK, 2002.

[3] Y. A. Erlangga, C. Vuik, and C. W. Oosterlee, "On a class of preconditioners for solving the Helmholtz equation," *Applied Numerical Mathematics*, vol. 50, no. 3-4, pp. 409–425, 2004.

[4] Y. A. Erlangga, C. Vuik, and C. W. Oosterlee, "Comparison of multigrid and incomplete LU shifted-Laplace preconditioners for the inhomogeneous Helmholtz equation," *Applied Numerical Mathematics*, vol. 56, no. 5, pp. 648–666, 2006.

[5] Y. A. Erlangga, C. W. Oosterlee, and C. Vuik, "A novel multigrid based preconditioner for heterogeneous Helmholtz problems," *SIAM Journal on Scientific Computing*, vol. 27, no. 4, pp. 1471–1492, 2006.

[6] M. B. van Gijzen, Y. A. Erlangga, and C. Vuik, "Spectral analysis of the discrete Helmholtz operator preconditioned with a shifted Laplacian," *SIAM Journal on Scientific Computing*, vol. 29, no. 5, pp. 1942–1958, 2007.

[7] N. Umetani, S. P. MacLachlan, and C. W. Oosterlee, "A multigrid-based shifted Laplacian preconditioner for a fourth-order Helmholtz discretization," *Numerical Linear Algebra with Applications*, vol. 16, no. 8, pp. 603–626, 2009.

[8] T. Airaksinen, E. Heikkola, A. Pennanen, and J. Toivanen, "An algebraic multigrid based shifted-Laplacian preconditioner for the Helmholtz equation," *Journal of Computational Physics*, vol. 226, no. 1, pp. 1196–1210, 2007.

[9] L. N. Olson and J. B. Schroder, "Smoothed aggregation for Helmholtz problems," *Numerical Linear Algebra with Applications*, vol. 17, no. 2-3, pp. 361–386, 2010.

[10] Y. Notay, "An aggregation-based algebraic multigrid method," *Electronic Transactions on Numerical Analysis*, vol. 37, pp. 123–146, 2010.

[11] Y. Notay, "Aggregation-based algebraic multilevel preconditioning," *SIAM Journal on Matrix Analysis and Applications*, vol. 27, no. 4, pp. 998–1018, 2006.

[12] I. Singer and E. Turkel, "High-order finite difference methods for the Helmholtz equation," *Computer Methods in Applied Mechanics and Engineering*, vol. 163, no. 1–4, pp. 343–358, 1998.

[13] H. A. van der Vorst, "Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 13, no. 2, pp. 631–644, 1992.

[14] C. W. Oosterlee, C. Vuik, W. A. Mulder, and R.-E. Plessix, "Shifted-laplacian preconditioners for heterogeneous helmholtz problems," *Lecture Notes in Electrical Engineering*, vol. 71, pp. 21–46, 2010.

[15] C. Wagner, "Introduction to algebraic multigrid," Course Note of an Algebraic Multigrid Course at the university of Heidelberg in the Wintersemester, 1998.

Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 307939, 12 pages
doi:10.1155/2012/307939

Research Article

An Alternative HSS Preconditioner for the Unsteady Incompressible Navier-Stokes Equations in Rotation Form

Jia Liu

Department of Mathematics and Statistics, University of West Florida, Pensacola, FL 32514, USA

Correspondence should be addressed to Jia Liu, jliu@uwf.edu

Received 2 November 2011; Accepted 27 January 2012

Academic Editor: Kok Kwang Phoon

We study the preconditioned iterative method for the unsteady Navier-Stokes equations. The rotation form of the Oseen system is considered. We apply an efficient preconditioner which is derived from the Hermitian/Skew-Hermitian preconditioner to the Krylov subspace-iterative method. Numerical experiments show the robustness of the preconditioned iterative methods with respect to the mesh size, Reynolds numbers, time step, and algorithm parameters. The preconditioner is efficient and easy to apply for the unsteady Oseen problems in rotation form.

1. Introduction

We study the numerical solution methods of the incompressible viscous fluid problems with the following form:

$$\frac{\partial \mathbf{u}}{\partial t} - \nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega \times (0, \Gamma], \tag{1.1}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \times [0, \Gamma], \tag{1.2}$$

$$\mathcal{B}\mathbf{u} = \mathbf{g} \quad \text{on } \partial\Omega \times [0, \Gamma], \tag{1.3}$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0 \quad \text{in } \Omega. \tag{1.4}$$

Equations (1.1) to (1.4) are also known as the Navier-Stokes equations. Here Ω is an open set of $\mathbb{R}^d$, where $d = 2$, or $d = 3$, with boundary $\partial\Omega$; the variable $\mathbf{u} = \mathbf{u}(\mathbf{x}, t) \in \mathbb{R}^d$ is a vector-valued function representing the velocity of the fluid, and the scalar function

$p = p(\mathbf{x}, t) \in \mathbb{R}$ represents the pressure. The pressure field, p, is determined up to an additive constant. To uniquely determine p, we may impose some additional condition, such as

$$\int_{\Omega} p \, d\mathbf{x} = 0. \tag{1.5}$$

The source function $\mathbf{f}$ is given on Ω. Here $\nu > 0$ is a given constant called the kinematic viscosity, which is $\nu = O(\mathrm{Re}^{-1})$. Re is the Reynolds number: $\mathrm{Re} = VL/\nu$, where V denotes the mean velocity, and L is the diameter of Ω, see [1]. Also, Δ is the (vector) Laplacian operator in d dimensions, ∇ is the gradient operator, and $\nabla\cdot$ is the divergence operator. In (1.3) $\mathcal{B}$ is some boundary operator; for example, the Dirichlet boundary condition $\mathbf{u} = \mathbf{g}$; Neumann boundary condition $\partial\mathbf{u}/\partial\mathbf{n} = \mathbf{g}$, where $\mathbf{n}$ denotes the outward-pointing normal to the boundary, or a mixture of the two.

We use fully implicit time discretization and picard linearization to obtain a sequence of Oseen problems, that is, linear problems of the form

$$\alpha\mathbf{u} - \nu\Delta\mathbf{u} + (\mathbf{v}\cdot\nabla)\mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega, \tag{1.6}$$

$$\nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega, \tag{1.7}$$

$$\mathcal{B}\mathbf{u} = \mathbf{g} \quad \text{on } \partial\Omega. \tag{1.8}$$

Here $\mathbf{v}$ is a known divergence-free vector obtained from the previous linearized step (e.g., $\mathbf{v} = \mathbf{u}_k$). We call the vector $\mathbf{v}$ the wind function. In addition, $\alpha = O(1/\delta t)$ where δt is the time step. Equations (1.6)–(1.8) are referred to as the Oseen problem.

We can use either finite element or finite different methods to discretize (1.6)–(1.8). The resulting discrete system $\mathcal{A}\mathbf{u} = \mathbf{b}$ has the form

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & -\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ g \end{bmatrix}. \tag{1.9}$$

In this paper, we are interested in an alternative linearization of the steady-state Navier-Stokes equation. Based on the identity

$$(\mathbf{u}\cdot\nabla)\mathbf{u} = \frac{1}{2}\nabla(\mathbf{u}\cdot\mathbf{u}) + (\nabla\times\mathbf{u})\times\mathbf{u}. \tag{1.10}$$

In order to linearize it, we replace $\mathbf{u}$ in one place with a known divergence-free vector $\mathbf{v}$ which can be the solution obtained from the previous Picard iteration. In this case we have

$$(\mathbf{v}\cdot\nabla)\mathbf{u} \approx \frac{1}{2}\nabla(\mathbf{u}\cdot\mathbf{u}) + (\nabla\times\mathbf{v})\times\mathbf{u}. \tag{1.11}$$

After substituting the right-hand side into (1.6), we find that the corresponding linearized equations have the following form:

$$\frac{\partial \mathbf{u}}{\partial t} - \nu \Delta \mathbf{u} + \mathbf{w} \times \mathbf{u} + \nabla P = \mathbf{f} \quad \text{in } \Omega \times (0, T], \tag{1.12}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \times [0, T], \tag{1.13}$$

$$\mathcal{B}\mathbf{u} = \mathbf{g} \quad \text{on } \partial\Omega \times [0, T], \tag{1.14}$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0 \quad \text{in } \Omega, \tag{1.15}$$

where $P = p + (1/2)\|\mathbf{u}\|_2^2$ is the so-called *Bernoulli pressure*. For the two-dimensional case

$$\mathbf{w}\times = \begin{pmatrix} 0 & w \\ -w & 0 \end{pmatrix}, \tag{1.16}$$

where $w = \nabla \times \mathbf{v} = -\partial v_1/\partial x_2 + \partial v_2/\partial x_1$ is a scalar function.

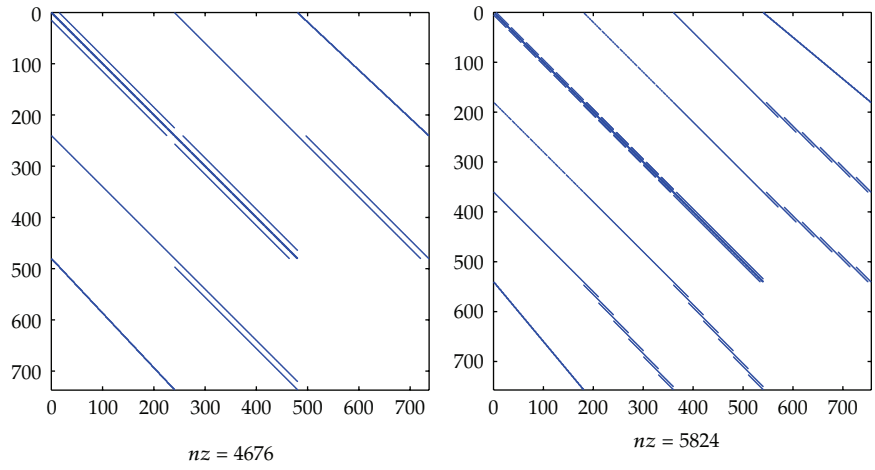
In the three-dimensional case, we have

$$\mathbf{w}\times = \begin{pmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{pmatrix}, \tag{1.17}$$

here $(w_1, w_2, w_3) = \mathbf{w} = \nabla \times \mathbf{v}$, where w_i denotes the ith component of $\nabla \times \mathbf{v}$. Assume $\mathbf{v} = (v_1, v_2, v_3)$, then we have the formal expression of w

$$\nabla \times \mathbf{v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \dfrac{\partial}{\partial x} & \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \\ v_1 & v_2 & v_3 \end{vmatrix}. \tag{1.18}$$

Here the divergence-free vector field $\mathbf{v}$ again denotes the approximate velocity from the previous Picard iteration. Note that when the "wind" function $\mathbf{v}$ is irrotational ($\nabla \times \mathbf{v} = 0$), (1.12)–(1.14) reduce to the Stokes problem. It is not difficult to see that the linearizations (1.6)–(1.8) and (1.12)–(1.14), although both conservative, are not mathematically equivalent. The momentum equation (1.12) is called the *rotation form*. We can see that no first-order terms in the velocities appear in (1.12); on the other hand, the velocities in the d scalar equations comprising (1.12) are now coupled due to the presence of the term $\mathbf{w}\times\mathbf{u}$. The disappearance of the convective terms suggests that the *rotation form* (1.12) of the momentum equations may be advantageous over the standard form (1.6) from the linear solution point of view. This observation was first made by Olshanskii and his coworkers in [2–5]. In their papers, they showed the advantages of the rotation form over the standard convection form in several aspects.



(a) 2D case using MAC (b) 3D case using MAC

Figure 1: Sparsity patterns for different types of the Oseen problem in rotation form.

After we discretize the Oseen problem in rotation form (1.12)–(1.14), we obtain the sparse linear system $\mathcal{A}\mathbf{x} = \mathbf{b}$, where

$$\mathcal{A} = \begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix}. \tag{1.19}$$

Here $\mathbf{A} = \mathbf{L} + \mathbf{K}$, where $\mathbf{L}$ is the discretization of the operator $\alpha - \nu\Delta$, and matrix $\mathbf{K}$ is the discretization of the term $\mathbf{w}\times$, where $\mathbf{w} = \nabla \times \mathbf{v}$. In the 2D case,

$$\mathbf{K} = \begin{bmatrix} \mathbf{0} & \mathbf{D} \\ -\mathbf{D} & \mathbf{0} \end{bmatrix}. \tag{1.20}$$

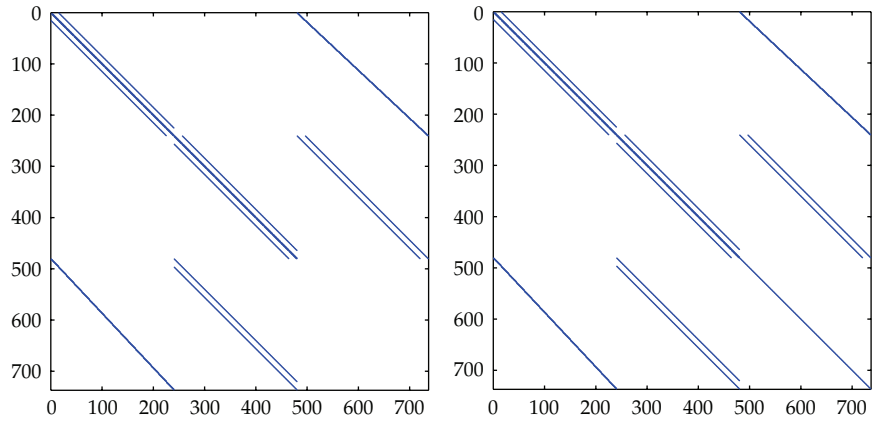
The rectangular matrix $\mathbf{B}$ is the discretization of the negative divergence, and $\mathbf{B}^T$ is the discretization of the gradient.

If we use a finite difference method, like Mac and Cell (MAC), see [6], then $\mathbf{D}$ is a diagonal matrix where its diagonal elements are the values of $\mathbf{w}$ evaluated at the grid edges. Matrix $\mathbf{D}$ is a weighted mass matrix if a finite element method is used. In the 3D case, we have

$$\mathbf{K} = \begin{bmatrix} \mathbf{0} & -\mathbf{D}_3 & \mathbf{D}_2 \\ \mathbf{D}_3 & \mathbf{0} & -\mathbf{D}_1 \\ -\mathbf{D}_2 & \mathbf{D}_1 & \mathbf{0} \end{bmatrix}. \tag{1.21}$$

Again matrices $\mathbf{D}_1, \mathbf{D}_2$, and $\mathbf{D}_3$ are all diagonal matrices or weighted mass matrices. Typical sparsity patterns for $\mathcal{A}$ in the 2D and 3D case are displayed in Figures 1(a) and 1(b).

For some discretization methods, a stabilization matrix needs to be added to the (2,2) block of $\mathcal{A}$, namely, a matrix $-\mathbf{C}$, where $\mathbf{C}$ is a symmetric positive semidefinite diagonal or



(a) without stabilization (b) with stabilization

Figure 2: Sparsity patterns for different types of the 2D Oseen problem in convection form.

scaled mass matrix, or scaled Laplacian with small norm. Figures 2(a) and 2(b) show the sparsity pattern for the coefficient matrix $\mathcal{A}$ with or without stabilization term in the 2D case. Such a stabilization is not necessary for the MAC discretization.

To solve the system $\mathcal{A}\mathbf{x} = \mathbf{b}$, we can consider the Krylov subspace methods with the preconditioning. Many powerful preconditioning techniques have been explored for the generalized Oseen problems, for example, Uzawa-type preconditioner, block and approximate schur complement preconditioner, pressure preconditioner, and so forth, see [7–11] for more details. However, there is no "best" preconditioner for the saddle point system. To find the "best" preconditioner, we would like to find a preconditioner P, such that the rate of convergence of the preconditioned Krylov subspace matrix is low and bounded independent of the mesh size, viscosity ν and time step α. In addition, the cost of the preconditioning steps must be low. In this paper, we describe such a new preconditioner that satisfies the above requirements in most of cases and demonstrate its utility.

A summary of the paper is as follows. Section 2 demonstrates the Alternative Hermitian and Skew-Hermitian (AHSS) preconditioner; studies some of its convergence properties and the application of the HSS preconditioner for Krylov subspace methods; Section 3 shows the results of a series of numerical experiments. Finally, section 4 summarizes the approach and future work.

2. The Alternative HSS Preconditioner

The alternative HSS preconditioner is based on the nonsymmetric formulation

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ -\mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ -g \end{bmatrix}. \tag{2.1}$$

We have analyzed the advantages of the nonsymmetric formulation in [12]. We gain positive (semi)-definiteness in this case. By changing the sign in front of the $(2,1)$ and $(2,2)$ blocks,

we obtain an equivalent linear system with a matrix whose spectrum is entirely contained in the half-plane $\Re(z) > 0$. (Here we use $\Re(z)$ to denote the real part of $z \in \mathbb{C}$).The spectra of the nonsymmetric formulation is more friendly to the convergence of Krylov subspace iterations. For an example, GMRES methods, see [13, 14].

We have investigated the preconditioner based on the Hermitian and Skew-Hermitian splitting methods for the Navier-Stokes problem, see [12, 15, 16]. However, the HSS preconditioner still has some problems. When the time step is not small enough or the viscosity is relative larger, the iteration number increases a lot. Therefore, we are trying to find another preconditioner which works better than the HSS preconditioner. We find out that if we use a different splitting of the coefficient matrix, we can get a very good results. The following splitting is the new preconditioner we will introduce in the paper. Letting $H \equiv (1/2)(A + A^T)$ and $K \equiv (1/2)(A - A^T)$, we have the following splitting of $\mathcal{A}$ into two parts:

$$\mathcal{A} = \begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ -\mathbf{B} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{H} & \mathbf{B}^T \\ -\mathbf{B} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{2.2}$$

We denote

$$\mathcal{H} = \begin{bmatrix} \mathbf{H} & \mathbf{B}^T \\ -\mathbf{B} & \mathbf{0} \end{bmatrix}, \qquad \mathcal{K} = \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{2.3}$$

Therefore, we defined the preconditioner as the following:

$$P_\rho = \frac{1}{2\rho}(\mathcal{H} + I_{n+m})(\mathcal{K} + I_{n+m}). \tag{2.4}$$

Here I_{n+m} denotes the identity matrix of order $n + m$, and $\rho > 0$ is a parameter.

Similar in spirit to the classical ADI (alternating-direction implicit) method, we consider the following two splittings of $\widehat{\mathcal{A}}$:

$$\mathcal{A} = (\mathcal{H} + \rho\mathcal{O}) - (\rho\mathcal{O} - \mathcal{K}), \qquad \mathcal{A} = (\mathcal{K} + \rho\mathcal{O}) - (\rho\mathcal{O} - \mathcal{H}). \tag{2.5}$$

Here $\mathcal{O}$ denotes the identity matrix of order $n + m$. Note that

$$\mathcal{H} + \rho\mathcal{O} = \begin{bmatrix} \mathbf{H} + \rho\mathbf{I}_n & \mathbf{B}^T \\ \mathbf{B} & \rho\mathbf{I}_m \end{bmatrix} \tag{2.6}$$

is the shifted discretized Stokes problem, where I_n denotes the identity matrix of order n, and I_m denotes the identity matrix of order m. We obtian that

$$\mathcal{K} + \rho\mathcal{O} = \begin{bmatrix} \mathbf{K} + \rho\mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \rho\mathbf{I}_m \end{bmatrix} \tag{2.7}$$

is nonsingular and has positive definite symmetric part.

Alternating between these two splittings leads to the the following iteration:

$$\begin{aligned}(\mathcal{H}+\rho\mathcal{D})\mathbf{u}_{k+1/2} &= (\rho\mathcal{D}-\mathcal{K})\mathbf{u}_k+\widehat{\mathbf{b}},\\ (\mathcal{K}+\rho\mathcal{D})\mathbf{u}_{k+1} &= (\rho\mathcal{D}-\mathcal{H})\mathbf{u}_{k+1/2}+\widehat{\mathbf{b}},\end{aligned} \tag{2.8}$$

($k = 0,1,\ldots$). Here $\widehat{\mathbf{b}}$ denotes the right-hand side of (1.9); the initial guess $\mathbf{u}_0$ is chosen arbitrarily. Elimination of $\mathbf{u}_{k+1/2}$ from (2.8) leads to a stationary (fixed-point) iteration of the form

$$\mathbf{u}_{k+1} = \mathcal{T}_\rho \mathbf{u}_k + \mathbf{c}, \quad k = 0,1,\ldots, \tag{2.9}$$

where $\mathcal{T}_\rho = (\mathcal{K}+\rho\mathcal{D})^{-1}(\rho\mathcal{D}-\mathcal{H})(\mathcal{H}+\rho\mathcal{D})^{-1}(\rho\mathcal{D}-\mathcal{K})$ is the iteration matrix and $\mathbf{c} := (\mathcal{S}+\rho\mathcal{D})^{-1}(\rho\mathcal{D}-\mathcal{H})(\mathcal{H}+\rho\mathcal{D})^{-1}(\rho\mathcal{D}-\mathcal{S})$. The iteration converges for arbitrary initial guesses $\mathbf{u}_0$ and right-hand sides $\mathbf{b}$ to the solution $\mathbf{u}_* = \widehat{\mathcal{A}}^{-1}$ if and only if $\varrho(\mathcal{T}_\rho) < 1$, where $\varrho(\mathcal{T}_\rho)$ denotes the spectral radius of $\mathcal{T}_\rho$.

Theorem 2.1. *Consider the problem* (2.1)*, that is,*

$$\begin{bmatrix}\mathbf{A} & \mathbf{B}^T\\ -\mathbf{B} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ p\end{bmatrix} = \begin{bmatrix}\mathbf{f}\\ -g\end{bmatrix}. \tag{2.10}$$

We assume that $\mathbf{A}$ *is positive real, and* $\mathbf{B}$ *has full rank. Then the iteration* (2.9) *from the splitting* (2.3) *is unconditionally convergent; that is,* $\varrho(\mathcal{T}_\rho) < 1$ *for all* $\rho > 0$ *and* $\alpha \geq 0$.

Proof. Consider the splitting (2.3). The iteration matrix $\mathcal{T}_\rho$ is similar to

$$\begin{aligned}\widehat{\mathcal{T}}_\rho &:= (\rho\mathcal{D}-\mathcal{H})(\mathcal{H}+\rho\mathcal{D})^{-1}(\rho\mathcal{D}-\mathcal{K})(\mathcal{K}+\rho\mathcal{D})^{-1},\\ &= \mathcal{R}\mathcal{U},\end{aligned} \tag{2.11}$$

where $\mathcal{R} := (\rho\mathcal{D}-\mathcal{H})(\mathcal{H}+\rho\mathcal{D})^{-1}$ is symmetric and $\mathcal{U} := (\rho\mathcal{D}-\mathcal{K})(\mathcal{K}+\rho\mathcal{D})^{-1}$.

By Kellogg's lemma, $\|\mathcal{R}\| = \|(\rho\mathcal{D}-\mathcal{H})(\mathcal{H}+\rho\mathcal{D})^{-1}\| \leq 1$ since $\mathcal{H}$ is positive semidefinite. $\mathcal{U}$ is the unitary matrix so $\|\mathcal{U}\| = \|(\rho\mathcal{D}-\mathcal{K})(\mathcal{K}+\rho\mathcal{D})^{-1}\| = 1$. Therefore,

$$\varrho(\mathcal{T}_\rho) \leq 1. \tag{2.12}$$

We claim that $\varrho(\mathcal{T}_\rho) \neq 1$.

Assume that λ is one eigenvalue of the preconditioned linear system $P^{-1}\mathcal{A}x = P^{-1}\mathbf{b}$. We have

$$\begin{aligned}\mathcal{A}\mathbf{x} &= \lambda\frac{1}{2\rho}(\mathcal{H}+\rho\mathcal{O})(\mathcal{K}+\rho\mathcal{O})\mathbf{x}\\ &= \frac{\lambda}{2\rho}\left(\mathcal{H}\mathcal{K}+\rho\mathcal{A}+\rho^2\mathcal{O}\right)\mathbf{x}\\ &= \frac{\lambda}{2\rho}\mathcal{H}\mathcal{K}\mathbf{x}+\frac{\lambda}{2}\mathcal{A}\mathbf{x}+\frac{\rho\lambda}{2}\mathbf{x}.\end{aligned} \tag{2.13}$$

Therefore,

$$\left(1-\frac{\lambda}{2}\right)\mathcal{A}\mathbf{x} = \frac{\rho\lambda}{2}\left(\mathcal{O}+\frac{1}{\rho^2}\mathcal{H}\mathcal{K}\right)\mathbf{x}. \tag{2.14}$$

We claim that $1-\lambda/2\neq 0$, otherwise, $(\mathcal{O}+(1/\rho^2)\mathcal{H}\mathcal{K})\mathbf{x}=0$. It turns out that

$$\begin{bmatrix}\frac{1}{\rho^2}\mathbf{HS}+\mathbf{I} & \mathbf{0}\\ -\mathbf{BS} & \mathbf{I}\end{bmatrix}\mathbf{x}=0. \tag{2.15}$$

However, $\rho((1/\rho^2)\mathbf{HS}+\mathbf{I}) = 1+\rho((1/\rho^2)\mathbf{HS})$, and $\mathbf{HS}$ is orthogonal similar with the matrix $\mathbf{H}^{-1/2}\mathbf{S}\mathbf{H}^{1/2}$ which is a skew symmetric matrix with only pure imaginary eigenvalues. Thus, if $\mathbf{B}$ is full rank, we claim that $\lambda\neq 2$.

We define that $\theta=\lambda\rho/(2-\lambda)$. Since $\lambda\neq 2$, θ is welldefined. Thus, with $\lambda = 2\theta/(\theta+2)$, we consider the following equation:

$$\mathcal{A}\mathbf{x} = \theta\left(\mathcal{O}+\frac{1}{\rho^2}\mathcal{H}\mathcal{K}\right)\mathbf{x}. \tag{2.16}$$

If $|\lambda|<1$, then, $|2\theta/(\theta+2)|<1$. Since $|1-(2\theta/(\theta+2))| = |(\rho-\theta)/(\theta+\rho)|\leq 1$, we need to show $|\rho-\theta|/|\rho+\theta|\neq 1$, which means that θ is not a pure imaginary number.

Next we will prove that θ is not a pure imaginary number.

Consider the system

$$\begin{bmatrix}\mathbf{A} & \mathbf{B}^T\\ -\mathbf{B} & 0\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ -p\end{bmatrix} = \begin{bmatrix}\frac{1}{\rho^2}\mathbf{HS}+\mathbf{I} & \mathbf{0}\\ -\frac{1}{\rho^2}\mathbf{BS} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ -p\end{bmatrix}. \tag{2.17}$$

We can obtain the following system of the equations:

$$\begin{aligned}\mathbf{Au}+\mathbf{B}^T p &= \theta\mathbf{u}+\frac{\theta}{\rho^2}\mathbf{HSu},\\ -\mathbf{Bu} &= -\frac{\theta}{\rho^2}\mathbf{BSu}+\theta p.\end{aligned} \tag{2.18}$$

We solve p from the second equation $p = (1/\theta)\mathbf{B}((\theta/\rho^2)\mathbf{S}-\mathbf{I})\mathbf{u}$. Plug in p into the first equation, we have

$$\mathbf{Au}+\frac{1}{\rho^2}\mathbf{B}^T\mathbf{BSu}-\theta\mathbf{u}-\frac{\theta}{\rho^2}\mathbf{HSu}=\frac{1}{\theta}\mathbf{B}^T\mathbf{Bu}. \tag{2.19}$$

Applying $\theta\mathbf{u}^H$ to the both sides, we can obtain the following equation:

$$\theta\mathbf{u}^H\mathbf{Au}+\frac{\theta}{\rho^2}\mathbf{u}^H\mathbf{B}^T\mathbf{BSu}-\theta^2\mathbf{u}^H\mathbf{u}-\frac{\theta^2}{\rho^2}\mathbf{u}^H\mathbf{HSu}=\mathbf{u}^H\mathbf{B}^T\mathbf{Bu}=\|\mathbf{Bu}\|_2\geq 0. \tag{2.20}$$

Therefore, $\theta\mathbf{A}+(\theta/\rho^2)\mathbf{B}^T\mathbf{BS}-\theta^2\mathbf{I}-(\theta^2/\rho^2)\mathbf{HS}$ is a Hermitian matrix. Suppose that $\mathrm{Re}(\theta)=0$, that is, $\theta = t\mathbf{i}$, where $t\neq 0$. We denote that $\mathbf{G}=\theta\mathbf{A}+(\theta/\rho^2)\mathbf{B}^T\mathbf{BS}-\theta^2\mathbf{I}-(\theta^2/\rho^2)\mathbf{HS}=t\mathbf{i}\mathbf{A}+(ti/\rho^2)\mathbf{B}^T\mathbf{BS}+t^2\mathbf{I}+(t^2/\rho^2)\mathbf{HS}$. While $\mathbf{G}^H=-t\mathbf{i}\mathbf{A}^H+(ti/\rho^2)\mathbf{B}^T\mathbf{BS}+t^2\mathbf{I}+(t^2/\rho^2)\mathbf{HS}=\mathbf{G}$, which leads to $\mathbf{A}=\mathbf{A}^H$. A contradiction. Because $\mathbf{A}$ is the discretization of the Oseen problem which is not Hermitian.

Thus, θ is not a pure imaginary number. □

3. Application of the Preconditioner

To solve the preconditioner $P_\alpha z_k = r_k$, we first solve the system

$$\mathscr{H}w_k = r_k, \tag{3.1}$$

for w_k, followed by

$$\mathscr{K}z_k = w_k. \tag{3.2}$$

The first system requires solving systems with coefficient matrix $\mathscr{H}$, which is a system from discretized Stokes problem. We have many efficient solvers to solve this type of the system, see [17–19].

The second system requires solving the sparse tridiagonal matrix $K+\rho I_n$. This can be done by sparse LU factorization, preconditioned GMRES method. Notice that since $K+\rho I$ is a tridiagnoal matrix, it is very easy to solve this system. In practice, we can solve (3.1) and (3.2) with inexact solvers. Our experience is that the rate of convergence of the outer Krylov subspace iteration is scarcely affected by the use of inexact inner solves. We can only use 1 step of pcg method for (3.1) and 1 step of gmres method for (3.2).

4. Numerical Experiments

In this section we report on several numerical experiments meant to illustrate the behavior of the HSS preconditioner on a wide range of model problems. We consider both Stokes and Oseen-type problems, steady and unsteady, in 2D. The use of inexact solves is also discussed. Our results include iteration counts, as well as some timings and eigenvalue plots.

All results were computed in Matlab 7.6.0 on one processor of an Intel core i7 with 8 GB of memory.

In all experiments, a symmetric diagonal scaling was applied before forming the preconditioner, so as to have all the nonzeros diagonal entries of the saddle point matrix equal to 1. We found that this scaling is beneficial to convergence, and it makes finding (nearly) optimal values of the shift ρ easier. Of course, the right-hand side and the solution vector were scaled accordingly. We found, however, that without further preconditioning Krylov subspace solvers converge extremely slowly on all the problems considered here. Right preconditioning was used in all cases.

Here we consider linear systems arising from the discretization of the linearized Navier-Stokes equations in rotation form. The computational domain is the unit square $\Omega = [0,1] \times [0,1]$. Homogeneous Dirichlet boundary conditions are imposed on the velocities; experiments with different boundary conditions were also performed, with results similar to those reported below. We experimented with different forms of the divergence-free field $\mathbf{v}$ appearing (via $\mathbf{w} = \nabla \times \mathbf{v}$) in the rotation form of the unsteady Oseen problem. Here we present results for the choice $w = 16x(x-1) + 16y(y-1)$ (2D case) and $\mathbf{w} = (-4z(1-2x),\ 4z(2y-1),\ 2y(1-y))$ (3D case). The 2D case corresponds to the choice of $\mathbf{v}$. The equations were discretized with a Marker-and-Cell (MAC) scheme with a uniform mesh size h. The outer iteration (full GMRES) was stopped when

$$\frac{\|r_k\|_2}{\|r_0\|_2} < 10^{-6}, \tag{4.1}$$

where r_k denotes the residual vector at step k. For the results presented in this section, we use the exact solver to solve the two systems.

In Tables 1, 2, and 3, we present results for unsteady problems with $\alpha = 10$ to $\alpha = 100$ for different values of ν. We can see from the table that AHSS preconditioning results in fast convergence in all cases, and that the rate of convergence is virtually h-independent. Here as in all other unsteady (or quasi-steady) problems that we have tested, the rate of convergence is not overly sensitive to the choice of ρ, especially for small ν. A good choice is $\rho \approx 0.01$ for the two most grids.

5. Conclusions

In this paper, we have considered preconditioned iterative methods applied to discretizations of the Navier-Stokes equations in rotation form. We focus on the unsteady case of the linearized Navier-Stokes problem. We have compared the performance of the alternative HSS (AHSS) preconditioners with regard to the mesh size, the Reynolds number, the time step, and other problem parameters. We find that the AHSS preconditioner has a robust behavior especially for unsteady Oseen problems. Although our computational experience has been

Table 1: Results for 2D unsteady Oseen problem different values of α (exact solves) and $\nu = 0.1$.

Grid	$\alpha = 10$	$\alpha = 20$	$\alpha = 50$	$\alpha = 100$
8×8	7	5	4	3
16×16	8	6	4	4
32×32	8	6	5	4
64×64	8	6	5	4
128×128	8	6	5	4

Table 2: Results for 2D unsteady Oseen problem different values of α (exact solves) and $\nu = 0.05$.

Grid	$\alpha = 10$	$\alpha = 20$	$\alpha = 50$	$\alpha = 100$
8×8	4	4	3	3
16×16	6	5	3	3
32×32	8	6	4	3
64×64	9	7	5	5
128×128	10	7	5	5

Table 3: Results for 2D unsteady Oseen problem different values of α (exact solves) and $\nu = 0.01$.

Grid	$\alpha = 10$	$\alpha = 20$	$\alpha = 50$	$\alpha = 100$
8×8	3	3	2	2
16×16	4	3	3	2
32×32	5	4	3	3
64×64	8	5	3	3
128×128	9	6	4	3

limited to uniform MAC discretizations and simple geometries, the preconditioner should be applicable to more complicated problems and discretizations, including unstructured grids.

Compared with HSS (Hermitian and Skew-Hermitian preconditioner), the AHSS preconditioner works better for relative large viscosity. For an example, $\nu > 0.05$. For the smaller viscosity, $\nu < 0.01$, HSS preconditioner will be recommended.

In the future study, we will investigate the performance the AHSS preconditioner based using the inexact solvers for the inner iteration. Also the picard's iteration will be tested.

Acknowledgment

The authors would like to thank Michele Benzi for helpful suggestions.

References

[1] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, NY, USA, 2005.

[2] M. A. Olshanskii, "An iterative solver for the Oseen problem and numerical solution of incompressible Navier-Stokes equations," *Numerical Linear Algebra with Applications*, vol. 6, no. 5, pp. 353–378, 1999.

[3] M. A. Olshanskii, "A low order Galerkin finite element method for the Navier-Stokes equations of steady incompressible flow: a stabilization issue and iterative methods," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 47-48, pp. 5515–5536, 2002.

[4] M. A. Olshanskii, "Preconditioned iterations for the linearized Navier-Stokes system in the rotation form," in *Proceedings of the Second MIT Conference on Computational Fluid and Solid Mechanics*, K. J. Bathe, Ed., pp. 1074–1077, 2003.

[5] M. A. Olshanskii and A. Reusken, "Navier-Stokes equations in rotation form: a robust multigrid solver for the velocity problem," *SIAM Journal on Scientific Computing*, vol. 23, no. 5, pp. 1683–1706, 2002.

[6] F. H. Harlow and J. E. Welch, "Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface," *Physics of Fluids*, vol. 8, no. 12, pp. 2182–2189, 1965.

[7] H. C. Elman, D. J. Silvester, and A. J. Wathen, "Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations," *Numerische Mathematik*, vol. 90, no. 4, pp. 665–688, 2002.

[8] M. Benzi, M. J. Gander, and G. H. Golub, "Optimization of the Hermitian and skew-Hermitian splitting iteration for saddle-point problems," *BIT. Numerical Mathematics*, vol. 43, pp. 881–900, 2003.

[9] M. Benzi and G. H. Golub, "A preconditioner for generalized saddle point problems," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 1, pp. 20–41, 2004.

[10] M. Benzi, G. H. Golub, and J. Liesen, "Numerical solution of saddle point problems," *Acta Numerica*, vol. 14, pp. 1–137, 2005.

[11] M. Benzi and V. Simoncini, "On the eigenvalues of a class of saddle point matrices," *Numerische Mathematik*, vol. 103, no. 2, pp. 173–196, 2006.

[12] J. Liu, *Preconditioned Krylov subspace methods for incompressible flow problems*, Ph.D. thesis, Emory University, Ann Arbor, Mich, USA, 2006.

[13] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, Pa, USA, 2nd edition, 2003.

[14] Y. Saad and M. H. Schultz, "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 3, pp. 856–869, 1986.

[15] Z.-Z. Bai, G. H. Golub, and M. K. Ng, "Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems," *SIAM Journal on Matrix Analysis and Applications*, vol. 24, no. 3, pp. 603–626, 2003.

[16] V. Simoncini and M. Benzi, "Spectral properties of the Hermitian and skew-Hermitian splitting preconditioner for saddle point problems," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 2, pp. 377–389, 2004/05.

[17] O. A. Karakashian, "On a Galerkin-Lagrange multiplier method for the stationary Navier-Stokes equations," *SIAM Journal on Numerical Analysis*, vol. 19, no. 5, pp. 909–923, 1982.

[18] J. Peters, V. Reichelt, and A. Reusken, "Fast iterative solvers for discrete Stokes equations," *SIAM Journal on Scientific Computing*, vol. 27, no. 2, pp. 646–666, 2005.

[19] G. M. Kobelkov and M. A. Olshanskii, "Effective preconditioning of Uzawa type schemes for a generalized Stokes problem," *Numerische Mathematik*, vol. 86, no. 3, pp. 443–470, 2000.

Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 564132, 7 pages
doi:10.1155/2012/564132

Research Article

A Note on the Eigenvalue Analysis of the SIMPLE Preconditioning for Incompressible Flow

Shi-Liang Wu,[1,2] Feng Chen,[1] and Xiao-Qi Niu[1]

[1] *School of Mathematics and Statistics, Anyang Normal University, Anyang 455000, China*
[2] *College of Mathematics, Chengdu University of Information Technology, Chengdu 610255, China*

Correspondence should be addressed to Shi-Liang Wu, wushiliang1999@126.com

Received 21 November 2011; Accepted 11 January 2012

Academic Editor: Kok Kwang Phoon

We consider the SIMPLE preconditioning for block two-by-two generalized saddle point problems; this is the general nonsymmetric, nonsingular case where the (1,2) block needs not to equal the transposed (2,1) block, and the (2,2) block may not be zero. The eigenvalue analysis of the SIMPLE preconditioned matrix is presented. The relationship between the two different formulations spectrum of the SIMPLE preconditioned matrix is established by using the theory of matrix eigenvalue, and some corresponding results in recent article by Li and Vuik (2004) are extended.

1. Introduction

Consider the two-by-two generalized saddle point problems

$$\mathcal{A}\begin{bmatrix} x \\ y \end{bmatrix} \equiv \begin{bmatrix} A & B^T \\ C & -D \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$ is nonsingular, $B, C \in \mathbb{R}^{m\times n}$ $(m \le n)$, $D \in \mathbb{R}^{m\times m}$.

Systems of the form (1.1) arise in a variety of scientific and engineering applications, such as linear elasticity, fluid dynamics, electromagnetics, and constrained quadratic programming [1–4]. We refer the reader to [5] for more applications and numerical solution techniques of (1.1).

Since the coefficient matrix of (1.1) is often large and sparse, it may be attractive to use iterative methods. In particular, Krylov subspace methods might be used. As known, Krylov subspace methods are considered as one kind of the important and efficient iterative techniques for solving the large sparse linear systems because these methods are cheap to be implemented and are able to fully exploit the sparsity of the coefficient matrix. It is well

known that the convergence speed of Krylov subspace methods depends on the eigenvalue distribution of the coefficient matrix [6]. Since the coefficient matrix of (1.1) is often extremely ill-conditioned and highly indefinite, the convergence speed of Krylov subspace methods can be unacceptably slow. In this case, Krylov subspace methods are not competitive without a good preconditioner. That is, preconditioning technique is a key ingredient for the success of Krylov subspace methods in applications.

To efficiently and accurately solve (1.1), Semi-implicit method for pressure linked equations (SIMPLE) were presented in [7] by Patankar. Subsequently, combining the SIMPLE(R) algorithm and Krylov subspace method GCR [8], Vuik et al. [9] proposed the GCR-SIMPLE(R) algorithm for solving (1.1). In this algorithm, the SIMPLE iteration is used as a preconditioner in the GCR method. Numerical experiments show that the SIMPLE(R) preconditioning is effective and competitive.

It is well known that the spectral properties of the preconditioned matrix give important insight in the convergence behavior of the preconditioned Krylov subspace methods. In [10], the eigenvalue analysis was given for the SIMPLE preconditioned matrix with $B = C$ and $D = 0$, and two different formulations spectrum of the preconditioned matrix were derived. The relationship between the two different formulations has been built by using the theory of matrix singular value decomposition. If $B \neq C$ and $D \neq 0$, using matrix singular value decomposition to establish the relationship between the two different formulations is invalid. On this occasion, we present the relationship between the two different formulations by using the theory of matrix eigenvalue and overcome the shortcomings of [10]. Some corresponding results in [10] are extended to two-by-two generalized saddle point problems.

2. Spectral Analysis

For simplicity, $\sigma(\cdot)$ denotes the set of all eigenvalues of a matrix, and the diagonal entries of A are not equal to zero. If the SIMPLE algorithm is used as preconditioning, it is equivalent to choose the preconditioner $\mathcal{P}$ as

$$\mathcal{P} = \mathcal{M}\mathcal{B}^{-1}, \tag{2.1}$$

where

$$\mathcal{B} = \begin{bmatrix} I & -Q^{-1}B^T \\ 0 & I \end{bmatrix}, \qquad \mathcal{M} = \begin{bmatrix} A & 0 \\ C & R \end{bmatrix}, \qquad Q = \operatorname{diag}(A), \qquad R = -\left(D + CQ^{-1}B^T\right). \tag{2.2}$$

On the nonsingular of $\mathcal{A}$ and $\mathcal{P}$ we have the following proposition.

Proposition 2.1. *The matrices $\mathcal{A}$ and $\mathcal{P}$, respectively, in* (1.1) *and* (2.1) *are nonsingular if and only if the Schur complements $-(D + CA^{-1}B^T)$ and $-(D + CQ^{-1}B^T)$, respectively, are nonsingular.*

In this paper, we assume that $\mathcal{A}$ and $\mathcal{P}$ are nonsingular and that B and C are of full rank.

Proposition 2.2. *If the right preconditioner $\mathcal{P}$ is defined by* (2.1)*, then the preconditioned matrix is*

$$\tilde{\mathcal{A}} = \mathcal{A}\mathcal{P}^{-1} = \begin{bmatrix} I - (I - AQ^{-1})B^T R^{-1} C A^{-1} & (I - AQ^{-1})B^T R^{-1} \\ 0 & I \end{bmatrix}. \tag{2.3}$$

Therefore, the spectrum of the SIMPLE preconditioned matrix $\tilde{\mathcal{A}}$ is

$$\sigma\left(\tilde{\mathcal{A}}\right) = \{1\} \cup \sigma\left(I - \left(I - AQ^{-1}\right)B^T R^{-1} C A^{-1}\right). \tag{2.4}$$

Proof. By simple computations, it is easy to verify that

$$\begin{aligned}
\mathcal{M}^{-1} &= \begin{bmatrix} A^{-1} & 0 \\ -R^{-1}CA^{-1} & R^{-1} \end{bmatrix} \\
\sigma\left(\tilde{\mathcal{A}}\right) &= \begin{bmatrix} A & B^T \\ C & -D \end{bmatrix} \begin{bmatrix} I & -Q^{-1}B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ -R^{-1}CA^{-1} & R^{-1} \end{bmatrix} \\
&= \begin{bmatrix} I - (I - AQ^{-1})B^T R^{-1} C A^{-1} & (I - AQ^{-1})B^T R^{-1} \\ 0 & I \end{bmatrix}.
\end{aligned} \tag{2.5}$$

Further, it is easy to find that the form of the spectrum of $\sigma(\tilde{\mathcal{A}})$ is described by (2.4). □

By the similarity invariance of the spectrum of the matrix, we have

$$\begin{aligned}
\sigma\left(I - \left(I - AQ^{-1}\right)B^T R^{-1} C A^{-1}\right) &= \sigma\left(I - \left(A^{-1} - Q^{-1}\right)B^T R^{-1} \mathrm{C}\right) \\
&= \sigma\left(I - Q^{-1}(Q - A)A^{-1}B^T R^{-1} C\right) \\
&= \sigma\left(I - JA^{-1}B^T R^{-1} C\right),
\end{aligned} \tag{2.6}$$

where the matrix $J = Q^{-1}(Q - A)$ is the Jacobi iteration matrix of the matrix A. Further, we have the following proposition.

Proposition 2.3. *For the SIMPLE preconditioned matrix $\tilde{\mathcal{A}}$,*

(1) *1 is an eigenvalue with multiplicity at least of m,*

(2) *the remaining eigenvalues are $1 - \mu_i$, $i = 1, 2, \ldots, n$, where μ_i is the ith eigenvalue of*

$$ZEx = \mu x, \tag{2.7}$$

where

$$Z = JA^{-1} \in \mathbb{R}^{n\times n}, \qquad E = B^T R^{-1} C \in \mathbb{R}^{n\times n}. \tag{2.8}$$

In fact, we also have the following result.

Proposition 2.4. *For the SIMPLE preconditioned matrix* $\widetilde{\mathcal{A}}$,

(1) *1 is an eigenvalue with (algebraic and geometric) multiplicity of* n,

(2) *the remaining eigenvalues are defined by the generalized eigenvalue problem*

$$Sx = \lambda Rx, \tag{2.9}$$

where $S = -(D + CA^{-1}B^T)$ *is the Schur complement of the matrix* $\mathcal{A}$.

Proof. Note that $\mathcal{A}\mathcal{P}^{-1}$ is the same spectrum as $\mathcal{P}^{-1}\mathcal{A}$. So, it is only needed to consider the following generalized eigenvalue problem

$$\mathcal{A}x = \lambda \mathcal{P}x, \tag{2.10}$$

where

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ C & -D \end{bmatrix}, \qquad \mathcal{P} = \begin{bmatrix} A & AQ^{-1}B^T \\ C & -D \end{bmatrix}. \tag{2.11}$$

The generalized eigenvalue problem (2.10) can be written as

$$\begin{bmatrix} A & B^T \\ C & -D \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \lambda \begin{bmatrix} A & AQ^{-1}B^T \\ C & -D \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix}, \tag{2.12}$$

that is,

$$Au + B^T p = \lambda\left(Au + AQ^{-1}B^T p\right), \tag{2.13}$$

$$Cu - Dp = \lambda(Cu - Dp). \tag{2.14}$$

From (2.13) and (2.14), it is easy to see that $\lambda = 1$ is an eigenvalue of (2.12). If the matrix $Q^{-1} - A^{-1}$ is nonsingular with $\lambda = 1$ and rank $(B^T) = m$, from (2.13) we have $p = 0$. Therefore, the eigenvectors corresponding to eigenvalue 1 are

$$v_i = \begin{bmatrix} u_i \\ 0 \end{bmatrix}, \quad u_i \in \mathbb{R}^n,\ i = 1, 2, \ldots, n, \tag{2.15}$$

where $\{u_i\}_i^n$ is a basis of $\mathbb{R}^n$.

For $\lambda \neq 1$, from (2.13) we obtain

$$u = \frac{1}{1-\lambda} A^{-1}\left(\lambda AQ^{-1}B^T p - B^T p\right). \tag{2.16}$$

Substituting it into (2.14) yields

$$Sp = \lambda Rp, \tag{2.17}$$

where $S = -(D + CA^{-1}B^T)$ is the Schur complement of the matrix $\mathcal{A}$. □

From Propositions 2.3 and 2.4, two different generalized eigenvalue problems (2.7) and (2.9) have been derived to describe the spectrum of $\widetilde{\mathcal{A}}$. Subsequently, we will investigate the relationship between both spectral formulations for the nonsymmetric case. Here we will make use of the theory of matrix eigenvalue to establish the relationship of the two different formulations spectrum of the SIMPLE preconditioned matrix. To this end, the following lemma is required.

Lemma 2.5 (See [11]). *Suppose that $M \in \mathbb{R}^{m\times n}$ and $N \in \mathbb{R}^{n\times m}$ with $m \le n$. Then NM has the same eigenvalues as MN, counting multiplicity, together with an additional $n - m$ eigenvalues equal to 0.*

By (2.7), it follows that

$$\begin{aligned} ZE &= Z^{n\times n}\left(B^T\right)^{n\times m}\left(R^{-1}\right)^{m\times m}C^{m\times n} \in \mathbb{R}^{n\times n}, \\ &\left(R^{-1}\right)^{m\times m}C^{m\times n}Z^{n\times n}\left(B^T\right)^{n\times m} \in \mathbb{R}^{m\times m}. \end{aligned} \tag{2.18}$$

From Lemma 2.5, we have

$$\sigma(ZE) = \{0\} \cup \sigma\left(R^{-1}CZB^T\right), \tag{2.19}$$

where the eigenvalue 0 is with multiplicity of $n - m$ and

$$\begin{aligned} \sigma\left(R^{-1}CZB^T\right) &= \sigma\left(R^{-1}CQ^{-1}(Q - A)A^{-1}B^T\right) \\ &= \sigma\left(R^{-1}C\left(A^{-1} - Q^{-1}\right)B^T\right) \\ &= \sigma\left(R^{-1}\left(CA^{-1}B^T - CQ^{-1}B^T\right)\right) \\ &= \sigma\left(R^{-1}\left[\left(CA^{-1}B^T + D\right) - \left(CQ^{-1}B^T + D\right)\right]\right) \\ &= \sigma\left(R^{-1}(R - S)\right) \\ &= \sigma\left(I - R^{-1}S\right) \\ &= \cup\{1 - \lambda_i\}, \quad i = 1, 2, \ldots, m. \end{aligned} \tag{2.20}$$

These relations lead to the following proposition.

Proposition 2.6. *For two generalized eigenvalue problems* (2.7) *and* (2.9)*, suppose that* $\mu_i \in \sigma(ZE)$*,* $i = 1, 2, \ldots, n$*, and* $\lambda_i \in \sigma(R^{-1}S)$*,* $i = 1, 2, \ldots, m$*, the relationship between two problems is that* $\mu = 0$ *is an eigenvalue of* (2.7) *with multiplicity of* $n - m$*, which can be denoted as* $\mu_{m+1} = \mu_{m+2} = \cdots = \mu_n = 0$*, and that* $\lambda_i = 1 - \mu_i$*,* $i = 1, 2, \ldots, m$*, holds for the remaining* m *eigenvalues.*

Some remarks on Proposition 2.6 are given as follows.

(i) In [10], the relationship between two different formulations spectrum of the preconditioned matrix with $B = C$ and $D = 0$ was built by using the theory of matrix singular value decomposition, but for the nonsymmetric case, the above strategy is invalid. Whereas, using the theory of matrix eigenvalue not only establishes the relationship between the two different formulations, but also overcomes the shortcomings of [10]. In this way, Propositions 2.2–2.6 can be regarded as the extension of Propositions 2–5 [10].

(ii) In [10], the diagonal entries of matrix A must be positive. But, in this paper, the diagonal entries of Q are only not equal to zero. Clearly, this assumption is weaker than that of [10]. If the diagonal entries of matrix A are complex and not equal to zero, then the diagonal entries of Q take the absolute diagonal entries of A. This idea is based on an absolute diagonal scaling technique, which is cheaply easy to implement, reducing computation times and amount of memory.

(iii) Recently, although Li et al. in [12] discussed the SIMPLE preconditioning for the generalized nonsymmetric saddle point problems and provided some results above the spectrum of the SIMPLE preconditioned matrix, some conditions of the supporting propositions may be defective. In fact, if A is nonsingular with rank $(B^T) = \text{rank}\ (C) = m$, then R and $\mathcal{D}$ may be singular. For a counterexample, we take $C = [1\ 0]$, $A = Q = 2I$ and $B = [0\ 1]$, then $R = CQ^{-1}B^T = 0$. That is, this paper corrects some results in [12].

(iv) In fact, Q is not necessary the diagonal entries of A; in this case, the diagonal entries of A can be equal to zero. In actual implements, the choice of matrix Q is that the eigenvalue of the generalized eigenvalue problem (2.9) is close to one; Krylov subspace methods such as GMRES will converge quickly.

3. Conclusion

In this paper, the SIMPLE preconditioner for the nonsymmetric generalized saddle point problems is discussed. The relationship of the two different formulations spectrum of the SIMPLE preconditioned matrix has been built by using the theory of matrix eigenvalue.

Acknowledgments

The authors would like to thank Editor Professor Phoon and two anonymous referees for their helpful suggestions, which greatly improve the paper. This research of this author is supported by NSFC Tianyuan Mathematics Youth Fund (11026040).

References

[1] O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems*, Computer Science and Applied Mathematics, Academic Press Inc., Orlando, Fla, USA, 1984.
[2] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, vol. 15, Springer, New York, NY, USA, 1991.
[3] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers*, Oxford University Press, Oxford, UK, 2003.
[4] W. Zulehner, "Analysis of iterative methods for saddle point problems: a unified approach," *Mathematics of Computation*, vol. 71, no. 238, pp. 479–505, 2002.
[5] M. Benzi, G. H. Golub, and J. Liesen, "Numerical solution of saddle point problems," *Acta Numerica*, vol. 14, pp. 1–137, 2005.
[6] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, Mass, USA, 1996.
[7] S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York, NY, USA, 1980.
[8] S. C. Eisenstat, H. C. Elman, and M. H. Schultz, "Variational iterative methods for nonsymmetric systems of linear equations," *SIAM Journal on Numerical Analysis*, vol. 20, no. 2, pp. 345–357, 1983.
[9] C. Vuik, A. Saghir, and G. P. Boerstoel, "The Krylov accelerated SIMPLE(R) method for flow problems in industrial furnaces," *International Journal for Numerical Methods in Fluids*, vol. 33, no. 7, pp. 1027–1040, 2000.
[10] C. Li and C. Vuik, "Eigenvalue analysis of the SIMPLE preconditioning for incompressible flow," *Numerical Linear Algebra with Applications*, vol. 11, no. 5-6, pp. 511–523, 2004.
[11] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, Mass, USA, 1990.
[12] L. J. Li, Z. J. Zhang, and L. Li, "A note on the SIMPLE preconditioner for nonsymmetric saddle point problems," *Journal of Information and Computing Science*, vol. 5, pp. 153–160, 2010.

Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 352081, 15 pages
doi:10.1155/2012/352081

Research Article

Applications of Symmetric and Nonsymmetric MSSOR Preconditioners to Large-Scale Biot's Consolidation Problems with Nonassociated Plasticity

Xi Chen[1] and Kok Kwang Phoon[2]

[1] *Department of Geotechnical Engineering, School of Civil Engineering, Beijing Jiaotong University, Beijing 100044, China*
[2] *Department of Civil and Environmental Engineering, National University of Singapore, E1A-07-14, Blk E1A, 07-03, 1 Engineering Drive 2, Singapore 117576*

Correspondence should be addressed to Xi Chen, chenxi@bjtu.edu.cn
and Kok Kwang Phoon, kkphoon@nus.edu.sg

Received 12 October 2011; Revised 14 December 2011; Accepted 15 December 2011

Academic Editor: Massimiliano Ferronato

Two solution schemes are proposed and compared for large 3D soil consolidation problems with nonassociated plasticity. One solution scheme results in the nonsymmetric linear equations due to the Newton iteration, while the other leads to the symmetric linear systems due to the symmetrized stiffness strategies. To solve the resulting linear systems, the QMR and SQMR solver are employed in conjunction with nonsymmetric and symmetric MSSOR preconditioner, respectively. A simple footing example and a pile-group example are used to assess the performance of the two solution schemes. Numerical results disclose that compared to the Newton iterative scheme, the symmetric stiffness schemes combined with adequate acceleration strategy may lead to a significant reduction in total computer runtime as well as in memory requirement, indicating that the accelerated symmetric stiffness method has considerable potential to be exploited to solve very large problems.

1. Introduction

The wide applications of large-scale finite element analyses for soil consolidation settlements demand efficient iterative solutions. Among various iterative solvers the Krylov subspace iterative methods have enjoyed great popularity so that it was ranked as one of top ten algorithms invented in the 20th century [1]. In large-scale finite element computations, the success of Krylov subspace iterative methods may be partially attributed to the matrix-vector

multiplications which can be implemented sparsely in the iterative process, but it is the preconditioning technique that makes Krylov subspace iterative methods become practically useful.

For an indefinite linear system some preconditioning methods (such as incomplete LU or ILU) originally proposed for general problems may encounter slow convergence or even breakdown during the iterative process (e.g., [2]). In the past decade, many researchers have focused on constructing preconditioners for indefinite problems either by revising a standard preconditioner or by designing a block preconditioner according to the block matrix structure. For example, for the indefinite linear equation stemming from Biot's consolidation a generalized Jacobi (GJ) preconditioner was proposed by Phoon et al. [3] to overcome the unbalanced diagonal scaling of standard Jacobi, while to obviate unstable triangular solves a modified symmetric successive overrelaxation (MSSOR) preconditioner [4] was designed to enhance the pivots embedded within the standard symmetric successive overrelaxation (SSOR) factorization. The successes of GJ and MSSOR undoubtedly are attributed to the fact that both of them were developed based on the block matrix structure. On a separate track, many other researchers have been working on block preconditioners. Perugia and Simoncini [5] proposed symmetric indefinite block diagonal preconditioners for mixed finite element formulations, and, furthermore, Simoncini [6] proposed block triangular preconditioners to couple with symmetric quasiminimal residual (SQMR) [7] for symmetric saddle-point problems. Keller et al. [8] and Bai et al. [9] developed their own block constraint preconditioners for indefinite linear systems. For three types of block preconditioners, an in-depth comparison and eigenvalue study has been carried out by Toh et al. [10], and it was concluded that the block constraint preconditioner and the diagonal GJ preconditioner could be more promising for large-scale Biot's linear system. Recently, more attentions have been paid to the constraint block structure. For instance, Bergamaschi et al. [11], Ferronato et al. [12], and Haga et al. [13] defined their own block constraint preconditioners, respectively, and validated the good performance of their block constraint preconditioners. Furthermore, Janna et al. [14] developed a parallel block preconditioner for symmetric positive definite (SPD) matrices by coupling the generalized factored sparse approximate inverse (FSAI) with ILU factorization, and it was concluded that in many realistic cases the FSAI-ILU preconditioner is more effective than both FSAI and ILU preconditioners. To resolve the soil-structure problems in which material stiffness contrast is significant, the partitioned block diagonal preconditioners were proposed by Chaudhary [15], and numerical results indicate that the proposed preconditioners are not sensitive to the material stiffness contrast ratio. Based on the above review, it seems that constructing a preconditioner based on the block structure of the coefficient matrix has become a popular strategy for large-scale Biot's indefinite linear systems.

Soil consolidation accompanied with soil dilatancy may be frequently encountered in practice, and commonly the soil dilatancy may be modeled by nonassociated plasticity. The main contribution of the article is that for coupled consolidation problems involving nonassociated plasticity, two solution strategies are proposed and evaluated, respectively, by employing a strategy of combining a nonlinear iterative scheme with a corresponding linear iterative method (e.g., [16, 17]). The paper is organized as follows. In Section 2, the nonlinear iterative methods are formulated for elastoplastic soil consolidation problems, and the coupled 2×2 nonsymmetric indefinite Biot's finite element linear system of equation due to nonassociated plastic flow is derived. In Section 3, the recently proposed block preconditioners are reviewed and commented. In Section 4, both symmetric version and nonsymmetric

version of MSSOR are implemented within the proposed two schemes, respectively, to solve the soil consolidation examples with nonassociated plasticity, and their performances are investigated and compared. Finally, some useful observations and conclusions are summarized in Section 5.

2. Coupled Linear System of Equation Arising from Elastoplastic Biot's Consolidation Problems

Soil consolidation is a soil settlement process coupled with dissipation of excess pore water pressure, and this process may be physically modeled by the widely accepted Biot's consolidation theory [18]. Recall that for a fully saturated porous media, the volumetric fluid content variation within the soil skeleton is solely related to the deformation of solid skeleton, that is,

$$\nabla \cdot \boldsymbol{\sigma}(\mathbf{x}, t) - \gamma \mathbf{b} = 0, \quad (\mathbf{x}, t) \in \Omega \times (0, T], \tag{2.1}$$

in which the Terzaghi's effective stress principle $\boldsymbol{\sigma} = \boldsymbol{\sigma}' + p$ is applied; $\mathbf{b} = [0, 0, 1]^T$ is the unit body force vector; γ is the total unit weight; Ω is the solution domain; T is the total analyzing time. For the fully saturated porous media, the pore fluid should comply with the fluid continuity equation or the mass-conservation equation, that is,

$$\nabla \cdot \dot{\mathbf{u}}(\mathbf{x}, t) + \nabla \cdot \mathbf{v}_f(\mathbf{x}, t) = 0, \tag{2.2}$$

where $\nabla \cdot \dot{\mathbf{u}} = \dot{\varepsilon}_v$ with ε_v as the volumetric strain, the dot symbol over any symbols means time differentiation, and $\mathbf{u}$ is the displacement vector; the fluid velocity $\mathbf{v}_f$ is described by Darcy's law

$$\mathbf{v}_f = -\frac{[\mathbf{k}]}{\gamma_f}(\nabla p - \gamma_f \mathbf{b}). \tag{2.3}$$

Here, γ_f is the unit weight of pore fluid; $[\mathbf{k}] = [\mathbf{k}_s]$ is the hydraulic conductivity tensor (it is a diagonal matrix $[\mathbf{k}_s] = \mathrm{diag}(k_{s,x}, k_{s,y}, k_{s,z})$ when orthogonal hydraulic conductivity properties are assumed) for saturated flow. The solution domain Ω has the complementary boundary conditions may be given as $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ and $\partial\Omega_D \cap \partial\Omega_N = \emptyset$ with $\partial\Omega_D$ for the Dirichlet (or prescribed displacement and pore pressure) boundary and $\partial\Omega_N$ for the Neumann (or prescribed force and flux) boundary. After discretizing (2.1) and (2.2) in space and time domain, respectively, the incremental form of Biot's finite element equation is derived as (e.g., [19]):

$$\begin{bmatrix} \mathbf{K} & \mathbf{L} \\ \mathbf{L}^T & -\theta \Delta t \mathbf{G} \end{bmatrix} \begin{Bmatrix} \Delta \mathbf{u} \\ \Delta \mathbf{p}^{ex} \end{Bmatrix} = \begin{Bmatrix} \Delta \mathbf{f} \\ \Delta t \mathbf{G} \mathbf{p}_n^{ex} \end{Bmatrix} \tag{2.4}$$

in which the subblocks within the coupled matrix are defined, respectively, as

$$\begin{aligned}\mathbf{K} &= \sum_e \left(\int_{V_e} \mathbf{B}_u^T \mathbf{D}_{ep} \mathbf{B}_u dV \right), \\ \mathbf{L} &= \sum_e \left(\int_{V_e} \mathbf{B}_u^T \mathbf{1} \mathbf{N}_p dV \right), \\ \mathbf{G} &= \sum_e \left(\int_{V_e} \mathbf{B}_p^T \frac{[\mathbf{k}]}{\gamma_f} \mathbf{B}_p dV \right),\end{aligned} \tag{2.5}$$

where $\mathbf{D}_{ep}$ is the constitutive matrix; $\mathbf{B}_u$, $\mathbf{B}_p$ is the gradient of shape function $\mathbf{N}_u$ and $\mathbf{N}_p$ used for interpolating the displacement $\mathbf{u}$ and the excess pore pressure $\mathbf{p}^{ex}$, respectively; $\mathbf{1} = [1\ 1\ 1\ 0\ 0\ 0]^T$; Δt is the time increment; θ is the time integration parameter ranging from 0 to 1; θ =1/2 corresponds to the second-order accuracy Crank-Nicolson method, while θ =1 leads to the fully implicit method possessing the first-order accuracy.

Equation (2.4) is the discretized finite element equation for each time increment, and to be convenient the following weak form of the residual equation is used to derive the nonlinear iterative process,

$$\mathbf{R}(\mathbf{u}; \mathbf{p}^{ex}) = \begin{bmatrix} \mathbf{R}_u \\ \mathbf{R}_p \end{bmatrix} = \begin{bmatrix} \mathbf{F}_u^{\text{ext}} \\ \mathbf{F}_p^{\text{ext}} \end{bmatrix} - \begin{bmatrix} \mathbf{F}_u^{\text{int}} \\ \mathbf{F}_p^{\text{int}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{2.6}$$

where $\mathbf{R}$ is the residual or out-of-balance force vector derived from the total potential energy $\Phi(\mathbf{u}; \mathbf{p}^{ex})$; $\mathbf{F}^{\text{ext}}$ is the applied external force at current time increment, and its two parts corresponding to $\mathbf{u}$ and $\mathbf{p}^{ex}$, respectively, have been presented in the RHS of (2.4). Noting that the deformation of soil skeleton is solely caused by the effective stress, the internal forces $\mathbf{F}_u^{\text{int}}$ and $\mathbf{F}_p^{\text{int}}$ may be expressed, respectively, as below,

$$\begin{aligned}\mathbf{F}_u^{\text{int}} &= \int_V \mathbf{B}_u^T \boldsymbol{\sigma}' dV + \int_V \mathbf{B}_u^T \mathbf{p}^{ex} dV, \\ \mathbf{F}_p^{\text{int}} &= \int_V \mathbf{N}_p^T \boldsymbol{\varepsilon} dV + \theta \Delta t \int_V \mathbf{B}_p^T \mathbf{v}_f dV.\end{aligned} \tag{2.7}$$

Applying the first-order Taylor expansion to (2.6) leads to the following Newton-Raphson (NR) iteration:

$$\mathbf{R}\left(\mathbf{u}_{k-1}; \mathbf{p}_{k-1}^{ex}\right) + \mathbf{A}_{k-1} \begin{bmatrix} \delta \mathbf{u}_k \\ \delta \mathbf{p}_k^{ex} \end{bmatrix} = 0, \tag{2.8}$$

$$\begin{bmatrix} \mathbf{u}_k \\ \mathbf{p}_k^{ex} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_{k-1} \\ \mathbf{p}_{k-1}^{ex} \end{bmatrix} + \begin{bmatrix} \delta \mathbf{u}_k \\ \delta \mathbf{p}_k^{ex} \end{bmatrix}, \tag{2.9}$$

where

$$\mathbf{A}_{k-1} = \begin{bmatrix} \left.\dfrac{\partial \mathbf{R}_u}{\partial \mathbf{u}}\right|_{k-1} & \dfrac{\partial \mathbf{R}_u}{\partial \mathbf{p}^{ex}} \\ \dfrac{\partial \mathbf{R}_p}{\partial \mathbf{u}} & \dfrac{\partial \mathbf{R}_p}{\partial \mathbf{p}^{ex}} \end{bmatrix}, \qquad \frac{\partial \mathbf{R}_u}{\partial \mathbf{p}^{ex}} = \left[\frac{\partial \mathbf{R}_p}{\partial \mathbf{u}}\right]^T = -\mathbf{L}, \quad \frac{\partial \mathbf{R}_p}{\partial \mathbf{p}^{ex}} = \theta \Delta t \mathbf{G} \tag{2.10}$$

in which it is apparent that $\mathbf{L}$ is independent of time increment and iterative process due to small-strain finite element computation, and $\mathbf{G}$ is independent of time increment and iterative process due to the saturated soil assumption. In $\mathbf{A}_{k-1}$, the gradient of $\mathbf{R}_u$ about $\mathbf{u}$ is the tangential solid stiffness matrix,

$$\mathbf{K}_T = -\frac{\partial \mathbf{R}_u}{\partial \mathbf{u}} = \frac{\partial \mathbf{F}_u^{\text{int}}(\mathbf{u})}{\partial \mathbf{u}}, \tag{2.11}$$

and $\mathbf{K}_T$ is assembled by the element stiffness $\mathbf{k}_{ep}$, that is,

$$\mathbf{K}_T = \sum_{\text{elements}} \frac{\partial\left(\int_{V^e} \mathbf{B}^T \mathbf{D}_{ep} \mathbf{B} dV\, \mathbf{u}_e\right)}{\partial \mathbf{u}} = \sum_{\text{elements}} \mathbf{k}_{ep} \quad \text{with } \mathbf{k}_{ep} = \int_{V^e} \mathbf{B}^T \mathbf{D}_{ep} \mathbf{B} dV. \tag{2.12}$$

Evidently, the symmetry of $\mathbf{K}_T$ or the 2×2 coupled matrix $\mathbf{A}_{k-1}$ in (2.10) is governed by $\mathbf{D}_{ep}$, which may be expressed as

$$\mathbf{D}_{ep} = \mathbf{D}_e - \mathbf{D}_p = \mathbf{D}_e - \frac{\mathbf{D}_e \mathbf{b} \mathbf{a}^T \mathbf{D}_e}{\mathbf{a}^T \mathbf{D}_e \mathbf{b} - \mathbf{c}^T \mathbf{h}}, \tag{2.13}$$

where $\mathbf{a} = \partial_\sigma F$ represents the yield surface normal vector, $\mathbf{b} = \partial_\sigma G$ denotes the plastic flow direction vector, while $\mathbf{c} = \partial_{\mathbf{q}} F$ is the gradient of yield function (i.e., F) about the internal variables of the soil model. In conventional elastoplastic soil models, the physically associated plastic flow leads to symmetric $\mathbf{D}_{ep}$ due to $\mathbf{b} = \mathbf{a}$, while the nonassociated plastic flow results in nonsymmetric $\mathbf{D}_{ep}$ due to $\mathbf{b} \neq \mathbf{a}$. For convenience, (2.8) may be further expressed as

$$\begin{bmatrix} \mathbf{K}_{k-1} & \mathbf{L} \\ \mathbf{L}^T & -\theta \Delta t \mathbf{G} \end{bmatrix} \begin{Bmatrix} \delta \mathbf{u} \\ \delta \mathbf{p}^{ex} \end{Bmatrix}_k = \mathbf{R}_{k-1} = \mathbf{F}^{\text{ext}} - \mathbf{F}_{k-1}^{\text{int}}, \tag{2.14}$$

where the superscript k signifies the nonlinear iteration count within each time increment. To distinguish the nonlinear iterative process from the Krylov subspace iterative process, the iteration count for the first process as described by (2.14) is called nonlinear iteration count, while the second Krylov subspace process is a linear iterative process, and the iteration count for this process is called linear iteration count.

In the nonlinear iterative process, the search direction $\mathbf{d}_k$ may be computed inexactly (i.e., $\mathbf{d}_k \approx [\delta \mathbf{u}; \delta \mathbf{p}^{ex}]_k^T$), leading to the so-called inexact Newton method whose convergence is governed by (e.g., [16, 17]),

$$\|\mathbf{R}_{k-1} + \mathbf{A}_{k-1} \mathbf{d}_k\| \leq \eta_{k-1} \|\mathbf{R}_{k-1}\|, \tag{2.15}$$

where $\eta_{k-1} \in [0, 1)$ is the forcing term. With the computed search direction $\mathbf{d}_k$, the displacement is updated by

$$\begin{bmatrix} \mathbf{u}_k \\ \mathbf{p}_k^{ex} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_{k-1} \\ \mathbf{p}_{k-1}^{ex} \end{bmatrix} + \chi_k \begin{bmatrix} \delta\mathbf{u}_k \\ \delta\mathbf{p}_k^{ex} \end{bmatrix} \tag{2.16}$$

in which χ_k is the step-length parameter, which may be determined by some optimization strategies. The inexact computation of $\mathbf{d}_k$ may arise from the approximate solve of $\mathbf{d}_k$ or the approximation to $\mathbf{A}_{k-1}$.

For ease of presentation, the linear system of equation at each nonlinear iteration as shown by (2.14) is expressed concisely as

$$\begin{bmatrix} K & B \\ B^T & -C \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{Bmatrix} f \\ g \end{Bmatrix} \quad \text{with } A = \begin{bmatrix} K & B \\ B^T & -C \end{bmatrix}_{N\times N}. \tag{2.17}$$

3. Block-Structured Preconditioners

Here, the block-structured preconditioners are defined as those developed according to the coefficient matrix block structure, and hence they include the block preconditioners and those modified preconditioners inspired by the matrix block structure.

3.1. Block Preconditioners for Biot's Linear System of Equation

Block preconditioners can be classified into three types as mentioned previously. According to Toh et al. [10], it could be convenient to derive the preconditioners based on the matrix inverse. Note that the inverse of the 2×2 matrix in (2.17) may be expressed as

$$\begin{bmatrix} K & B_2 \\ B_1^T & -C \end{bmatrix}^{-1} = \begin{bmatrix} K^{-1} - K^{-1}B_2S^{-1}B_1^TK^{-1} & K^{-1}B_2S^{-1} \\ S^{-1}B_1^TK^{-1} & -S^{-1} \end{bmatrix} \tag{3.1}$$

in which $S = C + B_1^TK^{-1}B_2$ (with $B_1 = B_2$) is the Schur complement matrix. Given a vector $[u; v]$ and a block preconditioner $M \approx A$ with the approximation $\widehat{K}$ and $\widehat{S}$ to K and S, respectively. Hence, the preconditioning step may be written as the following.

Preconditioning Step $M^{-1}[u; v]$*:*

solve $p = \widehat{K}^{-1}u$,

solve $q = \widehat{S}^{-1}(B_1^Tp - v)$,

compute $M^{-1}[u,\ v]^T = [\widehat{K}^{-1}(u - B_2q); q]$.

The above preconditioning step has already been introduced in [4, 10] for the block constraint preconditioners; however, it is highlighted here as a unified computational framework for all block preconditioners. It can be observed that within the above preconditioning step, canceling the terms associated with off-diagonal subblocks B_1 and B_2 corresponds

Table 1: Block preconditioners recently proposed for Biot's linear system.

Authors	Type	Block preconditioner $\widehat{\mathbf{K}}$	Block preconditioner $\widehat{\mathbf{S}}$
Phoon et al. [3]	Diagonal	Diag(K) Direct inverse	diag[$C + B^T \operatorname{diag}(K)^{-1} B$] Direct inverse
Simoncini [6]	Triangular	K Incomplete Cholesky	$C + B^T B$ Incomplete Cholesky
Toh et al. [10]	All three types	diag(K) Direct inverse	$C + B^T \widehat{K}^{-1} B$ Incomplete or sparse Cholesky
Bergamaschi et al. [11]	Constraint	$L_K L_K{}^T$ & $(ZZ^T)^{-1}$ Incomplete Cholesky and approximate inverse	$C + B^T \widehat{K}^{-1} B$ Incomplete Cholesky
Haga et al. [13]	Constraint	diag(K) & K AMG	diag[$C + B^T \operatorname{diag}(K)^{-1} B$] & $C + B^T \operatorname{diag}(K)^{-1} B$ Direct inverse & AMG
Chaudhary [15]	Diagonal	Partitioned block diagonal Cholesky & direct inverse	diag[$C + B^T \operatorname{diag}(K)^{-1} B$] Direct inverse

to the block diagonal preconditioning, while canceling either B_1 or B_2 corresponds to the block triangular types. Depending on how $\widehat{K}$ and $\widehat{S}$ are approximated, various block preconditioners may be developed. Regardless of the type of preconditioners adopted, solving the linear systems associated with $\widehat{K}$ and $\widehat{S}$ can not be obviated. Consequently, how $\widehat{K}$ and $\widehat{S}$ are approximated and how these linear systems are solved efficiently are the key features distinguishing recently proposed block preconditioners. For example, Phoon et al. [3] proposed the diagonal approximations to both $\widehat{K}$ and $\widehat{S}$ so that the inverses of $\widehat{K}$ and $\widehat{S}$ can be directly attained. While in the block constraint preconditioner proposed by Toh et al. [10], because it is observed that the size of K is significantly larger than that of S, a diagonal approximation is employed for K, and the incomplete or sparse Cholesky factorization is recommended for $\widehat{S}$. In another version of block constraint preconditioner proposed by Bergamaschi et al. [11, 12], the incomplete Cholesky factorization and factorized approximate inverse are adopted for $\widehat{K}$ and $\widehat{S}$, respectively. Different from the previous studies, Haga et al. [13] recommended solving the linear systems $\widehat{K}$ and $\widehat{S}$ using the algebraic multigrid (AMG) method. To summarize the key differences, these block preconditioners with the proposed $\widehat{K}$ and $\widehat{S}$ as well as the corresponding solution schemes are tabulated in Table 1. In the table, all block preconditioners are categorized into three types including the block diagonal preconditioners, block triangular preconditioners, and block constraint preconditioners which are denoted as "diagonal," "constraint," and "triangular," respectively.

3.2. Modified Preconditioners Based on Matrix Block Structure

There are some advantages to developing preconditioners based on the matrix block structure, particularly in the case of block indefinite linear systems. It is natural to hope that modifying some standard or general preconditioning techniques may lead to improved versions. For instance, the generalized Jacobi preconditioner [3] may be viewed as an

improved version of standard Jacobi preconditioner by referring to the block diagonal. The MSSOR preconditioner [4] may also be regarded as a good example that demonstrates how to develop a preconditioner based on a standard preconditioner to suit indefinite problems. The general nonsymmetric form of MSSOR may be expressed as below

$$M_{\mathrm{MSSOR}} = \left(L_A + \frac{D}{\omega}\right)\left(\frac{D}{\omega}\right)^{-1}\left(U_A + \frac{D}{\omega}\right) = \left(L_A + \tilde{D}\right)\tilde{D}^{-1}\left(U_A + \tilde{D}\right), \tag{3.2}$$

where $\omega \in [1, 2]$ is the relaxation parameter; L_A, U_A is the strictly lower and upper part of matrix A, respectively, that is, $A = L_A + U_A + D_A$. For symmetric linear equation, $L_A = U_A^T$ leads to the symmetric MSSOR. D is the modified diagonal and it is recommended to take GJ, that is,

$$D = M_{\mathrm{GJ}} = \begin{bmatrix} \mathrm{diag}(K) & 0 \\ 0 & \alpha\ \mathrm{diag}\left(\hat{S}\right) \end{bmatrix} \tag{3.3}$$

in which α is a scaling factor which is recommended to be -4 according to the eigenvalue study. It is clear that MSSOR preconditioner can be implemented efficiently when combined with the Eisenstat trick [20]. In the following part, both symmetric version and nonsymmetric version of MSSOR preconditioner will be employed for large-scale nonlinear soil consolidation involving nonassociated plasticity.

4. Numerical Experiments

To simulate soil dilatancy during the soil consolidation process, an elastoplastic soil model with nonassociated plasticity should be used [21]. The nonassociated plasticity theory is a generalization of the classical elastoplastic theory by introducing a new plastic potential function [22], and in the past decades the nonassociated plasticity theory is widely applied in practical finite element analyses.

It is also well known that when the nonassociated plastic soil models are employed in finite element analysis, the resulting linear systems of equations are nonsymmetric. Solving the linear equations separately from the nonlinear iterative procedure may not be wise, because an appropriate combination between the outer nonlinear iterative scheme and an inner linear iterative solution may lead to a significant reduction in computer runtime without sacrificing accuracy [17]. In this work, two solution schemes are proposed and compared for the target problem.

Scheme 1. Applying the Newton-Krylov iterative method. As shown by (2.8), the resultant linear system is nonsymmetric at each Newton iteration, a nonsymmetric Krylov subspace iterative method, such as quasiminimal residual (QMR) [23], is adopted, and hence the nonsymmetric MSSOR preconditioner is used to accelerate its convergence.

Scheme 2. Compared to the nonsymmetric linear systems, solving the symmetric linear systems could lead to a saving in required memory and computer runtime. Therefore,

accelerated nonlinear solution schemes with symmetrized stiffness are attempted here, that is, at each nonlinear iteration one attempts to solve

$$\mathbf{R}\left(\mathbf{u}_{k-1};\mathbf{p}_{k-1}^{ex}\right)+\widehat{\mathbf{A}}_{k-1}\begin{bmatrix}\delta\mathbf{u}_k\\ \delta\mathbf{p}_k^{ex}\end{bmatrix}=0, \tag{4.1}$$

and the solution vector may be updated according to (2.16). The difficulty associated with the scheme is how to construct the symmetric linear systems. In the accelerated symmetric stiffness method proposed by Chen and Cheng [24], the idea is to construct the symmetric constitutive matrix as

$$\operatorname{sym}\left(\mathbf{D}_{ep}\right)=\mathbf{D}_e-\operatorname{sym}\left(\mathbf{D}_p\right), \tag{4.2}$$

where $\operatorname{sym}(\cdot)$ is a symmetrizing symbol. When $\operatorname{sym}(\mathbf{D}_{ep})=\mathbf{D}_e$, it corresponds to the initial stiffness (IS) method proposed by Zienkiewicz et al. [25]. When it is defined that

$$\operatorname{sym}\left(\mathbf{D}_{ep}\right)=\mathbf{D}_{ep}^{G}=\mathbf{D}_e-\frac{\mathbf{D}_e\mathbf{b}\mathbf{b}^T\mathbf{D}_e}{\mathbf{b}^T\mathbf{D}_e\mathbf{b}-\mathbf{c}_G^T\mathbf{h}_G} \tag{4.3}$$

which corresponds to the so-called accelerated K_G approach. In (2.16), the step-length parameter χ_k can be determined by some optimization strategies such as the Chen's method [26] and Thomas' method [27]. Chen's method is derived by minimizing the out-of-balance load at next iteration in the least-squares sense, while the Thomas method is proposed by minimizing the "symmetric" displacement at the next iteration in the least-squares sense. In this study, the solution vector consists of coupled displacement and excess pore water pressure degrees of freedom (DOFs). The Chen method may be more straightforward to apply in this case, which will be demonstrated by the following numerical experiments. As a result, by using the present scheme, the symmetric MSSOR preconditioner may be adopted to accelerate the convergence of the SQMR solver.

4.1. Convergence Criteria

In the examples to be studied, the nonlinear iteration is terminated if the relative residual force criterion is satisfied,

$$\frac{\|\mathbf{R}_k\|_2}{\|\mathbf{F}^{\text{ext}}\|_2}\le \text{Tol\_NL}\quad\text{or}\quad k\ge \text{Maxit\_NL},\qquad (k=0,1,\ldots) \tag{4.4}$$

in which Tol_NL and Maxit_NL are the prescribed tolerance and the maximum nonlinear iterative number, respectively, and in this study Tol_NL and Maxit_NL are set as 0.01 and 50. For the employed Krylov subspace iterative method, the relative residual convergence criterion is adopted,

$$\frac{\left\|\mathbf{R}_{k-1}-\mathbf{A}_{k-1}[\delta\mathbf{u};\delta\mathbf{p}^{ex}]_{k,j}^T\right\|_2}{\|\mathbf{R}_{k-1}\|_2}\le \text{Tol\_Lin},\quad\text{or}\quad j\ge \text{Maxit\_Lin},\qquad (j=0,1,\ldots), \tag{4.5}$$

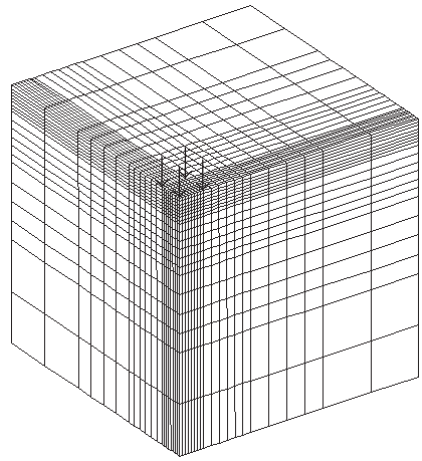


Figure 1: The 20 × 20 × 20 finite element mesh of the flexible footing.

in which zero initial guess is assumed; j is the linear iterative index; Tol_Lin and Maxit_Lin are the prescribed tolerance and the maximum iterative number, respectively, and in this study Maxit_Lin is set as 50000. While to achieve a better performance for the adopted nonlinear scheme, a combined tolerance (in which Tol_Lin = 1.E − 5 is used if the residual load is the external applied force, or Tol_Lin = 1.E − 3 is used if the residual load is the out-of-balance force) is employed, as recommended by Chen and Cheng [24]. Based on our numerical experiences, this recommendation is reasonable because the deformation of a soil body induced by an external load is usually large and should be solved more accurately, while corrections of the deformation to resolve out-of-balance loads are relatively smaller in magnitude and hence may be solved with a less strict tolerance.

In addition, it should be mentioned that the uniform substepping method with $n$substep = 500 (i.e., the number of substeps) is adopted for stress-strain integration. For more advanced automatic substepping algorithms, see Abbo [28]. In the present study, an ordinary personal computer platform equipped with a 2.4 GHz Intel Core(TM)2 Duo CPU and 3 GB physical memory is used.

4.2. Homogeneous Flexible Footing

To investigate and compare the two schemes proposed, a simple flexible footing problem with homogeneous soil material is simulated. For the homogenous soil property, the hydraulic conductivity is assumed to be $k_{s,x} = k_{s,y} = k_{s,z} = 10^{-9}$ m/s, the effective Young's modulus E' as 10 MPa, and the Poisson's ratio as $\nu' = 0.3$. The nonassociated Mohr-Coulomb soil model is used to simulate soil plasticity with the cohesion c'=10 kPa, the internal friction angle $\phi' = 30°$ and the dilatancy angle $\varphi = 5°$. The K_0 approach is used to generate the initial field stress with $K_0 = 1 - \sin\phi'$ and soil unit weight is $\gamma = 18\,\text{kN/m}^3$. Due to the geometric symmetry, only a quadrant of the footing is modeled with the solution domain discretized by 8000 (20 × 20 × 20) 20-node hexahedral consolidation elements as shown in Figure 1, and the resultant total number of DOFs is 107180. For the boundary conditions, standard displacement fixities are assumed, and only the ground surface is drained. The uniform pressure load is applied on a

Table 2: Performance of different solution schemes for the flexible footing example.

Load step no.	Number of nonlinear iteration	Average iterations for each linear system	Total solution time (hours)
Scheme 1: Newton + QMR preconditioned by nonsymmetric MSSOR			
1	1	1940	0.137
2	4	1755	0.494
3	5	1756	0.624
4	5	1732	0.625
5	6	1723	0.751
Total	21	—	2.662
Scheme 2: IS (Chen) method + SQMR preconditioned by symmetric MSSOR			
1	1	1060	0.061
2	4	780	0.180
3	5	780	0.241
4	6	793	0.308
5	6	753	0.317
Total	22	—	1.129
Scheme 2: K_G (Chen) method + SQMR preconditioned by symmetric MSSOR			
1	1	1060	0.060
2	4	845	0.182
3	4	865	0.189
4	4	855	0.200
5	5	860	0.256
Total	18	—	0.903

$1 \times 1\,\text{m}^2$ area. It is increased incrementally using a total of 5 load steps. Each load increment is 0.1 MPa per second.

Table 2 provides the numerical performance of two solution schemes proposed above, that is, the Newton-Krylov (i.e., Newton-QMR) solution scheme and the Chen accelerated symmetric stiffness scheme. In the two schemes, the nonsymmetric MSSOR and symmetric MSSOR are used in conjunction with QMR and SQMR solver, respectively. Because the combined tolerance is used, the required linear iterative count for the first nonlinear iteration is higher than the following nonlinear iterations. When comparing the two schemes, it is found that the average iterative count required by MSSOR-preconditioned QMR is about two times of that required by MSSOR preconditioned SQMR solver, explaining the final results. In addition, the fact that two matrix-vector products are required in each iteration of QMR contributes to the longer computer runtime. Note that the average iterative count required by SQMR in Chen accelerated IS method is slightly smaller than that required by SQMR in Chen accelerated K_G method, indicating that the initial stiffness (i.e., the elastic stiffness) could be better conditioned. When observing the required nonlinear iterations by each scheme, it is interesting to note that three solution strategies (i.e., the Newton scheme, the Chen accelerated IS method, and Chen accelerated K_G method) exhibit similar nonlinear iterative behavior, although the Chen accelerated K_G method leads to slightly less nonlinear iterations (i.e., 18) than the other two counterparts. The exhibited similarity in nonlinear convergence indicates that the nonlinearity in the present problem is not strong. Obviously, for the homogeneous flexible footing problem the Chen accelerated K_G method coupled with

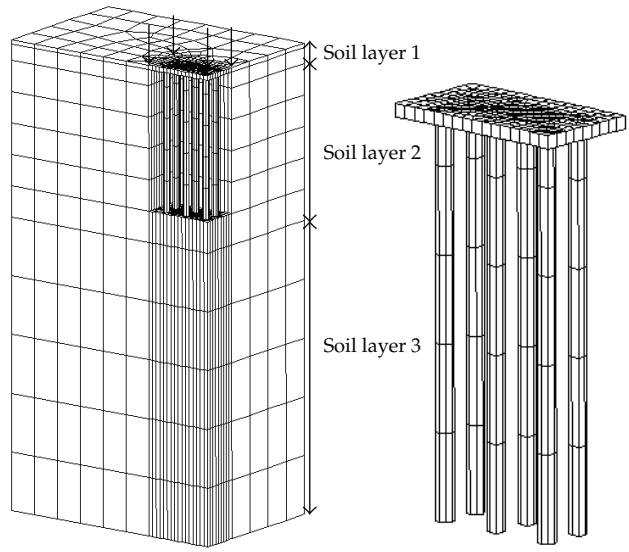


Figure 2: The finite element mesh of heterogeneous pile-group example.

Table 3: Material properties for the pile-group foundation.

Material	E' (MPa)	ν'	k (m/s)	c' (kPa)	ϕ' (°)	φ (°)
Soil layer 1	20	0.3	10^{-5}	5	35	5
Soil layer 2	8	0.3	10^{-9}	50	30	8
Soil layer 3	50	0.3	10^{-6}	10	35	7
Pile and pile cap concrete	3×10^7	0.18	10^{-14}	—	—	—

MSSOR preconditioned SQMR solver may lead to a 66% reduction in total computer runtime than the Newton-QMR scheme preconditioned by nonsymmetric MSSOR. Even for the Chen accelerated IS method, a 57% reduction of total computer runtime can be achieved. It is noteworthy that the Thomas' acceleration strategy appears to be more effective in a past study [24]. However, for the coupled consolidation problem discussed in the present study, the Thomas accelerated IS method requires 1, 4, 7, 8, 8 nonlinear iterations, respectively, for the five load steps, while the Thomas accelerated K_G method takes 1, 5, 8, 6 nonlinear iterations, respectively, for the first four load steps, but it fails to converge at the last load step.

4.3. Heterogeneous Soil-Structure Interaction Example

A pile-group example is used to demonstrate the interaction between soil and structure. Because of the significant contrast in material stiffness between soils and concrete, the performance of the MSSOR preconditioner may deteriorate with the increasing contrast ratio, as noticed and investigated by Chaudhary [15]. As shown in Figure 2, the pile-group finite element mesh has total 4944 elements, in which 430 are concrete elements. The material properties for soils and concrete are tabulated in Table 3. From the table, it is seen that the concrete material is modeled by linear elastic consolidation elements with extremely small permeability, but the soils are still simulated by the Mohr-Coulomb model.

Table 4: Performance of different solution schemes for the pile-group example.

Load step no.	Number of nonlinear iteration	Average iterations for each linear system	Total solution time (hours)
Scheme 1: Newton + QMR preconditioned by nonsymmetric MSSOR			
1	4	7400	0.951
2	5	7944	1.276
3	6 (1)	13967	2.601
Total	15	—	4.823
Scheme 2: IS (Chen) method + SQMR preconditioned by symmetric MSSOR			
1	8	3300	0.464
2	13	3095	0.707
3	12	3020	0.636
Total	33	—	1.801
Scheme 2: K_G (Chen) method + SQMR preconditioned by symmetric MSSOR			
1	4	3410	0.242
2	5	3180	0.283
3	6	3284	0.349
Total	15	—	0.874

Table 4 provides numerical results of these solution schemes for the pile-group example. To be efficient for the accelerated symmetric stiffness methods, the combined tolerance introduced in Section 4.1 is still adopted. In the pile-group example, three uniform load steps are simulated and the pressure increment of 50 kPa is applied on the pile cap for a 2-day time increment. Similar to the flexible footing example, the iteration number spent by nonsymmetric MSSOR-preconditioned QMR is about two times of that of MSSOR-preconditioned SQMR solver. From the nonlinear iteration counts, it is seen that the Chen accelerated K_G method achieves a similar convergence rate as that of the Newton method, but at the third load step QMR solver dose not converge at one nonlinear iteration, which denoted by the bracketed number. Hence, at that load step the average iterative number for each linear system is remarkably increased because of Maxit_Lin = 50000. Compared to the Newton method, the Chen accelerated symmetric stiffness methods show better convergence behaviors, while the Chen accelerated IS method may be markedly slower than Newton method, indicating that the soil-structure interaction involving significant contrast in material stiffness may produce a stronger nonlinearity than the simple homogeneous footing problem. Upon close examination of the computer runtime, it is noticed that even though it is slower in convergence, the Chen accelerated IS method may lead to a reduction of 63% in computer runtime compared with the Newton method. When using K_G symmetric stiffness, the resultant convergence rate is similar to the Newton method, the saving in computer runtime is more impressive and a reduction of 82% may be attained. Furthermore, the Thomas acceleration scheme is examined for the initial stiffness and the K_G stiffness matrix, respectively. Both nonlinear solution strategies associated with the Thomas acceleration scheme fail, indicating that the Thomas acceleration scheme may not be suitable for problems involving coupled pore water pressure and the displacement degrees of freedom.

5. Conclusions

Soil consolidation accompanied with soil dilatancy may be frequently encountered in practice, and usually it can be modeled by nonassociated soil models. Due to the nonassociated soil model, the resultant finite element linear equation is nonsymmetric. To solve the nonsymmetric coupled Biot's linear systems of equations, two schemes in conjunction with MSSOR preconditioner are proposed and examined. Some useful observations are summarized as follows.

(1) Two schemes are proposed for the Biot's consolidation problem involving nonassociated plasticity. Depending on the discretized linear equations, both nonsymmetric and symmetric MSSOR preconditioners are adopted for such problems,

(2) In the accelerated symmetric stiffness methods for coupled consolidation problems, the Thomas' acceleration strategy does not exhibit better convergence behaviors as observed in the previous studies. On the other hand, the Chen's acceleration strategy is more effective, and hence it is the recommended approach for such coupled consolidation problems.

(3) Compared to the Newton solution scheme which adopts the QMR solver preconditioned by nonsymmetric MSSOR preconditioner, the Chen accelerated symmetric stiffness approaches, which use the SQMR solver preconditioned by the symmetric MSSOR, may lead to a significant reduction in computer runtime. Based on the above numerical experiments, it may be concluded that the Chen accelerated symmetric stiffness methods have considerable potential to be exploited for solution of large-scale Biot's consolidation problems.

Acknowledgment

The research is supported in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, no. (2010) 1561, the Fundamental Research Funds for the Central Universities, no. 2011JBM070, and the Research Fund Program of State Key Laboratory of Hydroscience and Engineering, no. SKL-2010-TC-2.

References

[1] B. A. Cipra, "The best of the 20th century: Editors name Top 10 Algorithms," *SIAM News*, vol. 33, no. 4, pp. 1–2, 2000.

[2] E. Chow and Y. Saad, "Experimental study of ILU preconditioners for indefinite matrices," *Journal of Computational and Applied Mathematics*, vol. 86, no. 2, pp. 387–414, 1997.

[3] K. K. Phoon, K. C. Toh, S. H. Chan, and F. H. Lee, "An efficient diagonal preconditioner for finite element solution of Biot's consolidation equations," *International Journal for Numerical Methods in Engineering*, vol. 55, no. 4, pp. 377–400, 2002.

[4] X. Chen, K. C. Toh, and K. K. Phoon, "A modified SSOR preconditioner for sparse symmetric indefinite linear systems of equations," *International Journal for Numerical Methods in Engineering*, vol. 65, no. 6, pp. 785–807, 2006.

[5] I. Perugia and V. Simoncini, "Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations," *Numerical Linear Algebra with Applications*, vol. 7, no. 7-8, pp. 585–616, 2000.

[6] V. Simoncini, "Block triangular preconditioners for symmetric saddle-point problems," *Applied Numerical Mathematics*, vol. 49, no. 1, pp. 63–80, 2004.

[7] R. W. Freund and N. M. Nachtigal, "A new Krylov-subspace method for symmetric indefinite linear systems," in *Proceedings of the 14th IMACS World Congress on Computational and Applied Mathematics*, W. F. Ames, Ed., pp. 1253–1256, Atlanta, Ga, USA, 1994.

[8] C. Keller, N. I. M. Gould, and A. J. Wathen, "Constraint preconditioning for indefinite linear systems," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1300–1317, 2000.

[9] Z.-Z. Bai, M. K. Ng, and Z.-Q. Wang, "Constraint preconditioners for symmetric indefinite matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 2, pp. 410–433, 2009.

[10] K.-C. Toh, K.-K. Phoon, and S.-H. Chan, "Block preconditioners for symmetric indefinite linear systems," *International Journal for Numerical Methods in Engineering*, vol. 60, no. 8, pp. 1361–1381, 2004.

[11] L. Bergamaschi, M. Ferronato, and G. Gambolati, "Mixed constraint preconditioners for the iterative solution of FE coupled consolidation equations," *Journal of Computational Physics*, vol. 227, no. 23, pp. 9885–9897, 2008.

[12] M. Ferronato, L. Bergamaschi, and G. Gambolati, "Performance and robustness of block constraint preconditioners in finite element coupled consolidation problems," *International Journal for Numerical Methods in Engineering*, vol. 81, no. 3, pp. 381–402, 2010.

[13] J. B. Haga, H. Osnes, and H. P. Langtangen, "Efficient block preconditioners for the coupled equations of pressure and deformation in highly discontinuous media," *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 35, no. 13, pp. 1466–1482, 2011.

[14] C. Janna, M. Ferronato, and G. Gambolati, "A block Fsai-Ilu parallel preconditioner for symmetric positive definite linear systems," *SIAM Journal on Scientific Computing*, vol. 32, no. 5, pp. 2468–2484, 2010.

[15] K. B. Chaudhary, *Preconditioners for soil-structure interaction problems with significant material stiffness contrast*, Ph.D. thesis, National University of Singapore, 2010.

[16] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, "Inexact Newton methods," *SIAM Journal on Numerical Analysis*, vol. 19, no. 2, pp. 400–408, 1982.

[17] P. N. Brown and Y. Saad, "Convergence theory of nonlinear Newton-Krylov algorithms," *SIAM Journal on Optimization*, vol. 4, no. 2, pp. 297–330, 1994.

[18] M. A. Biot, "General theory of three-dimensional consolidation," *Journal of Applied Physics*, vol. 12, no. 2, pp. 155–164, 1941.

[19] I. M. Smith and D. V. Griffiths, *Programming the Finite Element Method*, John Wiley & Sons, Chichester, UK, 2nd edition, 1988.

[20] S. C. Eisenstat, "Efficient implementation of a class of preconditioned conjugate gradient methods," *SIAM Journal on Scientific and Statistical Computing*, vol. 2, no. 1, pp. 1–4, 1981.

[21] P. A. Vermeer and R. de Borst, "Non-associated plasticity for soils, concrete and rock," *Heron*, vol. 29, no. 3, pp. 1–62, 1984.

[22] O. C. Zienkiewicz, C. Humpheson, and R. W. Lewis, "Associated and non-associated visco-plasticity and plasticity in soil mechanics," *Geotechnique*, vol. 25, no. 4, pp. 671–689, 1975.

[23] R. W. Freund and N. M. Nachtigal, "QMR: a quasi-minimal residual method for non-Hermitian linear systems," *Numerische Mathematik*, vol. 60, no. 3, pp. 315–339, 1991.

[24] X. Chen and Y.-G. Cheng, "On accelerated symmetric stiffness techniques for non-associated plasticity with application to soil problems," *Engineering Computations*, vol. 28, no. 8, pp. 1044–1063, 2011.

[25] O. C. Zienkiewicz, S. Valliappan, and I. P. King, "Elasto-plastic solutions of engineering problems "initial stress", finite element approach," *International Journal for Numerical Methods in Engineering*, vol. 1, no. 1, pp. 75–100, 1969.

[26] C. N. Chen, "Efficient and reliable accelerated constant stiffness algorithms for the solution of non-linear problems," *International Journal for Numerical Methods in Engineering*, vol. 35, no. 3, pp. 481–490, 1992.

[27] J. N. Thomas, "An improved accelerated initial stress procedure for elasto-plastic finite element analysis," *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 8, no. 4, pp. 359–379, 1984.

[28] A. J. Abbo, *Finite element algorithms for elastoplasticity and consolidation*, Ph.D. thesis, University of Newcastle, 1997.

Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 638546, 12 pages
doi:10.1155/2012/638546

Research Article

A Direct Eigenanalysis of Multibody System in Equilibrium

Cheng Yang,[1] Dazhi Cao,[2] Zhihua Zhao,[2] Zhengru Zhang,[1] and Gexue Ren[2]

[1] *School of Mathematical Sciences, Beijing Normal University, Beijing 100875, China*
[2] *School of Aerospace, Tsinghua University, Beijing 100084, China*

Correspondence should be addressed to Cheng Yang, yangchengabcd@126.com

Received 4 September 2011; Accepted 2 November 2011

Academic Editor: Massimiliano Ferronato

This paper presents a direct eigenanalysis procedure for multibody system in equilibrium. The first kind Lagrange's equation of the dynamics of multibody system is a set of differential algebraic equations, and the equations can be used to solve the equilibrium of the system. The vibration of the system about the equilibrium can be described by the linearization of the governing equation with the generalized coordinates and the multipliers as the perturbed variables. But the multiplier variables and the generalize coordinates are not in the same dimension. As a result, the system matrices in the perturbed vibration equations are badly conditioned, and a direct application of the mature eigensolvers does not guarantee a correct solution to the corresponding eigenvalue problem. This paper discusses the condition number of the problem and proposes a method for preconditioning the system matrices, then the corresponding eigenvalue problem of the multibody system about equilibrium can be smoothly solved with standard eigensolver such as ARPACK. In addition, a necessary frequency shift technology is also presented in the paper. The importance of matrix conditioning and the effectiveness of the presented method for preconditioning are demonstrated with numerical examples.

1. Introduction

Modal analysis of multibody system [1, 2] has important application in structure analysis and modal reduction method [3–6]. The eigenanalysis methods for constrained multibody system can be mainly divided into two categories, namely, the eigenanalysis method by transforming the equations of motion from DAEs to ODEs by selecting independent coordinates [7–9] and the direct eigenanalysis method [10–12]. Compared with structural systems, the governing equation of constrained multibody system is differential algebraic equations, instead of the ordinary differential systems. In the reduction methods, the differential

algebraic equations of multibody system are reduced into ordinary differential equations for independent general coordinates, through looking for a set of orthogonal basis of constraint Jacobian matrix to eliminate the constraints from the system governing equations. The reduction method is accurate, efficient, and stable provided that the degree of freedom of the system and the number of constraints are small. However, the reduction method is considered not suitable for large scale problems due to the spoilage of sparse matrices [13].

In the direct methods, the nonlinear differential and algebraic equations are simultaneously linearized about a given state of the system, that is, the algebraic constraint equations are kept during the eigenanalysis and the sparsity of system matrices is preserved. Depending on how the problem is actually formulated, the enlarged mass matrix may become structurally singular (eigenvalues associated to constraints become infinite); other approaches may result in a structurally singular enlarged stiffness matrix (eigenvalues associated to constraints become 0). Usually a frequency shift is applied, and structurally nonsingular enlarged matrices are resulted. But in most cases, matrices' condition number of the frequency shifted system is very large and numerical singular due to the constraints and a corresponding precondition is needed. The root cause for numerical singularity can be understood by the fact that the dynamic equations and the Lagrange multipliers carry a dimension of force while the constraint equations and the general coordinate a dimension of length or angular. The literature [14, 15] suggested multiplying a parameter to the constraints' corresponding rows and columns of system matrices as a preconditioning. The time step integration takes a parameter associate with time step and integration format to do the precondition. But the eigenanalysis about a static equilibrium has nothing to do with time integration, and we need to find an appropriate parameter.

For the direct eigenanalysis of constrained multibody system in static equilibrium, this paper proposes a procedure and a necessary preconditioning method of system matrices to guarantee correct eigensolution. This paper is organized as follows. Section 2 schematically describes the formulation of constrained multibody system based on Lagrange equation with multipliers and the direct eigenvalue problem about equilibrium. Section 3 presented a detailed condition number analysis of the system matrices and a preconditioning scaling technique. Section 4 introduces the selection of eigensolver and numerical examples.

2. Governing Equations of Multibody System and the Direct Eigenvalue Problem

The dynamic equations of a general multibody system are usually established with Lagrange's equations with multipliers [16]. The system equations are

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{x}}}\right) - \frac{\partial L}{\partial \mathbf{x}} &= \left(\frac{\partial \boldsymbol{\Phi}}{\partial \mathbf{x}}\right)^{T} \boldsymbol{\lambda} + \mathbf{f}, \\ \boldsymbol{\Phi}(\mathbf{x}) &= \mathbf{0}, \end{aligned} \tag{2.1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of general coordinates, $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the vector of Lagrange multipliers, L is the Lagrange function associated with $(\dot{\mathbf{x}}, \mathbf{x})$, $\mathbf{f}$ is the vector of general nonconservative

forces which are functions of $(\dot{\mathbf{x}}, \mathbf{x}, t)$, $\mathbf{\Phi}$ is the vector of constraint functions associated with $(\mathbf{x})$, and $(\partial\mathbf{\Phi}/\partial\mathbf{x})^T\boldsymbol{\lambda}$ is the vector of generalized constraint forces. Denote

$$\begin{aligned}\mathbf{F}(\ddot{\mathbf{x}}, \dot{\mathbf{x}}, \mathbf{x}, t) &= \left(-\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{x}}}\right) + \frac{\partial L}{\partial \mathbf{x}} + \mathbf{f}\right)(\ddot{\mathbf{x}}, \dot{\mathbf{x}}, \mathbf{x}, t),\\ \mathbf{G}(\mathbf{x}) &= \left(\frac{\partial \mathbf{\Phi}(\mathbf{x})}{\partial \mathbf{x}}\right)^T.\end{aligned} \tag{2.2}$$

Then (2.1) becomes

$$\begin{aligned}\mathbf{F}(\ddot{\mathbf{x}}, \dot{\mathbf{x}}, \mathbf{x}, t) + \mathbf{G}(\mathbf{x})\boldsymbol{\lambda} &= \mathbf{0},\\ \mathbf{\Phi}(\mathbf{x}) &= \mathbf{0}.\end{aligned} \tag{2.3}$$

The equilibrium state $(\mathbf{x_0}, \boldsymbol{\lambda_0}, t_0)$ satisfies

$$\begin{aligned}\mathbf{F}(\mathbf{0}, \mathbf{0}, \mathbf{x_0}, t) + \mathbf{G}(\mathbf{x_0})\boldsymbol{\lambda_0} &= \mathbf{0},\\ \mathbf{\Phi}(\mathbf{x_0}) &= \mathbf{0}.\end{aligned} \tag{2.4}$$

Assume small vibration about the equilibrium $(\delta\ddot{\mathbf{x}}, \delta\dot{\mathbf{x}}, \mathbf{x}_0 + \delta\mathbf{x}, \boldsymbol{\lambda}_0 + \delta\boldsymbol{\lambda}, t)$, then

$$\begin{aligned}\mathbf{F}(\delta\ddot{\mathbf{x}}, \delta\dot{\mathbf{x}}, \mathbf{x_0} + \delta\mathbf{x}, t) + \mathbf{G}(\mathbf{x}_0 + \delta\mathbf{x})(\boldsymbol{\lambda}_0 + \delta\boldsymbol{\lambda}) &= \mathbf{0},\\ \mathbf{\Phi}(\mathbf{x_0} + \delta\mathbf{x}) &= \mathbf{0}.\end{aligned} \tag{2.5}$$

Carry out the Taylor expansion and reserve linear terms, we have

$$\widehat{\mathbf{M}}(\mathbf{x}_0)\delta\ddot{\mathbf{y}} + \widehat{\mathbf{C}}(\mathbf{x}_0)\delta\dot{\mathbf{y}} + \widehat{\mathbf{K}}(\mathbf{x}_0, \boldsymbol{\lambda}_0)\delta\mathbf{y} = \mathbf{0}, \tag{2.6}$$

where $\widehat{\mathbf{M}}(\mathbf{x}_0) = \begin{bmatrix}\mathbf{M}(\mathbf{x}_0) & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}$, $\widehat{\mathbf{C}}(\mathbf{x}_0)\begin{bmatrix}\mathbf{C}(\mathbf{x}_0) & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}$, $\widehat{\mathbf{K}}(\mathbf{x}_0, \boldsymbol{\lambda}_0) = \begin{bmatrix}\mathbf{K}(\mathbf{x}_0, \boldsymbol{\lambda}_0) & \mathbf{G}(\mathbf{x}_0)\\ \mathbf{G}^T(\mathbf{x}_0) & \mathbf{0}\end{bmatrix}$, $\mathbf{M}(\mathbf{x}_0) = (\partial\mathbf{F}/\partial\ddot{\mathbf{x}})(\mathbf{x}_0)$, $\mathbf{C}(\mathbf{x}_0) = (\partial\mathbf{F}/\partial\dot{\mathbf{x}})(\mathbf{x}_0)$, $\mathbf{K}(\mathbf{x}_0, \boldsymbol{\lambda}_0) = (\partial(\mathbf{F} + \mathbf{G}\boldsymbol{\lambda})/\partial\mathbf{x})(\mathbf{x}_0, \boldsymbol{\lambda}_0)$, $\delta\mathbf{y} = \begin{bmatrix}\delta\mathbf{x}\\ \delta\boldsymbol{\lambda}\end{bmatrix}$.

Write the solution as $\delta\mathbf{y}(t) = e^{rt}\mathbf{V}$, where r is one of the eigenvalues, $\mathbf{V}$ is the corresponding eigenvector, and the eigenvalue problem is

$$\left(r^2\widehat{\mathbf{M}} + r\widehat{\mathbf{C}} + \widehat{\mathbf{K}}\right)\mathbf{V} = \mathbf{0}. \tag{2.7}$$

A frequency shift is applied to (2.7) by let $\overline{r} = r - \alpha$, $\alpha \in \mathbb{R}^+$, and then

$$\left(\overline{r}^2\overline{\mathbf{M}} + \overline{r}\overline{\mathbf{C}} + \overline{\mathbf{K}}\right)\mathbf{V} = \mathbf{0}, \tag{2.8}$$

where $\overline{\mathbf{M}} = \widehat{\mathbf{M}}$, $\overline{\mathbf{K}} = \alpha^2\widehat{\mathbf{M}} + \alpha\widehat{\mathbf{C}} + \widehat{\mathbf{K}}$, $\overline{\mathbf{C}} = 2\alpha\widehat{\mathbf{M}} + \widehat{\mathbf{C}}$. The frequency shift makes us calculate the eigenvalues about α and makes sure $\overline{\mathbf{K}}$'s reversibility (the probability of singular is zero when

$\alpha \neq 0$) as the system's energy must be in the form of kinetic energy or deformation energy. The eigenvalue problem can be transformed into the state space form as the following:

$$\overline{r}\widetilde{\mathbf{M}}\mathbf{Y} = \widetilde{\mathbf{K}}\mathbf{Y}, \tag{2.9}$$

where $\mathbf{Y} = \begin{pmatrix} \mathbf{y} \\ \overline{r}\mathbf{y} \end{pmatrix}$, $\widetilde{\mathbf{M}} = \begin{bmatrix} -\overline{\mathbf{C}} & -\overline{\mathbf{M}} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}$, $\widetilde{\mathbf{K}} = \begin{bmatrix} \overline{\mathbf{K}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$.

3. Ill-Condition in the Matrices of the Perturbed Vibration Equations and the Preconditioning

Consider the shifted eigenvalue problem (2.8) in the undamped case

$$\left[\overline{r}^2\overline{\mathbf{M}} + \overline{\mathbf{K}}\right]\mathbf{V} = \mathbf{0}. \tag{3.1}$$

It is more convenient to consider the standard eigenvalue problem than the original generalized one

$$\left[\overline{\mathbf{K}}^{-1}\overline{\mathbf{M}} + \overline{r}^{-2}\mathbf{I}\right]\mathbf{V} = \mathbf{0}. \tag{3.2}$$

As the inverse of matrix $\overline{\mathbf{K}}$ is used, cases in large condition number of $\overline{\mathbf{K}}$ could make the direct solution of $\overline{\mathbf{K}}^{-1}\overline{\mathbf{M}}$ result in erroneous eigensolution. As shown in Section 4, the eigenvalue problems (2.8) obtained by the linearization of the Lagrange equations with multipliers for practical multibody systems usually encounter serious condition number problem. The reason mainly lies in the different dimensions of the multiplier variables and constraint equations with that of generalized coordinates and governing equations.

Inspired by the eigenanalysis method through transforming the equations of motion from DAEs to ODEs by selecting independent coordinates, we divide the matrix into blocks and calculate its inverse by Gaussian elimination. Let $\alpha^2\mathbf{M} + \mathbf{K} = c\mathbf{D}$ where $c = \|\alpha^2\mathbf{M} + \mathbf{K}\|_\infty$. Without loss of generality, let the first m (the number of constraints) rows of $\mathbf{G}$ be linearly independent and be denoted as $\mathbf{G}_1 \in \mathbb{R}^{m\times m}$. Divide the matrix as

$$\overline{\mathbf{K}} = \begin{bmatrix} c\mathbf{D}_{11} & c\mathbf{D}_{12} & \mathbf{G}_1 \\ c\mathbf{D}_{21} & c\mathbf{D}_{22} & \mathbf{G}_2 \\ \mathbf{G}_1^T & \mathbf{G}_2^T & \mathbf{0} \end{bmatrix}, \tag{3.3}$$

where $\mathbf{D}_{11} \in \mathbb{R}^{m\times m}$, $\mathbf{D}_{22} \in \mathbb{R}^{(n-m)\times(n-m)}$, $\mathbf{D}_{12} \in \mathbb{R}^{m\times(n-m)}$, $\mathbf{D}_{21} \in \mathbb{R}^{(n-m)\times m}$, $\mathbf{G}_2 \in \mathbb{R}^{(n-m)\times m}$, n is the number of freedom of general coordinates.

Then

$$\overline{\mathbf{K}}^{-1} = c^{-1}\begin{bmatrix} \mathbf{G}_1^{-T}\mathbf{G}_2^T\mathbf{P}^{-1}\mathbf{G}_2\mathbf{G}_1^{-1} & -\mathbf{G}_1^{-T}\mathbf{G}_2^T\mathbf{P}^{-1} & c\left(\mathbf{G}_1^{-T} + \mathbf{G}_1^{-T}\mathbf{G}_2^T\mathbf{P}^{-1}\mathbf{R}\right) \\ -\mathbf{P}^{-1}\mathbf{G}_2\mathbf{G}_1^{-1} & \mathbf{P}^{-1} & -c\mathbf{P}^{-1}\mathbf{R} \\ c\left(\mathbf{G}_1^{-1} + \mathbf{Q}\mathbf{P}^{-1}\mathbf{G}_2\mathbf{G}_1^{-1}\right) & -c\mathbf{Q}\mathbf{P}^{-1} & c^2\left(\mathbf{Q}\mathbf{P}^{-1}\mathbf{R} - \mathbf{N}\right) \end{bmatrix}, \tag{3.4}$$

where $\mathbf{N} = \mathbf{G}_1^{-1}\mathbf{D}_{11}\mathbf{G}_1^{-T}$, $\mathbf{P} = \mathbf{D}_{22} + \mathbf{G}_2\mathbf{N}\mathbf{G}_2^T - \mathbf{D}_{21}\mathbf{G}_1^{-T}\mathbf{G}_2^T - \mathbf{G}_2\mathbf{G}_1^{-1}\mathbf{D}_{12}$, $\mathbf{Q} = \mathbf{G}_1^{-1}\mathbf{D}_{12} - \mathbf{N}\mathbf{G}_2^T$, $\mathbf{R} = \mathbf{D}_{21}\mathbf{G}_1^{-T} - \mathbf{G}_2\mathbf{N}$. See appendix for detail of Gaussian elimination.

Assume

$$\|\mathbf{N}\|_\infty \approx \left\|\mathbf{P}^{-1}\right\|_\infty \approx \|\mathbf{Q}\|_\infty \approx \|\mathbf{R}\|_\infty = O(1). \tag{3.5}$$

Then it shows $\|\overline{\mathbf{K}}^{-1}\|_\infty = O(c)$ and by taking condition number $\mathrm{cond}(\cdot) = \|\cdot\|_\infty \|\cdot^{-1}\|_\infty$

$$\mathrm{cond}\left(\overline{\mathbf{K}}\right) \approx O\left(c^2\right). \tag{3.6}$$

Equation (3.5) can be expected when the elements of eigenvector of the underlying ODE system have about the same magnitude, that is, the elastic parameters of the physical system are about the same magnitude in physical description, and this is often the case. The elastic parameters mean the Young' modulus of flexible bodies, stiffness of spring forces, and so forth.

The condition number of the original system can be changed by scaling the multiplier variables and constraint equations. Let $\widehat{\boldsymbol{\lambda}} = k^{-1}\boldsymbol{\lambda}$ and multiply the constraint equations by k in (2.1) we have

$$\begin{gathered}\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{x}}}\right) - \frac{\partial L}{\partial \mathbf{x}} = k\left(\frac{\partial \boldsymbol{\Phi}}{\partial \mathbf{x}}\right)^T \widehat{\boldsymbol{\lambda}} + \mathbf{f}, \\ k\boldsymbol{\Phi}(\mathbf{x}) = \mathbf{0}.\end{gathered} \tag{3.7}$$

The corresponding block matrices are

$$\begin{gathered}\overline{\mathbf{K}} = \begin{bmatrix} c\mathbf{D}_{11} & c\mathbf{D}_{12} & k\mathbf{G}_1 \\ c\mathbf{D}_{21} & c\mathbf{D}_{22} & k\mathbf{G}_2 \\ k\mathbf{G}_1^T & k\mathbf{G}_2^T & \mathbf{0} \end{bmatrix}, \\ \overline{\mathbf{K}}^{-1} = c^{-1}\begin{bmatrix} \mathbf{G}_1^{-T}\mathbf{G}_2^T\mathbf{P}^{-1}\mathbf{G}_2\mathbf{G}_1^{-1} & -\mathbf{G}_1^{-T}\mathbf{G}_2^T\mathbf{P}^{-1} & k^{-1}c\left(\mathbf{G}_1^{-T} + \mathbf{G}_1^{-T}\mathbf{G}_2^T\mathbf{P}^{-1}\mathbf{R}\right) \\ -\mathbf{P}^{-1}\mathbf{G}_2\mathbf{G}_1^{-1} & \mathbf{P}^{-1} & -k^{-1}c\mathbf{P}^{-1}\mathbf{R} \\ k^{-1}c\left(\mathbf{G}_1^{-1} + \mathbf{Q}\mathbf{P}^{-1}\mathbf{G}_2\mathbf{G}_1^{-1}\right) & -k^{-1}c\mathbf{Q}\mathbf{P}^{-1} & k^{-2}c^2\left(\mathbf{Q}\mathbf{P}^{-1}\mathbf{R} - \mathbf{N}\right) \end{bmatrix}. \end{gathered} \tag{3.8}$$

Then the condition number analysis shows

$$\mathrm{cond}\left(\overline{\mathbf{K}}\right) \approx O\left(\max(k, c) \times \max\left(k^{-1}c, k^{-2}c^2\right)\right). \tag{3.9}$$

Choosing $k \approx c$, the estimation of condition number decreases from about $O(c^2)$ to about $O(1)$.

The elements of matrix $\mathbf{K}$ stand for the elastic coefficients of the system and the elements of $\mathbf{G}$ and $\mathbf{G}^T$ work similarly as the *stiffness of constraints*. Taking $\overline{\mathbf{K}}$ as an enlarged stiffness matrix, the scaling is equivalent to changing the stiffness of constraints from $O(1)$ to the same magnitude of physical stiffness.

Note that the condition cannot be improved by only scaling multipliers or by only scaling constraint equations. It is suggested that any different dimension in the procedure of modeling could lead to bad condition of numerical calculation.

The complete process for direct eigenanalysis of constrained deformable multibody system is thus as follows.

Step 1. Calculate the static equilibrium $(\mathbf{x}_0, \boldsymbol{\lambda}_0)$ from (2.4).

Step 2. Calculate $\widehat{\mathbf{M}}$, $\widehat{\mathbf{C}}$, $\widehat{\mathbf{K}}$ at $(\mathbf{x}_0, \boldsymbol{\lambda}_0)$.

Step 3. Precondition $\widehat{\mathbf{K}}$ with c which is the maximum absolute element of $\widehat{\mathbf{K}}$.

Step 4. Frequency shift with parameter $\alpha(1 \sim 10)$ with (2.8).

Step 5. Enlarge the matrix into state space as (2.9).

Step 6. Solve the eigenvalues and eigenvectors form (2.9) with eigensolver such as ARPACK.

Step 7. Add α to the eigenvalues and multiply the eigenvector components of Lagrange multipliers with c.

4. Selection of Eigensolver and Numerical Examples

The famous iteration method Implicitly Restarted Arnoldi Method (IRAM) [17, 18] is used in the numerical solution of eigenvalue problem (2.9) in the paper. Arnoldi method is one of the Krylov subspace methods [19] which calculate the largest few eigenvalues of a matrix. The presented numerical examples have verified the efficiency of IRAM in practical calculation. The famous parallel sparse linear system solver PARDISO [20, 21] is used to provide $\widetilde{\mathbf{K}}^{-1}$ of (2.9) in the paper. A cantilever and a four-bar linkage are calculated to illustrate the effect of the condition number of the system matrix and the effectiveness of the presented preconditioning method. It is shown that the numerical results are physically meaningless without scaling and the precondition presented in the paper is very effective.

4.1. A Cantilever

In this example, beam element and 3-dimensional solid element are separately used to model the cantilever in the multibody frame, as shown in Figure 1. The parameters of the cantilever are shown in Table 1. The models are computed with the presented direct eigenanalysis method in two cases, that is, with and without axial tension, and the results are compared with analytical results based on the mathematical physics, see Tables 2 and 3.

The governing equation and boundary conditions for the single-mode-free vibration of the cantilever under axial tension is as the following:

$$\begin{gathered} EI\frac{d^4y}{dx^4} - F\frac{d^2y}{dx^2} - \rho\omega^2 y = 0, \\ y\big|_0 = 0 \qquad \left.\frac{dy}{dx}\right|_0 = 0 \qquad \left.\frac{d^2y}{dx^2}\right|_l = 0 \qquad \left.\frac{d^3y}{dx^3}\right|_l = a\left.\frac{dy}{dx}\right|_l, \end{gathered} \tag{4.1}$$

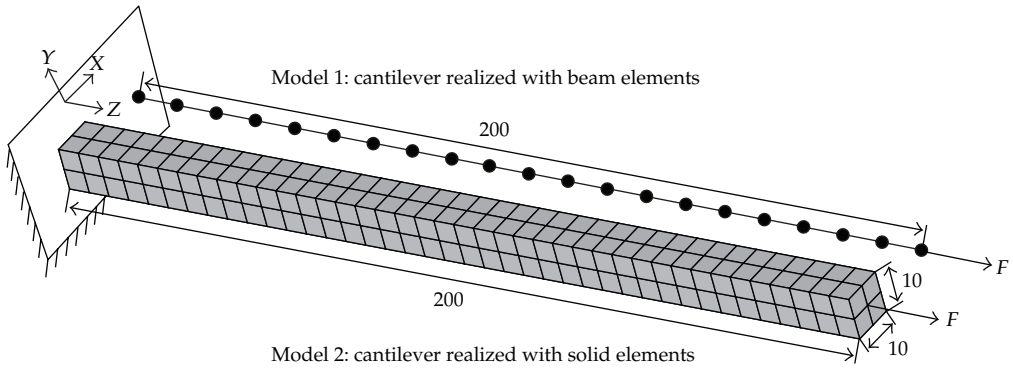


Figure 1: Two models of cantilever.

Table 1: Parameters for the cantilever.

Item	Density	Young's modulus	Poisson's ratio	Beta damping	Size	Area	Inertia moment
Value	7860	$2.06e11$	0	0.0	10*10*200	10^2	$(10)^4/12$
Unit	kg/m^3	N/m	—	—	m*m*m	m^2	m^4

Table 2: Eigenvalues comparison for cantilever ($F = 0$ Newton).

Frequency order	Unscaled model 1	Unscaled model 2	Euler-Bernoulli beams Analytical	Scaled model 1		Scaled model 2	
				value	error	value	error
1st-X	$0.15 + 0.43i$	$-0.01 + 0.26i$	$0.2067i$	$0.2065i$	−0.10%	$0.2208i$	6.82%
1st-Y	$-0.47 + 0.49i$	$-0.49 + 0.33i$	$0.2067i$	$0.2065i$	−0.10%	$0.2208i$	6.82%
2nd-X	$-0.76 + 0.03i$	$-0.38 + 0.95i$	$1.2957i$	$1.2906i$	−0.40%	$1.3701i$	5.74%
2nd-Y	$-0.87 + 0.01i$	$-0.73 + 1.35i$	$1.2957i$	$1.2906i$	−0.40%	$1.3701i$	5.74%
3rd-X	$-1.45 + 1.17i$	$-0.41 + 1.49i$	$3.6280i$	$3.6049i$	−0.64%	$3.7803i$	4.20%
3rd-Y	$-1.44 + 0.38i$	$-2.68 + 0.11i$	$3.\ 6280i$	$3.6049i$	−0.64%	$3.7803i$	4.20%
Condition number	$3.9409e + 22$	$2.3198e + 19$		$9.3813e + 06$		$3.4183e + 07$	

Note: Error = (Algorithm−Analytical)/Analytical.

where x is the material coordinate, y is the lateral vibration displacement mode, E is the Young's modulus, I is the sectional moment of inertia, F is the axial tension, ρ is the linear density, and ω is the circular frequency of vibration. l is the length of the cantilever and $a = F/EI$. With the conventional mathematical procedure, characteristic equation for frequency ω is

$$\frac{\left[r_1^2\cosh(r_1 l) + r_2^2\cos(r_2 l)\right]}{\left\{r_1^3\sinh(r_1 l) - r_2^3\sin(r_2 l) + a\left[-r_1\sinh(r_1 l) - r_2\sin(r_2 l)\right]\right\}} = \frac{\left[r_1^2\sinh(r_1 l) + r_1 r_2\sin(r_2 l)\right]}{\left\{r_1^3\cosh(r_1 l) + r_1 r_2^2\cos(r_2 l) + a\left[-r_1\cosh(r_1 l) - r_1\cos(r_2 l)\right]\right\}}, \tag{4.2}$$

where $b = \rho\omega^2/EI$, $r_1 = \sqrt{(\sqrt{a^2 + 4b} + a)/2}$, and $r_2 = \sqrt{(\sqrt{a^2 + 4b} - a)/2}$.

Table 3: Eigenvalues comparison for cantilever ($F = 1e10$ Newton).

Frequency order	Unscaled model 1	Unscaled model 2	Euler-Bernoulli beam Analytical	Scaled model 1		Scaled model 2	
				value	error	value	error
1st-X	$0.12 + 0.42i$	$-0.03 + 0.28i$	$0.2796i$	$0.2792i$	−0.14%	$0.2906i$	3.93%
1st-Y	$-0.87 + 0.19i$	$-0.16 + 0.60i$	$0.2796i$	$0.2792i$	−0.14%	$0.2906i$	3.93%
2nd-X	$-1.14 + 0.05i$	$0.07 + 1.35i$	$1.3923i$	$1.3863i$	−0.43%	$1.4611i$	4.94%
2nd-Y	$1.27 + 0.22i$	$1.45 + 0.88i$	$1.3923i$	$1.3863i$	−0.43%	$1.4611i$	4.94%
3rd-X	$-0.18 + 1.38i$	$-1.06 + 1.41i$	$3.7127i$	$3.6876i$	−0.68%	$3.8602i$	3.97%
3rd-Y	$-1.26 + 0.71i$	$-0.23 + 2.73i$	$3.7127i$	$3.6876i$	−0.68%	$3.8602i$	3.97%
Condition number	$4.3103e + 21$	$7.9065e + 18$		$5.1001e + 06$		$6.4188e + 07$	

Note: error = (Algorithm−Analytical)/Analytical.

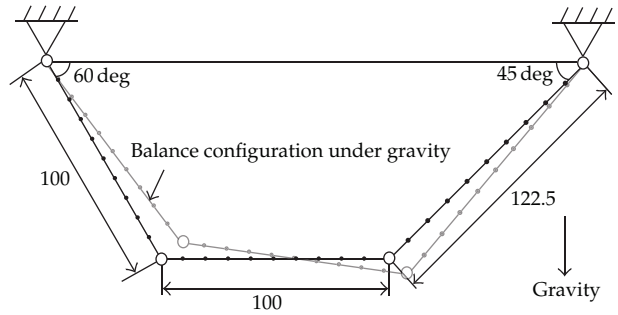


Figure 2: Four-bar linkage discrete model.

It is shown in Tables 2 and 3 that the results without the scaling precondition are totally erroneous in both cases. In the two models, through computation $c = 1.39167e + 06$ for $F = 0$ and $c = 1.39252e + 06$ for $F = 1e + 10$. Precondition parameter k is chosen the same as c and the condition numbers in the two computed cases are decreased about $O(c^2)$ times after the precondition. The results show that the presented precondition procedure could guarantee correct results for the direct eigenanalysis, and the unscaled models give out meaningless results by ARPACK as the condition numbers are too large for the nonphysical contribution of constraints. Note that model 2 is used to illustrate the precondition's effectiveness for models including other flexible elements such as 3D solid elements, and the error of scaled model 2 to Euler beam analytical results is partly because the solid model does not agree with the plan cross section assumption of Euler-Bernoulli beam.

4.2. Four-Bar Linkage

The four-bar linkage is shown in Figure 2, and the parameters of the bar are shown in Table 4. The model is modeled with Cartesian coordinates, and the constraints in the model are all revolute joints. The beam element model is used in this example, as shown in Figure 3. The equilibrium under gravity is first calculated with dynamic relaxation method, [22] and the eigenvalues are calculated about the deformed configuration. The equilibrium configuration of the four-bar system is also shown in Figure 3; the computed results with and without the preconditioning scaling are shown in Table 5 compared with results of FEA software

Table 4: Parameters for the linkage.

Item	Density	Young's modulus	Poisson's ratio	Beta damping	Area	Moment of inertia	Gravity
Value	7860	$2.06e11$	0	0	10^2	$(10)^4/12$	10
Unit	kg/m^3	N/m	—	—	m^2	m^4	m/s^2

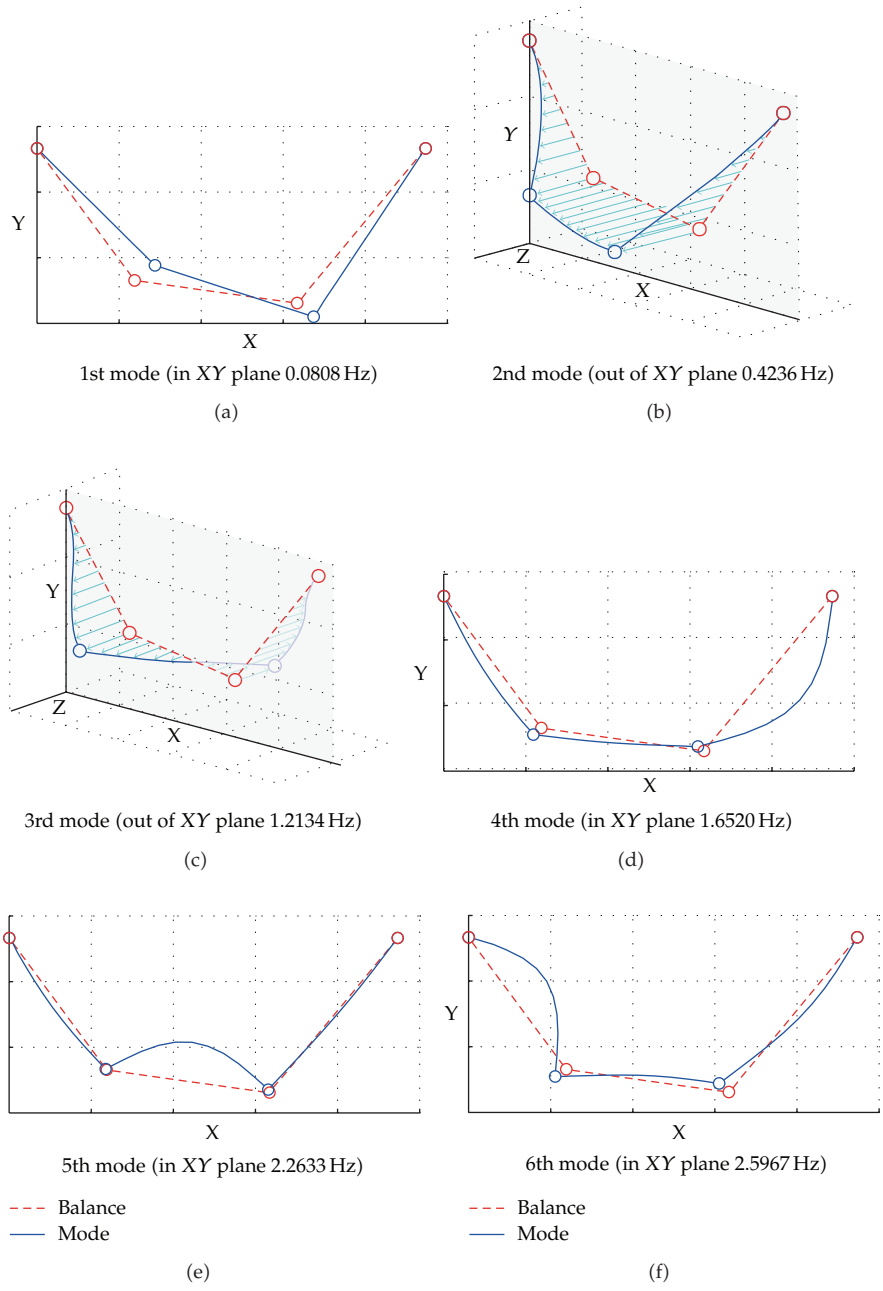


Figure 3: Modes of the four-bar linkage.

Table 5: Eigenvalues with or without scaling.

Frequency order	ABAQUS	Unscaled model		Scaled model	
		value	error	value	error
1st	$0.0805i$	$-1.09 + 0.02i$	—	$0.0808i$	0.37%
2nd	$0.4215i$	$-1.06 + 0.34i$	—	$0.4236i$	0.50%
3rd	$1.1922i$	$-1.20 + 0.35i$	—	$1.2134i$	1.78%
4th	$1.6457i$	$-1.30 + 0.08i$	—	$1.6520i$	0.38%
5th	$2.2631i$	$-1.30 + 0.03i$	—	$2.2633i$	0.01%
6th	$2.6006i$	$-1.34 + 0.23i$	—	$2.5967i$	−0.15%
Condition number		$3.3416e + 19$		$1.1560e + 10$	

Note: error = (Algorithm−ABAQUS)/ABAQUS.

ABAQUS with the same model. Again the unscaled results in the case are all meaningless as they have nonphysical real parts. In this problem $c = 9.11047e + 13$ and precondition parameter k is chosen, the same value as c. The condition numbers are reduced about $O(c)$ times after the preconditioning, and correct results are obtained with the direct eigenanalysis method.

5. Summary and Conclusions

In this paper, the bad condition of the eigenvalue problem formulated from the direct linearization of the differential-algebraic equations of multibody system is reviewed and then discussed. The root cause may be attributed to the different dimensions of constraint multipliers and equations with physical system. Through analysis, a simple scale of the constraint multiplier variables and equations according to the stiffness of the elastic elements in the system can reduce the condition number of the system matrix to be inverted. After the preconditioning scale, eigensolvers such as ARPACK can be directly applied to the preconditioned system and yield correct result. The soundness of proposed approach and applicability are illustrated by comparing the results of a cantilever and a four-bar linkage with those obtained from either analysis or ABAQUS.

Appendix

A. Gaussian elimination to get (3.4) from (3.3)

$$\begin{bmatrix} c\mathbf{D}_{11} & c\mathbf{D}_{12} & \mathbf{G}_1 \\ c\mathbf{D}_{21} & c\mathbf{D}_{22} & \mathbf{G}_2 \\ \mathbf{G}_1^T & \mathbf{G}_2^T & \mathbf{0} \end{bmatrix} \sim \begin{bmatrix} \mathbf{I}_m & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-m} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_m \end{bmatrix}.$$

1. Unitize 1st and 3rd line by left multiplying $\mathbf{G}_1^{-1}$to 1st line and $\mathbf{G}_1^{-T}$ 3rd line.
2. Exchange 1st and 3rd line.

$$\longrightarrow$$

$$\begin{bmatrix} \mathbf{I}_m & \mathbf{G}_1^{-T}\mathbf{G}_2^T & \mathbf{0} \\ c\mathbf{D}_{21} & c\mathbf{D}_{22} & \mathbf{G}_2 \\ c\mathbf{G}_1^{-1}\mathbf{D}_{11} & c\mathbf{G}_1^{-1}\mathbf{D}_{12} & \mathbf{I}_m \end{bmatrix} \sim \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{G}_1^{-T} \\ \mathbf{0} & \mathbf{I}_{n-m} & \mathbf{0} \\ \mathbf{G}_1^{-1} & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

1. Left multiply $-c\mathbf{D}_{21}$ to 1st line and add to 2nd line.
2. Left multiply $-c\mathbf{G}_1^{-1}\mathbf{D}_{11}$ to 1st line and add to 3rd line.
3. Left multiply $-\mathbf{G}_2$ to 3rd line and add to 2nd line.

$$\longrightarrow$$

$$\begin{bmatrix} \mathbf{I}_m & \mathbf{G}_1^{-T}\mathbf{G}_2^T & \mathbf{0} \\ \mathbf{0} & c\left(\mathbf{D}_{22}+\mathbf{G}_2\mathbf{G}_1^{-1}\mathbf{D}_{11}\mathbf{G}_1^{-T}\mathbf{G}_2^T-\mathbf{D}_{21}\mathbf{G}_1^{-T}\mathbf{G}_2^T-\mathbf{G}_2\mathbf{G}_1^{-1}\mathbf{D}_{12}\right) & \mathbf{0} \\ \mathbf{0} & c\left(\mathbf{G}_1^{-1}\mathbf{D}_{12}-\mathbf{G}_1^{-1}\mathbf{D}_{11}\mathbf{G}_1^{-T}\mathbf{G}_2^T\right) & \mathbf{I}_m \end{bmatrix}$$

$$\sim \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{G}_1^{-T} \\ -\mathbf{G}_2\mathbf{G}_1^{-1} & \mathbf{I}_{n-m} & -c\left(\mathbf{D}_{21}\mathbf{G}_1^{-T}-\mathbf{G}_2\mathbf{G}_1^{-1}\mathbf{D}_{11}\mathbf{G}_1^{-T}\right) \\ \mathbf{G}_1^{-1} & \mathbf{0} & -c\mathbf{G}_1^{-1}\mathbf{D}_{11}\mathbf{G}_1^{-T} \end{bmatrix}.$$

$$\xrightarrow{\text{Let } \mathbf{N}=\mathbf{G}_1^{-1}\mathbf{D}_{11}\mathbf{G}_1^{-T},\ \mathbf{P}=\mathbf{D}_{22}+\mathbf{G}_2\mathbf{N}\mathbf{G}_2^T-\mathbf{D}_{21}\mathbf{G}_1^{-T}\mathbf{G}_2^T-\mathbf{G}_2\mathbf{G}_1^{-1}\mathbf{D}_{12},\ \mathbf{Q}=\mathbf{G}_1^{-1}\mathbf{D}_{12}-\mathbf{N}\mathbf{G}_2^T,\ \mathbf{R}=\mathbf{D}_{21}\mathbf{G}_1^{-T}-\mathbf{G}_2\mathbf{N}.}$$

$$\begin{bmatrix} \mathbf{I}_m & \mathbf{G}_1^{-T}\mathbf{G}_2^T & \mathbf{0} \\ \mathbf{0} & c\mathbf{P} & \mathbf{0} \\ \mathbf{0} & c\mathbf{Q} & \mathbf{I}_m \end{bmatrix} \sim \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{G}_1^{-T} \\ \mathbf{G}_2\mathbf{G}_1^{-1} & \mathbf{I}_{n-m} & -c\mathbf{R} \\ \mathbf{G}_1^{-1} & \mathbf{0} & -c\mathbf{N} \end{bmatrix}.$$

1. Unitize 2nd line by left multiplying $c^{-1}\mathbf{P}^{-1}$ (reversibility of the whole matrix ensure the existence of $\mathbf{P}^{-1}$).
2. Left multiply $-\mathbf{G}_1^{-T}\mathbf{G}_2^T$ to 2nd line and add to 1st line.
3. Left multiply $-c\mathbf{Q}$ to 2nd line and add to 3rd line.

$$\longrightarrow$$

$$\begin{bmatrix} \mathbf{I}_m & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-m} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_m \end{bmatrix} \sim c^{-1}\begin{bmatrix} \mathbf{G}_1^{-T}\mathbf{G}_2^T\mathbf{P}^{-1}\mathbf{G}_2\mathbf{G}_1^{-1} & -\mathbf{G}_1^{-T}\mathbf{G}_2^T\mathbf{P}^{-1} & c\left(\mathbf{G}_1^{-T}+\mathbf{G}_1^{-T}\mathbf{G}_2^T\mathbf{P}^{-1}\mathbf{R}\right) \\ -\mathbf{P}^{-1}\mathbf{G}_2\mathbf{G}_1^{-1} & \mathbf{P}^{-1} & -c\mathbf{P}^{-1}\mathbf{R} \\ c\left(\mathbf{G}_1^{-1}+\mathbf{Q}\mathbf{P}^{-1}\mathbf{G}_2\mathbf{G}_1^{-1}\right) & -c\mathbf{Q}\mathbf{P}^{-1} & c^2\left(\mathbf{Q}\mathbf{P}^{-1}\mathbf{R}-\mathbf{N}\right) \end{bmatrix}. \tag{A.1}$$

Acknowledgment

Cheng Yang and Zhengru Zhang were supported by National NSF of China under Grant 11071124.

References

[1] A. A. Shabana, "Flexible multibody dynamics: review of past and recent developments," *Multibody System Dynamics*, vol. 1, no. 2, pp. 189–222, 1997.
[2] A. A. Shabana, *Dynamics of Multibody Systems*, Cambridge University Press, Cambridge, UK, 2005.
[3] W. H. A. Schilders, H. A. Van Der Vorst, and J. Rommes, *Model Order Reduction: Theory, Research Aspects and Applications*, vol. 13 of *Mathematics in Industry*, Springer-Verlag, Berlin, Germany, 2008.
[4] M. Ripepi and P. Masarati, "Reduced order models using generalized eigenanalysis," *Proceedings of the Institution of Mechanical Engineers, Part K*, vol. 225, no. 1, pp. 52–65, 2011.
[5] K. Holm-Joslashrgensen and S. R.K. Nielsen, "A component mode synthesis algorithm for multibody dynamics of wind turbines," *Journal of Sound and Vibration*, vol. 326, no. 3–5, pp. 753–767, 2009.
[6] P. Koutsovasilis and M. Beitelschmidt, "Comparison of model reduction techniques for large mechanical systems: a study on an elastic rod," *Multibody System Dynamics*, vol. 20, no. 2, pp. 111–128, 2008.

[7] Y. Zhang, B. Wen, and S. Chen, "Eigenvalue problem of constrained flexible multibody systems," *Mechanics Research Communications*, vol. 24, no. 1, pp. 11–16, 1997.
[8] M. Da Lio and R. Lot, "Analisi modale di sistemi multibody descritti in coordinate naturali," pp. 6-C9, 1999.
[9] V. Cossalter, R. Lot, and F. Maggio, "The modal analysis of a motorcycle in straight running and on a curve," *Meccanica*, vol. 39, no. 1, pp. 1–16, 2004.
[10] P. Masarati, "Direct eigenanalysis of constrained system dynamics," *Proceedings of the Institution of Mechanical Engineers, Part K*, vol. 223, no. 4, pp. 335–342, 2010.
[11] V. Boschi, R. De Salvo, P. Masarati, G. Quaranta, and V. Sannibale, "Seismic attenuation system synthesis by reduced order models from multibody analysis," in *Proceedings of the Multibody Dynamics*, Milano, Italy, June 2007.
[12] J. L. Escalona and R. Chamorro, "Stability analysis of vehicles on circular motions using multibody dynamics," *Nonlinear Dynamics*, vol. 53, no. 3, pp. 237–250, 2008.
[13] M. Géradin and A. Cardona, *Flexible Multibody Dynamics: A Finite Element Approach*, Wiley & Sons, New York, NY, USA, 2001.
[14] O. A. Bauchau, A. Epple, and C. L. Bottasso, "Scaling of constraints and augmented lagrangian formulations in multibody dynamics simulations," *Journal of Computational and Nonlinear Dynamics*, vol. 4, no. 2, pp. 1–9, 2009.
[15] C. L. Bottasso, D. Dopico, and L. Trainelli, "On the optimal scaling of index three DAEs in multibody dynamics," *Multibody System Dynamics*, vol. 19, no. 1-2, pp. 3–20, 2008.
[16] C. Lanczos, *The Variational Principles of Mechanics*, Dover Publications, New York, NY, USA, 1970.
[17] R. B. Lehoucq and D. C. Sorensen, "Deflation techniques for an implicitly restarted Arnoldi iteration," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 4, pp. 789–821, 1996.
[18] H. Saberi Najafi and E. Khaleghi, "A new restarting method in the Arnoldi algorithm for computing the eigenvalues of a nonsymmetric matrix," *Applied Mathematics and Computation*, vol. 156, no. 1, pp. 59–71, 2004.
[19] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK, 1992.
[20] O. Schenk and K. Gartner, "Solving unsymmetric sparse systems of linear equations with PARDISO," in *Computational Science-ICCS 2002, Part II*, P. Sloot, C. J. K. Tan, J. J. Dongarra, and A. G. Hoekstra, Eds., vol. 2330, pp. 355–363, Springer, Berlin, Germany, 2002.
[21] O. Schenk and K. Gartner, "On fast factorization pivoting methods for sparse symmetric indefinite systems," *Electronic Transactions on Numerical Analysis*, vol. 23, pp. 158–179, 2006.
[22] W. J. Lewis, *Tension Structures: Form and Behaviour*, Thomas Telford Services, 2003.

Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 245051, 16 pages
doi:10.1155/2012/245051

Research Article

Finite Element Preconditioning on Spectral Element Discretizations for Coupled Elliptic Equations

JongKyum Kwon, Soorok Ryu, Philsu Kim, and Sang Dong Kim

Department of Mathematics, Kyungpook National University, Daegu 702-701, Republic of Korea

Correspondence should be addressed to Sang Dong Kim, skim@knu.ac.kr

Received 8 August 2011; Accepted 2 November 2011

Academic Editor: Massimiliano Ferronato

The uniform bounds on eigenvalues of $\widehat{\mathsf{B}}_{h^2}^{-1}\widehat{\mathsf{A}}_{N^2}$ are shown both analytically and numerically by the p_1 finite element preconditioner $\widehat{\mathsf{B}}_{h^2}^{-1}$ for the Legendre spectral element system $\widehat{\mathsf{A}}_{N^2}\underline{\mathbf{u}} = \underline{\mathbf{f}}$ which is arisen from a coupled elliptic system occurred by an optimal control problem. The finite element preconditioner is corresponding to a leading part of the coupled elliptic system.

1. Introduction

Optimal control problems constrained by partial differential equations can be reduced to a system of coupled partial differential equations by Lagrange multiplier method ([1]). In particular, the needs for accurate and efficient numerical methods for these problems have been important subjects. Many works are reported for solving coupled partial differential equations by finite element/difference methods; or finite element least-squares methods ([2–5], etc.). But, there are a few literature (for examples, [6, 7]) on coupled partial differential equations using the spectral element methods (SEM) despite of its popularity and accuracy (see, e.g., [8]).

One of the goals in this paper is to investigate a finite element preconditioner for the SEM discretizations. The induced nonsymmetric linear systems by the SEM discretizations from such coupled elliptic partial differential equations have the condition numbers which are getting larger incredibly not only as the number of elements and degrees of polynomials increases but also as the penalty parameter δ decreases (see [5] and Section 4). Hence, an efficient preconditioner is necessary to improve the convergence of a numerical method whose number of iterations depends on the distributions of eigenvalues (see [9–12]).

Particularly, the lower-order finite element/difference preconditioning methods for spectral collocation/element methods have been reported ([9, 10, 13–17], etc.).

The target-coupled elliptic type equations are as follows:

$$\begin{aligned} -\Delta u + \mathbf{b}\cdot\nabla u + u + \frac{1}{\delta}v &= 0 \quad \text{in } \Omega, \\ -\Delta v - \nabla\cdot(v\mathbf{b}) + v - u &= -\widehat{u} \quad \text{in } \Omega, \\ u = v &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{1.1}$$

which is the result of Lagrange multiplier rule applied to a L^2 optimal control problem subject to an elliptic equation (see [1]). Applying the $\mathcal{P}_1$ finite element preconditioner to our coupled elliptic system discretized by SEM using LGL (Legendre-Gauss-Lobatto) nodes, we show that the preconditioned linear systems have uniformly bounded eigenvalues with respect to elements and degrees.

The field of values arguments will be used instead of analyzing eigenvalues directly because the matrix representation of the target operator, even with zero convection term, is not symmetric. We will show that the real parts of eigenvalues are positive, uniformly bounded away from zero, and the absolute values of eigenvalues are uniformly bounded whose bounds are only dependent on the penalty parameter δ in (1.1) and the constant vector $\mathbf{b}$ in (1.1). Because of this result, one may apply a lower-order finite element preconditioner to a real optimal control problem subject to Stokes equations which requires an elliptic type solver.

This paper is organized as follows: in Section 2, we introduce some preliminaries and notations. The norm equivalences of interpolation operators are reviewed to show the norm equivalence of an interpolation operator using vector basis. The preconditioning results are presented theoretically and numerically in Section 3 and Section 4, respectively. Finally, we add the concluding remarks in Section 5.

2. Preliminaries

2.1. Coupled Elliptic System

Because we are going to deal with a coupled elliptic system, the vector Laplacian, gradient and divergence operators for a vector function $\underline{\mathbf{u}} = [u, v]^T$, where T denotes the transpose, are defined by

$$\Delta\underline{\mathbf{u}} := \begin{bmatrix} u_{xx} + u_{yy} \\ v_{xx} + v_{yy} \end{bmatrix}, \qquad \nabla\underline{\mathbf{u}} := \begin{bmatrix} u_x & v_x \\ u_y & v_y \end{bmatrix}, \qquad \nabla\cdot\underline{\mathbf{u}} := \begin{bmatrix} u_x + u_y \\ v_x + v_y \end{bmatrix}. \tag{2.1}$$

With the usual L^2 inner product $\langle\cdot,\cdot\rangle$ and its norm $\|\cdot\|$, for vector functions $\underline{\mathbf{u}} := [u, v]^T$ and $\underline{\mathbf{w}} := [w, z]^T$, define

$$\langle\underline{\mathbf{u}}, \underline{\mathbf{w}}\rangle := \langle u, w\rangle + \langle v, z\rangle, \qquad \left\|\underline{\mathbf{u}}\right\|^2 := \|u\|^2 + \|v\|^2 \tag{2.2}$$

and, for matrix functions U and V, define

$$\langle U, V\rangle = \sum_{k=1}^{4}\langle u_k, v_k\rangle, \quad \|U\|^2 := \sum_{k=1}^{4}\|u_k\|^2, \quad \text{where } U = \begin{bmatrix} u_1 & u_2 \\ u_3 & u_4 \end{bmatrix},\ V = \begin{bmatrix} v_1 & v_2 \\ v_3 & v_4 \end{bmatrix}. \tag{2.3}$$

We use the standard Sobolev spaces like $H^1(\Omega)$ and $H_0^1(\Omega)$ on a given domain Ω with the usual Sobolev seminorm $|\cdot|_1$ and norm $\|\cdot\|_1$. The main content of this paper is to provide an efficient low-order preconditioner for the system (1.1).

Multiplying the second equation by $1/\delta$, the system (1.1) can be expressed as

$$\mathcal{A}\underline{\mathbf{u}} := -\mathsf{A}\Delta\underline{\mathbf{u}} + \mathsf{B}\big(\mathbf{b}\cdot\nabla\underline{\mathbf{u}}\big)^T + (\mathsf{A}+\mathsf{C})\underline{\mathbf{u}} = \underline{\mathbf{f}} \quad \text{in } \Omega, \tag{2.4}$$

where

$$\mathsf{A} = \begin{bmatrix} 1 & 0 \\ 0 & \dfrac{1}{\delta} \end{bmatrix}, \qquad \mathsf{B} = \begin{bmatrix} 1 & 0 \\ 0 & -\dfrac{1}{\delta} \end{bmatrix}, \qquad \mathsf{C} = \begin{bmatrix} 0 & \dfrac{1}{\delta} \\ -\dfrac{1}{\delta} & 0 \end{bmatrix}, \qquad \underline{\mathbf{f}} = \begin{bmatrix} 0 \\ -\dfrac{1}{\delta}\widehat{u} \end{bmatrix}, \tag{2.5}$$

with the zero boundary condition $\underline{\mathbf{u}} = \underline{\mathbf{0}}$ on $\partial\Omega$. Let $\mathcal{B}$ be another decoupled uniformly elliptic operator such that

$$\mathcal{B}\underline{\mathbf{u}} := -\mathsf{A}\Delta\underline{\mathbf{u}} + \mathsf{A}\underline{\mathbf{u}} \quad \text{in } \Omega \tag{2.6}$$

with the zero boundary condition.

2.2. LGL Nodes, Weights, and Function Spaces

Let $\{\eta_k\}_{k=0}^N$ and $\{\omega_k\}_{k=0}^N$ be the reference LGL nodes and its corresponding LGL weights in $I = [-1,1]$, respectively, arranged by $-1 =: \eta_0 < \eta_1 < \cdots < \eta_{N-1} < \eta_N := 1$. We use $\{t_j\}_{j=0}^E$ as the set of knots in the interval I such that $-1 =: t_0 < t_1 < \cdots < t_{E-1} < t_E := 1$. Here E denotes the number of subintervals of I. Denote N by the degree of a polynomial on each subinterval $I_j := [t_{j-1}, t_j]$ and $\mathcal{G} := \{\xi_{j,k}\}_{j=1,k=0}^{E,N}$ by the set of kth-LGL nodes $\xi_{j,k}$ in each subinterval I_j $(j = 1,\ldots,E)$ arranged by

$$\xi_{j,0} := t_{j-1} < \xi_{j,1} < \cdots < \xi_{j,N-1} < t_j =: \xi_{j,N}, \tag{2.7}$$

where

$$\xi_{j,k} = \frac{h_j}{2}\eta_k + \frac{1}{2}\big(t_{j-1} + t_j\big), \quad h_j = t_j - t_{j-1}, \tag{2.8}$$

and the corresponding LGL weights $\{\rho_{j,k}\}_{j=1,k=0}^{E,N}$ are given by

$$\rho_{j,k} := \begin{cases} \dfrac{h_j}{2}\omega_N + \dfrac{h_{j+1}}{2}\omega_0, & k = N,\ j = 1,\ldots,E-1, \\ \rho_{j-1,N,} & k = 0,\ j = 2,\ldots,E, \\ \dfrac{h_j}{2}\omega_k, & \text{otherwise.} \end{cases} \tag{2.9}$$

Let $\mathcal{P}_k$ be the space of all polynomials defined on I whose degrees are less than or equal to k. The Lagrange basis for $\mathcal{P}_N$ on I is given by $\{\widehat{\phi}_i(t)\}_{i=0}^{N}$ satisfying

$$\widehat{\phi}_i(\eta_j) = \delta_{ij} \quad \text{for } i,j = 0,1,\ldots,N, \tag{2.10}$$

where δ_{ij} denotes the Kronecker delta function.

We define $\mathcal{P}_N^h$ as the subspace of continuous functions whose basis $\{\phi_\mu\}_{\mu=0}^{\mathfrak{N}+1}$ is piecewise continuous Lagrange polynomials of degree N on I_j with respect to $\mathcal{G}$ and define $\mathcal{V}_N^h$ as the space of all piecewise Lagrange linear functions $\{\psi_\mu\}_{\mu=0}^{\mathfrak{N}+1}$ with respect to $\mathcal{G}$. Note that these basis functions have a proper support. See [9, 18] for detail. Let $\mathcal{P}_{N,h}^0 := H_0^1(I) \cap \mathcal{P}_N^h$ and $\mathcal{V}_{N,h}^0 := H_0^1(I) \cap \mathcal{V}_N^h$. With the notation $\mathcal{V} \times \mathcal{V} := \{[u,v]^T \mid u,v \in \mathcal{V}\}$, define $\underline{\mathbf{P}}_{N,h}^0$ and $\underline{\mathbf{V}}_{N,h}^0$ as subspaces of $\mathcal{P}_{N,h}^0 \times \mathcal{P}_{N,h}^0$ and $\mathcal{V}_{N,h}^0 \times \mathcal{V}_{N,h}^0$, respectively. The basis functions for $\underline{\mathbf{P}}_{N,h}^0$ and $\underline{\mathbf{V}}_{N,h}^0$ are given respectively by

$$\underline{\phi}_p(t) := \begin{cases} [\phi_p(t),0]^T, & \text{for } p \le \mathfrak{N}, \\ [0,\phi_{p-\mathfrak{N}}(t)]^T, & \text{for } p > N, \end{cases} \qquad \underline{\psi}_p(t) := \begin{cases} [\psi_p(t),0]^T, & \text{for } p \le \mathfrak{N}, \\ [0,\psi_{P-\mathfrak{N}}(t)]^T, & \text{for } p > N, \end{cases} \tag{2.11}$$

where $p = 1,2,\ldots,2\mathfrak{N}$. For two dimensional (2D) case, let $[[\mathcal{P}_{N,h}^0]] := \mathcal{P}_{N,h}^0 \otimes \mathcal{P}_{N,h}^0$ and $[[\mathcal{V}_{N,h}^0]] := \mathcal{V}_{N,h}^0 \otimes \mathcal{V}_{N,h}^0$ be tensor product function spaces of one-dimensional function spaces and let $[[\underline{\mathbf{P}}_{N,h}^0]]$ and $[[\underline{\mathbf{V}}_{N,h}^0]]$ be subspaces of $[[\mathcal{P}_{N,h}^0]] \times [[\mathcal{P}_{N,h}^0]]$ and $[[\mathcal{V}_{N,h}^0]] \times [[\mathcal{V}_{N,h}^0]]$, respectively. Now let us order the interior LGL points in Ω by horizontal lines as $\{\Xi_{\tilde{\mu}}\}_{\tilde{\mu}=1}^{\mathfrak{N}^2} := \{(\xi_\mu,\xi_\nu)\}_{\mu=1,\nu=1}^{\mathfrak{N},\mathfrak{N}}$, where $\tilde{\mu} = \mu + \mathfrak{N}(\nu-1)$ for $\mu,\nu = 1,2,\ldots,\mathfrak{N}$. Accordingly, the basis functions for $[[\underline{\mathbf{P}}_{N,h}^0]]$ and $[[\underline{\mathbf{V}}_{N,h}^0]]$ are also arranged as the same way. Then, with the notations $\phi_{\tilde{\mu}}(x,y) := \phi_\mu(x)\phi_\nu(y)$ and $\psi_{\tilde{\mu}}(x,y) := \psi_\mu(x)\psi_\nu(y)$, the basis functions of $[[\underline{\mathbf{P}}_{N,h}^0]]$ and $[[\underline{\mathbf{V}}_{N,h}^0]]$ are given, respectively, by

$$\begin{aligned} \underline{\Phi}_{\tilde{p}}(x,y) &:= \begin{cases} [\phi_{\tilde{p}}(x,y),0]^T, & \text{for } \tilde{p} \le \mathfrak{N}^2, \\ [0,\phi_{\tilde{p}-\mathfrak{N}^2}(x,y)]^{\mathrm{T}}, & \text{for } \tilde{p} > \mathfrak{N}^2, \end{cases} \\ \underline{\Psi}_{\tilde{p}}(x,y) &:= \begin{cases} [\psi_{\tilde{p}}(x,y),0]^T, & \text{for } \tilde{p} \le \mathfrak{N}^2, \\ [0,\psi_{\tilde{p}-\mathfrak{N}^2}(x,y)]^T, & \text{for } \tilde{p} > \mathfrak{N}^2, \end{cases} \end{aligned} \tag{2.12}$$

where $\tilde{p} = 1,\ldots,2\mathfrak{N}^2$.

2.3. Interpolation Operators

We denote $C(\Omega)$ as the set of continuous functions in $\Omega := I \times I$. Let $\mathcal{I}_{N^2} : C(\Omega) \to [[\mathcal{P}_N]] := \mathcal{P}_N \otimes \mathcal{P}_N$ be the usual reference interpolation operator such that $(\mathcal{I}_{N^2}u)(\eta_i, \eta_j) = u(\eta_i, \eta_j)$ for $u \in C(\Omega)$ (see, e.g., [17]). The global interpolation operator $\mathcal{I}^h_{N^2} : C(\Omega) \to [[\mathcal{P}^0_{N,h}]]$ is given by $\mathcal{I}^h_{N^2}u(x,y) = \sum_{\tilde{\mu}=1}^{\mathfrak{N}^2} u_{\tilde{\mu}}\phi_{\tilde{\mu}}(x,y)$, where $u_{\tilde{\mu}} = u(\xi_\mu, \xi_\nu)$. Hence, it follows that $(\mathcal{I}^h_{N^2}u)(\xi_\mu, \xi_\nu) = u(\xi_\mu, \xi_\nu)$ for $u \in C(\Omega)$. With this interpolation operator $\mathcal{I}^h_{N^2}$, let us define the vector interpolation operator $\underline{\mathbf{I}}^h_{N^2} : C(\Omega) \times C(\Omega) \to [[\underline{\mathbf{P}}^0_{N,h}]]$ such that, for $\underline{\mathbf{u}} := [u, v]^T \in C(\Omega) \times C(\Omega)$,

$$\left(\underline{\mathbf{I}}^h_{N^2}\underline{\mathbf{u}}\right)(\xi_\mu, \xi_\nu) := \left[\mathcal{I}^h_{N^2}u(\xi_\mu, \xi_\nu), \mathcal{I}^h_{N^2}v(\xi_\mu, \xi_\nu)\right]^T = \underline{\mathbf{u}}(\xi_\mu, \xi_\nu). \tag{2.13}$$

Let $\widehat{\Phi}_{\tilde{i}}(x,y) = \widehat{\phi}_{ij}(x,y) = \widehat{\phi}_i(x)\widehat{\phi}_j(y)$ and $\widehat{\Psi}_{\tilde{i}}(x,y) = \widehat{\varphi}_{ij}(x,y) = \widehat{\varphi}_i(x)\widehat{\varphi}_j(y)$, where $\tilde{i} = i + (N+1)j$ and $i, j = 0, \ldots, N$, be the basis of $[[\mathcal{P}_N]]$ and $[[\mathcal{U}_N]]$, respectively. Let us denote M_{N^2} and M_{h^2} by the mass matrices such that

$$\mathsf{M}_{N^2}\left(\tilde{i}, \tilde{j}\right) = \left\langle \widehat{\Phi}_{\tilde{i}}, \widehat{\Phi}_{\tilde{j}} \right\rangle, \qquad \mathsf{M}_{h^2}\left(\tilde{i}, \tilde{j}\right) = \left\langle \widehat{\Psi}_{\tilde{i}}, \widehat{\Psi}_{\tilde{j}} \right\rangle \tag{2.14}$$

and denote S_{N^2} and S_{h^2} by the stiffness matrices such that

$$\mathsf{S}_{N^2}\left(\tilde{i}, \tilde{j}\right) = \left\langle \nabla\widehat{\Phi}_{\tilde{i}}, \nabla\widehat{\Phi}_{\tilde{j}} \right\rangle, \qquad \mathsf{S}_{h^2}\left(\tilde{i}, \tilde{j}\right) = \left\langle \nabla\widehat{\Psi}_{\tilde{i}}, \nabla\widehat{\Psi}_{\tilde{j}} \right\rangle, \tag{2.15}$$

where $\tilde{i}, \tilde{j} = 1, \ldots, (N+1)^2$.

According to Theorems 5.4 and 5.5 in [17], there are two absolute positive constants c_0 and c_1 such that for any $U = [u_1, \ldots, u_{(N+1)^2}]^T$,

$$\begin{aligned} c_0\langle \mathsf{M}_{N^2}U, U\rangle \le \langle \mathsf{M}_{h^2}U, U\rangle \le c_1\langle \mathsf{M}_{N^2}U, U\rangle, \\ c_0\langle \mathsf{S}_{N^2}U, U\rangle \le \langle \mathsf{S}_{h^2}U, U\rangle \le c_1\langle \mathsf{S}_{N^2}U, U\rangle, \end{aligned} \tag{2.16}$$

and for all $u \in [[\mathcal{U}_N]]$,

$$c_0\|u\| \le \|\mathcal{I}_{N^2}u\| \le c_1\|u\|, \qquad c_0\|u\|_1 \le \|\mathcal{I}_{N^2}u\|_1 \le c_1\|u\|_1 . \tag{2.17}$$

The extension of (2.17) to the interpolation operator $\mathcal{I}^h_{N^2}$ leads to

$$c_0\|u\| \le \left\|\mathcal{I}^h_{N^2}u\right\| \le c_1\|u\|, \qquad c_0\|u\|_1 \le \left\|\mathcal{I}^h_{N^2}u\right\|_1 \le c_1\|u\|_1 \tag{2.18}$$

for all $u \in [[\mathcal{U}^0_{N,h}]]$, where the constants c_0 and c_1 are positive constants independent of E and N (see [18]).

Theorem 2.1. *For all* $\underline{\mathbf{u}} \in [[\underline{\mathbf{V}}^0_{N,h}]]$, *there are positive constants* c_0 *and* c_1 *independent of* E *and* N *such that*

$$c_0\|\underline{\mathbf{u}}\| \le \left\|\underline{\mathbf{I}}^h_{N^2}\underline{\mathbf{u}}\right\| \le c_1\|\underline{\mathbf{u}}\|, \qquad c_0\|\underline{\mathbf{u}}\|_1 \le \left\|\underline{\mathbf{I}}^h_{N^2}\underline{\mathbf{u}}\right\|_1 \le c_1\|\underline{\mathbf{u}}\|_1. \tag{2.19}$$

Proof. By the definitions of the interpolation operator $\underline{\mathbf{I}}^h_{N^2}$ and the norms, we have

$$\left\|\underline{\mathbf{I}}^h_{N^2}\underline{\mathbf{u}}\right\|^2 = \left\|\mathcal{I}^h_{N^2}u\right\|^2 + \left\|\mathcal{I}^h_{N^2}v\right\|^2, \qquad \left\|\underline{\mathbf{I}}^h_{N^2}\underline{\mathbf{u}}\right\|^2_1 = \left\|\mathcal{I}^h_{N^2}u\right\|^2_1 + \left\|\mathcal{I}^h_{N^2}v\right\|^2_1, \tag{2.20}$$

which completes the proof because of (2.18). □

3. Analysis on $\mathcal{P}_1$ Finite Element Preconditioner

The bilinear forms corresponding to (2.4) and (2.6) are given by

$$\alpha(\underline{\mathbf{u}},\underline{\mathbf{w}}) = \left\langle \mathsf{A}\nabla\underline{\mathbf{u}}^T, \nabla\underline{\mathbf{w}}^T \right\rangle + \left\langle \mathsf{B}(\mathbf{b}\cdot\nabla\underline{\mathbf{u}})^T, \underline{\mathbf{w}} \right\rangle + \langle (\mathsf{A}+C)\underline{\mathbf{u}}, \underline{\mathbf{w}}\rangle = \langle \underline{\mathbf{f}}, \underline{\mathbf{w}}\rangle, \tag{3.1}$$

$$\beta(\underline{\mathbf{u}},\underline{\mathbf{w}}) = \left\langle \mathsf{A}\nabla\underline{\mathbf{u}}^T, \nabla\underline{\mathbf{w}}^T \right\rangle + \langle \mathsf{A}\underline{\mathbf{u}}, \underline{\mathbf{w}}\rangle, \tag{3.2}$$

where $\underline{\mathbf{u}} := [u(x,y), v(x,y)]^T$, $\underline{\mathbf{w}} := [w(x,y), z(x,y)]^T$ and $\mathsf{A}, \mathsf{B}, \mathsf{C}$ are the same matrices in (2.4) and $\underline{\mathbf{f}} = [0, \widehat{u}(x,y)]^T$. Note that the bilinear form $\beta(\cdot,\cdot)$ in (3.2) is symmetric but the bilinear form $\alpha(\cdot,\cdot)$ in (3.1) is not symmetric. The following norm equivalence guarantees the existence and uniqueness of the solution in $H^1_0(\Omega)\times H^1_0(\Omega)$ for the variational problem (3.1).

Proposition 3.1. *For a real valued vector function* $\underline{\mathbf{u}}(x,y) = [u,v]^T \in H^1_0(\Omega)\times H^1_0(\Omega)$, *we have*

$$\|\underline{\mathbf{u}}\|_1 \le \alpha(\underline{\mathbf{u}},\underline{\mathbf{u}}) \le \frac{1}{\delta}\|\underline{\mathbf{u}}\|^2_1. \tag{3.3}$$

Proof. Since $\mathbf{b}$ is a constant vector, $[u,v]^T \in H^1_0(\Omega)\times H^1_0(\Omega)$, and $\underline{\mathbf{u}}$ is a real valued vector function, we have

$$\begin{aligned}
\alpha(\underline{\mathbf{u}},\underline{\mathbf{u}}) &= \beta(\underline{\mathbf{u}},\underline{\mathbf{u}}) + \left\langle \mathsf{B}(\mathbf{b}\cdot\nabla\underline{\mathbf{u}})^T, \underline{\mathbf{u}} \right\rangle + \langle \mathsf{C}\underline{\mathbf{u}}, \underline{\mathbf{u}}\rangle \\
&= \beta(\underline{\mathbf{u}},\underline{\mathbf{u}}) + \langle \mathbf{b}\cdot\nabla u, u\rangle - \frac{1}{\delta}\langle \mathbf{b}\cdot\nabla v, v\rangle + \frac{1}{\delta}(\langle v,u\rangle - \langle u,v\rangle) \\
&= \beta(\underline{\mathbf{u}},\underline{\mathbf{u}}) = \|u\|^2_1 + \frac{1}{\delta}\|v\|^2_1.
\end{aligned} \tag{3.4}$$

Hence, (3.3) is proved because $0 < \delta \le 1$. □

Lemma 3.2. *If* $\underline{\mathbf{u}} = [u, v]^T$ *is a complex valued function in* $H_0^1(\Omega) \times H_0^1(\Omega)$*, then we have the following estimates:*

$$\begin{aligned} \operatorname{Re}\big(\alpha(\underline{\mathbf{u}}, \underline{\mathbf{u}})\big) &= \beta(\underline{\mathbf{u}}, \underline{\mathbf{u}}), \\ \big|\alpha(\underline{\mathbf{u}}, \underline{\mathbf{u}})\big| &\le \left(1 + \frac{1}{2}|\mathbf{b}| + \frac{1}{\delta}\right)\beta(\underline{\mathbf{u}}, \underline{\mathbf{u}}), \end{aligned} \tag{3.5}$$

where $|\mathbf{b}| = \sqrt{b_1^2 + b_2^2}$.

Proof. Let us decompose u and v in $\underline{\mathbf{u}}$ as $u(x, y) = p(x, y) + iq(x, y)$ and $v(x, y) = r(x, y) + is(x, y)$, where p, q, r, and s are real functions and $i^2 = -1$. Then we have

$$\langle \mathbf{b} \cdot \nabla u, u \rangle = i \int_\Omega b_1 (q_x p - p_x q) + b_2 (p_y q - q_y p) d\Omega, \tag{3.6}$$

$$-\frac{1}{\delta}\langle \mathbf{b} \cdot \nabla u, u \rangle = -\frac{i}{\delta} \int_\Omega b_1 (s_x r - r_x s) + b_2 (r_y s - s_y r) d\Omega, \tag{3.7}$$

$$\frac{1}{\delta}(\langle v, u \rangle - \langle u, v \rangle) = i\left(\frac{2}{\delta} \int_\Omega sp - qr \, d\Omega\right). \tag{3.8}$$

Hence, one may see that the real part of $\alpha(\underline{\mathbf{u}}, \underline{\mathbf{u}})$ is $\beta(\underline{\mathbf{u}}, \underline{\mathbf{u}})$ and the pure imaginary part is the sum of (3.6), (3.7), and (3.8). By Cauchy-Schwarz inequality, ϵ-inequality, and the range of $0 < \delta \le 1$, we have

$$\begin{aligned} &\left|\langle \mathbf{b} \cdot \nabla u, u \rangle - \frac{1}{\delta}\langle \mathbf{b} \cdot \nabla v, v \rangle + \frac{1}{\delta}(\langle v, u \rangle - \langle u, v \rangle)\right| \\ &\quad \le |\mathbf{b}| \|\nabla u\| \|u\| + \frac{1}{\delta}|\mathbf{b}| \|\nabla v\| \|v\| + \frac{1}{\delta}(2\|v\| \|u\|) \\ &\quad \le \frac{1}{2}|\mathbf{b}|\left(\|\nabla u\|^2 + \|u\|^2 + \frac{1}{\delta}\left(\|\nabla v\|^2 + \|v\|^2\right)\right) + \frac{1}{\delta}\left(\|u\|^2 + \|v\|^2\right) \\ &\quad \le \left(\frac{1}{2}|\mathbf{b}| + \frac{1}{\delta}\right)\beta(\underline{\mathbf{u}}, \underline{\mathbf{u}}). \end{aligned} \tag{3.9}$$

Hence (3.5) is proved. □

Let $\sigma(\mathsf{A})$ and $\mathcal{F}(\mathsf{A})$ be the spectrum (or set of eigenvalues) and field of values of the square matrix A, respectively. Let $\widehat{\mathsf{A}}_{N^2}$ and $\widehat{\mathsf{B}}_{h^2}$ be the two dimensional stiffness matrices on the spaces $[[\underline{\mathbf{P}}_{N,h}^0]]$ and $[[\underline{\mathbf{V}}_{N,h}^0]]$ induced by (3.1) and (3.2), respectively. Then, we have the following.

Lemma 3.3. *For $U(\neq \mathbf{0}) \in \mathbb{C}^{2\mathfrak{N}^2}$, one has*

$$\sigma\left(\widehat{\mathsf{B}}_{h^2}^{-1}\widehat{\mathsf{A}}_{N^2}\right) \subset \mathcal{W}, \tag{3.10}$$

where

$$\mathcal{W} := \left\{ \frac{\left\langle \widehat{\mathsf{A}}_{N^2}U, U\right\rangle}{\left\langle \widehat{\mathsf{B}}_{h^2}U, U\right\rangle} \mid U \neq \mathbf{0} \right\}. \tag{3.11}$$

Proof. Since $\widehat{\mathsf{B}}_{h^2}$ is symmetric positive definite, there exists a unique positive definite square root $\widehat{\mathsf{B}}_{h^2}^{1/2}$ of $\widehat{\mathsf{B}}_{h^2}$. So, we have

$$\frac{\left\langle \widehat{\mathsf{A}}_{N^2}U, U\right\rangle}{\left\langle \widehat{\mathsf{B}}_{h^2}U, U\right\rangle} = \frac{\left\langle \widehat{\mathsf{A}}_{N^2}U, U\right\rangle}{\left\langle \widehat{\mathsf{B}}_{h^2}^{1/2}U, \widehat{\mathsf{B}}_{h^2}^{1/2}U\right\rangle} = \frac{\left\langle \widehat{\mathsf{B}}_{h^2}^{-1/2}\widehat{\mathsf{A}}_{N^2}\widehat{\mathsf{B}}_{h^2}^{-1/2}V, V\right\rangle}{\langle V, V\rangle}, \quad \text{where } V = \widehat{\mathsf{B}}_{h^2}^{1/2}U, \tag{3.12}$$

for $U(\neq \mathbf{0}) \in \mathbb{C}^{2\mathfrak{N}^2}$. Now let $\widehat{\mathsf{B}}_{h^2}^{-1/2}$ be a short-hand notation for $\left(\widehat{\mathsf{B}}_{h^2}^{1/2}\right)^{-1}$. Therefore, from the relation of spectrum and field of values (see [19] or [11]), it follows that

$$\sigma\left(\widehat{\mathsf{B}}_{h^2}^{-1}\widehat{\mathsf{A}}_{N^2}\right) = \sigma\left(\widehat{\mathsf{B}}_{h^2}^{-1/2}\widehat{\mathsf{A}}_{N^2}\widehat{\mathsf{B}}_{h^2}^{-1/2}\right) \subset \mathcal{F}\left(\widehat{\mathsf{B}}_{h^2}^{-1/2}\widehat{\mathsf{A}}_{N^2}\widehat{\mathsf{B}}_{h^2}^{-1/2}\right) = \mathcal{W}, \tag{3.13}$$

which completes the proof. □

The following theorem shows the uniform bounds of eigenvalues which is independent of both N and E for our preconditioned system

$$\widehat{\mathsf{B}}_{h^2}^{-1}\widehat{\mathsf{A}}_{N^2}\underline{\mathbf{u}}^N = \widehat{\mathsf{B}}_{h^2}^{-1}\underline{\mathbf{f}}^N. \tag{3.14}$$

Theorem 3.4. *Let $\{\lambda_p\}_{p=1}^{2\mathfrak{N}^2}$ be the set of eigenvalues of*

$$\widehat{\mathsf{B}}_{h^2}^{-1}\widehat{\mathsf{A}}_{N^2}, \tag{3.15}$$

then, there are constants c_0, C_0, and Λ_δ independent of E and N, such that

$$0 < c_0 < \mathrm{Re}\,(\lambda_p) < C_0, \quad |\lambda_p| \le \Lambda_\delta, \tag{3.16}$$

where $\Lambda_\delta = C(1 + \|\mathbf{b}\| + 1/\delta)$.

Proof. Let $\underline{\mathbf{u}}(x,y) \in [[\underline{\mathbf{V}}^0_{N,h}]]$ be represented as $\underline{\mathbf{u}}(x,y) = \sum_{p=1}^{2\mathfrak{N}^2} u_p \underline{\Psi}_p(x,y)$. Then, its piecewise polynomial interpolation can be written as

$$\left(\underline{\mathbf{I}}^h_{N^2}\underline{\mathbf{u}}\right)(x,y) = \sum_{p=1}^{2\mathfrak{N}^2} u_p \underline{\Phi}_p(x,y). \tag{3.17}$$

Let $U = [u_1, \ldots, u_{2\mathfrak{N}^2}]^T$. From the definitions of the bilinear forms, we have

$$\left\langle \widehat{\mathsf{A}}_{N^2} U, U \right\rangle = \alpha\left(\underline{\mathbf{I}}^h_{N^2}\underline{\mathbf{u}}, \underline{\mathbf{I}}^h_{N^2}\underline{\mathbf{u}}\right), \qquad \left\langle \widehat{\mathsf{B}}_{h^2} U, U \right\rangle = \beta(\underline{\mathbf{u}}, \underline{\mathbf{u}}). \tag{3.18}$$

This implies that

$$w_\lambda := \frac{\left\langle \widehat{\mathsf{A}}_{N^2} U, U \right\rangle}{\left\langle \widehat{\mathsf{B}}_{h^2} U, U \right\rangle} = \frac{\alpha\left(\underline{\mathbf{I}}^h_{N^2}\underline{\mathbf{u}}, \underline{\mathbf{I}}^h_{N^2}\underline{\mathbf{u}}\right)}{\beta(\underline{\mathbf{u}}, \underline{\mathbf{u}})}. \tag{3.19}$$

By Theorem 2.1 and Lemma 3.2, the real part of w_λ satisfies

$$\mathrm{Re}(w_\lambda) = \frac{\beta\left(\underline{\mathbf{I}}^h_{N^2}\underline{\mathbf{u}}, \underline{\mathbf{I}}^h_{N^2}\underline{\mathbf{u}}\right)}{\beta(\underline{\mathbf{u}}, \underline{\mathbf{u}})} \sim 1, \tag{3.20}$$

where the notation $a \sim b$ means the equivalence of two quantities a and b which does not depend on E and N. Again, from Theorem 2.1 and Lemma 3.2, the absolute value of w_λ satisfies

$$\begin{aligned} |w_\lambda| = \frac{\left|\alpha\left(\underline{\mathbf{I}}^h_{N^2}\underline{\mathbf{u}}, \underline{\mathbf{I}}^h_{N^2}\underline{\mathbf{u}}\right)\right|}{\beta(\underline{\mathbf{u}}, \underline{\mathbf{u}})} &\le \frac{(1 + (1/2)|\mathbf{b}| + (1/\delta))\ \beta\left(\underline{\mathbf{I}}^h_{N^2}\underline{\mathbf{u}}, \underline{\mathbf{I}}^h_{N^2}\underline{\mathbf{u}}\right)}{\beta(\underline{\mathbf{u}}, \underline{\mathbf{u}})} \\ &\le C\left(1 + |\mathbf{b}| + \frac{1}{\delta}\right). \end{aligned} \tag{3.21}$$

Therefore, from Lemma 3.3, the real parts and the absolute values of eigenvalues of $\widehat{\mathsf{B}}^{-1}_{h^2}\widehat{\mathsf{A}}_{N^2}$ satisfy (3.16). □

Remark 3.5. Let the one dimensional (1D) bilinear forms for $\underline{\mathbf{u}}, \underline{\mathbf{w}} \in H_0^1(-1,1) \times H_0^1(-1,1)$ be

$$\alpha_1(\underline{\mathbf{u}}, \underline{\mathbf{w}}) = \langle \mathsf{A}\underline{\mathbf{u}}', \underline{\mathbf{w}}' \rangle + \langle \mathsf{B}\, b\underline{\mathbf{u}}', \underline{\mathbf{w}} \rangle + \langle (\mathsf{A}+\mathsf{C})\underline{\mathbf{u}}, \underline{\mathbf{w}} \rangle, \tag{3.22}$$

$$\beta_1(\underline{\mathbf{u}}, \underline{\mathbf{w}}) = \langle \mathsf{A}\underline{\mathbf{u}}', \underline{\mathbf{w}}' \rangle + \langle \mathsf{A}\underline{\mathbf{u}}, \underline{\mathbf{w}} \rangle, \tag{3.23}$$

and denote $\widehat{\mathsf{A}}_N$ and $\widehat{\mathsf{B}}_h$ by the 1D stiffness matrices on the spaces $\underline{\mathbf{P}}_{N,h}^0$ and $\underline{\mathbf{V}}_{N,h}^0$ corresponding to the bilinear forms (3.22) and (3.23), respectively. Then one can easily get the same results as Theorem 3.4 for 1D case. Since the proof is similar to 2D case, we omit the statements.

Now, for actual numerical computations, we need 2D stiffness and mass matrices expressed as the tensor products of 1D matrices (see [20] for details). For this, let us denote 1D spectral element matrices as

$$\mathsf{S}_N = \left\langle \phi_\mu'(t), \phi_\nu'(t) \right\rangle, \qquad \mathsf{R}_N = \left\langle \phi_\mu'(t), \phi_\nu(t) \right\rangle, \qquad \mathsf{M}_N = \langle \phi_\mu(t), \phi_\nu(t) \rangle \tag{3.24}$$

and 1D finite element matrices

$$\mathsf{S}_h = \left\langle \psi_\mu'(t), \psi_\nu'(t) \right\rangle, \qquad \mathsf{R}_h = \left\langle \psi_\mu'(t), \psi_\nu(t) \right\rangle, \qquad \mathsf{M}_h = \langle \psi_\mu(t), \psi_\nu(t) \rangle, \tag{3.25}$$

where $\{\phi_\mu\}_{\mu=1}^{\mathfrak{N}}$ and $\{\psi_\nu\}_{\nu=1}^{\mathfrak{N}}$ are the Lagrange basis of the spaces $\mathcal{P}_{N,h}^0$ and $\mathcal{V}_{N,h}^0$, respectively. For actual computations for S_N, R_N, and M_N, the inner product $\langle \cdot, \cdot \rangle$ on the space $\mathcal{P}_{N,h}^0$ will be computed using LGL quadrature rule at LGL nodes. Without any confusion, such approximate matrices can be denoted by same notations in the next section. We note that the approximate matrices S_N, R_N, and M_N and the exact matrices S_N, R_N, and M_N are equivalent, respectively, because of the equivalence of LGL quadrature on the polynomial space we used. For example, the mass matrix M_N can be computed using the LGL weights $\{\rho_{j,k}\}$ only.

4. Numerical Tests of Preconditioning

4.1. Matrix Representation

In this section we discuss effects of the proposed finite element preconditioning for the spectral element discretizations to the coupled elliptic system (1.1). For this purpose, first we set up one dimensional matrices $\widehat{\mathsf{A}}_N$ and $\widehat{\mathsf{B}}_h$ corresponding to (3.22) and (3.23) using the matrices in (3.24). One may have

$$\begin{aligned} \widehat{\mathsf{A}}_N &= \begin{bmatrix} \mathsf{S}_N + b\mathsf{R}_N + \mathsf{M}_N & \dfrac{1}{\delta}\mathsf{M}_N \\ -\dfrac{1}{\delta}\mathsf{M}_N & \dfrac{1}{\delta}(\mathsf{S}_N - b\mathsf{R}_N + \mathsf{M}_N) \end{bmatrix}_{2\mathfrak{N}\times 2\mathfrak{N}}, \\ \widehat{\mathsf{B}}_h &= \begin{bmatrix} \mathsf{S}_h + \mathsf{M}_h & \mathbf{0} \\ \mathbf{0} & \dfrac{1}{\delta}(\mathsf{S}_h + \mathsf{M}_h) \end{bmatrix}_{2\mathfrak{N}\times 2\mathfrak{N}}. \end{aligned} \tag{4.1}$$

Let $\mathbf{b} = [b_1, b_2]^T$ be a constant vector in (1.1), then the 2D matrices $\widehat{\mathsf{A}}_{N^2}$ and $\widehat{\mathsf{B}}_{h^2}$ can be expressed as

$$\widehat{\mathsf{A}}_{N^2} = \begin{bmatrix} \mathsf{S}_{N^2} + \mathsf{R}_{N^2} + \mathsf{M}_{N^2} & \dfrac{1}{\delta}\mathsf{M}_{N^2} \\ -\dfrac{1}{\delta}\mathsf{M}_{N^2} & \dfrac{1}{\delta}(\mathsf{S}_{N^2} - \mathsf{R}_{N^2} + \mathsf{M}_{N^2}) \end{bmatrix}_{2\mathfrak{N}^2 \times 2\mathfrak{N}^2},$$
$$\widehat{\mathsf{B}}_{h^2} = \begin{bmatrix} \mathsf{S}_{h^2} + \mathsf{M}_{h^2} & \mathbf{0} \\ \mathbf{0} & \dfrac{1}{\delta}(\mathsf{S}_{h^2} + \mathsf{M}_{h^2}) \end{bmatrix}_{2\mathfrak{N}^2 \times 2\mathfrak{N}^2}, \tag{4.2}$$

where

$$\mathsf{S}_{k^2} = \mathsf{M}_k(y) \otimes \mathsf{S}_k(x) + \mathsf{S}_k(y) \otimes \mathsf{M}_k(x), \qquad \mathsf{M}_{k^2} = \mathsf{M}_k(y) \otimes \mathsf{M}_k(x) \quad (k = N \text{ or } h),$$
$$\mathsf{R}_{N^2} = b_1 \mathsf{M}_N(y) \otimes \mathsf{R}_N(x) + b_2 \mathsf{R}_N(y) \otimes \mathsf{M}_N(x). \tag{4.3}$$

4.2. Numerical Analysis on Eigenvalues

The linear system $\widehat{\mathsf{A}}_{N^2}\underline{\mathbf{u}}^N = \underline{\mathbf{f}}^N$ and the preconditioned linear system $\widehat{\mathsf{B}}_{h^2}^{-1}\widehat{\mathsf{A}}_{N^2}\underline{\mathbf{u}}^N = \widehat{\mathsf{B}}_{h^2}^{-1}\underline{\mathbf{f}}^N$ will be compared in the sense of the distribution of eigenvalues. As proved in Section 3, it is shown that the behaviors of spectra of $\widehat{\mathsf{B}}_{h^2}^{-1}\widehat{\mathsf{A}}_{N^2}$ are independent of the number of elements and degrees of polynomials.

One may also see the condition numbers of these discretized systems by varying the penalty parameter δ. The condition numbers of $\widehat{\mathsf{A}}_{N^2}$ are presented in Figure 1 for fixed $\delta = 1$ (left) and fixed $E = 3$ (right) as increasing the degrees N of polynomials. It shows that such condition numbers depend on N, E, and δ. In particular, the smaller δ is, the larger condition number it yields.

Figures 2 and 4 show the spectra of the resulting preconditioned operator $\widehat{\mathsf{B}}_{h^2}^{-1}\widehat{\mathsf{A}}_{N^2}$ for the polynomials of degrees $N = 4, 6, 8, 10$ and $E = 4$ when $\delta = 1$ and $\delta = 10^{-4}$, respectively. Also, Figures 3 and 5 show the spectra of $\widehat{\mathsf{B}}_{h^2}^{-1}\widehat{\mathsf{A}}_{N^2}$ for $E = 4, 6, 8, 10$ and $N = 4$ for $\delta = 1$ and $\delta = 10^{-4}$, respectively. The same axis scales are presented for the same δ when $\mathbf{b} = [1, 1]^T$. As proven in Theorem 3.4, the eigenvalues of $\widehat{\mathsf{B}}_{h^2}^{-1}\widehat{\mathsf{A}}_{N^2}$ are independent of N and E, but they depend still on the penalty parameter δ.

By choosing the convection coefficient $\mathbf{b} = [10, 10]^T$, in Figure 6, the distributions of eigenvalues of $\widehat{\mathsf{A}}_{N^2}$(left) and $\widehat{\mathsf{B}}_{h^2}^{-1}\widehat{\mathsf{A}}_{N^2}$(right) are presented for penalty parameters $\delta = 1, 10^{-3}$ to examine their dependence. In this figure, we see that the distributions of eigenvalues (both real and imaginary part) of $\widehat{\mathsf{A}}_{N^2}$ are strongly dependent on δ. The real parts of such eigenvalues are increased in proportion to $1/\delta$. On the other hand, as predicted by Theorem 3.4, the real parts of the eigenvalues of $\widehat{\mathsf{B}}_{h^2}^{-1}\widehat{\mathsf{A}}_{N^2}$ are positive and uniformly bounded away from 0. Moreover, the real parts are not dependent on δ and $\mathbf{b}$ (see Figures 2–6), and their moduli are uniformly bounded. The numerical results show that the imaginary parts of the eigenvalues are bounded by some constants which are dependent on δ and $\mathbf{b}$. These phenomena support the theory proved in Theorem 3.4.

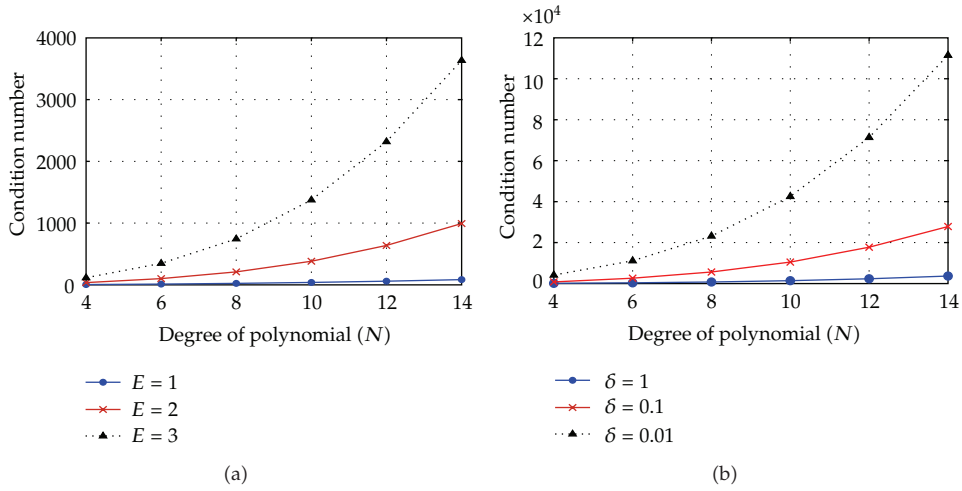


Figure 1: The condition numbers of the matrix $\widehat{A}_{N^2}$ when $\mathbf{b} = [1, 1]^T$.

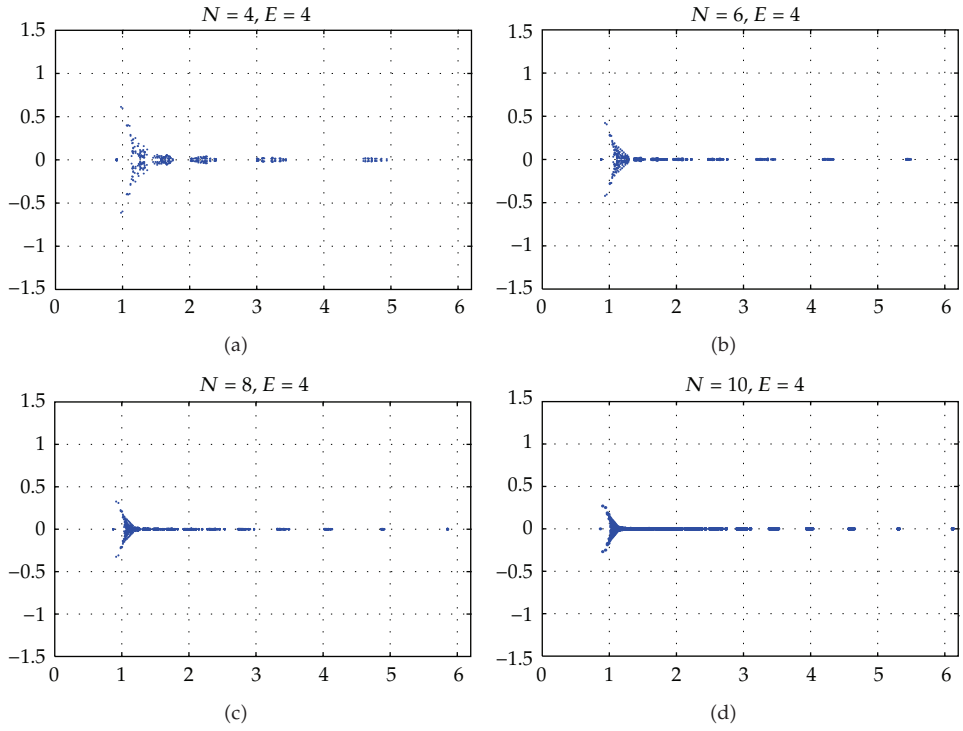


Figure 2: The eigenvalues of $\widehat{B}_{h^2}^{-1}\widehat{A}_{N^2}$ for $N = 4, 6, 8, 10$ and $E = 4$ when $\delta = 1, \mathbf{b} = [1, 1]^T$.

5. Concluding Remarks

An optimal control problem subject to an elliptic partial differential equation yields coupled elliptic differential equations (1.1). Any kind of discretizations leads to a nonsymmetric linear system which may require Krylov subspace methods to solve the system. In this paper, the

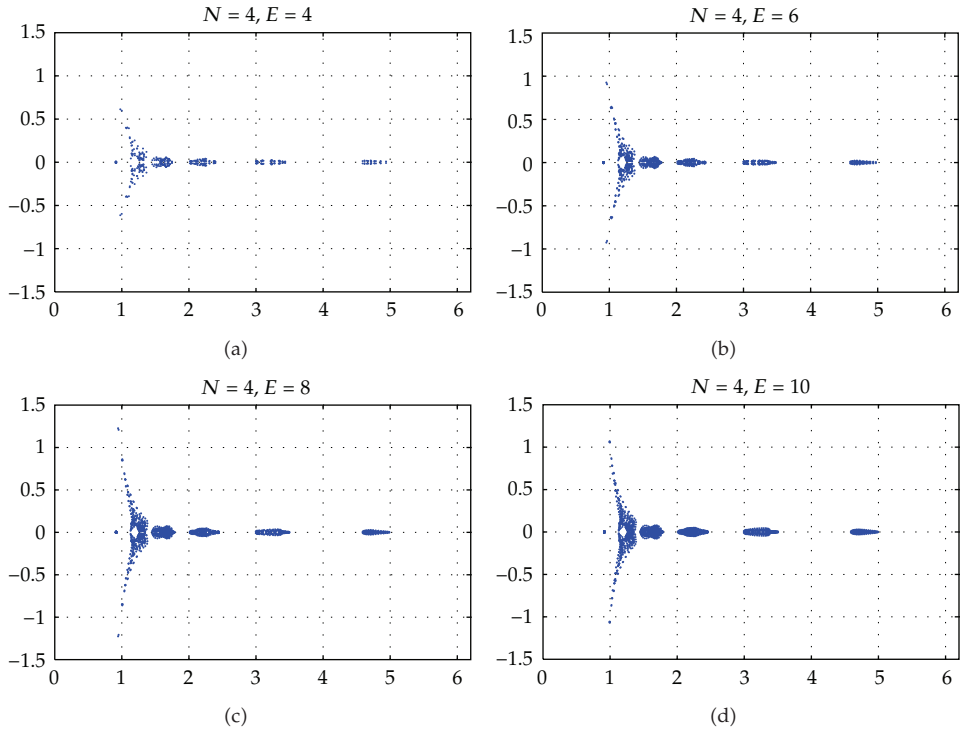


Figure 3: The eigenvalues of $\widehat{B}_{h^2}^{-1}\widehat{A}_{N^2}$ for $E = 4, 6, 8, 10$ and $N = 4$ when $\delta = 1, \mathbf{b} = [1, 1]^T$.

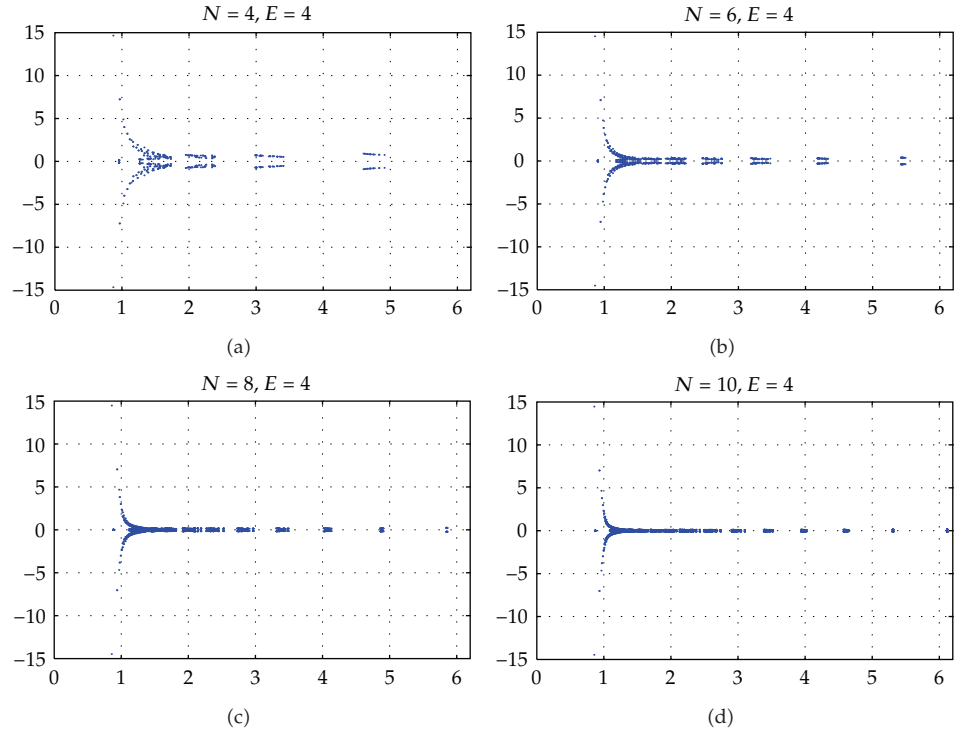


Figure 4: The eigenvalues of $\widehat{B}_{h^2}^{-1}\widehat{A}_{N^2}$ for $N = 4, 6, 8, 10$ and $E = 4$ when $\delta = 10^{-4}, \mathbf{b} = [1, 1]^T$.

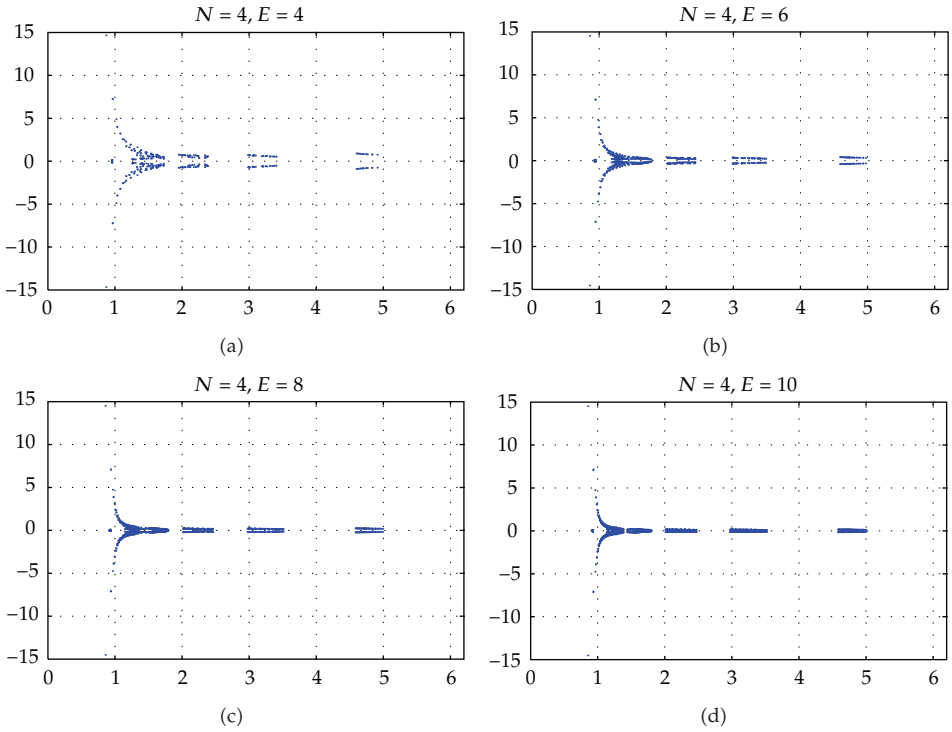


Figure 5: The eigenvalues of $\widehat{\mathrm{B}}_{h^2}^{-1}\widehat{\mathrm{A}}_{N^2}$ for $E = 4, 6, 8, 10$ and $N = 4$ when $\delta = 10^{-4}$, $\mathbf{b} = [1, 1]^T$.

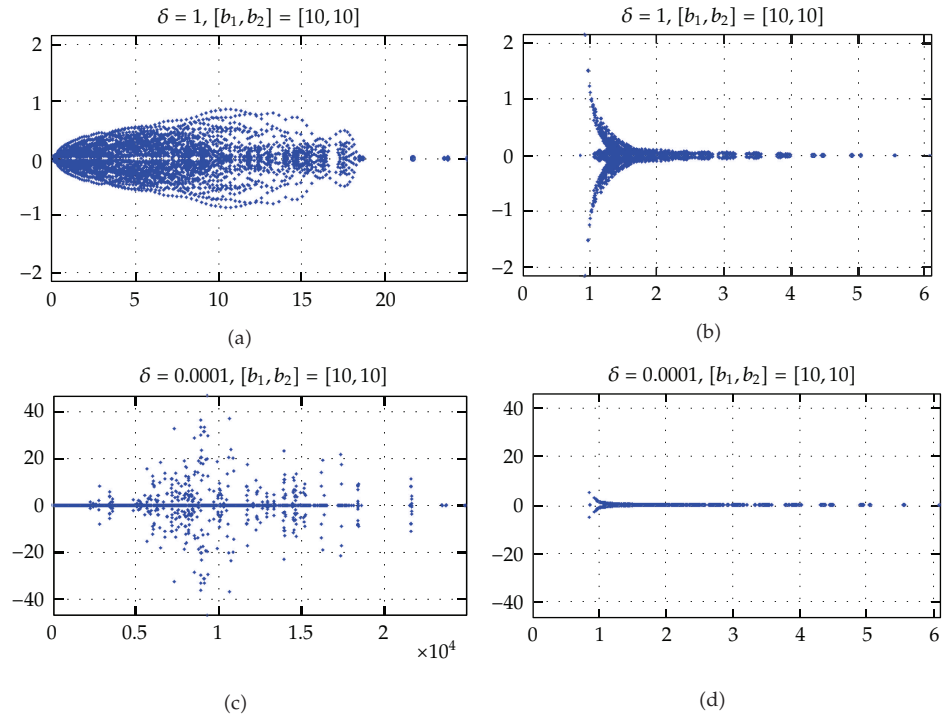


Figure 6: The eigenvalues of $\widehat{\mathrm{A}}_{N^2}$(left) and $\widehat{\mathrm{B}}_{h^2}^{-1}\widehat{\mathrm{A}}_{N^2}$(right) for $\delta = 1$ (top) and $\delta = 10^{-3}$ (bottom) when $\mathbf{b} = [10, 10]^T$ and $N = 12$, $E = 4$.

spectral element discretization is chosen because it is very accurate and popular, but the resulting linear systems have large condition numbers. This situation now becomes one of disadvantages if one aims at a fast and efficient numerical simulation for an optimal control problem subject to even a simple elliptic differential equation. To overcome such a disadvantage, the lower-order finite element preconditioner is proposed so that the preconditioned linear system has uniformly bounded condition numbers independent of the degrees of polynomials and the mesh sizes. One may also take various degrees of polynomials on subintervals with different mesh sizes. In this case, similar results can be obtained without any difficulties. This kind of finite element preconditioner may be used for an optimal control problem subject to Stokes flow (see, e.g., [13]).

Acknowledgments

The authors would like to thank Professor. Gunzburg for his kind advice to improve this paper. This work was supported by KRF under contract no. C00094.

References

[1] M. D. Gunzburger, *Perspectives in Flow Control and Optimization*, vol. 5 of *Advances in Design and Control*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2003.
[2] R. Li, W. Liu, H. Ma, and T. Tang, "Adaptive finite element approximation for distributed elliptic optimal control problems," *SIAM Journal on Control and Optimization*, vol. 41, no. 5, pp. 1321–1349, 2002.
[3] P. Bochev and M. D. Gunzburger, "Least-squares finite element methods for optimality systems arising in optimization and control problems," *SIAM Journal on Numerical Analysis*, vol. 43, no. 6, pp. 2517–2543, 2006.
[4] A. Borzì, K. Kunisch, and D. Y. Kwak, "Accuracy and convergence properties of the finite difference multigrid solution of an optimal control optimality system," *SIAM Journal on Control and Optimization*, vol. 41, no. 5, pp. 1477–1497, 2003.
[5] S. Ryu, H.-C. Lee, and S. D. Kim, "First-order system least-squares methods for an optimal control problem by the Stokes flow," *SIAM Journal on Numerical Analysis*, vol. 47, no. 2, pp. 1524–1545, 2009.
[6] Y. Chen, N. Yi, and W. Liu, "A Legendre-Galerkin spectral method for optimal control problems governed by elliptic equations," *SIAM Journal on Numerical Analysis*, vol. 46, no. 5, pp. 2254–2275, 2008.
[7] R. Ghanem and H. Sissaoui, "A posteriori error estimate by a spectral method of an elliptic optimal control problem," *Journal of Computational Mathematics and Optimization*, vol. 2, no. 2, pp. 111–135, 2006.
[8] A. T. Patera, "A spectral element method for fluid dynamics: Laminar flow in a channel expansion," *Journal of Computational Physics*, vol. 54, pp. 468–488, 1984.
[9] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods : Fundamentals in Single Domains*, Springer, Berlin, Germany, 2006.
[10] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods : Evolution to Complex Geometries and Applications to Fluid Dynamics*, Springer, Berlin, Germany, 2006.
[11] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Ruplishing Company, Boston, Mass, USA, 1966.
[12] L. N. Trefethen and D. B. Bau, *Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 1997.
[13] P. F. Fischer, "An overlapping Schwarz method for spectral element solution of the incompressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 133, no. 1, pp. 84–101, 1997.
[14] S. Kim and S. D. Kim, "Preconditioning on high-order element methods using Chebyshev-Gauss-Lobatto nodes," *Applied Numerical Mathematics*, vol. 59, no. 2, pp. 316–333, 2009.
[15] S. D. Kim and S. V. Parter, "Preconditioning Chebyshev spectral collocation method for elliptic partial differential equations," *SIAM Journal on Numerical Analysis*, vol. 33, no. 6, pp. 2375–2400, 1996.
[16] S. D. Kim and S. V. Parter, "Preconditioning Chebyshev spectral collocation by finite-difference operators," *SIAM Journal on Numerical Analysis*, vol. 34, no. 3, pp. 939–958, 1997.

[17] S. V. Parter and E. E. Rothman, "Preconditioning Legendre spectral collocation approximations to elliptic problems," *SIAM Journal on Numerical Analysis*, vol. 32, no. 2, pp. 333–385, 1995.
[18] S. D. Kim, "Piecewise bilinear preconditioning of high-order finite element methods," *Electronic Transactions on Numerical Analysis*, vol. 26, pp. 228–242, 2007.
[19] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
[20] M. O. Deville, P. F. Fischer, and E. H. Mund, *High-Order Methods for Incompressible Fluid Flow*, vol. 9 of *Cambridge Monographs on Applied and Computational Mathematics*, Cambridge University Press, Cambridge, UK, 2002.

Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 365124, 9 pages
doi:10.1155/2012/365124

Research Article

A Modified SSOR Preconditioning Strategy for Helmholtz Equations

Shi-Liang Wu[1] and Cui-Xia Li[2]

[1] *College of Mathematics, Chengdu University of Information Technology, Chengdu 610225, China*
[2] *School of Mathematics and Statistics, Anyang Normal University, Anyang 455002, China*

Correspondence should be addressed to Shi-Liang Wu, wushiliang1999@126.com

Received 22 August 2011; Revised 7 November 2011; Accepted 18 November 2011

Academic Editor: Kok Kwang Phoon

The finite difference method discretization of Helmholtz equations usually leads to the large spare linear systems. Since the coefficient matrix is frequently indefinite, it is difficult to solve iteratively. In this paper, a modified symmetric successive overrelaxation (MSSOR) preconditioning strategy is constructed based on the coefficient matrix and employed to speed up the convergence rate of iterative methods. The idea is to increase the values of diagonal elements of the coefficient matrix to obtain better preconditioners for the original linear systems. Compared with SSOR preconditioner, MSSOR preconditioner has no additional computational cost to improve the convergence rate of iterative methods. Numerical results demonstrate that this method can reduce both the number of iterations and the computational time significantly with low cost for construction and implementation of preconditioners.

1. Introduction

The finite difference method is one of the most effective and popular techniques in computational electromagnetics and seismology, such as time-harmonic wave propagations, scattering phenomena arising in acoustic and optical problems, and electromagnetics scattering from a large cavity. More information about applications of this method in electromagnetics can be found in [1–5].

In this paper, we focus on the following form of the complex Helmholtz equation:

$$\begin{aligned} -\Delta u - pu + iqu &= f \quad \text{in } \Omega, \\ u &= g \quad \text{on } \partial\Omega. \end{aligned} \tag{1.1}$$

Here Ω is a bounded region in $\mathbb{R}^2$. p and q are real continuous coefficient functions on $\overline{\Omega}$, while f and g are given continuous functions on $\overline{\Omega}$ and $\partial\Omega$, respectively.

To conveniently find numerical solutions of (1.1), the Laplace operator is approximated by using the second-order accurate 5-point difference stencil:

$$L_h \doteq \frac{1}{h^2}\begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix}_h. \tag{1.2}$$

Making use of the above stencil, the following linear system is obtained:

$$\mathcal{A}x = \left(A - h^2 D + ih^2 E\right)x = b, \tag{1.3}$$

where $D(E)$ is a diagonal matrix whose diagonal elements are just the values of $p(q)$ at the mesh points and A is the symmetric positive definite M-matrix arising from the discrete Laplace operator and is of the block tridiagonal form

$$A = \begin{bmatrix} G_1 & F_2 & & \\ E_2 & G_2 & \ddots & \\ & \ddots & \ddots & F_m \\ & & E_m & G_m \end{bmatrix}, \tag{1.4}$$

with

$$G_k = \begin{bmatrix} 4 & -1 & & \\ -1 & 4 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{bmatrix}_{n_k \times n_k}, \quad k = 1, 2, \ldots, m,$$

$$E_k = F_k^T = \begin{cases} \left(O_{n_k \times p_k} \; -I_{n_k} \, O_{n_k \times q_k}\right), & p_k, q_k \geq 0, \\ p_k + q_k = n_{k-1} - n_k & \text{if } n_k \leq n_{k-1}, \\ \left(O_{n_{k-1} \times s_k} \; -I_{n_{k-1}} \, O_{n_{k-1} \times t_k}\right)^T, & s_k, t_k \geq 0, \\ s_k + t_k = n_k - n_{k-1} & \text{if } n_k > n_{k-1}, \end{cases} \quad k = 2, 3, \ldots, m. \tag{1.5}$$

Obviously, from the linear systems (1.3), it is not difficult to find that the matrix $\mathcal{A}$ is a complex symmetric coefficient matrix. Matrix $\mathcal{A}$ becomes highly indefinite and ill-conditioned as p is a sufficiently large positive number. So, large amount of computation times and memory are needed in order to solve the linear systems (1.3) efficiently.

As is well known, direct methods and iterative methods can be employed to solve the linear systems (1.3). The former is widely employed when the order of the coefficient matrix $\mathcal{A}$ is not too large and is usually regarded as robust methods. The memory and the computational requirements for solving the large sparse linear systems may seriously challenge the most efficient direct solution method available today. Currently, the latter employed to solve the large sparse linear systems is popular. The reason is that iterative methods are easier to implement efficiently on high performance computers than direct methods. In practice, a natural choice is that we make use of iterative methods instead of direct methods to solve the large sparse linear systems.

At present, Krylov subspace methods are considered as one kind of the important and efficient iterative techniques for solving the large sparse linear systems because these methods are cheap to be implemented and are able to fully exploit the sparsity of the coefficient matrix. It is well known that the convergence speed of Krylov subspace methods depends on the distribution of the eigenvalues of the coefficient matrix [6]. When the coefficient matrix is typically extremely ill-conditioned and highly indefinite, the convergence of Krylov subspace methods can be unacceptably slow. In this case, Krylov subspace methods are not competitive without a good preconditioner. That is, preconditioning technique is a key ingredient for the success of Krylov subspace methods in applications. The idea of preconditioning technique is based on consideration of the linear system with the same solution as the original equation. The problem is that each preconditioning technique is suited for a different type of problem. Until current days, no robust preconditioning technique appears for all or at least much types of problems. Finding a good preconditioner to solve a given large sparse linear systems is often viewed as a combination of art and science.

In recent years, a great deal of effort has been invested in solving indefinite linear systems from the discrete Helmholtz equations. Most of the work has been aimed at developing effective preconditioning techniques. In general, there exist two classes of preconditioners for Helmholtz equations: the "operator-based" preconditioning technique and the "matrix-based" preconditioning technique.

The former is built based on an operator, such as the Laplace preconditioner [2, 3, 7–9], Analytic ILU [10], the Separation-of-variables [11]. The purpose of this class of preconditioners is that the spectrum of the corresponding preconditioned matrix is favorably clustered. Its advantage is that this operator does not have to be a representation of the inverse of the Helmholtz operator.

The latter is established based on an approximation of the inverse of the coefficient matrix. For this class, one of the natural and simplest ways of structuring a preconditioner is to employ a diagonal or block diagonal of the coefficient matrix as a preconditioner [12]. The above two diagonal preconditioners have no remarkable reduce with respect to the iterative number and CPU time. Another one of the simplest preconditioners is to perform an incomplete factorization (ILU) of the coefficient matrix [1]. The main idea of ILU factorizations depends on the implementation of Gaussian elimination which is used, see the survey [13] and the related references therein.

When the coefficient matrix of the linear systems (1.1) is complex symmetric and indefinite, it is difficult to solve iteratively. Using the symmetric successive overrelaxation (SSOR) as a preconditioner preserves the symmetry of the iterative matrix and also taking little initialization cost, which in some cases makes it preferable over other factorization methods such as ILU. So far, some variant SSOR preconditioning techniques have been proposed to improve the convergence rate of the corresponding iterative method for solving the linear systems. Mazzia and Alan [14] introduced a shift parameter to develop a shifted

SSOR preconditioner for solving the linear systems from an electromagnetics application. Bai in [15] used a (block) diagonal matrix instead of the diagonal matrix of the coefficient matrix to establish a modified (block) SSOR preconditioner for the second-order selfadjoint elliptic boundary value problems. Chen et al. [16] used a diagonal matrix with a relaxation parameter instead of the diagonal matrix of the coefficient matrix to establish a modified block SSOR preconditioner for the Biot's consolidation equations. Ran and Yuan in [17] also discussed a class of modified (block) SSOR preconditioners for linear systems from steady incompressible viscous flow problems. We refer the reader to [18, 19] for a general discussion.

Although the SSOR preconditioner with Krylov subspace methods can improve convergence improvement, the disadvantage of the SSOR preconditioner is that the convergence rate may still remain unsatisfactorily slow in some cases, especially in indefinite linear systems. In this paper, a modified symmetric successive overrelaxation (MSSOR) preconditioning strategy is presented, which can significantly improve the convergence speed and CPU time. Our motivation for this method arises from the solution of complex symmetric and indefinite linear systems from the discrete Helmholtz equations. The idea is to increase the values of diagonal elements of the coefficient matrix to obtain better preconditioners for the original linear systems, which is different from [14–17]. This modification does not require any significant computational cost as compared with the original SSOR preconditioner and also requires no additional storage cost.

The remainder of this paper is organized as follows. In Section 2, the MSSOR preconditioner for solving the resulting linear system is presented. In Section 3, numerical experiments are given to illustrate the efficiency of the presented preconditioner. Finally, in Section 4 some conclusions are drawn.

2. Modified SSOR Preconditioner

To improve the convergence rate of iterative methods, an appropriate preconditioner should be incorporated. That is, it is often preferable to solve the preconditioned linear system as follows:

$$P^{-1}\mathcal{A}x = P^{-1}b, \tag{2.1}$$

where P, called the preconditioner, is a nonsingular matrix. The choice of the preconditioner P plays an important role in actual implements. In general, the preconditioner P is chosen such that the condition number of the preconditioned matrix $P^{-1}\mathcal{A}$ is less than that of the original matrix $\mathcal{A}$. Based on the excellent survey of [13] by Benzi, a good preconditioner should meet the following requirements:

(1) the preconditioned system should be easy to solve;

(2) the preconditioner should be cheap to construct and apply.

Certainly, the best choice for P^{-1} is the inverse of $\mathcal{A}$. However, it is unpractical in actual implements because the cost of the computation of $\mathcal{A}^{-1}$ may be high. If $\mathcal{A}$ is a symmetric positive definite matrix, the approximation of $\mathcal{A}^{-1}$ can be replaced by SSOR or multigrid. However, in fact, the Helmholtz equation often results in an indefinite linear system, for which SSOR or multi-grid may be not guaranteed to converge.

To introduce the modified SSOR preconditioner, a brief review of the classical and well-known SSOR preconditioner is needed. The SSOR preconditioner is established by

the SSOR iterative method, which is a symmetric version of the well-known SOR iterative method. Based on matrix splitting, the coefficient matrix $\mathcal{A}$ is split as follows:

$$\mathcal{A} = \mathfrak{D} + \mathcal{L} + \mathcal{L}^T, \tag{2.2}$$

where $\mathfrak{D}$ and $\mathcal{L}$ are the diagonal parts and strictly lower triangular of $\mathcal{A}$. According to the foregoing matrix splitting (2.2), the standard SSOR preconditioner [6, 20] is defined by

$$P_{\mathrm{SSOR}} = (\mathfrak{D} + \mathcal{L})\mathfrak{D}^{-1}\left(\mathfrak{D} + \mathcal{L}^T\right). \tag{2.3}$$

It is difficult to show theoretically the behavior of a preconditioner when the coefficient matrix $\mathcal{A}$ is a large, sparse, and symmetric indefinite. The SSOR iterative method is not convergent, but P_{SSOR} may be still used as a preconditioner. By a simple modification on the original indefinite linear systems (1.3), we establish the following coefficient matrix:

$$\overline{\mathcal{A}} = A + h^2 D + h^2 Ei = \widetilde{\mathfrak{D}} + \mathcal{L} + \mathcal{L}^T, \tag{2.4}$$

where

$$\widetilde{\mathfrak{D}} = \mathrm{diag}(A) + h^2 D + h^2 Ei. \tag{2.5}$$

Obviously, $\overline{\mathcal{A}}$ is a symmetric and positive stable H-matrix. To increase the values of diagonal elements of the coefficient matrix to obtain better preconditioners for the original linear systems and reduce computation times and amount of memory, based on (2.4), the MSSOR preconditioner is defined by

$$P_{\mathrm{MSSOR}} = \left(\overline{\mathfrak{D}} + \mathcal{L}\right)\overline{\mathfrak{D}}^{-1}\left(\overline{\mathfrak{D}} + \mathcal{L}^T\right), \tag{2.6}$$

with

$$\overline{\mathfrak{D}} = \mathrm{diag}\left(\left|\widetilde{\mathfrak{D}}\right|\right), \qquad \left|\widetilde{\mathfrak{D}}\right| = \left|\widetilde{d}_{ii}\right| \quad (i = 1, 2, \ldots, n). \tag{2.7}$$

This idea is based on an absolute diagonal scaling technique, which is cheap and easy to implement.

Since the coefficient matrix of the linear systems (1.3) is neither positive definite nor Hermitian with p being a sufficiently large positive number, the Conjugate Gradient (CG) method [21] may breakdown. To solve the complex symmetric linear systems, van der Vorst and Melissen [22] proposed the conjugate orthogonal conjugate gradient (COCG) method, which is regarded as an extension of CG method.

To solve the linear systems (1.3) efficiently, (1.3) is transformed into the following form with the preconditioner P_{MSSOR}, that is,

$$P_{\mathrm{MSSOR}}\left(A - h^2 D + ih^2 E\right)x = P_{\mathrm{MSSOR}}b. \tag{2.8}$$

Then the MSSOR preconditioned COCG (PCOCG) method can be employed to solve the preconditioned linear systems (2.8).

In the following, we give the MSSOR preconditioned COCG method for solving the linear systems (2.8). The MSSOR preconditioned COCG (PCOCG) algorithm is described as follows

Algorithm PCOCG [22]: given an initial guess x_0

(1) $v_0 = b - \mathcal{A}x_0$;

(2) $p_{-1} = 0$; $\beta_{-1} = 0$;

(3) $w_0 = P_{\mathrm{MSSOR}}^{-1} v_0$; $\rho_0 = v_0^T w_0$;

(4) for $i = 0, 1, 2, \ldots$

(5) $p_i = w_i + \beta_{i-1} p_{i-1}$;

(6) $u_i = \mathcal{A}p_i$;

(7) $\mu_i = u_j^T p_j$; if $\mu_i = 0$ then quit (failure);

(8) $\alpha = \rho_i / \mu_i$;

(9) $x_{i+1} = x_i + \alpha p_i$; $v_{i+1} = v_i - \alpha u_i$

(10) if x_{i+1} is accurate enough, then quit (convergence);

(11) $w_{i+1} = P_{\mathrm{MSSOR}}^{-1} v_{i+1}$;

(12) $\rho_{i+1} = v_{i+1}^T w_{i+1}$; if $|\rho_{i+1}|$ too small, then quit (failure);

(13) $\beta_i = \rho_{i+1} / \rho_i$;

(14) End for i.

It is not difficult to find that the main computation of algorithm PCOCG involves one matrix-vector multiplication and two triangular linear systems. These computations are very easy to implement. The main advantage is no extra computational cost in construction of MSSOR preconditioner.

Note that the *transpose* in all dot products in this algorithm is essential [23]. Meanwhile, note that two different breakdowns of this algorithm may occur: one is that if $v_{i+1}^T P_{\mathrm{MSSOR}}^{-1} v_{i+1}$ is too small, but v_{i+1} exists (line 12), algorithm PCOCG breaks down and the other is that when the search direction $p_i \neq 0$, but $p_i^T \mathcal{A} p_i = 0$ (line 7), algorithm PCOCG breaks down. The breakdown can be fixed to some extent by restarting the process [22], such as the restarted process in GMRES [24]. However, breakdown scarcely happens in the actual computation of the Helmholtz equation.

3. Numerical Experiments

In this section, some numerical experiments are given to demonstrate the performance of both preconditioner P_{SSOR} and preconditioner P_{MSSOR} for solving the Helmholtz equation.

Example 3.1 (see [25]). Consider the following complex Helmholtz equation:

$$\begin{aligned} -\Delta u - pu + iqu &= f \quad \text{in } \Omega, \\ u &= g \quad \text{on } \partial\Omega, \end{aligned} \tag{3.1}$$

Table 1: Iteration numbers for the two-dimensional Helmholtz equations for $h = 1/19$ meshes, using COCG with the preconditioner P_{SSOR} and P_{MSSOR} for solving the complex symmetric indefinite linear systems.

h	Preconditioner		(p, q) (800, 10)	(800, 20)	(800, 30)	(800, 40)	(800, 60)
1/19	P_{SSOR}	IT	246	242	239	214	196
		CPU(s)	0.8438	0.8125	0.7813	0.7344	0.7031
	P_{MSSOR}	IT	138	133	126	117	96
		CPU(s)	0.25	0.2188	0.2031	0.1875	0.1719

Table 2: Iteration numbers for the two-dimensional Helmholtz equations for $h = 1/34$ meshes, using COCG with the preconditioner P_{SSOR} and P_{MSSOR} for solving the complex symmetric indefinite linear systems.

h	Preconditioner		(p, q) (1400, 40)	(1500, 40)	(1600, 40)	(1700, 40)	(1800, 40)
1/34	P_{SSOR}	IT	230	274	310	336	399
		CPU(s)	3.2813	3.9375	4.8438	4.7188	5.6719
	P_{MSSOR}	IT	214	228	237	249	260
		CPU(s)	1.4375	1.5156	1.6563	1.6250	1.7188

Table 3: Iteration numbers for the two-dimensional Helmholtz equations for $h = 1/64$ and $p = 4100$, using COCG with the preconditioner P_{SSOR} and P_{MSSOR}.

h	Preconditioner		q 100	120	150	160	180
1/64	P_{SSOR}	IT	471	456	374	349	309
		CPU(s)	32.9344	30.9688	27.9844	26.0156	21.0938
	P_{MSSOR}	IT	373	366	347	326	293
		CPU(s)	12.4688	12.2500	11.6094	10.9063	9.4219

Table 4: Iteration numbers for the two-dimensional Helmholtz equations for $h = 1/119$ and $q = 2000$, using COCG with the preconditioner P_{SSOR} and P_{MSSOR}.

h	Preconditioner		p 15000	15500	16000	16500	17000
1/119	P_{SSOR}	IT	148	153	147	165	179
		CPU(s)	55.3594	57.4531	57.8594	52.4844	67.625
	P_{MSSOR}	IT	134	137	139	144	146
		CPU(s)	22.625	22.9063	23.2813	23.7344	24.625

where $\Omega = [0,1] \times [0,1]$ and $p \geq 0$ and $q \geq 0$ are real constants. Discretizing (3.1) with the approach above in introduction, we obtain the complex symmetric indefinite linear systems $\mathcal{A}x = (A - h^2D + ih^2E)\ x = b$, and f and g are adjusted such that $b = \mathcal{A}e$ $(e = (1, 1, \ldots, 1)^T)$.

All tests are started from the zero vector, preformed in MATLAB with machine precision 10^{-16}. The COCG iteration terminates if the relative residual error satisfies $\|r^{(k)}\|_2 / \|r^{(0)}\|_2 < 10^{-6}$ or the iteration number is more than 500.

In Tables 1, 2, 3, and 4, we present some iteration results to illustrate the convergence behaviors of the COCG method preconditioned by P_{SSOR} and P_{MSSOR} to solve the complex

symmetric indefinite linear systems with the different values of p and q. In Tables 1–4, $(\cdot,\cdot)$ denotes the values of p and q. "CPU(s)" denotes the time (in seconds) required to solve a problem. "IT" denotes the number of iteration.

From Tables 1–4, it is not difficult to find that when the COCG method preconditioned by P_{SSOR} and P_{MSSOR} is used to solve the complex symmetric indefinite linear systems, the convergence rate of the preconditioner P_{MSSOR} is more efficient than that of the preconditioner P_{SSOR} by the iteration numbers and CPU time. That is, the preconditioner P_{MSSOR} outperforms the preconditioner P_{SSOR} under certain conditions. Compared with the preconditioner P_{SSOR}, the preconditioner P_{MSSOR} may be the "preferential" choice under certain conditions.

4. Conclusions

In this paper, MSSOR preconditioned COCG algorithm has been applied for solving the complex symmetric indefinite systems arising from Helmholtz equations. Due to the reduction of the iteration numbers and CPU time, the MSSOR preconditioner presented is feasible and effective. Without extra costs, MSSOR preconditioner is more efficient than SSOR preconditioner.

Acknowledgments

The authors are grateful to the referees and the editors for their helpful suggestions to improve the quality of this paper. The research of this author was supported by NSFC Tianyuan Mathematics Youth Fund (11026040).

References

[1] C.-H. Guo, "Incomplete block factorization preconditioning for linear systems arising in the numerical solution of the Helmholtz equation," *Applied Numerical Mathematics*, vol. 19, no. 4, pp. 495–508, 1996.

[2] Y. A. Erlangga, C. Vuik, and C. W. Oosterlee, "On a class of preconditioners for solving the Helmholtz equation," *Applied Numerical Mathematics*, vol. 50, no. 3-4, pp. 409–425, 2004.

[3] A. Bayliss, C. I. Goldstein, and E. Turkel, "An iterative method for the Helmholtz equation," *Journal of Computational Physics*, vol. 49, no. 3, pp. 443–457, 1983.

[4] G. Bao and W. W. Sun, "A fast algorithm for the electromagnetic scattering from a large cavity," *SIAM Journal on Scientific Computing*, vol. 27, no. 2, pp. 553–574, 2005.

[5] Y. Wang, K. Du, and W. W. Sun, "Preconditioning iterative algorithm for the electromagnetic scattering from a large cavity," *Numerical Linear Algebra with Applications*, vol. 16, no. 5, pp. 345–363, 2009.

[6] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS, Boston, Mass, USA, 1996.

[7] J. Gozani, A. Nachshon, and E. Turkel, "Conjugate gradient comipled with multi-grid for an indefinite problem," in *Advances in Computer Methods for Partial Differential Equations*, R. Vichnevestsky and R. S. Tepelman, Eds., vol. 5, pp. 425–427, IMACS, New Brunswick, NJ, USA, 1984.

[8] A. L. Laird, "Preconditioned iterative solution of the 2D Helmholtz equation," First Year's Report 02/12, Hugh's College, Oxford, UK, 2002.

[9] Y. A. Erlangga, C. Vuik, and C. W. Oosterlee, "Comparison of multigrid and incomplete LU shifted-Laplace preconditioners for the inhomogeneous Helmholtz equation," *Applied Numerical Mathematics*, vol. 56, no. 5, pp. 648–666, 2006.

[10] M. Gander, "AILU for Helmholtz problems: a new preconditioner based on the analytic parabolic factorization," *Journal of Computational Acoustics*, vol. 9, no. 4, pp. 1499–1506, 2001.

[11] R. E. Plessix and W. A. Mulder, "Separation-of-variables as a preconditioner for an iterative Helmholtz solver," *Applied Numerical Mathematics*, vol. 44, no. 3, pp. 385–400, 2003.

[12] M. M. M. Made, R. Beauwens, and G. Warzée, "Preconditioning of discrete Helmholtz operators perturbed by a diagonal complex matrix," *Communications in Numerical Methods in Engineering*, vol. 16, no. 11, pp. 801–817, 2000.
[13] M. Benzi, "Preconditioning techniques for large linear systems: a survey," *Journal of Computational Physics*, vol. 182, no. 2, pp. 418–477, 2002.
[14] F. Mazzia and R. Alan McCoy, "Numerical experimetns with a shifted SSOR preconditioner for symmetric matrices," type TR/PA/98/12, CERFACS, 1998.
[15] Z.-Z. Bai, "Mofidied block SSOR preconditioners for symmetric positive definite linear systems," *Annals of Operations Research*, vol. 103, pp. 263–282, 2001.
[16] X. Chen, K. C. Toh, and K. K. Phoon, "A modified SSOR preconditioner for sparse symmetric indefinite linear systems of equations," *International Journal for Numerical Methods in Engineering*, vol. 65, no. 6, pp. 785–807, 2006.
[17] Y.-H. Ran and L. Yuan, "On modified block SSOR iteration methods for linear systems from steady incompressible viscous flow problems," *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3050–3068, 2010.
[18] O. Axelsson, "A generalized SSOR method," *BIT*, vol. 18, pp. 443–467, 1972.
[19] A. Hadjidimos, "Successive overrelaxation (SOR) and related methods," *Journal of Computational and Applied Mathematics*, vol. 123, no. 1-2, pp. 177–199, 2000.
[20] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, New York, NY, USA, 1995.
[21] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436, 1952.
[22] H. A. van der Vorst and J. B. M. Melissen, "A Petrov-Galerkin type method for solving $Ax = b$, where A is symmetric complex," *IEEE Transactions on Magnetics*, vol. 26, pp. 706–708, 1990.
[23] R. W. Freund, "Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices," *SIAM Journal on Scientific Computing*, vol. 13, no. 1, pp. 425–448, 1992.
[24] Y. Saad and M. H. Schultz, "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific Computing*, vol. 7, no. 3, pp. 856–869, 1986.
[25] S.-L. Wu, T.-Z. Huang, L. Li, and L.-L. Xiong, "Positive stable preconditioners for symmetric indefinite linear systems arising from Helmholtz equations," *Physics Letters A*, vol. 373, no. 29, pp. 2401–2407, 2009.