A NEW PRECONDITIONER FOR SOLVING LINEAR SYSTEMS WITH SYMMETRIC Z-MATRICES

By
Jesús M. Cajigas Santiago

A thesis submitted in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE
in
APPLIED MATHEMATICS

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
May, 2012

Approved by:

VASQUEZ Pedro , Sc.D Member, Graduate Committee	Date
CASTILLO Paul , Ph.D Member, Graduate Committee	Date
YONG Xuerong , Ph.D President, Graduate Committee	Date
GONZALEZ Antonio , Ph.D Graduate Studies Representative	Date
COLON Omar , Ph.D Chairperson of the Department	Date

Abstract Presented to the Graduate School

of the University of Puerto Rico in Partial Fulfillment of the

Requirements for the Degree of Master of Science

A NEW PRECONDITIONER FOR SOLVING LINEAR SYSTEMS WITH SYMMETRIC Z-MATRICES

By

Jesús M. Cajigas Santiago

May 2012

Chair: YONG, Xuerong

Major Department: Mathematical Sciences

There are many preconditioners for linear systems, for example, $P = I + S_{max}$ developed by Kotakemori in [9] and its extension $\tilde{P}$ developed by Arenas & Yong in [1]. These preconditioners were built to speed up the convergence of the method when solving it. Symmetry is a useful property to preserve when applying a preconditioner. Preserving symmetry is advantageous since there are known methods that ensure convergence if the coefficient matrix is symmetric, among other things. Due to this fact, preconditioners that improve the convergence of the method at the same time that keeps symmetry are interesting and useful. This thesis introduces a new preconditioner, named P_{SYM}, that preserves symmetricity of the coeficient matrix and improves the convergence of the Symmetric Gauss-Seidel method when applied. This new preconditioner was based on the one proposed by Arenas & Yong in [1] using the idea proposed by Kotakemori in [9]. This one is to turn the largest entry (in terms of absolute value) above the main diagonal, by row, into zero when the preconditioner is

applied.

By applying the new preconditioner, a clear reduction on the number of iterations until convergence and the elapsed time in the Symmetric Gauss-Seidel method is observed. In addition, a reduction of the condition number of the coefficient matrix is noticed. Even though this method is not as general as the one proposed by Arenas & Yong ([1]) or Kotakemori ([9]) when comparing them by considering the coefficient matrix as a symmetric one, the new method shows an improvement. On the other hand, other preconditioners, with the exception of, $\tilde{P}$, are not pose as iterative, the preconditioner proposed for this thesis, P_{SYM} is. The preconditioner $\tilde{P}$ if applied iteratively, transforms the matrix into a lower triangular matrix; meanwhile, P_{SYM} improves this. By applying the preconditioner P_{SYM} iteratively, the matrix turns into a diagonal matrix.

Resumen Presentado a Escuela Graduada

de la Universidad de Puerto Rico como requisito parcial de los

Requerimientos para el grado de Maestría en Ciencias

UN NUEVO PRECONDICIONADOR PARA RESOLVER SISTEMAS LINEALES CON Z-MATRICES SIMÉTRICAS

Por

JESÚS M. CAJIGAS SANTIAGO

mayo 2012

Consejero: YONG, Xuerong

Departamento: Ciencias Matemáticas

Existen muchos precondicionadores para sistemas lineales, por ejemplo, $P = I + S_{max}$ desarrollado por Kotakemori en [9] y su extensión $\tilde{P}$ desarrollada por Arenas & Yong en [1]. Estos precondicionadores fueron construidos para acelerar la convergencia del método al resolverlo. Es útil preservar la simetría cuando se aplica un precondicionador. Preservar la simetría es ventajoso ya que existen métodos que aseguran la convergencia si la matriz coeficiente es simétrica, entre otras cosas. Dado este hecho, precondicionadores que mejoran la convergencia del método al mismo tiempo que mantienen la simetría son interesantes y útiles. Esta tesis introduce un nuevo precondicionador, llamado P_{SYM}, que preserva la simetría de la matriz coeficiente y mejora la convergencia del método Gauss-Seidel Simétrico. Este nuevo precondicionador fue basado en el propuesto por Arenas & Yong in [1] utilizando la idea propuesta por Kotakemori en [9]. Ésta es volver la entrada más grande (en términos de valor absoluto) sobre la diagonal principal, por fila, en cero cuando se aplica el precondicionador.

Aplicando el nuevo precondicionador, se observa una clara reducción en el número de iteraciones hasta la convergencia asi como del tiempo de ejecución del método Gauss-Seidel Simétrico. En adición, se puede ver una reducción en el número de condicionamiento de la matriz coeficiente. Aunque este método no es tan general como los propuestos por Arenas & Yong ([1]) ó Kotakemori ([9]), cuando se comparan estos considerando la matriz coeficiente como una simétrica, el nuevo método muestra una mejora. Por otro lado, otros precondicionadores, con la excepción de, $\tilde{P}$, no se plantean como iterativos, el precondicionador propuesto para ésta tesis, P_{SYM}, lo es. Si el precondicionador $\tilde{P}$ es aplicado iterativamente, transforma la matriz en una triangular inferior; P_{SYM} mejora esto. Aplicando el precondicionador P_{SYM} iterativamente, la matriz se convierte en una matriz diagonal.

**

Dedicated to:

My family, friends, professors, and everyone that, in one way or another, have encouraged me to reach my goals in life.

ACKNOWLEDGMENTS

- To God for letting me live day by day.
- To my family for encouraging me to keep studying and working towards my dreams.
- To my friend and classmate, Isnardo Arenas Navarro for his disposition and helpful ideas.
- To my advisor Dr. Xuerong Yong for giving me the opportunity to work with him, for his patience, and all his advise.
- To Dr. Paul Castillo and Dr. Pedro Vasquez, for revising my work and making valuable remarks.
- At last, but not least, to Andrea Ramírez Vélez for the grammatical corrections of this work.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Preconditioners are studied to improve the convergence of various systems, for example, linear systems. Systems of the form:

$$Ax = b$$

where A is an $n \times n$ matrix and x and b are $n \times 1$ vectors, are very popular and of great interest for applied mathematicians and engineers. It is known that systems of this form can be solved if A is a non-singular matrix. The solutions to these kind of systems can be computed with a sheet of paper and a pencil if the dimensions of the given matrix are small, however, this is not the general case.

Many popular methods like, Jacobi, Gauss-Seidel, and SSOR are used to approximate the solution of linear systems. Nevertheless, when solving them one should consider many things. For example, the time that the method takes to converge and the accuracy of the solution are important factors when solving a system. To improve the convergence of these systems preconditioners are applied. A preconditioner is a form of modification of a linear system that conditions it into a form that is more suitable for a numerical solution. Roughly speaking, a preconditioner can be seen as "good" if it speeds up the convergence of the method used.

One of the most reliable methods is the Gauss-Seidel. For now, to avoid defining the method (which will be done in Chapter 2) let's just say that it converges for any initial value when the norm of the largest eigenvalue of the matrix $M^{-1}N$ is less than one, i.e., $\rho(M^{-1}N) < 1$. So, a good preconditioner will decrease the spectral radius of $M^{-1}N$ and the time that the method takes to converge. At this point, it should be obvious what this work is aiming for, but before further discussing its main idea, lets talk briefly about its predecessors.

For many years, people like Kotakemori ([9], [10]), Gunawardena ([5]), Niki ([16]), Kohno ([8], [7]), Morimoto ([15]), Van Der Vorst ([12]), Ji-cheng Li ([11]), Noutsos ([17]), and, Hailong Shen ([21]) have work with preconditioners to improve the convergence of different methods like, for example, the Gauss-Seidel. Among all investigations on preconditioners of different people, this thesis is based and motivated by the work done by Arenas & Yong in [1]. This last one is an extension of the work done by Kotakemori in [9]. Kotakemori's idea of vanishing the maximum entry (in terms of absolute value) above the main diagonal per row of a matrix is adopted in this thesis. In their work, Arenas & Yong extend Kotakemori's preconditioner applying it iteratively, and they left the following as an open problem:

"If the matrix A is symmetric, PA is not symmetric. Thus, for future work, one could try to obtain a preconditioner derived from P that keeps the symmetry, ..."

Hence, following the open problem left by Arenas & Yong in [1], the main idea and goal of this work was established. The main idea and goal of this work was to construct a preconditioner, that would keep symmetricity of the coefficient matrix and improves the convergence of the Symmetric Gauss-Seidel method when apply.

The goal was achieved with the construction of the preconditioner P_{SYM}. This one acts over a linear system, where the coefficient matrix satisfies some conditions; for example, the matrix is assumed to be a Symmetric Z-Matrix. The preconditioner is validated in Chapter 4 with examples that give numerical proof that in fact the number of iterations until convergence and the running time of the Symmetric Gauss-Seidel method improves. In addition, the condition number of the given matrix is reduced when P_{SYM} is applied. Since the convergence of a method can be characterized by the condition number of the matrix, trying to get a smaller condition number will yields a better convergence.

An extension of the preconditioner is discussed in Chapter 3 and validated in Chapter 4 where numerical evidence shows that the preconditioner P_{SYM} can be applied iteratively. Finally, a result that describes what happens when P_{SYM} is applied iteratively is given. It was proved that in a finite number of times that P_{SYM} is applied, the matrix can be reduced to a diagonal.

CHAPTER 2

PRELIMINARIES

This chapter introduces the concepts and definitions that will be used in further chapters or are needed in order to understand certain ideas later on. Most of the concepts that will be explained are basic and are taught in Linear Algebra. In addition, books [4], [19], [13], and [6] help to understand the concepts used in the thesis.

2.1 Basic Properties of the Matrices

First, it is necessary to define and describe the type of matrices that will be used in this thesis. Examples will be provided for a better understanding of the following concepts.

Definition 2.1. *Let $A = (a_{ij})$ be a real $n \times n$ matrix, A is called a Z-Matrix if, $a_{ij} \leq 0$ for all $i \neq j$.*

Definition 2.2. *Let $A = (a_{ij})$ be a real $n \times n$ matrix. A is called diagonally dominant if,*

$$|a_{ii}| \geq \sum_{j=1\, j\neq i}^{n} |a_{ij}|, \quad \forall i = 1, \ldots, n$$

It is said to be strictly diagonally dominant if

$$|a_{ii}| > \sum_{j=1\, j\neq i}^{n} |a_{ij}|, \quad \forall i = 1, \ldots, n$$

Let the following serve as an example of a strictly diagonally dominant matrix.

Example. 2.1. *Let*

$$A = \begin{pmatrix} -3 & -1 & 0 & -1 \\ 0 & 6 & -3 & -2 \\ -2 & 0 & 4 & -1 \\ -1 & -2 & -4 & -8 \end{pmatrix}$$

Since for each row $|a_{jj}| > \sum_{i=1\, i\neq j}^{n} |a_{ij}|$, A is a strictly diagonally dominant matrix.

2.1.1 Coefficient Matrix Properties

In this thesis the coefficient matrix, A, of the linear system $Ax = b$ is assumed to be:

- A Non-Singular Symmetric Z-Matrix.
- Diagonally Dominant with Positive Main Diagonal.

2.2 Gauss-Seidel Method (G-S Method)

In numerical linear algebra, the Gauss-Seidel method (similar to the Jacobi Method), also known as the Liebmann method or the method of successive displacement, is an iterative method for solving systems of the form $Ax = b$. Sufficient conditions to ensure the convergence of the method are to take the coefficient matrix as either diagonally dominant, or symmetric positive definite. The following are the definitions of the Gauss-Seidel and the Symmetric Gauss-Seidel methods.

Definition 2.3. (Gauss-Seidel) Let $Ax = b$ and $A = M - N$ where M is the lower triangular part of A and N is the strictly upper part of $-A$, then the Gauss-Seidel iteration is given by:

$$x_{k+1} = M^{-1}Nx_k + M^{-1}b \tag{2.1}$$

since $N = M - A$, replacing it in Definition 2.1 yields,

$$x_{k+1} = x_k - M^{-1}Ax_k + M^{-1}b$$

Definition 2.4. (Symmetric Gauss-Seidel) Let $Ax = b$ and $A = M - N$ then, the Symmetric Gauss-Seidel iteration is given by:

$$x_{k+1} = x_k - M^{-1}Ax_k + M^{-1}b$$

where $M = LD^{-1}U$, $N = M - A$ and

$$L \ \textit{is the lower triangular part of } A \quad , \quad U \ \textit{is the upper triangular part of } A$$

$$D \ \textit{is the main diagonal of } A$$

As one can recall, the Gauss-Seidel Method is convergent for any initial vector if, and only if, $\rho(M^{-1}N) < 1$ (See[19], Theorem 4.1). The previously define methods are used in Chapter 4 to validate the new preconditioner.

2.3 Condition Number

A well known problem when talking about the convergence of a method is the quantity,

$$\kappa_p(A) = \|A\|_p \left\|A^{-1}\right\|_p \tag{2.2}$$

this value is called the condition number of the linear system with respect to the norm $\|.\|_p$, $p = 1, \ldots, \infty$. The condition number measures the sensitivity of the solution of a problem to perturbations in the data. This quantity is obtained by the following relation,

$$\begin{gathered} \|r\|_p = \|b - A\tilde{x}\|_p \\ \frac{\|x - \tilde{x}\|_p}{\|x\|_p} \leq \kappa_p(A) \frac{\|r\|_p}{\|b\|_p} \end{gathered} \tag{2.3}$$

when the condition number of a matrix is large, this could mean that even if the norm of the residual $\|r\|_p$ is small, the obtained approximation $\tilde{x}$ of the solution is not good.

When $p = 2$ the condition number is given by,

$$\kappa_2(A) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)} \tag{2.4}$$

where $\sigma_{max}(A)$ and $\sigma_{min}(A)$ are the maximum and minimum singular values of A.

According to Anna Pyzara, Beata Bylina, and Jaroslaw Bylina in [18] the accuracy of iterative methods, like Jacobi and Gauss-Seidel, depends on the condition number of the coefficient matrix describing the system. In their work they show, experimentally, that there exists a strong relationship between between the condition numberś value and the iterative methodś convergence. Iterative methods converge for well-conditioned matrices and diverge for ill-conditioned matrices. In Chapter 4 the condition number of various matrices are studied.

2.4 Preconditioning

Iterative methods that approximate the solution of the problem $Ax = b$ converge under certain theoretical conditions. These conditions vary for each iterative method. However, in practice, iterative methods suffer from slow convergence. Preconditioning is used to improve the convergence of these methods.

In computation, preconditioning is the procedure of transforming a system into one which has the same solution, but that is more suitable to solve with an iterative method. In other words, a preconditioner can be seen as any form of implicit or explicit modification of an original linear system that produces a system that is faster to solve than the given one.

The Jacobi preconditioner is one of the simplest forms of preconditioning and is very efficient for diagonally dominant matrices. This preconditioner requires less information of the given matrix and is easy to calculate since P is chosen to be the diagonal of the matrix A. Assuming $a_{ii} \neq 0, \forall i$, P^{-1} is given by:

$$P_{ij}^{-1} = \frac{1}{a_{ii}}$$

Another example of a preconditioner is to partition the given matrix as $A = M - R$, where R is a residual matrix and M can be, for example, lower triangular. The meaning of residual matrix in this context is that $R = M - A$. Instead of solving the system $Ax = b$ the system:

$$M^{-1}Ax = M^{-1}b$$

is solved in its place. To solve the last system the inverse of M is needed but instead of computing it explicitly, is more appropriate to write the product of M^{-1} and a vector as the solution of a linear system. When talking about the computational cost this process is less expensive.

Another way of defining a preconditioner is to perform an incomplete factorization of the given matrix A. This implies a decomposition of the form $A = LU - R$, where L is lower triangular and U is upper triangular. In practice, incomplete factorizations cannot always be achieved and even when the incomplete factorization is known or calculated, there is no guarantee that either the elapsed time of the method or the condition number of the matrix will improve. To solve $M^{-1}x = z$, where M is of the form $M = LU$, the following steps can be used:

1. Solve $Ly = x$.
2. Solve $Uz = y$.

The previous algorithm returns the value of the vector z.

One last typical preconditioner is the Symmetric Successive Overrelaxation Method (SSOR). This methods considers the splitting $A = D - E - F$, where D is the diagonal part of A, $-E$ is the strict lower triangular part of A, and $-F$ is the strict upper triangular part of A, so,

$$M_{SSOR} = (D - \omega E)D^{-1}(D - \omega F)$$

with $L \equiv (D - \omega E)D^{-1} = (I - \omega ED^{-1})$ and $U \equiv (D - \omega F)$. This preconditioner is called Symmetric Gauss-Seidel when $\omega = 1$. These preconditioners require that all entries of the diagonal of the matrix A are nonzero.

2.5 Some Preconditioners for Z-Matrices

This section aims to give a background of previous preconditioners that were and are used for Z-Matrices. It is important to say that none of the preconditioners that will be discuss in this section were specifically built for symmetric Z-matrices, neither they were built to preserve symmetry when applied. It is wise to preserve this property because methods like Gauss-Seidel, Conjugate Gradient and Biconjugate Gradient are known to converge if the given matrix is symmetric positive definite. Let A be diagonally dominant symmetric Z-matrix:

$$A = \begin{pmatrix} 1 & -\frac{1}{3} & -\frac{1}{6} & -\frac{1}{4} & -\frac{1}{5} \\ -\frac{1}{3} & 1 & 0 & -\frac{1}{2} & 0 \\ -\frac{1}{6} & 0 & 1 & 0 & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{2} & 0 & 1 & -\frac{1}{3} \\ -\frac{1}{5} & 0 & -\frac{1}{4} & -\frac{1}{3} & 1 \end{pmatrix} \tag{2.5}$$

this matrix will be used in each example of this section. For comparison purposes, the splitting of the Gauss-Seidel, $A = M - N$ (where M is the lower triangular part of A and N is the strictly upper tiangular part of $-A$) of this matrix will be calculated as well as the spectral radius of the matrix $M^{-1}N$. First, the splitting of A is the following:

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 & 0 & 0 \\ -\frac{1}{6} & 0 & 1 & 0 & 0 \\ -\frac{1}{4} & -\frac{1}{2} & 0 & 1 & 0 \\ -\frac{1}{5} & 0 & -\frac{1}{4} & -\frac{1}{3} & 1 \end{pmatrix} \quad , \quad N = \begin{pmatrix} 0 & \frac{1}{3} & \frac{1}{6} & \frac{1}{4} & \frac{1}{5} \\ 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} \\ 0 & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

for this splitting $\rho(M^{-1}N) = 0.7734$. Now, some preconditioners can be discussed.

2.5.1 Preconditioner $P = I + S_{max}$

This preconditioner was developed by Hisachi Kotakemori in [9] and is used with the Gauss-Seidel method as defined in 2.1. To be applied, the given matrix must be a Diagonally Dominant Z-matrix with unit main diagonal. The preconditioner is of the form:

$$P = I + S_{max}$$

where I is the $n \times n$ identity matrix and $S_{max} = s_{ij}$, is computed in the following way:

$$s_{ij} = \begin{cases} -a_{ij} & \text{if } j > i \text{ and } j = k_i \\ 0 & \text{otherwise} \end{cases}$$

and

$$k_i = \min\{j \mid |a_{ij}| = \max_{k>i} |a_{ik}|\}$$

The main idea on this preconditioner is to vanish the greatest entry (in terms of absolute value) of each row above the main diagonal of the given matrix. To illustrate how this preconditioner works, consider the following example.

Example. 2.2. *Let A be given, as in 2.5. Since it is a diagonally dominant Z-Matrix with unit main diagonal, then the preconditioner P can be apply and is given by:*

$$P = \begin{pmatrix} 1 & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 & 0 & \frac{1}{4} \\ 0 & 0 & 0 & 1 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

After applying the preconditioner to the matrix A, the following is obtained:

$$PA = \begin{pmatrix} \frac{8}{9} & 0 & -\frac{1}{6} & -\frac{5}{12} & -\frac{1}{5} \\ -\frac{11}{24} & \frac{3}{4} & 0 & 0 & -\frac{1}{6} \\ -\frac{13}{60} & 0 & \frac{15}{16} & -\frac{1}{12} & 0 \\ -\frac{19}{60} & -\frac{1}{2} & -\frac{1}{12} & \frac{8}{9} & 0 \\ -\frac{1}{5} & 0 & -\frac{1}{4} & -\frac{1}{3} & 1 \end{pmatrix}$$

Now, the matrices M and N for the splitting $A = M - N$, as given in 2.1, are as follows:

$$M = \begin{pmatrix} \frac{8}{9} & 0 & 0 & 0 & 0 \\ -\frac{11}{24} & \frac{3}{4} & 0 & 0 & 0 \\ -\frac{13}{60} & 0 & \frac{15}{16} & 0 & 0 \\ -\frac{19}{60} & -\frac{1}{2} & -\frac{1}{12} & \frac{8}{9} & 0 \\ -\frac{1}{5} & 0 & -\frac{1}{4} & -\frac{1}{3} & 1 \end{pmatrix} \quad , \quad N = \begin{pmatrix} 0 & 0 & \frac{1}{6} & \frac{5}{12} & \frac{1}{5} \\ 0 & 0 & 0 & 0 & \frac{1}{6} \\ 0 & 0 & 0 & \frac{1}{12} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

so the spectral radius of the product of $M^{-1}N$ is given by:

$$\rho(M^{-1}N) = 0.5966$$

2.5.2 Preconditioner $P = I + S$

This preconditioner was developed by Gunawardena in [5] and it is used with the Gauss-Seidel method as defined in 2.1. This preconditioner can be applied to the same type of matrices as $P = I + S_{max}$ and has the following structure:

$$P_S = I + S$$

where I is the $n \times n$ identity matrix and $S = s_{ij}$ is computed in the following way:

$$s_{ij} = \begin{cases} -a_{ij} & \text{if } j = i + 1 \\ 0 & \text{otherwise} \end{cases}$$

The main idea of this preconditioner is to vanish the first upper co-diagonal of the given matrix. To illustrate how this preconditioner works, consider the following example.

Example. 2.3. *Let A be given, as in 2.5. Since it is a Diagonally Dominant Z-Matrix with unit main diagonal, then the preconditioner P_S can be apply and is given by:*

$$P_S = \begin{pmatrix} 1 & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

After applying the preconditioner to the matrix A, the following is obtained:

$$P_S A = \begin{pmatrix} \frac{8}{9} & 0 & -\frac{1}{6} & -\frac{5}{12} & -\frac{1}{5} \\ -\frac{1}{3} & 1 & 0 & -\frac{1}{2} & 0 \\ -\frac{1}{6} & 0 & 1 & 0 & -\frac{1}{4} \\ -\frac{19}{60} & -\frac{1}{2} & -\frac{1}{12} & \frac{8}{9} & 0 \\ -\frac{1}{5} & 0 & -\frac{1}{4} & -\frac{1}{3} & 1 \end{pmatrix}$$

Now, the matrices M and N for the splitting $A = M - N$, as given in 2.1, are as follows:

$$M = \begin{pmatrix} \frac{8}{9} & 0 & 0 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 & 0 & 0 \\ -\frac{1}{6} & 0 & 1 & 0 & 0 \\ -\frac{19}{60} & -\frac{1}{2} & -\frac{1}{12} & \frac{8}{9} & 0 \\ -\frac{1}{5} & 0 & -\frac{1}{4} & -\frac{1}{3} & 1 \end{pmatrix} \quad , \quad N = \begin{pmatrix} 0 & 0 & \frac{1}{6} & \frac{5}{12} & \frac{1}{5} \\ 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

so the spectral radius of the product of M^{-1} and N is given by:

$$\rho(M^{-1}N) = 0.6805$$

2.5.3 Preconditioner $P_C = I + C$

This preconditioner was developed by Milaszewicz in [14] and it is used with the Gauss-Seidel method as defined in 2.1. It can be applied to the same type of matrices as $P = I + S_{max}$ and has the following structure:

$$P_C = I + C$$

where I is the $n \times n$ identity matrix and $C = c_{ij}$ is computed in the following way:

$$c_{ij} = \begin{cases} 0 & \text{if } j \neq 1 \text{ or } i = 1 \\ -a_{i1} & \text{otherwise} \end{cases}$$

The main idea of this preconditioner is to vanish the first column of the given matrix except for the entry a_{11}. To illustrate how this preconditioner works, consider the following example.

Example. 2.4. *Let A be given, as in 2.5. Since it is a Diagonally Dominant Z-Matrix with unit main diagonal, then the preconditioner P_C can be apply and is given by:*

$$P_C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 1 & 0 & 0 & 0 \\ \frac{1}{6} & 0 & 1 & 0 & 0 \\ \frac{1}{4} & 0 & 0 & 1 & 0 \\ \frac{1}{5} & 0 & 0 & 0 & 1 \end{pmatrix}$$

After applying the preconditioner to the matrix A, the following is obtained:

$$P_C A = \begin{pmatrix} 1 & -\frac{1}{3} & -\frac{1}{6} & -\frac{1}{4} & -\frac{1}{5} \\ 0 & \frac{8}{9} & -\frac{1}{18} & -\frac{7}{12} & -\frac{1}{15} \\ 0 & -\frac{1}{18} & \frac{35}{36} & -\frac{1}{24} & -\frac{17}{60} \\ 0 & -\frac{7}{12} & -\frac{1}{24} & \frac{15}{16} & -\frac{23}{60} \\ 0 & -\frac{1}{15} & -\frac{17}{60} & -\frac{23}{60} & \frac{24}{25} \end{pmatrix}$$

Now, the matrices M and N for the splitting $A = M - N$, as given in 2.1, are as follows:

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{8}{9} & 0 & 0 & 0 \\ 0 & -\frac{1}{18} & \frac{35}{36} & 0 & 0 \\ 0 & -\frac{7}{12} & -\frac{1}{24} & \frac{15}{16} & 0 \\ 0 & -\frac{1}{15} & -\frac{17}{60} & -\frac{23}{60} & \frac{24}{25} \end{pmatrix}, \quad N = \begin{pmatrix} 0 & \frac{1}{3} & \frac{1}{6} & \frac{1}{4} & \frac{1}{5} \\ 0 & 0 & \frac{1}{18} & \frac{7}{12} & \frac{1}{15} \\ 0 & 0 & 0 & \frac{1}{24} & \frac{17}{60} \\ 0 & 0 & 0 & 0 & \frac{23}{60} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

so the spectral radius of the product of M^{-1} and N is given by:

$$\rho(M^{-1}N) = 0.6971$$

CHAPTER 3

THE PROPOSED PRECONDITIONER

The idea for this preconditioner is born from the work done by Arenas & Yong in [1]. Their work is an extension of the work done by Hisachi Kotakemori in [9]. In fact, the preconditioner $\tilde{P}$ was tried to be used in a natural way ($\tilde{P}A\tilde{P}^T$) to preserve symmetry. Applying the preconditioner in this way preserves symmetry. Unfortunately, when preserving symmetry the idea of vanish the maximum entry (in terms of absolute value) above the main diagonal is lost. In order to produce better results, a new preconditioner has to be introduced. This one preserves the same idea of $\tilde{P}$, but fixes how it acts over the matrix. Before the discussion of the new preconditioner begins, let's give a brief explanation of why $\tilde{P}$ does not work.

3.1 Why $\tilde{P}$ does not Work as Expected

In this thesis $\tilde{P}$ is used instead of P because the conditions to apply $\tilde{P}$ are the same as those of P (see Section 2.5.1), except that it does not require A to have unitary main diagonal. In addition, Arenas & Yong in [1] proposed to apply P iteratively, fact that will be used in Chapter 4 were both preconditioners are applied iteratively. The structure of the preconditioner $\tilde{P}$ is as follows:

$$\tilde{P} = \begin{pmatrix} 1 & \dots & & & -\frac{a_{1,k_1}}{a_{k_1,k_1}} & & \dots \\ 0 & 1 & \dots & & -\frac{a_{2,k_2}}{a_{k_2,k_2}} & & \dots \\ 0 & 0 & 1 & \dots & -\frac{a_{3,k_3}}{a_{k_3,k_3}} & & \dots \\ \vdots & & & \ddots & & & \\ 0 & 0 & \dots & 0 & & 1 & -\frac{a_{n-1,k_{n-1}}}{a_{k_{n-1},k_{n-1}}} \\ 0 & 0 & \dots & 0 & & 0 & 1 \end{pmatrix}$$

where the k_i are computed as in 2.5.1. The problem involving $\tilde{P}$ was simple, $\tilde{P}$ is meant to be used as a product from the left, $\tilde{P}A$, and to preserve symmetry, the product $\tilde{P}A\tilde{P}^T$ needs to be considered. Now, $\tilde{P}$ acts over a matrix making the maximum entry (in terms of absolute value) of each row above the main diagonal of the original matrix zero. This does not happen when $\tilde{P}^T$ is multiplied from the right of A. This fact can be illustrated by the following example.

Example. 3.1. *Let A be a Diagonally Dominant Symmetric Z-Matrix, with positive main diagonal, given by:*

$$A = \begin{pmatrix} 6 & -1 & -2 & -1 \\ -1 & 7 & -3 & -2 \\ -2 & -3 & 8 & -1 \\ -1 & -2 & -1 & 8 \end{pmatrix}$$

then for this A, $\tilde{P}$ is as follows:

$$\tilde{P} = \begin{pmatrix} 1 & 0 & \frac{1}{4} & 0 \\ 0 & 1 & \frac{3}{8} & 0 \\ 0 & 0 & 1 & \frac{1}{8} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Now, in the product $B = \tilde{P}A = (b_{ij})$, one can guarantee that the entries $b_{1,3}$, $b_{2,3}$, and $b_{3,4}$ will be zero because of the construction of the preconditioner. The product $\tilde{P}A$ is as follows:

$$\tilde{P}A = \begin{pmatrix} 5.5 & -1.75 & 0 & -1.25 \\ -1.75 & 5.875 & 0 & -2.375 \\ -2.125 & -3.25 & 7.875 & 0 \\ -1 & -2 & -1 & 8 \end{pmatrix}$$

Until now everything is going as predicted by Arenas & Yong in [1], let's see what happens when P^T is multiplied by the right:

$$\tilde{P}A\tilde{P}^T = \begin{pmatrix} 5.5 & -1.75 & -0.1563 & -1.25 \\ -1.75 & 5.8750 & -0.2969 & -2.375 \\ -0.1563 & -0.2969 & 7.875 & 0 \\ -1.25 & -2.375 & 0 & 8 \end{pmatrix}$$

As one can easily notice, most of the entries $(\tilde{P}A)_{i,k_i}$, that were zero, now are not. This is not a real problem of the preconditioner, is just that it was tried to be used in a way it was not meant to be. The proposed preconditioner takes $\tilde{P}$ as a base and fixes it so that the entries $(\tilde{P}A\tilde{P}^T)_{i,k_i}$ end up being zeros.

3.2 The Proposed Preconditioner

Since the new preconditioner is taking as a base the one proposed by Arenas & Yong in [1], it should have a similar structure, but in this case, a new formula to calculate its entries must be derived. This new preconditioner is built to act over a matrix with the conditions described in Section 2.1.1. The following is the construction of the new preconditioner, P_{SYM}, in detail.

Since $\tilde{P}$ has been taken as a base, the structure of the new preconditioner should be as follows:

$$P_{SYM} = \begin{pmatrix} 1 & & & & p_{1,k_1} \\ 0 & \ddots & & & \vdots \\ \vdots & \ddots & 1 & & p_{n-2,k_{n-2}} \\ \vdots & & \ddots & 1 & p_{n-1,k_{n-1}} \\ 0 & & \dots & 0 & 1 \end{pmatrix}$$

where

$$k_i = \min\{j \mid |a_{ij}| = \max_{k>i} |a_{ik}|\}$$

In other words (i, k_i) indicates the position of the maximum (in terms of absolute value) value above the main diagonal in the ith row relative to the matrix A. An important detail is that k_i is always greater than i. Now, the values p_{i,k_i} must be computed. To do this, the product $P_{SYM}AP_{SYM}^T$ must be considered. So, let A be an $n \times n$ matrix (with all the conditions as described in Section2.1.1) given by:

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & \ddots & \ddots & \vdots \\ \vdots & & \ddots & a_{n-1,n} \\ a_{n,1} & \dots & a_{n,n-1} & a_{n,n} \end{pmatrix}$$

Now, the product $P_{SYM}A$ is given by:

$$P_{SYM}A = \begin{pmatrix} a_{1,1} + p_{1,k_1}a_{k_1,1} & a_{1,2} + p_{1,k_1}a_{k_1,2} & \dots & a_{1,n} + p_{1,k_1}a_{k_1,n} \\ a_{2,1} + p_{2,k_2}a_{k_2,1} & a_{2,2} + p_{2,k_2}a_{k_2,2} & \dots & a_{2,n} + p_{2,k_2}a_{k_2,n} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix}$$

When P_{SYM}^T is multiplied from the right the following is obtained:

$$B = \begin{pmatrix} a_{1,1} + p_{1,k_1}a_{k_1,1} + p_{1,k_1}(a_1, k_1 + p_{1,k_1}a_{k_1,k_1}) & \dots & \dots & \dots & a_{1,n} + p_{1,k_1} \times a_{k_1,n} \\ a_{2,1} + p_{2,k_2}a_{k_2,1} + p_{1,k_1}(a_2, k_1 + p_{2,k_2}a_{k_2,k_1}) & \dots & \dots & \dots & a_{2,n} + p_{2,k_2} \times a_{k_2,n} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ a_{1,n} + p_{1,k_1} \times a_{k_1,n} & \dots & \dots & \dots & a_{n,n} \end{pmatrix}$$

so the entries of $B = P_{SYM}AP_{SYM}^T$ are given by the following formula:

$$b_{i,j} = a_{i,j} + p_{i,k_i}a_{k_i,j} + p_{j,k_j}(a_{i,k_j} + p_{i,k_i}a_{k_i,k_j}) \tag{3.1}$$

Since the preconditioner is built with the idea of vanishing the (i, k_i) entry for each row i, B_{i,k_i} must be equal to zero for each i. Using this last fact $B_{1,k_1} = 0$. So,

$$a_{1,k_1} + p_{1,k_1}a_{k_1,k_1} + p_{k_1,k_{k_1}}(a_{1,k_1} + p_{1,k_1}a_{k_1,k_1}) = 0$$

Unfortunately, a problem arises; to calculate p_{i,k_i} information of $p_{k_1,k_{k_1}}$ is needed. In other words, the entries of P_{SYM} should be calculated recursively. To solve this issue a

p_{i,k_i} that can be calculated with less information must be found. Luckily $p_{n-1,k_{n-1}}$ can be calculated by defining $p_{n,k_n} = 0$.

- Computation of $p_{n-1,k_{n-1}}$.

  To compute this value, it must be recalled that the preconditioner should act over each row of the the matrix making the maximum value (in terms of absolute value) above the main diagonal equal to zero, i.e., $B_{n-1,k_{n-1}} = 0$. Since in the $n-1$ row the only entry above the main diagonal is in the n-th column then $k_{n-1} = n$. Now, $p_{n-1,k_{n-1}}$ can be computed.

$$\begin{aligned}
&B_{n-1,k_{n-1}} = 0 \\
&B_{n-1,n} = 0 \\
&a_{n-1,n} + p_{n-1,n}a_{n,n} = 0 \\
&p_{n-1,n}a_{n,n} = -a_{n-1,n} \\
&p_{n-1,n} = -\frac{a_{n-1,n}}{a_{n,n}}
\end{aligned}$$

  As it can be seen, setting $p_{n,k_n} = 0$, let us calculate $p_{n-1,k_{n-1}}$. This choice of p_{n,k_n} is not arbitrary, it is set to be zero because there is no entry over the main diagonal in the n-th row. With this information one can compute $p_{n-2,k_{n-2}}$.

- Computation of $p_{n-2,k_{n-2}}$.

  Like in the previous case, $B_{n-2,k_{n-2}} = 0$, but for this case two options of k_{n-2} are possible and will be dealt in two different cases. First, let's see what happens when $k_{n-2} = n-1$. This means that the maximum entry (in terms of absolute value) above the main diagonal of the matrix A in the $n-2$ row is in the $n-1$

column. Using this information one has that,

$$
\begin{aligned}
&B_{n-2,n-1} = 0 \\
&a_{n-2,n-1} + p_{n-2,n-1}a_{n-1,n-1} + p_{n-1,n}(a_{n-2,n} + p_{n-2,n-1}a_{n-1,n}) = 0 \\
&p_{n-2,n-1}(a_{n-1,n-1} + p_{n-1,n}a_{n-1,n}) = -(a_{n-2,n-1} + p_{n-1,n}a_{n-2,n}) \\
&p_{n-2,n-1} = -\frac{a_{n-2,n-1} + p_{n-1,n}a_{n-2,n}}{a_{n-1,n-1} + p_{n-1,n}a_{n-1,n}}
\end{aligned}
$$

Since A is assumed to be diagonally dominant and $p_{n-1,n} < 1$ then $a_{n-1,n-1} + p_{n-1,n}a_{n-1,n} \neq 0$. The second case is when $k_{n-2} = n$. This means that the maximum entry (in terms of absolute value) above the main diagonal of the matrix A in the $n-2$ row is in the n column. So as in the previous case, one entry must be zero and is the following one:

$$
\begin{aligned}
B_{n-2,n} &= 0 \\
a_{n-2,n} + p_{n-2,n}a_{n,n} &= 0 \\
p_{n-2,n}a_{n,n} &= -a_{n-2,n} \\
p_{n-2,n} &= -\frac{a_{n-2,n}}{a_{n,n}}
\end{aligned}
$$

considering both cases, one can say that

$$
p_{n-2,k_{n-2}} = -\frac{a_{n-2,k_{n-2}} + p_{k_{n-2},k_{k_{n-2}}}a_{n-2,k_{k_{n-2}}}}{a_{k_{n-2},k_{n-2}} + p_{k_{n-2},k_{k_{n-2}}}a_{k_{n-2},k_{k_{n-2}}}}
$$

where $p_{n,k_n} = 0$. One can continue this way until all the entries of P_{SYM} are found, but there is enough information to give a general formula for the preconditioner P_{SYM}. The same one is as follows:

$$
P_{SYM} = I + S_{SYM}
$$

where the entries of S_{SYM} are given by:

$$
S_{SYM}(i,j) = \begin{cases} -\frac{a_{i,k_i} + p_{k_i,k_{k_i}}a_{i,k_{k_i}}}{a_{k_i,k_i} + p_{k_i,k_{k_i}}a_{k_i,k_{k_i}}} & , \ j = k_i \\ 0 & , \ j \neq k_i \end{cases} \tag{3.2}
$$

and the k_i are given by:

$$k_i = \min\{j \mid |a_{i,j}| = \max_{k>i} |a_{i,k}|\}$$

Recalling Example 3.1, P_{SYM} can be put to the test. In this example $p_{4,k_4} = 0$ and the entries of the new preconditioner are calculated as follows:

$$\begin{aligned}
p_{3,k_3} &= -\frac{a_{3,k_3} + p_{k_3,k_{k_3}} a_{3,k_{k_3}}}{a_{k_3,k_3} + p_{k_3,k_{k_3}} a_{k_3,k_{k_3}}} \\
p_{3,4} &= -\frac{a_{3,4} + p_{4,k_4} a_{4,k_4}}{a_{4,4} + p_{4,k_4} a_{4,k_4}} \\
&= -\frac{a_{3,4}}{a_{4,4}} \\
&= \frac{1}{8} \\
p_{2,k_2} &= -\frac{a_{2,k_2} + p_{k_2,k_{k_2}} a_{2,k_{k_2}}}{a_{k_2,k_2} + p_{k_2,k_{k_2}} a_{k_2,k_{k_2}}} \\
p_{2,3} &= -\frac{a_{2,3} + p_{3,4} a_{2,4}}{a_{3,3} + p_{3,4} a_{3,4}} \\
&= -\frac{-3 + \frac{1}{8}(-2)}{8 + \frac{1}{8}(-1)} \\
&= \frac{\frac{13}{4}}{\frac{63}{8}} \\
&= \frac{26}{63}
\end{aligned}$$

$$\begin{aligned}
p_{1,k_1} &= -\frac{a_{1,k_1} + p_{k_1,k_{k_1}} a_{1,k_{k_1}}}{a_{k_1,k_1} + p_{k_1,k_{k_1}} a_{k_1,k_{k_1}}} \\
p_{1,3} &= -\frac{a_{1,3} + p_{3,4} a_{1,4}}{a_{3,3} + p_{3,4} a_{3,4}} \\
&= -\frac{-2 + \frac{1}{8}(-1)}{8 + \frac{1}{8}(-1)} \\
&= \frac{\frac{17}{8}}{\frac{63}{8}} \\
&= \frac{17}{63}
\end{aligned}$$

Now, the preconditioner P_{SYM} is given by:

$$P_{SYM} = \begin{pmatrix} 1 & 0 & \frac{17}{63} & 0 \\ 0 & 1 & \frac{26}{63} & 0 \\ 0 & 0 & 1 & \frac{1}{8} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and the product $P_{SYM}AP_{SYM}^T$ is as follows:

$$P_{SYM}AP_{SYM}^T = \begin{pmatrix} 5.5031 & -1.744 & 0 & -1.2698 \\ -1.744 & 5.8864 & 0 & -2.4127 \\ 0 & 0 & 7.875 & 0 \\ -1.2698 & -2.4127 & 0 & 8 \end{pmatrix}$$

With the re-calculation of the entries of $\tilde{P}$, a new preconditioner that keeps symmetry and follows the idea of vanishing the maximum entry (in terms of absolute value) above the main diagonal is born. Because of the of the complexity of the preconditioner, the following result is left as a conjecture.

Conjecture 3.1. *Let $A_{n\times n}$ be a Non-Singular Diagonally Dominant Symmetric M-Matrix and P_{SYM} the obtained preconditioner for the given matrix. Consider the splittings of the Symmetric Gauss-Seidel method, $A = M - N$ and $P_{SYM}AP_{SYM}^T = M_{SYM} - N_{SYM}$ then,*

$$\rho(M_p^{-1}N_p) < \rho(M^{-1}N) < 1$$

Even though this result is left as a conjecture, over 500 experiment were done to confirm it. The experiments were done using random matrices that satisfy the conditions in 2.1.1. The following is a sample of the results.

Matrix	$\rho(M^{-1}N)$	$\rho(M_p^{-1}N_p)$
Matx#1	0.8030	0.7816
Matx#2	0.8145	0.7952
Matx#3	0.7998	0.7785
Matx#4	0.8136	0.7947
Matx#5	0.8042	0.7839

Table 3.1: Spectral Radius of $M^{-1}N$ without Preconditioner and Applying the Preconditioner P_{SYM}.

As can be observed the spectral radius of $M^{-1}N$ reduces when P_{SYM} is applied. Even though only five experiments were taken all the experiments concerning the spectral radius show similiar results.

3.3 Special Case

A special case of this preconditioner arises when the maximum value over the main diagonal is in the last column for each row. When this happens, the preconditioner P_{SYM} is equal to the preconditioner $\tilde{P}$. To illustrate this, the following example can be considered.

Example. 3.2. *Let A be given by:*

$$A = \begin{pmatrix} 5 & -1 & -1 & -2 \\ -1 & 8 & -2 & -3 \\ -1 & -2 & 5 & -1 \\ -2 & -3 & -1 & 8 \end{pmatrix}$$

Note that A is a non-singular diagonally dominant symmetric Z-matrix with positive main diagonal. Then, the preconditioners P_{SYM} and $\tilde{P}$ are given by:

$$P_{SYM} = \begin{pmatrix} 1 & 0 & 0 & 0.25 \\ 0 & 1 & 0 & 0.375 \\ 0 & 0 & 1 & 0.125 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \tilde{P}$$

3.4 Applying P_{SYM} Iteratively

To apply the preconditioner iteratively, one must ensure that each time the preconditioner is applied, the resulting matrix shares all the properties of the original one. In other words the resulting matrix must be non-singular, symmetric Z-matrix, diagonally dominant and with positive main diagonal. Some of these results are left as conjectures since the structure of the resulting product, $P_{SYM}AP_{SYM}^T$, is very complex. Nonetheless, to provide support to them, examples will be given.

Since the structure of the entries of the matrix $B = P_{SYM}AP_{SYM}^T$ (see 3.1) depends on both i and j, it makes it very complicated to prove that B is diagonally dominant and a Z-Matrix.

Conjecture 3.2. *Let A be an $n \times n$ diagonally dominant symmetric M-Matrix then, $P_{SYM}AP_{SYM}^T$ is a diagonally dominant Z-Matrix.*

Over 500 experiments with random matrices that satisfy the conditions in 2.1.1 were done to confirm this conjecture. The results of this experiments were positive since when P_{SYM} was applied to the test matrices the resulting matrix was a diagonally dominant Z-Matrix.

Lemma 3.1. *If A is an $n \times n$ non-singular diagonally dominant symmetric Z-matrix with positive main diagonal, then $P_{SYM}AP_{SYM}^T$ is a non-singular symmetric matrix.*

Proof. Let $B = P_{SYM}AP_{SYM}^T$ then,

$$\begin{aligned} B^T &= (P_{SYM}AP_{SYM}^T)^T \\ &= (AP_{SYM}^T)^T P_{SYM}^T \\ &= P_{SYM}A^T P_{SYM}^T \end{aligned}$$

but since A is symmetric, then $A = A^T$ so B is symmetric. Now B is a non singular matrix since $det(P_{SYM}AP_{SYM}^T) = det(P_{SYM})det(A)det(P_{SYM}^T)$ and since neither of these determinants is zero then $det(B) \neq 0$ so B is non-singular. ■

Lemma 3.2. *If A is a positive definite symmetric matrix, and P_{SYM} its preconditioner then, $P_{SYM}AP_{SYM}^T$ is a positive definite matrix.*

Proof. Let x be an $n \times 1$ vector different than zero, then it must be shown that $xP_{SYM}AP_{SYM}^T x^T > 0$. Let $y = Px$, note that $y \neq 0$ since P^T is non-singular and $x^T \neq 0$ by hyphothesis. Now,

$$xP_{SYM}AP_{SYM}^T x^T = (xP_{SYM})A(xP_{SYM})^T = yAy^T$$

but $yAy^T > 0$ since A is positive definite, so $P_{SYM}AP_{SYM}^T$ is positive definite. ■

Lemma 3.3. *If $A_{n\times n}$ is a diagonally dominant symmetric Z-matrix with positive main diagonal, then $P_{SYM}AP_{SYM}^T$ has a positive main diagonal.*

Proof. Considering the entries of B as defined in 3.1, then,

$$b_{i,i} = a_{i,i} + p_{i,k_i}a_{k_i,i} + p_{i,k_i}(a_{i,k_i} + p_{i,k_i}a_{k_i,k_i})$$

but since A is a diagonally dominant Z-Matrix with positive main diagonal,

$$a_{i,i} + p_{i,k_i} a_{k_i,i} > 0 \tag{3.3}$$

and using p_{i,k_i} defined as in 3.2, the following is obtained for the sum of the last two terms:

$$\begin{aligned} a_{i,k_i} + p_{i,k_i} a_{k_i,k_i} &= a_{i,k_i} - \frac{a_{i,k_i} + p_{k_i,k_{k_i}} a_{i,k_{k_i}}}{a_{k_i,k_i} + p_{k_i,k_{k_i}} a_{k_i,k_{k_i}}} a_{k_i,k_i} \\ &= \frac{a_{i,k_i}(a_{k_i,k_i} + p_{k_i,k_{k_i}} a_{k_i,k_{k_i}}) - a_{k_i,k_i}(a_{i,k_i} + p_{k_i,k_{k_i}} a_{i,k_{k_i}})}{a_{k_i,k_i} + p_{k_i,k_{k_i}} a_{k_i,k_{k_i}}} \\ &= \frac{a_{i,k_i}(p_{k_i,k_{k_i}} a_{k_i,k_{k_i}}) - a_{k_i,k_i}(p_{k_i,k_{k_i}} a_{i,k_{k_i}})}{a_{k_i,k_i} + p_{k_i,k_{k_i}} a_{k_i,k_{k_i}}} \end{aligned}$$

again since A is a Z-Matrix with positive main diagonal $a_{i,j} < 0, \forall i \neq j$ and $a_{i,i} > 0$ so

$$a_{i,k_i} + p_{i,k_i} a_{k_i,k_i} > 0 \tag{3.4}$$

considering 3.3 and 3.4, one concludes that $b_{i,i} > 0$. So B has a positive main diagonal. The following section is going to provide numerical proof that the preconditioner P_{SYM} can be applied iteratively. ■

3.4.1 How Many Times P_{SYM} should be Applied?

The preconditioner P_{SYM} can be applied many times. Although the main idea in this work is not to apply the preconditioner iteratively, one can guess what will happen if one applies it enough times. The answer, "the matrix becomes diagonal", should pop in ones head. The proof to this is similar to that, given by Arenas in [1] so an adaptation of this is given.

Lemma 3.4. *Let A be a Non-Singular Diagonally Dominant $n \times n$ Symmetric-Matrix with positive main diagonal, then there exists $k_A \in \mathbb{N}$ such that A_{k_A} is a diagonal matrix ($A_{k_A} = P_{SYM_{A_k}} A_{k_A - 1} P^T_{SYM_{k_A}}$ and $A_0 = A$).*

Proof. For $n = 2$

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{1,2} & a_{2,2} \end{pmatrix}$$

, then

$$P_{SYM_1} = \begin{pmatrix} 1 & -\frac{a_{1,2}}{a_{2,2}} \\ 0 & 1 \end{pmatrix}$$

hence

$$A_1 = \begin{pmatrix} a_{1,1} - \frac{a_{1,2}a_{2,1}}{a_{2,2}} & 0 \\ 0 & a_{2,2} \end{pmatrix}$$

Now, suppose that for any Diagonally Dominant $A_{(n\times n)}$, Symmetric-Matrix with positive main diagonal there exists k_{A_n} such that, $A_{k_{A_n}}$ is a Diagonal Matrix.

It will be proved that for any $(n+1)\times(n+1)$ Diagonally Dominant Symmetric-Matrix with positive main diagonal, $A_{(n+1\times n+1)}$, there exists $k_{A_{n+1}}$ such that $A_{k_{A_{n+1}}}$ is a Diagonal Matrix. Let

$$A_{(n+1\times n+1)} = \left(\begin{array}{c|ccccc} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} & \dots & a_{0,k} \\ \hline a_{0,1} & a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ a_{0,2} & a_{2,1} & a_{2,2} & a_{2,3} & \dots & a_{2,n} \\ a_{0,3} & a_{3,1} & a_{3,2} & a_{3,3} & \dots & a_{3,n} \\ \vdots & \dots & & & \ddots & \\ a_{0,n} & a_{n,1} & a_{n,2} & a_{n,3} & \dots & a_{n,n} \end{array}\right)$$

Note that

$$P_{SYM_l} = \left(\begin{array}{c|cccccc} 1 & \dots & 0 & * & 0 & \dots \\ \hline 0 & & & & & \\ 0 & & & & & \\ 0 & & & \mathcal{P}_{\mathcal{SYM}_\updownarrow} & & \\ 0 & & & & & \end{array}\right)$$

where $\mathcal{P}_{\mathcal{SYM}_\updownarrow}$ is $n \times n$ preconditioner. By the induction hypothesis $A_{k_{A_n}}$ has a diagonal block and is as follows:

$$A_{k_{A_n}} = \left(\begin{array}{c|cccccc} * & * & * & * & \dots & * \\ \hline * & * & 0 & 0 & \dots & 0 \\ * & 0 & * & 0 & \dots & 0 \\ * & 0 & 0 & * & \ddots & 0 \\ \vdots & \dots & & & \ddots & 0 \\ * & 0 & 0 & 0 & \dots & * \end{array}\right)$$

since $A_{k_{A_n}}$ is a Symmetric-Matrix then $\left(A_{k_{A_n}}\right)_{0,n} = 0$ or $\left(A_{k_{A_n}}\right)_{0,n} \neq 0$.

- Case 1

  If $\left(A_{k_{A_n}}\right)_{0,n} = 0$, then $A_{k_{A_n}}$ will be as follows:

$$A_{k_{A_n}} = \left(\begin{array}{c|cccccc} * & * & * & * & \dots & 0 \\ \hline * & * & 0 & 0 & \dots & 0 \\ * & 0 & * & 0 & \dots & 0 \\ * & 0 & 0 & * & \ddots & 0 \\ \vdots & \dots & & & \ddots & 0 \\ * & 0 & 0 & 0 & \dots & * \end{array}\right)$$

  now, changing the partition of $A_{k_{A_n}}$ in the following way

$$A_{k_{A_n}} = \left(\begin{array}{ccccc|c} & & & & & 0 \\ & & & & & 0 \\ & & B_n & & & 0 \\ & & & & & \vdots \\ & & & & & 0 \\ \hline 0 & & 0 & 0 & \dots & * \end{array}\right)$$

and using the hypothesis of induction $A_{k_{A_n}+k_{B_n}}$, is a diagonal matrix.

- Case 2

  If $\left(A_{k_{A_n}}\right)_{0,n} \neq 0$, and if through $\tilde{k}$ iterations more with $\tilde{k} < k_{B_n}$ the entry becomes 0, then

$$A_{k_{A_n}+\tilde{k}} = \left(\begin{array}{ccccc|c} * & * & * & * & \dots & 0 \\ * & * & 0 & 0 & \dots & 0 \\ * & 0 & * & 0 & \dots & 0 \\ * & 0 & 0 & * & \ddots & \vdots \\ \vdots & \dots & & & \ddots & 0 \\ \hline * & 0 & 0 & 0 & \dots & * \end{array}\right).$$

  Hence, by the first case, $A_{k_{A_n}+k_{B_n}+\tilde{k}}$ is a diagonal matrix.

  On the other hand, if $\left(A_{k_{A_n}}\right)_{0,n} \neq 0$, and if through k_{B_n} iterations more the entry never becomes 0, then using the hypothesis of induction,

$$A_{k_{A_n}+k_{B_n}} = \left(\begin{array}{ccccc|c} * & 0 & 0 & 0 & \dots & * \\ 0 & * & 0 & 0 & \dots & 0 \\ 0 & 0 & * & 0 & \dots & 0 \\ 0 & 0 & 0 & * & \ddots & \vdots \\ \vdots & \dots & & & \ddots & 0 \\ \hline 0 & 0 & 0 & 0 & \dots & * \end{array}\right)$$

  so $A_{k_{A_n}+k_{B_n}+1}$ is a diagonal matrix.

Hence, $A_{k_{A_{n+1}}}$ becomes a diagonal matrix on a finite number of iterations that the preconditioner is applied. ■

The following example shows that the matrix becomes diagonal.

Example. 3.3. *Let A be given by:*

$$A = \begin{pmatrix} 0.5 & -0.25 & 0 & 0 \\ -0.25 & 0.5 & -0.25 & 0 \\ 0 & -0.25 & 0.5 & -0.25 \\ 0 & 0 & -0.25 & 0.5 \end{pmatrix}$$

note that this matrix is non-singular, diagonally dominant and is a Z-Matrix with positive main diagonal. After applying the preconditioner three times, the matrix becomes as follows:

$$\begin{pmatrix} 0.3125 & 0 & 0 & 0 \\ 0 & 0.333 & 0 & 0 \\ 0 & 0 & 0.375 & 0 \\ 0 & 0 & 0 & 0.5 \end{pmatrix}$$

Actually, it seems reasonable to think that if the preconditioner P_{SYM} is applied n to an $n \times n$ diagonally dominant symmetric Z-Matrix it will become diagonal, but this depends on whether the entries that are becoming zero are affected the next time the preconditioner is applied or not.

CHAPTER 4

EXPERIMENTS

Theoretically, preconditioners are built to give a better running time when solving a linear system. However, in practice things are not always as expected. For example, some preconditioners may take more time to converge than the original method, while others may take less.

This chapter aims to given numerical proof to show that P_{SYM} works. The chapter is divided in two parts: Non-Iterative Experiments on P_{SYM} and Iterative Experiments on P_{SYM}. The first part shows the data collected when P_{SYM} is applied only once to a linear system. The second part shows the data collected when P_{SYM} is applied iteratively to a linear system. For both parts experiments on different matrices and a comparison with the preconditioner $\tilde{P}$ is provided.

4.1 Non-Iterative Experiments on P_{SYM}

The following tables and graphs give numerical proof that P_{SYM} works. In addition, a comparison between solving the system without applying a preconditioner, applying the preconditioner $\tilde{P}$, and applying the preconditioner P_{SYM} is provided. For each of the cases and the preconditioners, the number of iterations until convergence of the Gauss-Seidel method and the Symmetric Gauss-Seidel method are supply. The criteria for convergence in all the experiments of this section is the residual norm

$$||b - Ax_n||_2 < tol$$

where the matrix A and the vector b are the input of the method, tol is a predefined tolerance, and x_n is the vector that comes from it on each iteration.

Also, the condition number when the preconditioners, $\tilde{P}$ and P_{SYM}, are applied, is compared. The two preconditioners are tested using the Laplacian matrix (in one, two, and three dimensions), Finite Volume matrices, and random matrices. These matrices have all the requirements described in Section 2.1.1 so the preconditioner can be applied to each one of them.

4.1.1 Test#1, Random Matrices

For this example random matrices of dimension 100×100 were used. These matrices have real entries and were built to satisfy all the conditions as described in 2.1.1 except that their diagonally dominance is strict. A maximum of $1,000$ iterations, a tolerance of 10^{-9}, and a starting value of $x_0 = [0, 0, \ldots, 0, 0]$ were used on each method. Also, a 100×1, non-zero, right-hand side was used for each of the experiments. As it was described in the beginning of this section, the Gauss-Seidel method and the Symmetric

Gauss-Seidel are the methods considered for these experiments. Over 500 experiments were made with random matrices, in this thesis only five are included.

Iterations Until Convergence

The iterations until convergence and the elapsed time of the Symmetric Gauss-Seidel Method using each one of the preconditioners for each random matrix is summarized in the following table.

Matrix	Without Precon	Elapsed . Time (seconds)	Arenas & Yong Precon. $\tilde{P}$	Elapsed Time (seconds)	Proposed Precon. P_{SYM}	Elapsed Time (seconds)
Matx#1	127	0.49	119	0.73	113	0.42
Matx#2	136	0.57	128	0.76	122	0.43
Matx#3	125	0.51	117	0.71	111	0.42
Matx#4	135	0.53	127	0.82	121	0.44
Matx#5	128	0.49	120	0.74	115	0.41

Table 4.1: Iterations until Convergence and Elapsed Time of the Symmetric Gauss-Seidel Method for Random Matrices.

Table 4.1 shows that, when applying any of the two preconditioners, the elapsed time and the number of iterations until convergence of the Symmetric Gauss-Seidel method improves. As it can be seen, using the preconditioner proposed for this work, P_{SYM}, the SG-S method converges in less iterations and in less time than when using the preconditioner proposed by Arenas & Yong in [1].

The following table shows the iterations until convergence and the elapsed time of the method, but this time for the Gauss-Seidel. Once again, each one of the preconditioners were applied to each one of the Random Matrices.

Matrix	Without Precon.	Elapsed Time (seconds)	Arenas& Yong Precon. $\tilde{P}$	Elapsed Time (seconds)	Proposed Precon. P_{SYM}	Elapsed Time (seconds)
Matx#1	182	0.42	175	0.39	161	0.34
Matx#2	199	0.41	192	0.41	177	0.39
Matx#3	183	0.37	176	0.35	161	0.34
Matx#4	196	0.41	190	0.40	174	0.37
Matx#5	188	0.37	182	0.39	167	0.36

Table 4.2: Iterations until Convergence and Elapsed Time of the Gauss-Seidel Method for Random Matrices.

The information obtained using the Gauss-Seidel method is similar to the one gained when the Symmetric version was used. What this means is that both preconditioners improve the convergence when using this method too, the convergence is faster, in terms of the elapsed time of the method, compared with the previous one. However, in each of the cases and for each of the methods P_{SYM} improves the convergence in terms of elapsed time and number of iterations further that $\tilde{P}$ does.

4.1.2 Test#2, One-Dimensional Laplacian Matrix

This example comes from the discretization of the equation:

$$-\frac{d^2u}{dx^2} = f(x)$$

using finite differences. This problem is considered with a Dirichlet boundary condition of $u = 0$ in all borders. For $h = \frac{1}{n}$ and values of n equal to 20, 40, 80, 160, 320 and 640. A maximum of $5,000$ iterations, a tolerance of 10^{-7}, and a starting value of $x_0 = [0, 0, \ldots, 0, 0]$ are used. To avoid trivial solutions a non-zero right-hand side is taken. For $n = 10$ the structure of the Laplacian matrix and of P_{SYM} is as follows:

$$L = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{pmatrix}$$

$$P_{SYM} = \begin{pmatrix} 1 & \frac{9}{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{8}{9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \frac{7}{8} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{6}{7} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{5}{6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \frac{4}{5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{3}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{2}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Iterations Until Convergence

The iterations until convergence and the elapsed time of the Symmetric Gauss-Seidel Method using each one of the preconditioners per value, "n", of the One-Dimensional Laplacian Matrix is summarized in the following table.

n	Without Precon.	Elapsed Time (seconds)	Arenas & Yong Precon. $\tilde{P}$	Elapsed Time (seconds)	Proposed Precon. P_{SYM}	Elapsed Time (seconds)
20	314	0.24	146	0.16	92	0.08
40	1091	1.73	503	0.94	301	0.51
80	3899	11.92	1796	6.66	1057	3.22
160	5000	30.52	5000	37.25	3794	23.27
320	5000	60.70	5000	81.84	5000	60.65
640	5000	123.36	5000	184.08	5000	123.39

Table 4.3: Iterations until Convergence and Elapsed Time of the Symmetric Gauss-Seidel Method for the One-Dimensional Laplacian Matrix.

Table 4.3 shows that, when applying any of the two preconditioners, the elapsed time and the number of iterations until convergence of the Symmetric Gauss-Seidel method improves at least for the convergent cases. As it can be seen, using the preconditioner proposed for this work, P_{SYM}, the SG-S method converges in less iterations and in less time than when using the preconditioner proposed by Arenas & Yong in [1].

The following table shows the iterations until convergence and the elapsed time of the method, but this time for the Gauss-Seidel. Once again, each one of the preconditioners were applied to the One-Dimensional Laplacian Matrix using the given values of n.

n	Without Precon.	Elapsed Time (seconds)	Arenas & Yong Precon. $\tilde{P}$	Elapsed Time (seconds)	Proposed Precon. P_{SYM}	Elapsed Time (seconds)
20	613	0.47	210	0.16	169	0.13
40	2168	3.42	746	1.13	587	0.93
80	5000	15.59	2685	8.38	2098	6.50
160	5000	30.93	5000	31.41	5000	31.28
320	5000	63.55	5000	63.35	5000	63.30
640	5000	131.06	5000	129.71	5000	129.77

Table 4.4: Iterations until Convergence and Elapsed Time of the Gauss-Seidel Method for the One-Dimensional Laplacian Matrix.

The information obtained using the Gauss-Seidel method is similar to the one gained when the Symmetric version was used. What this means is that both preconditioners improve the convergence for this method too, yet the convergence is slower, in terms of the elapsed time of the method, compared with the previous one. However, when P_{SYM} was applied, the convergence of the Gauss-Seidel method was faster that when $\tilde{P}$ was applied. The next section makes a comparison between the condition number when each preconditioner was applied.

Condition Number

The condition number of the resulting matrix when each one of the preconditioners was applied to the One-Dimensional Laplacian Matrix, of different dimensions, is shown in the next graph.

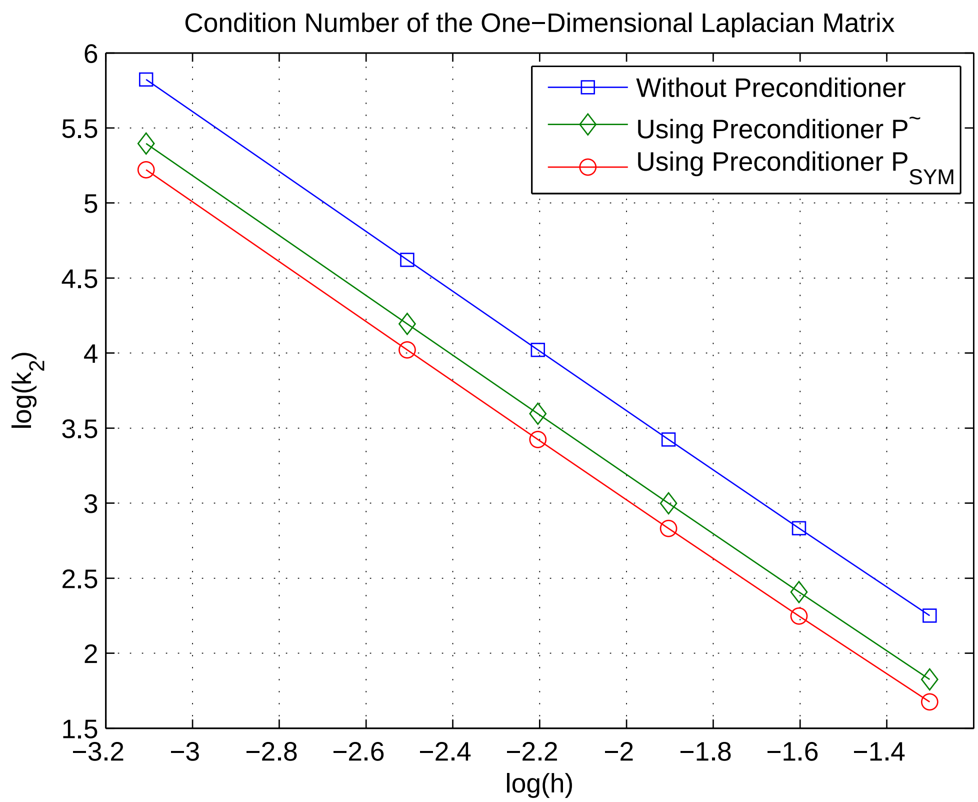


Figure 4.1: Condition Number for One-Dimensional Laplacian Matrix

Figure 4.1 shows clearly that both preconditioners, $\tilde{P}$ and P_{SYM}, improve the condition number of the given matrix when they are applied. Although there is not a significant difference between the condition number when both preconditioners are applied, P_{SYM} improves the condition number more that $\tilde{P}$ does.

4.1.3 Test#3, Two-Dimensional Laplacian Matrix

This example comes from the discretization of the equation:

$$-\left(\frac{\partial^2 u}{\partial x_1^2}+\frac{\partial^2 u}{\partial x_2^2}\right)=f(x_1,x_2)$$

using finite differences. This problem is considered with a Dirichlet boundary condition of $u = 0$ in all borders. For a $k \times k$ grid and for values of k equal to 5, 15, 25, 35 and 45. A maximum of $5,000$ iterations, a tolerance of 10^{-7}, and a starting value of $x = [0, 0, \ldots, 0, 0]$ were used. To avoid trivial solutions a non-zero right-hand side is taken. For $k = 3$ the structure of the Two-Dimensional Laplacian matrix and of P_{SYM} is as follows:

$$L=\begin{pmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{pmatrix}$$

$$P_{SYM}=\begin{pmatrix} 1 & \frac{59}{209} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{15}{56} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \frac{4}{15} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{15}{56} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{4}{15} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{4}{15} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Iterations Until Convergence

The iterations until convergence and the elapsed time of the Symmetric Gauss-Seidel Method using each one of the preconditioners per value,“k”, of the Two-Dimensional Laplacian Matrix is summarized in the following table.

k	Without Precon.	Elapsed Time (seconds)	Arenas & Yong Precon. $\tilde{P}$	Elapsed Time (seconds)	Proposed Precon. P_{SYM}	Elapsed Time (seconds)
5	35	0.05	26	0.03	22	0.02
15	213	1.82	157	1.82	118	1.10
25	540	13.05	395	16.75	322	7.69
35	1010	49.35	738	79.85	300	29.08
45	1619	134.56	1184	280.13	962	79.63

Table 4.5: Iterations until Convergence and Elapsed Time of the Symmetric Gauss-Seidel Method for the Two-Dimensional Laplacian Matrix.

Table 4.5 shows that, with the given conditions, when applying each preconditioner the SG-S method converges faster than the system without preconditioner for all cases. As it can be seen the SG-S method converges in almost the same number of iterations with any of the two preconditioners, $\tilde{P}$ and P_{SYM}. However, when P_{SYM} is applied the SG-S method converges in less time that when $\tilde{P}$ is applied.

The following table shows the iterations until convergence and the elapsed time for the method, but this time for the Gauss-Seidel. Once again, each one of the preconditioners are applied to the Two-Dimensional Laplacian Matrix using the given values of k.

k	No Precon.	Elapsed Time (seconds)	Arenas & Yong Precon. $\tilde{P}$	Elapsed Time (seconds)	Proposed Precon. P_{SYM}	Elapsed Time (seconds)
5	61	0.08	37	0.05	36	0.07
15	417	5.20	254	3.98	247	5.22
25	1070	38.06	651	29.25	634	40.60
35	2009	147.37	1224	114.73	1191	160.26
45	3228	419.04	1966	324.00	1913	455.85

Table 4.6: Iterations until Convergence and Elapsed Time of the Gauss-Seidel Method for the Two-Dimensional Laplacian Matrix.

As it can be observed, the number of iterations until convergence of the Gauss-Seidel method reduces when each one of the preconditioners is applied . Unfortunately, the elapsed time for each one of the cases is higher when comparing it with the time elapsed time when solving the system without preconditioner. Furthermore, the elapsed times obtained when the Gauss-Seidel method is used are higher that those obtained by the Symmetric Gauss-Seidel method.

Condition Number

The condition number of the resulting matrix when each one of the preconditioners was applied to the Two-Dimensional Laplacian Matrix, of different dimensions, is shown in the next graph.

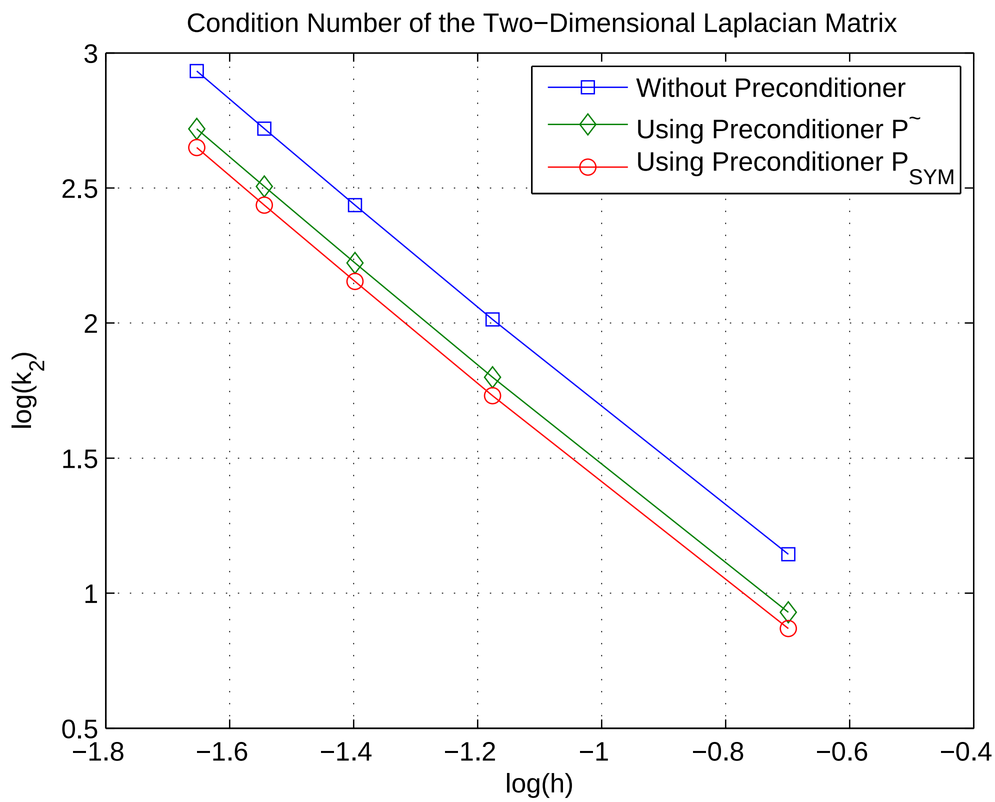


Figure 4.2: Condition Number for Two-Dimensional Laplacian Matrix

Figure 4.2 shows clearly that both preconditioners, $\tilde{P}$ and P_{SYM}, improve the condition number of the given matrix when applied. There is not a significant difference between the improvement of the condition number when each preconditioner was applied.

4.1.4 Test#4,Three-Dimensional Laplacian Matrix

This example comes from the discretization of the equation:

$$-\left(\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \frac{\partial^2 u}{\partial x_3^2}\right) = f(x_1, x_2, x_3)$$

using finite differences. This problem is considered with a Dirichlet boundary condition of $u = 0$ in all borders. Using a $k \times k \times k$ grid and for values of k equal to 10, 15, 20, 25, and 30. A maximum of $5,000$ iterations, a tolerance of 10^{-7}, and a starting value of $x = [0, 0, \ldots, 0, 0]$ were used. To avoid trivial solutions a non-zero right-hand side is taken. For $k = 2$ the structure of the Three-Dimensional Laplacian matrix and of P_{SYM} is as follows:

$$L = \begin{pmatrix} 6 & -1 & -1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 6 & 0 & -1 & 0 & -1 & 0 & 0 \\ -1 & 0 & 6 & -1 & 0 & 0 & -1 & 0 \\ 0 & -1 & -1 & 6 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 6 & -1 & -1 & 0 \\ 0 & -1 & 0 & 0 & -1 & 6 & 0 & -1 \\ 0 & 0 & -1 & 0 & -1 & 0 & 6 & -1 \\ 0 & 0 & 0 & -1 & 0 & -1 & -1 & 6 \end{pmatrix}$$

$$P_{SYM} = \begin{pmatrix} 1 & \frac{35}{204} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{6}{35} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \frac{6}{35} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \frac{1}{6} \\ 0 & 0 & 0 & 0 & 1 & \frac{6}{35} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \frac{1}{6} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{1}{6} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Iterations Until Convergence

The iterations until convergence and the elapsed time of the Symmetric Gauss-Seidel Method using each one of the preconditioners per value,“k”, of the Three-Dimensional Laplacian Matrix is summarized in the following table.

k	Without Precon.	Elapsed Time (seconds)	Arenas & Yong Precon. $\tilde{P}$	Elapsed Time (seconds)	Proposed Precon. P_{SYM}	Elapsed Time (seconds)
10	114	4.53	94	9.07	83	3.29
15	234	37.36	194	143.09	169	28.21
20	398	190.41	329	1133.91	286	136.33
25	605	718.42	499	6128.69	434	515.77
30	854	2336.54	704	24903.69	612	1685.92

Table 4.7: Iterations until Convergence and Elapsed Time of the Symmetric Gauss-Seidel Method for the Three-Dimensional Laplacian Matrix.

Table 4.5 shows that, with the given conditions, when applying each preconditioner the SG-S method converged in less iterations than the system without preconditioner for all cases. However, when P_{SYM} was applied the SG-S method converges in less time that when $\tilde{P}$ was applied. When $\tilde{P}$ was applied the SG-S method converged in more time than the system solved without applying a preconditioner.

The following table presents the iterations until convergence and the elapsed time for the method, but this time for the Gauss-Seidel. Once again, each one of the preconditioners are applied to the Three-Dimensional Laplacian Matrix using the given values of k.

k	Without Precon.	Elapsed Time (seconds)	Arenas & Yong Precon. $\tilde{P}$	Elapsed Time (seconds)	Proposed Precon. P_{SYM}	Elapsed Time (seconds)
10	219	16.08	158	16.11	157	21.69
15	460	149.40	332	151.89	329	199.38
20	788	789.22	568	793.64	564	1040.81
25	1201	3142	866	3194	859	4128
30	1700	10719	1226	10771	1216	13896

Table 4.8: Iterations until Convergence and Elapsed Time of the Gauss-Seidel Method for the Three-Dimensional Laplacian Matrix.

As it can be seen, the number of iterations until convergence of the Gauss-Seidel method reduces when each preconditioner is applied. Unfortunately, the elapsed time when the preconditioners were applied is higher that when the system is solved without preconditioner.

Condition Number

The condition number of the resulting matrix when each one of the preconditioners was applied to the Three-Dimensional Laplacian Matrix, of different dimensions, is shown in the next graph.

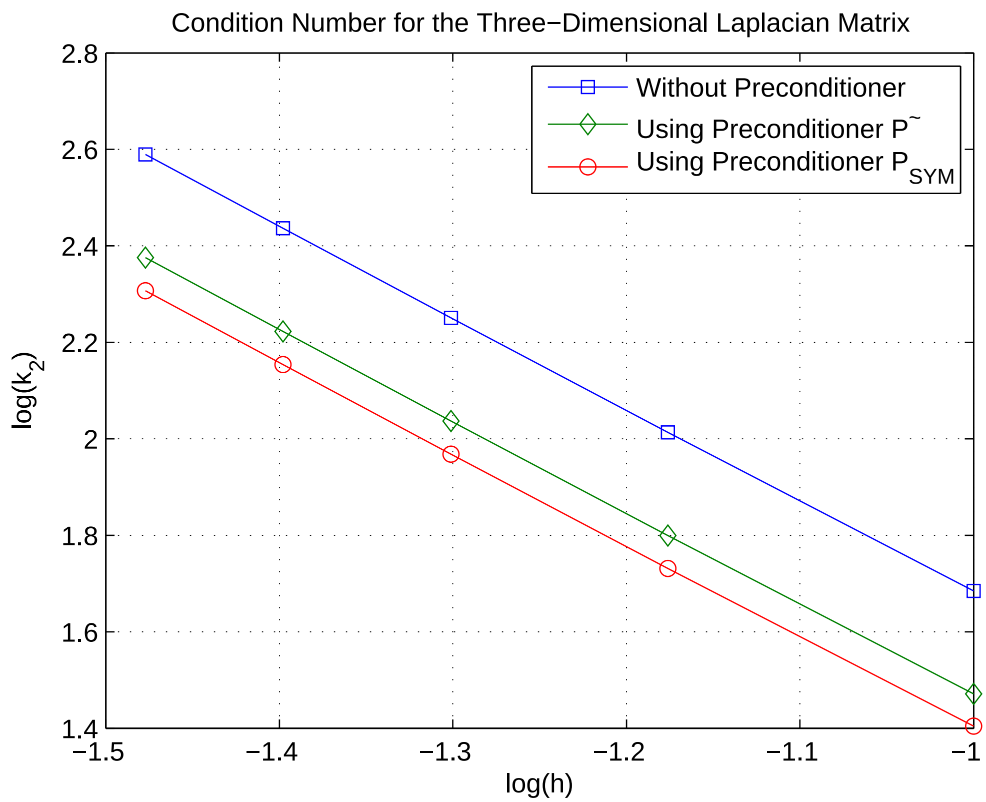


Figure 4.3: Condition Number for Three-Dimensional Laplacian

Once again Figure 4.3 shows that the condition number when applying P_{SYM} is better than when $\tilde{P}$ is applied.

4.1.5 Test#5, Finite Volume Matrices

This example comes from the discretization of the equation:

$$-\nabla\left(K\nabla u\right)=0$$

using finite differences. Using a $k\times k$ grid and for values of k equal to 10, 20, 40, and 80. For each grid a random set of symmetric materials was created. No flow boundary conditions are imposed at the right, top, and the bottom sides. A Dirichlet border condition of $u=1$ is imposed on the left side (see figure 4.4). A maximum of $7,000$ iterations, a tolerance of 10^{-6}, and a starting value of $x=[0,0,\ldots,0,0]$ are use on each case.

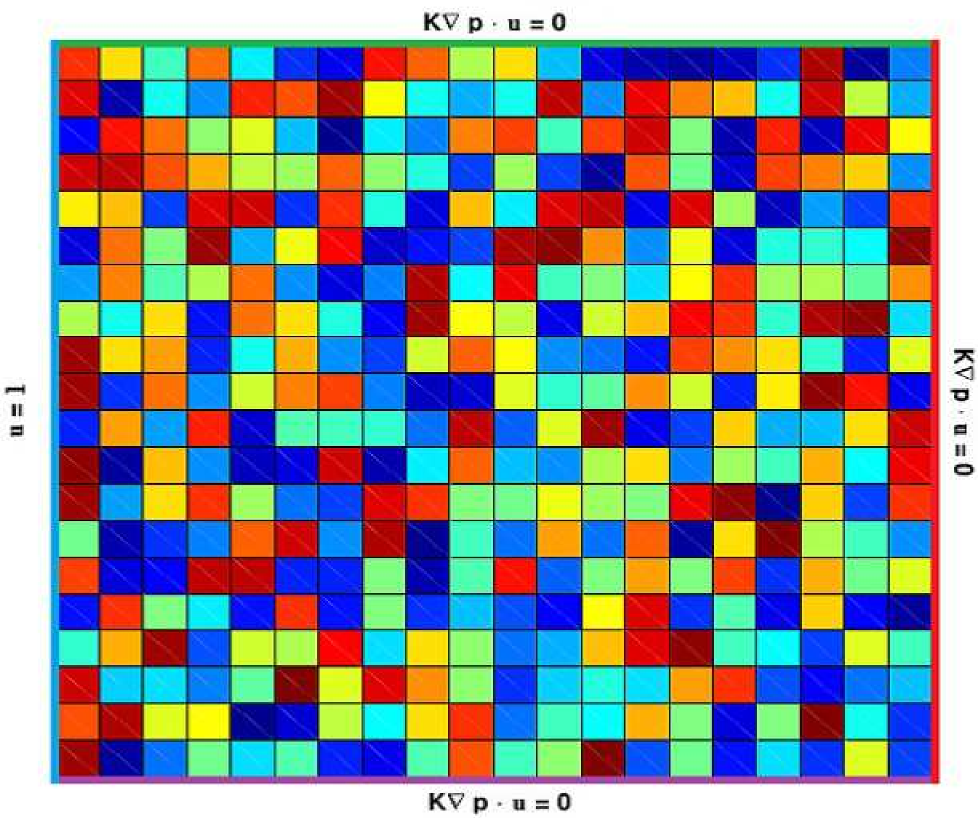


Figure 4.4: Representative Grid with Random Materials and Boundary Conditions

Iterations Until Convergence

The iterations until convergence and the elapsed time of the Symmetric Gauss-Seidel Method using each one of the preconditioners and for each matrix is summarized in the following table.

k	Without Precon.	Elapsed Time (seconds)	Arenas & Yong Precon. $\tilde{P}$	Elapsed Time (seconds)	Proposed Precon. P_{SYM}	Elapsed Time (seconds)
10	1316	5	689	3	564	2
20	2946	45	1533	38	1214	19
40	7000	452	6341	1357	5043	365
80	7000	2520	7000	35206	7000	2430

Table 4.9: Iterations until Convergence and Elapsed Time of the Symmetric Gauss-Seidel Method for Random Finite Volume Matrices.

Table 4.9 shows that, when applying any of the two preconditioners, the number of iterations until convergence of the Symmetric Gauss-Seidel method improves at least for the convergent cases (where the number of iterations did not reach $7,000$). As it can be seen, when P_{SYM} is applied the SG-S method converges in less iterations and in less time than when the preconditioner proposed by Arenas & Yong in [1] is apply.

The following table shows the iterations until convergence and the elapsed time of the method, but this time for the Gauss-Seidel. Once again, each one of the preconditioners were applied to each one of the Finite Volume Matrices.

k	Without Preconditioner	Elapsed Time (seconds)	Arenas & Yong Preconditioner $\tilde{P}$	Elapsed Time (seconds)	Proposed Preconditioner P_{SYM}	Elapsed Time (seconds)
10	1908	6	804	2	746	2
20	3964	56	1695	23	1551	21
40	7000	474	5718	389	5241	354
80	7000	3625	7000	3706	7000	3838

Table 4.10: Iterations until Convergence and Elapsed Time of the Gauss-Seidel Method for Random Finite Volume Matrices.

The information obtained using the Gauss-Seidel method is similar to the one gained when the Symmetric version was used. However, in each of the cases and for each of the methods P_{SYM} improves the convergence in terms of elapsed time and number of iterations further that $\tilde{P}$ does. The elapsed time of both the Symmetric Gauss-Seidel and the Gauss-Seidel were similar.

4.2 Iterative Experiments on P_{SYM}

The main interest of the following examples is to compare the number of iterations until convergence of the Symmetric Gauss-Seidel when the preconditioners, $\tilde{P}$ and P_{SYM}, are applied iteratively. The notion of convergence is the same as in the previous section, i.e.,

$$||b - Ax_n||_2 < tol$$

Also, a comparison between the spectral radius of the Symmetric Gauss-Seidel when both preconditioners are applied is provided for each experiment.

4.2.1 One-Dimensional Laplacian Matrix

Consider the hypothesis given in Section 4.1.2, but this time for $n = 160$ only. The next graphs show the residual error in the Symmetric Gauss-Seidel method when P_{SYM} and $\tilde{P}$ are applied iteratively, respectively.

As one can notice, when any of the two preconditioners are applied iteratively, the number of iterations until convergence on the Symmetric Gauss-Seidel method reduces. Comparing both preconditioners when they are applied iteratively, P_{SYM} reduces the number of iterations until the convergence of the method more than $\tilde{P}$ does. Also, for the first two times $\tilde{P}$ was applied, the method does not converge, but it does converge right after applying P_{SYM} for the first time.

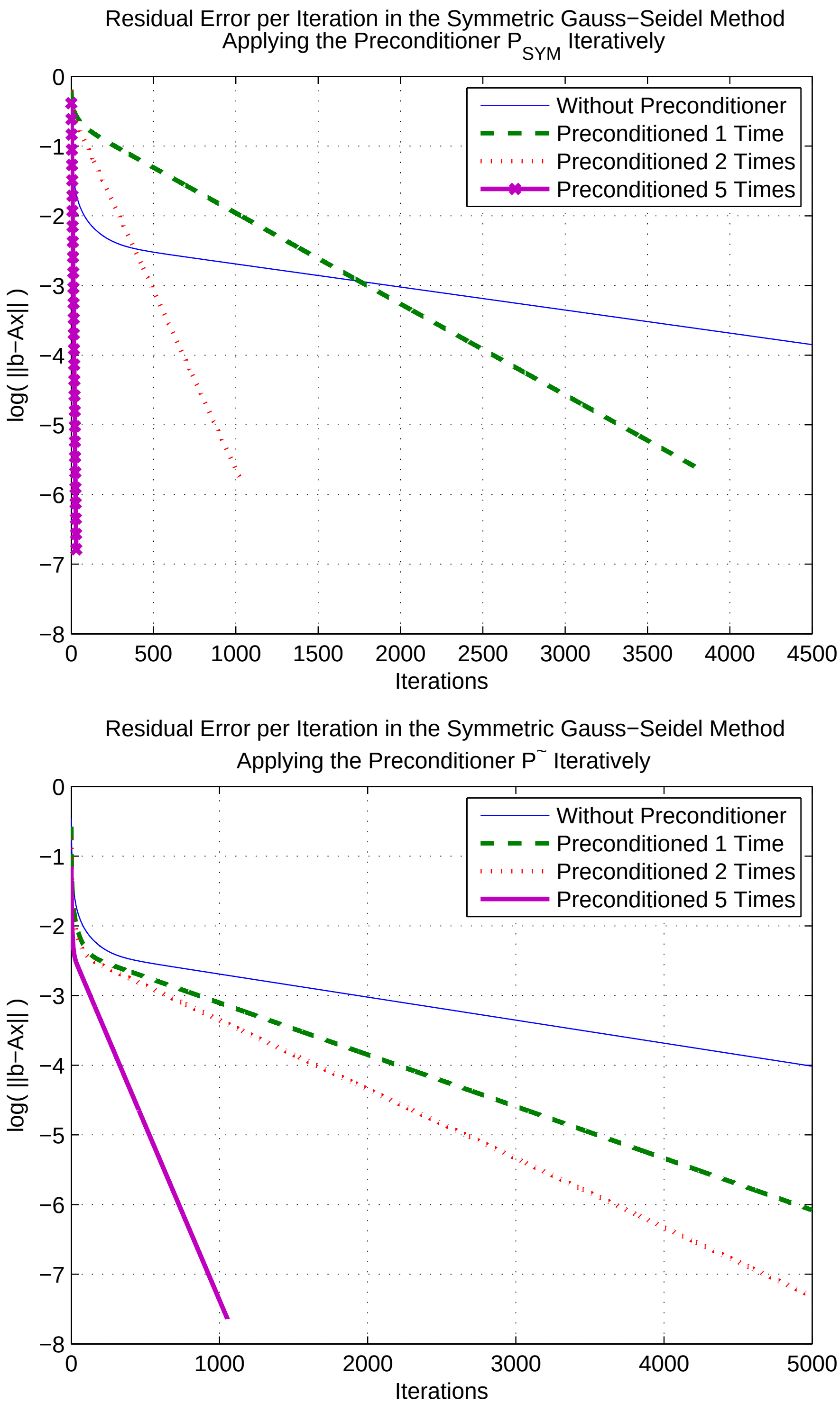


Figure 4.5: Residual Error of the Symmetric Gauss-Seidel method applying P_{SYM} and $\tilde{P}$ iteratively to the One-Dimensional Laplacian matrix, respectively.

The following table shows how the spectral radius of $(M^{-1}N)$ reduces when each preconditioner is applied iteratively. M and N came from the partition of the Symmetric Gauss-Seidel method as previously defined.

Times Preconditioned	$\rho(M^{-1}N)$ Applying $\tilde{P}$	$\rho(M^{-1}N)$ Applying P_{SYM}
0	0.9992	0.9992
1	0.9983	0.9970
2	0.9977	0.9882
5	0.9947	0.6014

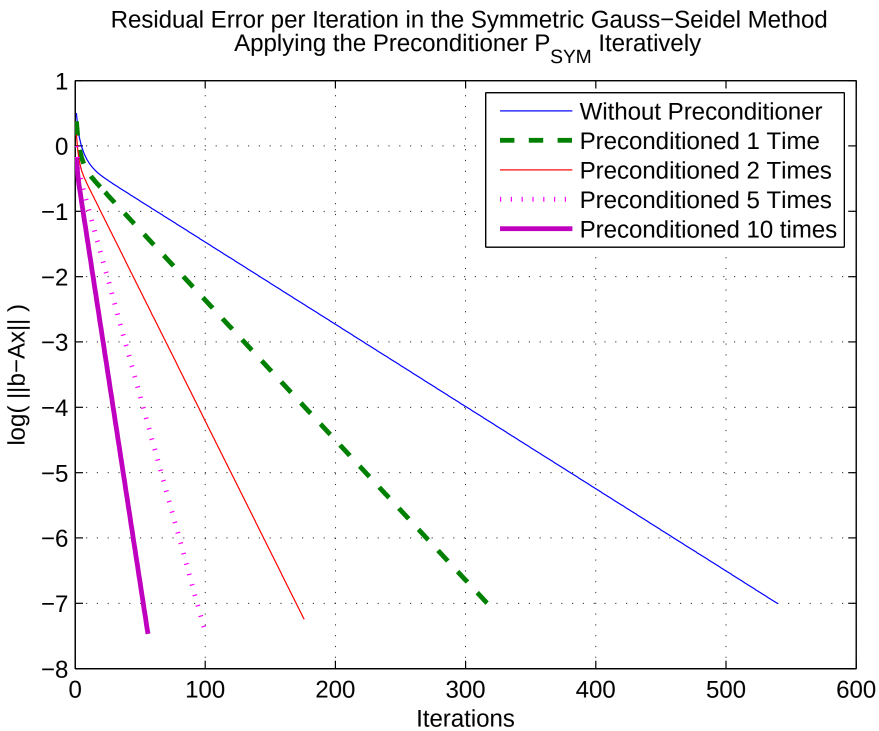
Table 4.11: Spectral Radius of $(M^{-1}N)$ Applying the Preconditioners P_{SYM} and $\tilde{P}$ Iteratively on the One-Dimensional Laplacian Matrix.

As it can be seen, when P_{SYM} is applied iteratively, $\rho(M^{-1}N)$ decreases faster than it does when $\tilde{P}$ is applied. This behaviour explains the fact that the Symmetric Gauss-Seidel method converges in less iteration when P_{SYM} is applied than when $\tilde{P}$ is.

4.2.2 Two-Dimensional Laplacian Matrix

Consider the hypothesis given in Section 4.1.3, but this time for $k = 25$ only. The next graphs show the residual error of the Symmetric Gauss-Seidel method when P_{SYM} and $\tilde{P}$ are applied iteratively, respectively.

As one can see, when any of the two preconditioners are applied iteratively, the number of iterations until convergence on the Symmetric Gauss-Seidel method reduces. Comparing both preconditioners when they are applied iteratively, P_{SYM} reduces the number of iterations until the convergence of the method more than $\tilde{P}$ does.



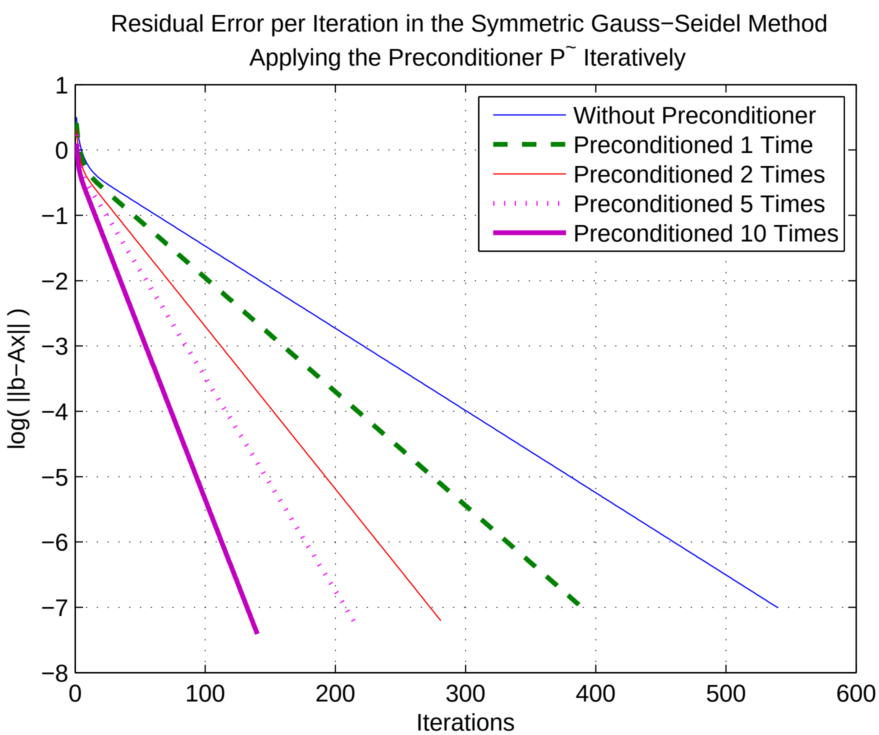


Figure 4.6: Residual Error of the Symmetric Gauss-Seidel method applying P_{SYM} and $\tilde{P}$ iteratively to the Two-Dimensional Laplacian matrix, respectively.

The following table shows how $\rho(M^{-1}N)$ reduces when each preconditioner is applied iteratively. M and N came from the partition of the Symmetric Gauss-Seidel method as previously defined.

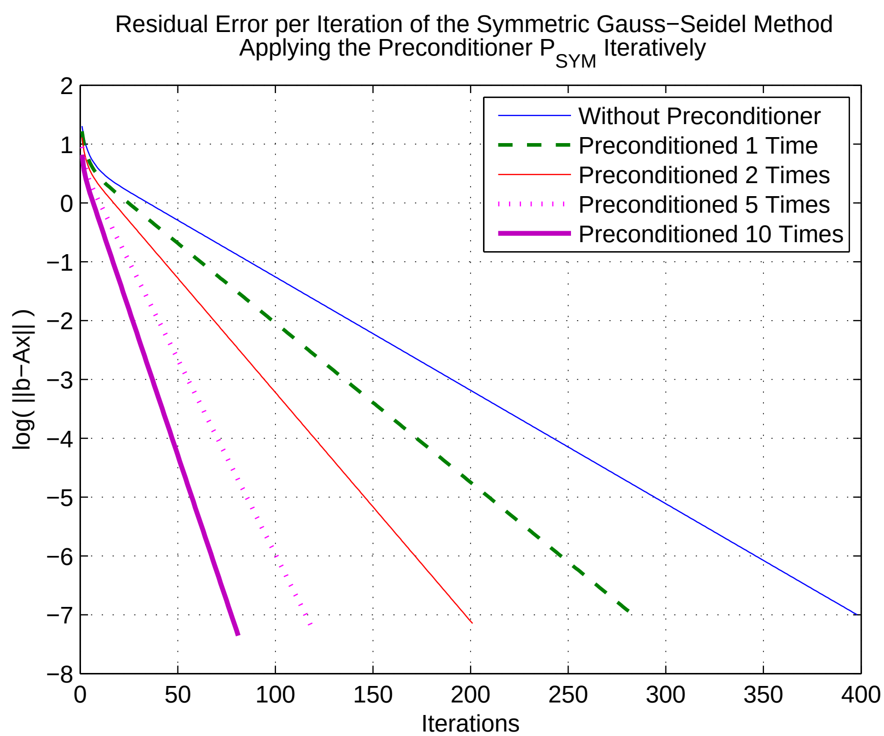
Times Preconditioned	$\rho(M^{-1}N)$ Applying $\tilde{P}$	$\rho(M^{-1}N)$ Applying P_{SYM}
0	0.9714	0.9714
1	0.9606	0.9518
2	0.9443	0.9121
5	0.9275	0.8466
10	0.8884	0.7421

Table 4.12: Spectral Radius of $(M^{-1}N)$ Applying the Preconditioners P_{SYM} and $\tilde{P}$ Iteratively on the Two-Dimensional Laplacian Matrix.

As can be seen, when P_{SYM} is applied iteratively, $\rho(M^{-1}N)$ decreases faster than it does when $\tilde{P}$ is applied. Nevertheless, the spectral radius does not reduce as fast as in the previous example (See Table 4.2.1).

4.2.3 Three-Dimensional Laplacian Matrix

Consider the hypothesis given in Section 4.1.4, but this time for $k = 20$ only. The next graphs show the residual error of the Symmetric Gauss-Seidel method when P_{SYM} and $\tilde{P}$ are applied iteratively, respectively. As it can be observed, the reduction of the number of iterations until convergence for this case is not that significant, but to begin with, the Symmetric Gauss-Seidel method did not take many iterations to converge as in the previous cases. Nevertheless, there is a reduction in the number of iterations.



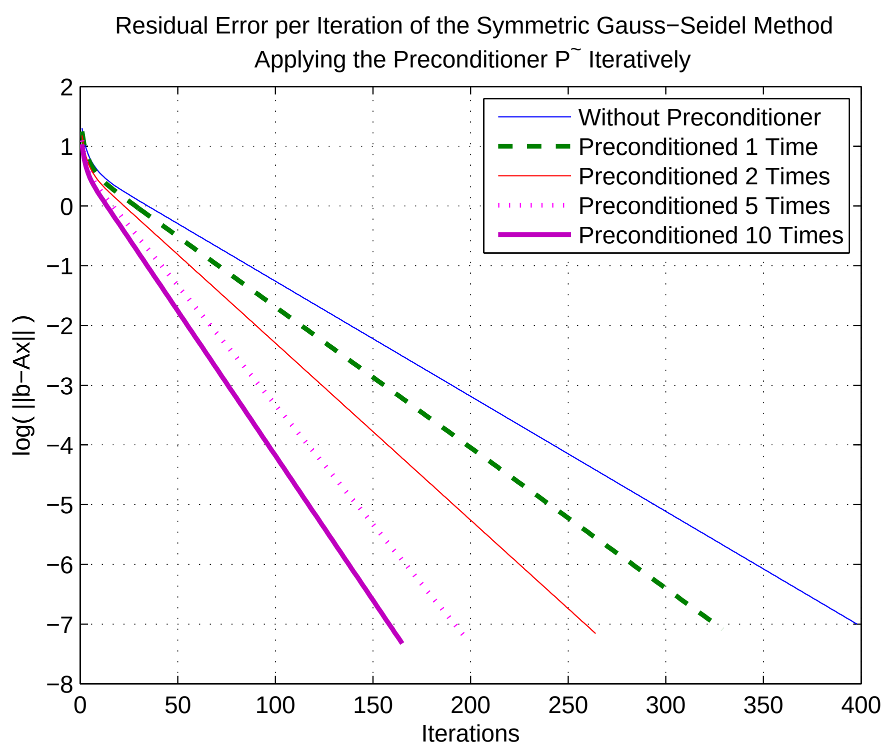


Figure 4.7: Residual Error of the Symmetric Gauss-Seidel method applying P_{SYM} and $\tilde{P}$ iteratively to the Three-Dimensional Laplacian matrix, respectively.

The following table shows how the spectral radius of $(M^{-1}N)$ reduces when each preconditioner is applied iteratively. M and N came from the partition of the Symmetric Gauss-Seidel method as previously defined.

Times Preconditioned	$\rho(M^{-1}N)$ Applying $\tilde{P}$	$\rho(M^{-1}N)$ Applying P_{SYM}
0	0.9566	0.9566
1	0.9472	0.9395
2	0.9340	0.9144
5	0.9120	0.8583
10	0.8944	0.7962

Table 4.13: Spectral Radius of $(M^{-1}N)$ Applying the Preconditioners P_{SYM} and $\tilde{P}$ Iteratively on the Three-Dimensional Laplacian Matrix.

As in the previous examples, the spectral radius reduces when the preconditioners are applied iteratively. When P_{SYM} is applied, a better spectral radius of $M^{-1}N$ is obtained for the Symmetric Gauss-Seidel method.

4.2.4 Finite Volume Matrices (Random Materials)

Consider the problem in Section 4.1.5, but this time for $k = 80$ only. The next graphs show the residual error in the Symmetric Gauss-Seidel method when P_{SYM} and $\tilde{P}$ are applied iteratively, respectively. As it can be observed, when P_{SYM} is applied 5 times the method converges. This does not happens when $\tilde{P}$ is applied. Even when $\tilde{P}$ is applied 25 times the method does not yield convergence.

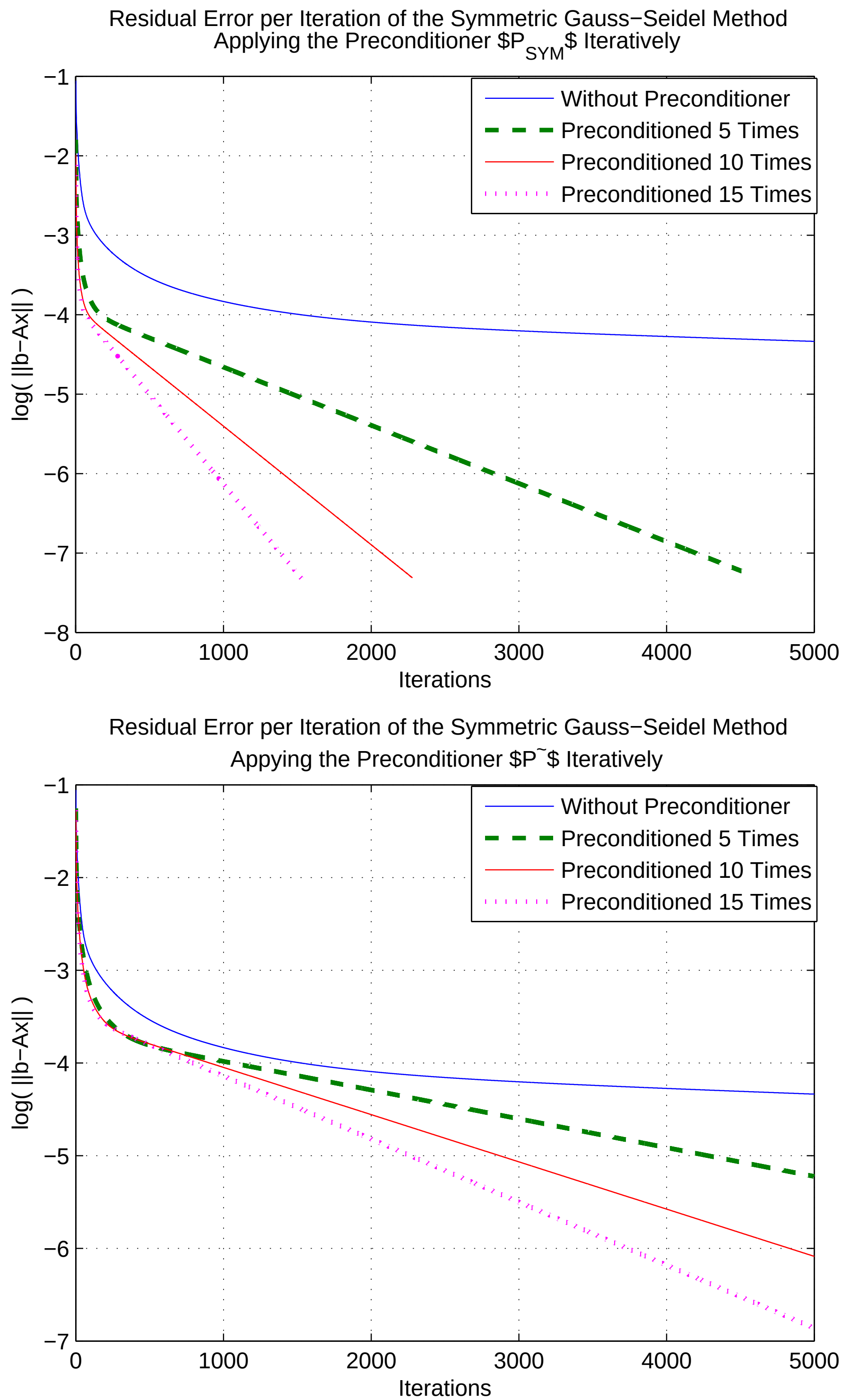


Figure 4.8: Residual Error of the Symmetric Gauss-Seidel method applying P_{SYM} and $\tilde{P}$ iteratively to a Finite Volume Matrix.

The following table shows how the spectral radius of $(M^{-1}N)$ reduces when each preconditioner is applied iteratively. M and N came from the partition of the Symmetric Gauss-Seidel method as previously defined.

Times Preconditioned	$\rho(M^{-1}N)$ Applying $\tilde{P}$	$\rho(M^{-1}N)$ Applying P_{SYM}
0	0.9998	0.9998
1	0.9997	0.9996
5	0.9992	0.9983
10	0.9988	0.9965
15	0.9984	0.9948

Table 4.14: Spectral Radius of $(M^{-1}N)$ Applying the Preconditioners P_{SYM} and $\tilde{P}$ Iteratively on the Three-Dimensional Laplacian Matrix.

As in the previous examples, the spectral radius reduces when the preconditioners are applied iteratively. When P_{SYM} is applied, a better spectral radius of $M^{-1}N$ is obtained for the Symmetric Gauss-Seidel method.

4.2.5 Finite Volume Matrices (Sand and Shales Problem)

This is a classical benchmark porous media problem proposed by J.L. Durlofsky in [3]. The permeability field consists of only two materials, sand and shales; and is defined on a grid consisting of 20×20 cells. Figure 4.9 illustrate the distribution of the permeability field.

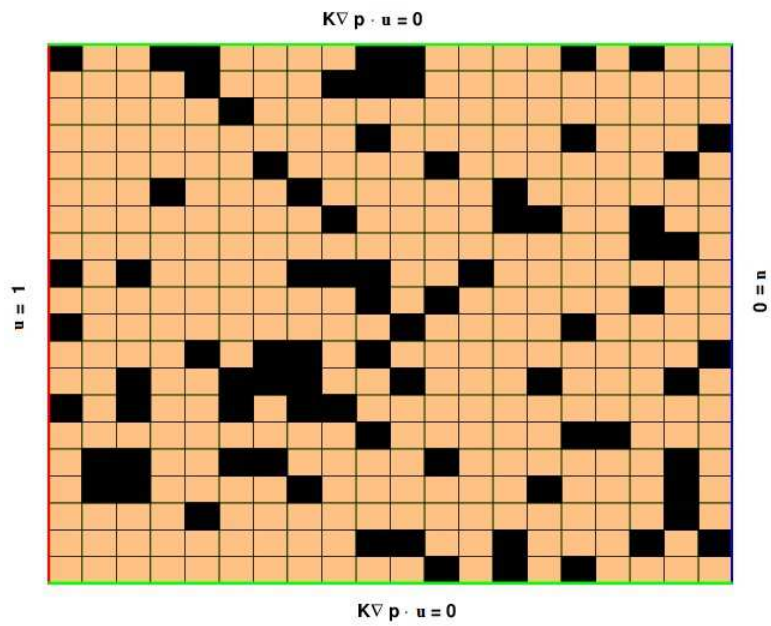


Figure 4.9: Representative Grid for the Sand and Shales Problem with Boundary Conditions.

Dark cells correspond to a permeability tensor $\mathcal{K} = 10^{-6} I_d$ and light cells to $\mathcal{K} = I_d$. Dirichlet boundary conditions are imposed at x = 0 where u = 1, and at x = 1, where u = 0. No flow boundary conditions are imposed at the top and the bottom.

This problem is considered refining the original mesh three times. To solve the systems two methods were considered: Gauss-Seidel, and Symmetric Gauss-Seidel. A maximum of $5,000$ iterations, a tolerance of 10^{-7}, and a starting value of $x = [0, 0, \ldots, 0, 0]$ were used for each method.

Number of preconditioning steps for P_{SYM}

cells	0	1	5	10	15	20	25
20×20	469	285	77	44	31	25	21
40×40	2161	1320	362	195	136	105	86
80×80	5000	5000	1474	803	564	436	354
160×160	5000	5000	5000	3143	2234	1750	1431

Table 4.15: Iterations until Convergence for the Symmetric Gauss-Seidel Method, applying the Preconditioner P_{SYM} Iteratively to the Sand and Shales Problem.

Number of preconditioning steps for $\tilde{P}$

cells	0	1	5	10	15	20	25
20×20	469	340	169	112	91	78	69
40×40	2161	1576	794	512	415	349	310
80×80	5000	5000	3220	2099	1715	1449	1301
160×160	5000	5000	5000	5000	5000	5000	5000

Table 4.16: Iterations until Convergence for the Symmetric Gauss-Seidel Method, applying the Preconditioner $\tilde{P}$ Iteratively to the Sand and Shales Problem.

Tables 4.2.5 and 4.2.5 show how both preconditioners reduce the number of iterations until convergence of the SG-S method when they are applied iteratively. However, when P_{SYM} is applied there is a further reduction on the number of iterations. When P_{SYM} is applied five times produce similiar results to that of $\tilde{P}$ applied twenty times. Analizing both preconditioners in the same preconditioning step, in most of the cases, the number of iterations until convergence when $\tilde{P}$ is applied is more than two times the number of iterations that P_{SYM} produces when applied in the same step.

The following tables show the same experiment but using the Gauss-Seidel method.

Number of preconditioning steps for P_{SYM}

cells	0	1	5	10	15	20	25
20×20	759	438	119	66	46	36	29
40×40	3338	1957	541	292	202	155	126
80×80	5000	5000	2135	1169	818	632	512
160×160	5000	5000	5000	4393	3118	2442	1992

Table 4.17: Iterations until Convergence for the Gauss-Seidel Method, applying the Preconditioner P_{SYM} Iteratively to the Sand and Shales Problem.

Number of preconditioning steps for $\tilde{P}$

cells	0	1	5	10	15	20	25
20×20	759	478	198	119	95	79	69
40×40	3338	2106	898	526	415	341	299
80×80	5000	5000	3521	2074	1646	1359	1208
160×160	5000	5000	5000	5000	5000	5000	4589

Table 4.18: Iterations until Convergence for the Gauss-Seidel Method, applying the Preconditioner $\tilde{P}$ Iteratively to the Sand and Shales Problem.

Tables in 4.2.5 and 4.2.5 show that when P_{SYM} is applied the number of iterations until convergence is lower than those obtained when $\tilde{P}$ is applied. The top performance of $\tilde{P}$ for this experiment is when it is used in with the Gauss-Seidel method. When P_{SYM} is applied it top performance is when it is used with the Symmetric Gauss-Seidel. Comparing both preconditioners at their best one can see that P_{SYM} produces better results. When the number of preconditioned steps is 25, P_{SYM} reduces the number of iterations until convergence found when $\tilde{P}$ was applied by a factor of 3.

The next set of tables show the behaviour of the spectral radius of $M^{-1}N$ for both methods when both preconditioners are applied iteratively. Tables 4.2.5 and 4.2.5 show the behaviour of the spectral radius of $M^{-1}N$ for the Symmetric Gauss-Seidel and tables 4.2.5 and 4.2.5 does it for the Gauss-Seidel. As can be seen for both methods there is a reduction on the spectral radius of $M^{-1}N$ and this reduction is greater when P_{SYM} is applied.

Number of preconditioning steps for P_{SYM}

cells	0	1	5	10	15	20	25
20×20	0.9744	0.9573	0.8452	0.7387	0.6523	0.5787	0.5143
40×40	0.9947	0.9912	0.9670	0.9382	0.9116	0.8856	0.8607
80×80	0.9987	0.9979	0.9923	0.9855	0.9791	0.9728	0.9664

Table 4.19: (SG-S) Spectral Radius Applying P_{SYM} to the Durlofsky Problem

Number of preconditioning steps for $\tilde{P}$

cells	0	1	5	10	15	20	25
20×20	0.9744	0.9641	0.9265	0.8894	0.8651	0.8429	0.8240
40×40	0.9947	0.9926	0.9849	0.9762	0.9705	0.9647	0.9601
80×80	0.9987	0.9982	0.9964	0.9945	0.9932	0.9919	0.9909

Table 4.20: (SG-S) Spectral Radius Applying $\tilde{P}$ to the Durlofsky Problem

Number of preconditioning steps for P_{SYM}

cells	0	1	5	10	15	20	25
20×20	0.9865	0.9763	0.9120	0.8433	0.7819	0.7242	0.6696
40×40	0.9972	0.9952	0.9821	0.9662	0.9510	0.9358	0.9207
80×80	0.9993	0.9989	0.9959	0.9923	0.9889	0.9855	0.9820

Table 4.21: (G-S) Spectral Radius Applying P_{SYM} to the Durlofsky Problem

Number of preconditioning steps for $\tilde{P}$

cells	0	1	5	10	15	20	25
20×20	0.9865	0.9782	0.9467	0.9116	0.8889	0.8675	0.8493
40×40	0.9972	0.9955	0.9892	0.9814	0.9761	0.9708	0.9666
80×80	0.9993	0.9989	0.9975	0.9957	0.9945	0.9933	0.9925

Table 4.22: (G-S) Spectral Radius Applying $\tilde{P}$ to the Durlofsky Problem

One last experiment was done with this problem. This one consist of exploring the non-zeroes of the coefficient matrix as the preconditioners are applied.

Number of preconditioning steps for P_{SYM}

cells	1	5	10	15	20	25
20×20	1.81	18.65	61.83	72.74	72.78	72.73
40×40	1.88	20.72	94.08	204.50	287.33	306.24
80×80	1.91	20.81	109.18	289.68	547.68	834.78
160×160	1.89	20.61	110.49	316.312	678.73	1200.67

Table 4.23: Non-zeros coefficients for the Durlofsky's problem when P_{SYM} is applied.

Number of preconditioning steps

cells	1	5	10	15	20	25
20×20	1.36	6.07	17.22	27.48	31.66	33.70
40×40	1.38	6.59	22.34	48.77	78.46	103.22
80×80	1.39	6.94	24.39	58.01	107.75	173.87
160×160	1.39	7.02	24.84	61.15	119.50	208.19

Table 4.24: Non-zeros coefficients for the Durlofsky's problem when $\tilde{P}$ is applied.

As one can see the non-zeroes are increasing faster when P_{SYM} is applied. This behavior will change eventually since it was shown that there exists a $k \in \mathbb{N}$ such that if we apply $PSYM$ k times to the coefficient matrix it will reduce to a diagonal. In the first case one can see that after P_{SYM} is applied 15 times the non-zeroes begin to stabilize.

4.2.6 Other Matrices

This next matrix comes from the Local Discontinuous Galerkin method studied by Castillo and Sequeira in [2] and in [20]. This one is a sparse matrix of real entries, with dimension (5380×5380), Symmetric and Positive Definite. A maximum of $5,000$ iterations, a tolerance of 10^{-7}, and a starting value of $x = [0, 0, \ldots, 0, 0]$ were used for the Symmetric Gauss-Seidel method.

The following table shows how the spectral radius of $(M^{-1}N)$ reduces when each preconditioner is applied iteratively. M and N came from the partition of the Symmetric Gauss-Seidel method as defined in Section 2.4.

Times Preconditioned	$\rho(M^{-1}N)$ Applying $\tilde{P}$	$\rho(M^{-1}N)$ Applying P_{SYM}
0	0.9870	0.9870
1	0.9776	0.9728
2	0.9716	0.9660
5	0.9561	0.9504
10	0.9260	0.9202

Table 4.25: Spectral Radius of $(M^{-1}N)$ Applying the Preconditioners P_{SYM} and $\tilde{P}$ Iteratively on a Matrix from the LDG Method.

As in the previous examples, the spectral radius reduce when the preconditioners were applied iteratively. There was not a significant difference between the spectral radius when each preconditioner was applied iteratively. The next graphs show the residual error of the Symmetric Gauss-Seidel method when P_{SYM} and $\tilde{P}$ are applied iteratively, respectively. As it can be seen, applying any of the two preconditioners will yield less iterations in the convergence of the Symmetric Gauss-Seidel method.

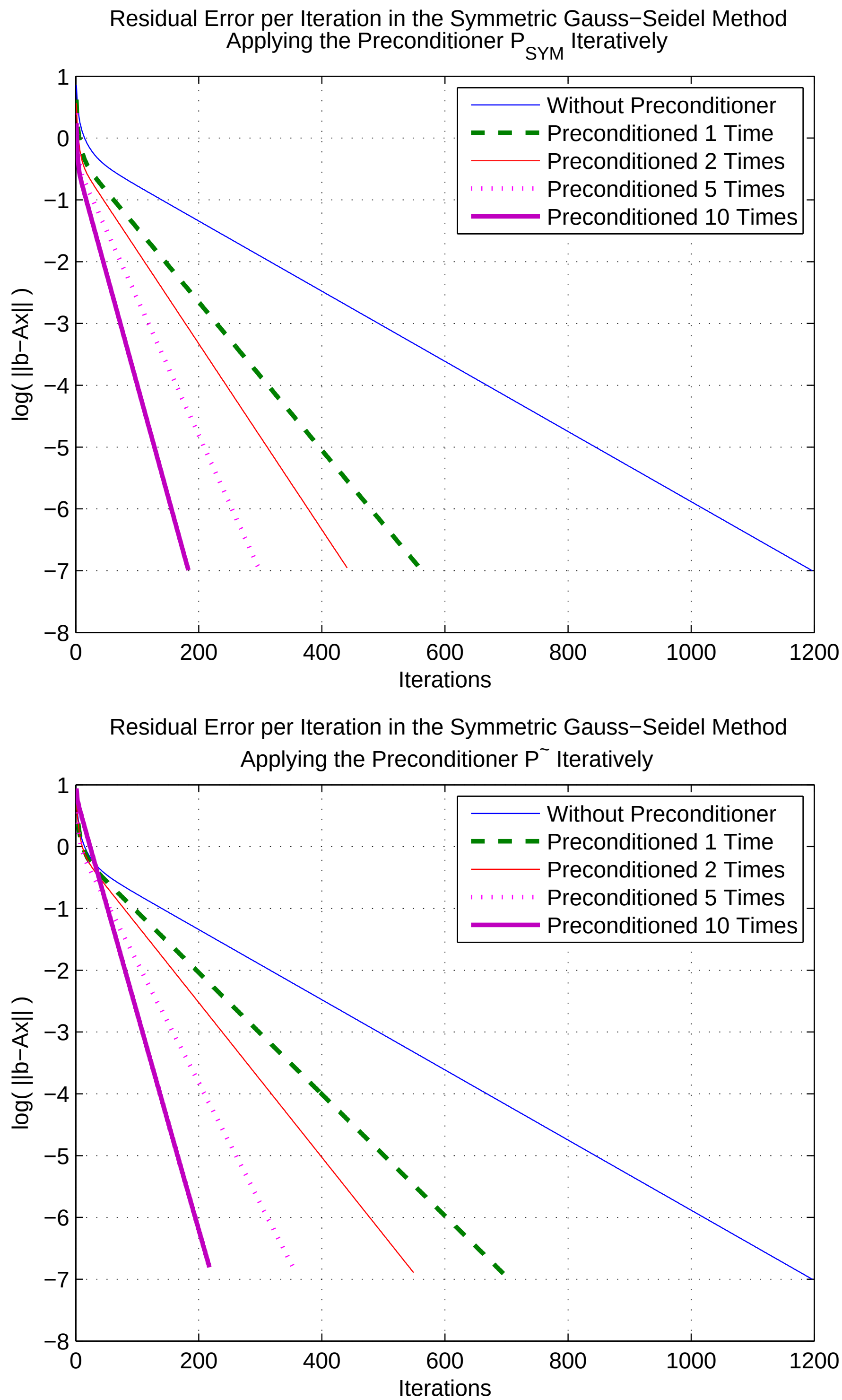


Figure 4.10: Residual Error of the Symmetric Gauss-Seidel method applying P_{SYM} and $\tilde{P}$ iteratively, respectively.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

- A new preconditioner, named P_{SYM}, that preserves symmetry is proposed.
- Numerical experiments show that the preconditioner improves the convergence of the Symmetric Gauss-Seidel method.
- The preconditioner was originally designed for diagonally dominant symmetric Z-Matrices, but experiments show a good performance on more general symmetric positive definite matrices.
- Experiments show that the preconditioner can be applied iteratively for better results.

- Numerical results shows that the spectral radius of $M^{-1}N$ on the Symmetric Gauss-Seidel and the Gauss-Seidel methods reduces after applying the preconditioner.

5.2 Future Work

- A formal proof to show that actually the spectral radius, reduces needs to be given. (Conjecture 3.1)
- A formal proof to show that when the preconditioner is applied to the given matrix the resulting one is a Z-Matrix and diagonally dominant must be given. (Conjecture 3.2)
- The computational cost of applying P_{SYM} can be analyzed.
- An upper bound on how many times P_{SYM} needs to be applied to obtain a diagonal matrix can be investigated.

BIBLIOGRAPHY

[1] Isnardo Arenas Navarro, *A new preconditioner for solving linear systems with ill-conditioned z-matrices*, University of Puerto Rico, Puerto Rico, 2011.

[2] P. Castillo and F. A. Sequeira, *Computational aspects of the Local Discontinuous Galerkin method on unstructured grids in 3D*, Mathematical and Computer Modelling **doi:10.1016/j.mcm.2011.07.032** (2011).

[3] Louis J Durlofsky, *Accuracy of mixed and control volume finite element approximations to darcy velocity and related quantities*, Water Resources Research **30** (1994), no. 4, 965.

[4] Gene H. Golub and Charles F. Van Loan, *Matrix computations*, 3rd ed., The Johns Hopkins University Press, 1996.

[5] Ananda D. Gunawardena, S.K. Jain, and Larry Snyder, *Modified iterative methods for consistent linear systems*, Linear Algebra and its Applications **154-156** (1991), 123 – 143.

[6] Roger A. Horn and Charles R. Johnson, *Matrix analysis*, repr. with corr. ed., Cambridge University Press, Cambridge [Cambridgeshire] ; New York, 1985 (English).

[7] Toshiyuki Kohno, Hisashi Kotakemori, Hiroshi Niki, and Masataka Usui, *Improving the modified Gauss-Seidel method for Z-matrices*, Linear Algebra and its Applications **267** (1997), 113 – 123.

[8] Toshiyuki Kohno and Hiroshi Niki, *A note on the preconditioner* $P_M = (I + S_M)$, Journal of Computational and Applied Mathematics **225** (2009), no. 1, 316 – 319.

[9] Hisashi Kotakemori, Kyouji Harada, Munenori Morimoto, and Hiroshi Niki, *A comparison theorem for the iterative method with the preconditioner* $(I + S_{max})$, Journal of Computational and Applied Mathematics **145** (2002), no. 2, 373 – 378.

[10] Hisashi Kotakemori, Hiroshi Niki, and Naotaka Okamoto, *Accelerated iterative method for Z-matrices*, Journal of Computational and Applied Mathematics **75** (1996), no. 1, 87 – 97.

[11] Ji-cheng Li and Wei Li, *The optimal preconditioner of strictly diagonally dominant z-matrix*, Acta Mathematicae Applicatae Sinica (English Series) **24** (2008), 305–312, 10.1007/s10255-006-6148-5.

[12] J.A. Meijerink and H.A¿ Van Der Vorst, *An iterative solution method for linear systems of wich the coefficient matrix is a symmetric m-matrix*, Mathematics of Computation **31** (1977), no. 137, 148 – 162.

[13] Carl D. Meyer (ed.), *Matrix analysis and applied linear algebra*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[14] J. P. Milaszewicz, *Improving Jacobi and Gauss-Seidel iterations*, Linear Algebra and its Applications **93** (1987), 161 – 170.

[15] Munenori Morimoto, Kyouji Harada, Michio Sakakihara, and Hideo Sawami, *The Gauss-Seidel iterative method with the preconditioning matrix* $(I + S + S_M)$, Japan Journal of Industrial and Applied Mathematics **21** (2004), 25–34, 10.1007/BF03167430.

[16] Hiroshi Niki, Kyouji Harada, Munenori Morimoto, and Michio Sakakihara, *The survey of preconditioners used for accelerating the rate of convergence in the Gauss-Seidel method*, Journal of Computational and Applied Mathematics **164-165** (2004), 587 – 600, Proceedings of the 10th International Congress on Computational and Applied Mathematics.

[17] D. Noutsos and M. Tzoumas, *On optimal improvements of classical iterative schemes for z-matrices*, J. Comput. Appl. Math. **188** (2006), 89–106.

[18] Anna Pyzara, Beata Bylina, and Jaroslaw Bylina, *The influence of a matrix condition number on iterative methods convergence.*, FedCSIS'11, 2011, pp. 459–464.

[19] Y. Saad, *Iterative methods for sparse linear systems, 2nd edition*, SIAM, Philadelpha, PA, 2003.

[20] F.A. Sequeira, *Aspectos computacionales del método Local Discontinuous Galerkin para mallas no estructuradas en 3D*, Master's thesis, University of Puerto Rico at Mayagüez, 2010.

[21] Hailong Shen, Xinhui Shao, Zhenxing Huang, and Chunji Li, *Preconditioned Gauss-Seidel iterative method for Z-matrices linear systems*, Bulletin of the Korean Mathematical Society **48** (2011), no. 2, 303 – 314.