KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

IMPROVING THE SOLVABILITY OF ILL-CONDITIONED SYSTEMS OF LINEAR EQUATIONS BY REDUCING THEIR CONDITION NUMBERS OF THEIR MATRICES.

By

MAHAMA, ABDUL SALAM

A THESIS SUBMITTED TO THE DEPARTMENT OF MATHEMATICS, KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE INDUSTRIAL MATHEMATICS

October 12, 2015

DECLARATION

I hereby declare that this submission is my own work towards the award of the M.Sc degree and that, to the best of my knowledge, it contains no material previously published by another person nor material which had been accepted for the award of any other degree of the university, except where due acknowledgement had been made in the text.

Mahama Abdul Salam	.....	.....
PG No: 20092255	Signature	Date

Certified by:

Mr. Kwaku Darkwah	.....	.....
Supervisor	Signature	Date

Certified by:

Prof. S.K Amponsah	.....	.....
Head of Department	Signature	Date

ABSTRACT

This thesis is concerned with the solution of a canonical example of ill-conditioned system called Hilbert Systems of Linear Equations (HSLE's) via the solution of an equivalent/transformed HSLE's which are well-conditioned. A matrix is first constructed from that of the given ill-conditioned system. Then, an adequate right-hand side is computed to make up the instance of an equivalent system. Formulae and algorithms for computing an instance of this equivalent HSLE and solving it will be given and illustrated. Analysis is made between the original Hilbert system and its equivalent/transformed system. Under original Hilbert system comparison is made between unperturbed and perturbed Hilbert system and under the equivalent/transformed Hilbert system comparison is made between unperturbed and perturbed transformed Hilbert system. The results established the fact that well conditioned solutions are more accurate and reliable than ill conditioned solutions due to their error margins and condition numbers.

DEDICATION

I dedicate this thesis to the Glory and Blessings of ALLAH.

ACKNOWLEDGMENT

I thank the almighty ALLAH for endowing in me everything to complete this academic work. My mother Mrs. Hajia Amamata Mahama - I will forever be indebted for your care, my wife Afisat; I say thank you for playing significant roles in my life. I also appreciate the role of my supervisor Mr. Kwaku Darkwah for taking his time to supervise me.

Contents

Chapter 1

INTRODUCTION

1.1 Background

Numerical stability is a desirable property of numerical algorithms. The precise definition of stability depends on the context, but it is derived from the accuracy of the algorithm. Sometimes a single calculation can be achieved in several ways, all of which are algebraically equivalent in terms of ideal real or complex numbers, but in practice when performed on digital computers yield different results. Some calculations might damp out approximation errors that occur; others might magnify such errors. Calculations that can be proven not to magnify approximation errors are called numerically stable. One of the common tasks of numerical analysis is to try to select algorithms which are robust - that is to say, have good numerical stability among other desirable properties.

An opposite phenomenon is instability. Typically, algorithms would approach the right solution in the limit, if there were no round-off or truncation errors, but depending on the specific computational method, errors can be magnified, instead of damped, causing the error to grow exponentially.

There are three central concepts in the analysis of numerical techniques and these are *Convergence* (whether the method approximates the solution), *Order* (how well it approximates the solution) and *Stability* (whether errors are damped out). The latter is the underpinning of this academic work.

One of the fundamental problems in many scientific and engineering applications is to solve an algebraic linear system $Ax = b$ for the unknown vector x when the coefficient matrix A and right-hand side vector b are known. Such systems arise naturally in various applications and one of the most popular tech-

niques for solving such systems is the Gaussian elimination method (also known as row reduction) which is a numerical algorithm for solving systems of linear equations. The approach is designed to solve a general set of n equations and n unknowns.

Two separate issues of stability in terms of accuracy in solving linear systems namely pivoting and condition number are studied in this thesis. The first, pivoting is a method that ensures that Gaussian elimination proceeds as accurately as possible; this can either be partial or complete pivoting. There are two pitfalls of the Gaussian elimination method namely: round-off errors (attributed to how computers store numbers as a finite strings of binary floating digits by truncating digits) and division by zero. Gaussian elimination can involve hundreds of arithmetic computations with the use of a digital computer, each of which can produce rounding error. When floating point arithmetic is used (In computing, floating point describes a method of representing an approximation of a real number in a way that can support a wide range of values. The numbers are, in general, represented approximately to a fixed number of significant digits, the *mantissa* and scaled using an exponent. The base for the scaling is normally 2, 10 or 16. The typical number that can be represented exactly is of the form: $\pm M * 10^k$ where k is an integer and the mantissa M satisfies the inequality $0.1 \leq M < 1$. Wikipedia (2014). Such large row multipliers tend to propagate rounding error. This type of error propagation can be lessened by appropriate row and or column interchanges that produce smaller multipliers by the use of partial and complete pivoting techniques.

It is well known that even for a nonsingular and well conditioned input matrix,Gaussian elimination fails in numerical computations with rounding errors as soon as it encounters a vanishing or nearly vanishing leading entry. In practice users avoid such encounters by applying pivoting strategies; partial and complete pivoting, that is an appropriate row and column interchanges, however, these takes its toll: pivoting usually degrades the performance. It interrupts the stream

of arithmetic operations with foreign operations of comparison, involves bookkeeping, compromises data locality, increases communication overhead and data dependence, and tends to destroy matrix structure. Pan et al. (2013).

In this thesis however, the efficacy of Gaussian elimination with no pivoting will be used to compute the solutions of Hilbert linear equations which will be transformed into an improved systems that are better conditioned using a direct method called the general approach algorithm. Gaussian elimination with no pivoting is used because ill conditioned systems are extremely sensitive to numerical errors and as such pivoting is not much of a help. In fact, Gaussian elimination with no pivoting is considered to be a stable method in practice.

Another facet of stability of an algorithm is that which gives the exact answer to a problem that is near to the original problem. Such algorithm is said to be backward stable. Algorithms that are not backward stable will tend to amplify roundoff errors present in the original data and make it inaccurate and instable. As a result, the solution produced by an algorithm that is not backward stable will not necessarily be the solution to a problem that is close to the original problem. Gaussian elimination with partial pivoting is said to be backward stable. If A is symmetric and positive definite, then Gaussian elimination without pivoting is also backward stable. Olson (2009).

The second, condition number, is a measure of how bad a matrix is. In other words it is the sensitivity of the solution with respect to errors in the data and it determines the loss in precision due to roundoff errors in Gaussian elimination and can be used to estimate the accuracy of results obtained from matrix inversion and linear equation solution. Ill condition means the solution of a system is unstable with respect to small changes in data and well condition also means the solution of a system is stable with respect to small changes in the data. If the condition number is close to one, the matrix is well conditioned which means its inverse can be computed with good accuracy. If the condition number is large, then the matrix is said to be ill-conditioned. Practically, such

a matrix is almost singular, and the computation of its inverse, or solution of a linear system of equations is prone to large numerical errors and the remedy is to resort to iterative techniques to avoid error. A matrix that is not invertible has the condition number equal to infinity.

Every problem that we try to solve is based on an expression of some form or another. To have confidence in the solution it is important to know that the expression is well conditioned, so that we would not get completely different results from slight changes in the input. If it is well-conditioned, a small change in the coefficient matrix or a small change in the right hand side results in a small change in the solution vector and if it is ill-conditioned a small change in the coefficient matrix or a small change in the right hand side results in a large change in the solution vector. The exact cutoff between well- and ill-conditioned depends on the context of the problem and the uses of the results.

The interest of an algorithm is the same as for an expression: it is desirable to have small changes in the input to only produce small changes in the output. An algorithm or numerical process is called stable if this is true and it is called unstable if large changes in the output are produced. Analyzing an algorithm for stability is more complicated than determining the condition of an expression, even if the algorithm simply evaluates the expression. This is because an algorithm consists of many basic calculations and each one must be analyzed and, due to roundoff error, it is necessary to consider the possibility of small errors being introduced in every computed value.

As Well-and Ill-conditioned refers to the problem; Stable/Unstable refer to an algorithm or the numerical process. If a problem is well-conditioned then there is a stable way to solve it and if the problem is ill-conditioned then it is difficult to solve it in a stable way. The difficulty has to be negotiated. Ill-conditioned systems of linear equations are notoriously difficult to solve to any useful accuracy. Their matrices are characterized by large condition numbers. Mixing roundoff-error with an unstable process is a recipe for disaster. With

exact arithmetic (no roundoff-error), stability is not a concern. Hence, even when a problem is well-conditioned, solving it with an unstable algorithm, the obtained results will be meaningless. (Farooq and Salhi, 2011).

1.1.1 Problem Statement

Hilbert systems are ill conditioned which have large condition numbers and are very sensitive to small changes in input data, resulting in a large change in the solution vector. Thus their solutions are unreliable, unstable and cannot be trusted to any degree of accuracy and to have any level of confidence in their solutions it is necessary to reduce the ill conditioned by solving an equivalent system in order to obtain a relatively stable and an improved solution closer to the exact solution.

1.2 Objectives of the study

1. To convert ill condition Hilbert system to an improved system.
2. To compute the solution of ill conditioned Hilbert system via the solution of an equivalent improved system.
3. To compare the relative stability and reliability of the ill conditioned Hilbert system and the improved system.

1.3 Methodology

Two methods are normally used for solving linear systems computationally namely direct and iterative methods. The direct methods consist of a finite number of steps that all must be performed for any given method before the solution is obtained. The basic idea behind all the direct methods is first to reduce the linear system $Ax = b$ to an equivalent triangular systems by finding triangular factors of the matrix A and then to solve the triangular system, which is much easier to

solve than the original problem. Some examples of the direct methods include the following: Gaussian elimination, QR factorization, Cholesky factorization.

On the other hand, iterative methods are based on computing a sequence of approximations to the solution x, and a user can stop whenever a certain desired accuracy is obtained or a certain number of iterations are completed. The iterative methods are used primarily for large and sparse systems. Some of the iterative methods include the following: Jacorbi method, Gauss-Seidel method, Successive over-relaxation method, Conjugate gradient method and General Minimal Residual method.

This thesis is a theoretical academic work and the matrix used for the analysis is a well-known ill conditioned matrix called the Hilbert matrix which is a square matrix $n \times n$ with entries being the unit fractions and it is characterize by large condition numbers. Gaussian elimination with no pivoting is considered as a stable method in practice and this motivates me to use it. A new system is constructed from the Hilbert matrix with a suitable right hand side, and it is expected that this transformed Hilbert system would have a better condition number and its result more accurate due to its relatively small error margin.

In the quest to achieve the objectives of this work, numerous research works, papers and articles both published and unpublished have been extensively scrutinize and the pertinent literatures drawn from it.

1.4 Justification of Work

The justifications and benefits of this thesis are amongst the following:

1. This study will help university lecturers to use the concept and approaches in the teaching of numerical errors.

2. It will serve as an introductory step for university students to develop interest in working at the topic.

3. It will also serve as a curricular material in which recommendations can be

made to enhance and broaden the horizons of teaching and learning of the research topic.

4. It will serve as a base or a reference material for other concerned students and researches to dive into the problem for onward suggestions and recommendations.

1.5 Thesis Organization

The thesis is organized into five chapters; each chapter is distinct from the other. Chapter one is the introduction of the study that comprise the following: Introduction, Problem statement, Objectives of the study, Methodology employed, Justification and thesis organization. Chapter two is the literature review that outlines the body of published work concerned with this particular thesis and in the quest to achieve the objectives of this work, numerous research works, papers and articles have been extensively scrutinize and the pertinent literatures have been reviewed. Chapter three is the methodology in which some theories about the thesis are presented in a methodical and organized manner. The fourth chapter concentrates on the analysis and discussions of the findings of the study. The final and fifth chapter is the conclusion and recommendation which is made up of summing up of all the points and a statement of opinion or decisions reached about the thesis.

Chapter 2

LITERATURE REVIEW

2.1 Literature Review

In the quest to achieve the objectives of this work, numerous research works, papers and articles have been extensively scrutinize and the pertinent literatures have been reviewed as follows:

Pan and Qian (2012) explains that a random matrix is likely to be well conditioned, and motivated by this well known property they employ random matrix multipliers to advance some fundamental matrix computations. This includes numerical stabilization of Gaussian elimination with no pivoting as well as block Gaussian elimination, approximation of the leading and trailing singular spaces of an ill conditioned matrix, associated with its largest and smallest singular values, respectively, and approximation of this matrix by low-rank matrices, with further extensions to Tensor Train approximation and the computation of the numerical rank of a matrix. The authors also formally support the efficiency of the proposed techniques where they employ Gaussian random multipliers, but their extensive tests have consistently produced the same outcome where instead they used sparse and structured random multipliers, defined by much fewer random parameters compared to the number of their entries.

Pan et al. (2013) proved that standard Gaussian random multipliers are expected to stabilize numerically both Gaussian elimination with no pivoting and block Gaussian elimination. The authors also explained that Gaussian elimination fails in numerical computations with rounding errors as soon as it encounters a vanishing or nearly vanishing leading (that is north- western) entry and they avoided such encounters by applying Gaussian elimination with partial pivot-

ing and has some limited formal but ample empirical support. Their tests show similar results where the authors applied circulant random multipliers instead of Gaussian ones.

Higham and Higham (1989) looked at how growth factor plays an important role in the error analysis of Gaussian elimination. The authors put forward that it is a fact when partial pivoting or complete pivoting is used the growth factor is usually small, but it can be large. The examples of large growth usually quoted involve contrived matrices that are unlikely to occur in practice. They present real and complex n x n matrices arising from practical applications that, for any pivoting strategy, yield growth factors bounded below by $n/2$ and n, respectively. These matrices enable the authors to improve the known lower bounds on the largest possible growth factor in the case of complete pivoting. For partial pivoting, the authors classify the set of real matrices for which the growth factor is 2^{n-1}. Finally, they show that large element growth does not necessarily lead to a large backward error in the solution of a particular linear system, and they commented on the practical implications of this result.

Higham (2009) explored the works done by Wilkinson who put Gaussian elimination on a sound numerical footing in the 1960's when he showed that with partial pivoting the method is stable in the sense of yielding a small backward error . He also derived bounds proportional to the condition number $\kappa(A)$ for the forward error $\| x - \hat{x} \|$, where $\hat{x}$ is the computer solution to $Ax = b$. More recent work has furthered the understanding of Gaussian Elimination, largely through the use component wise rather than norm wise analysis. The author of this paper surveyed what is known about the accuracy of Gaussian Elimination in both the forward and the backward error senses. Particular topics include: classes of matrix for which it is advantages not to pivot; how to estimate or compute the backward error; iterative refinement in single precision; and how to compute efficiently a bound on the forward error.

Skeel (1980) depicts that Gaussian elimination with pivoting is a stable algorithm for solving linear systems of equations in the sense that the computed solution exactly satisfies a linear system whose coefficient matrix differs slightly in norm from the given matrix. For this reason it is often thought that iterative refinement is not worthwhile unless either the data are known with great accuracy or one wishes to detect ill-conditioning. The author further explains that because of scaling problems, Gaussian elimination with pivoting is not always as accurate as one might reasonably expect. It is shown that stability is possible if an appropriate implicit scaling of the rows and/or columns is used with the pivoting. Unfortunately the proper scaling requires estimates of the solution components. It is shown that the effects of improper scaling can be eliminated by performing iterative refinement even if the residuals are not accumulated in double precision. Therefore, iterative refinement would be worthwhile for problems that may not be scaled properly for Gaussian elimination. The computational cost is often small, but this is not always true due to the necessity of storing the original matrix.

Mead et al. (2001) introduced variant form of Gaussian elimination with partial pivoting which is achieved by adding the pivot row to the k^{th} row at step k,. In their paper it is shown that the growth factor of this partial pivoting algorithm is bounded above by $\mu_n < 3^{n-1}$, as compared to 2^{n-1} for the standard partial pivoting. This bound μ_n, close to 3^{n-2} is attainable for a class of near-singular matrices. Moreover, for the same matrices the growth factor is small under partial pivoting.

Trefethen and Schreiber (1990) posit that Gaussian elimination with partial pivoting is unstable in the worst case: the *growth factor* can be as large as $2^n - 1$, where n is the matrix dimension, resulting in a loss of n bits of precision. It is proposed that an average-case analysis can help explain why it is nevertheless stable in practice. The results presented begin with the observation that for many distributions of matrices, the matrix elements after the first few steps of elimination are approximately normally distributed. From here, with the

aid of estimates from extreme value statistics, reasonably accurate predictions of the average magnitudes of elements, pivots, multipliers, and growth factors are derived. For various distributions of matrices with dimensions $n \leq 1024$, the average growth factor (normalized by the standard deviation of the initial matrix elements) is within a few percent of $n^{2/3}$ for partial pivoting and approximately $n^{1/2}$ for complete pivoting. The average maximum element of the residual with both kinds of pivoting appears to be of magnitude $O(n)$, as compared with $O(n^{1/2})$ for QR factorization. The experiments and analysis presented show that small multipliers alone are not enough to explain the average-case stability of Gaussian elimination; it is also important that the correction introduced in the remaining matrix at each elimination step is of rank 1. Because of this low-rank property, the signs of the elements and multipliers in Gaussian elimination are not independent, but are interrelated in such a way as to retard growth. By contrast, alternative pivoting strategies involving high-rank corrections are sometimes unstable even though the multipliers are small.

Foster (1994) probe that even though Gaussian elimination with partial pivoting is very widely used, $n \times n$ matrices can be constructed where the error growth in the algorithm is proportional to 2^{n-1}. Thus for moderate or large n, in theory, there is a potential for disastrous error growth. However, the author posits that prior to 1993 no reports of such an example in a practical application had appeared in the literature. Examples are presented that arise naturally from integral and differential equations and that lead to disastrous error growth in Gaussian elimination with partial pivoting. The author further presented a class of practical examples where the growth factors do grow exponentially. Volterra integral equations are considered and the growth factors of their matrices are closer to the theoretical limit and the results are apply to boundary value problem. Quadrature method is also used to numerically solve certain Volterra integral equations where large growth factors resulted.

Foster (1997) further indicates that Gaussian elimination is among the most widely used tools in scientific computing and Gaussian elimination with partial pivoting requires only $O(n^2)$ comparisons beyond the work required in Gaussian elimination with no pivoting but can, in principle, have error growth that is exponential in the matrix size n. Gaussian elimination with complete pivoting, on the other hand, cannot have exponential error growth but requires $O(n^2)$ comparisons beyond the work required by Gaussian elimination with no pivoting. Numerical experiments is conducted and it did suggest that Gaussian elimination with rook pivoting is between partial pivoting and complete pivoting in terms of efficiency and accuracy. In the paper it is proven that rook pivoting cannot have exponential error growth. The author introduce a combination of partial pivoting and rook pivoting and call it Gaussian elimination with partial rook pivoting and it is proven that partial rook pivoting cannot have exponential error growth and the numerical experiments showing that on a serial computer the run times for rook pivoting are almost always close to those of partial pivoting and the run times for partial rook pivoting appear to be the same as those of partial pivoting.

Cortes and Pena (2006) examine and compare several definitions of growth factors for Gaussian elimination some new pivoting strategies, intermediate between partial pivoting and rook pivoting, are introduced. For random matrices, an approximation of the average normalized growth factor associated with several pivoting strategies is computed and analyzed. A stationary behavior of the expected growth factors of the new pivoting strategies is observed. Bounds for the growth factors of these pivoting strategies are provided. It is also shown that partial pivoting by columns produces small growth factors for matrices appearing in practical observations and for which the growth factors produced by partial pivoting are very large.

Yeung and Chan (1997) explicate the numerical instability of Gaussian elimination is proportional to the size of the L and U factors that it produces. The

worst-case bounds are well known. For the case without pivoting, breakdowns can occur and it is not possible to provide a priori bounds for L and U. For the partial pivoting case, the worst-case bound is $O(2^m)$, where m is the size of the system. Yet these worst-case bounds are seldom achieved, and in particular Gaussian elimination with partial pivoting is extremely stable in practice. Surprisingly, there has been relatively little theoretical study of the *average* case behavior. The purpose of our paper is to provide a probabilistic analysis of the case without pivoting. The distribution we use for the entries of A is the normal distribution with mean 0 and unit variance. We first derive the distributions of the entries of L and U. Based on this, we prove that the probability of the occurrence of a pivot less than ϵ in magnitude is $O(\epsilon)$. We also prove that the probabilities Prob($\| U \|_\infty$ / $\| A \|_\infty > m^{2.5}$) and Prob($\| L \|_\infty > m^3$) decay algebraically to zero as m tends to infinity. Numerical experiments are presented to support the theoretical results.

Sankar (2004) presented a smoothed analysis of Gaussian elimination, both with partial pivoting and without pivoting. Two matrices namely A and B were used where A is any matrix and B be a slight random perturbation of A. The author proved that it is unlikely that B has large condition number. Using this result, the author also proved it is unlikely that B has large growth factor under Gaussian elimination without pivoting. By combining these results, the author bounded the smoothed precision needed to perform Gaussian elimination without pivoting. The results improve the average-case analysis of Gaussian elimination without pivoting performed by Yeung and Chan. The result was extended on the growth factor to the case of partial pivoting, and present the first analysis of partial pivoting that gives a sub-exponential bound on the growth factor. In particular, it is showed that if the random perturbation is Gaussian with a variance, then the growth factor is bounded with very high probability.

Higham (2011) explains that the standard method for solving systems of linear equations, Gaussian elimination (GE) is one of the most important and

ubiquitous numerical algorithms. However, its successful use relies on understanding its numerical stability properties and how to organize its computations for efficient execution on modern computers. Higham gives an overview of GE, ranging from theory to computation. He explains why GE computes an LU factorization and the various benefits of this matrix factorization viewpoint. Pivoting strategies for ensuring numerical stability are described. Special properties of GE for certain classes of structured matrices are summarized. How to implement GE in a way that efficiently exploits the hierarchical memories of modern computers is discussed. He also describe block LU factorization, corresponding to the use of pivot blocks instead of pivot elements, and explain how iterative refinement can be used to improve a solution computed by GE.

Ballard et al. (2005) posit that high performance for numerical linear algebra often comes at the expense of stability. Computing the LU decomposition of a matrix via Gaussian Elimination can be organized so that the computation involves regular and efficient data access. However, maintaining numerical stability via partial pivoting involves row interchanges that lead to inefficient data access patterns. To optimize communication efficiency throughout the memory hierarchy the authors confront two seemingly contradictory requirements: partial pivoting is efficient with column-major layout, whereas a block-recursive layout is optimal for the rest of the computation. The authors resolve this by introducing a shape morphing procedure that dynamically matches the layout to the computation throughout the algorithm, and show that Gaussian Elimination with partial pivoting can be performed in a communication efficient and cache-oblivious way. The technique extends to QR decomposition, where computing Householder vectors prefers a different data layout than the rest of the computation.

Khabou (2013) focuses on a widely used linear algebra kernel to solve linear systems, that is the LU decomposition. Usually, to perform such a computation one uses the Gaussian elimination with partial pivoting (GEPP). The backward stability of GEPP depends on a quantity which is referred to as the

growth factor, it is known that in general GEPP leads to modest element growth in practice. However its parallel version does not attain the communication lower bounds. To improve the upper bound of the growth factor, the author study a new pivoting strategy based on strong rank revealing QR factorization and develop a new block algorithm for the LU factorization. This algorithm has a smaller growth factor upper bound compared to Gaussian elimination with partial pivoting. The strong rank revealing pivoting is then combined with tournament pivoting strategy to produce a communication avoiding LU factorization that is more stable. Also two recursive algorithms were studied based on the communication avoiding LU algorithm, which are more suitable for architectures with multiple levels of parallelism. For an accurate and realistic cost analysis of these hierarchical algorithms, a hierarchical parallel performance model that takes into account processor and network hierarchies.

Uhling (1992), determined a scaling for the linear system $Ax = b$ through the two equations $D(AF)y = Db$, $y = F^{-1}x$. When scaling is implemented along with partial pivoting (PP) to solve $Ax = b$ by Gaussian elimination (GE), it is well known that certain ordered pairs (D, F) produce better computed solutions than those obtained in the absence of scaling, while others produce worse solutions. The two most common explanations of this fact are (D, F) modifies (magnifies or reduces) the classical condition number of A, and (D, F) modifies the magnitudes of the elements of A. In latter case, if a scaling yields entries of approximately the same magnitude, it is called an equilibration. Where the underlying hyperplane geometry of both the sweep out phase and the back-substitution phase of GE is used to achieve a new level of understanding. Uhling presented what we believe to be a better explanation of how scaling or equilibration influences PP in the selection of pivot equations, a process critical to both phases of GE.

Dekker et al. (1994) explained that the solution of linear systems continues to play an important role in scientific computing. The problems to be solved often are of very large size, so that solving them requires large computer resources.

To solve these problems, at least supercomputers with large shared memory or massive parallel computer systems with distributed memory are needed. Dekker et al. (1994) gave a survey of research on parallel implementation of various direct methods to solve dense linear systems. In particular they considered: Gaussian elimination, Gauss-Jordan elimination and a variant due to Huard (1979), and an algorithm due to Enright (1978)Enright (1978), designed in relation to solving (stiff) ODES, such that stepsize and other method parameters can easily be varied. Some theoretical results are mentioned, including a new result on error analysis of Huard's algorithm. Moreover, practical considerations and results of experiments on supercomputers and on a distributed-memory computer system are presented.

Yeung (2004) considered Gaussian elimination without pivoting applied to complex Gaussian matrices $X * **$. Yeung studied some independence properties of the elements of the LU factors of $\mathbf{X}$. Based on this, Yeung derived the probability distributions for all the L and U elements and obtain bounds for the probabilities of the occurrence of small pivots and large growth factors. Numerical experiments are presented to support the theoretical results and discussions are made to relate the results to the crucial practical problems of numerical stability of GE.

Xue et al. (2000) presented a new algorithm to directly solve the linear algebraic system $Ax = b$, where A is an $n \times n$ coefficient matrix which may be singular or ill-conditioned. By writing the system as an expanded matrix $A^{'} = [A \vdots b \vdots E]$, where E is an $n \times n$ unitary matrix, one can transform A into a unitary matrix through the row-transformations with complete pivoting and proper zeroing, It is shown that the algorithm can provide a solution in the non-null subspace of the solution space, if matrix A is singular. The criteria for curing ill-conditions are related to the numerical precision of computers. Numerical examples demonstrate the power of the new algorithm.

Castel et al. (1998) studied non-stationary multi splitting algorithms for

the solution of linear systems. Convergence of these algorithms is analyzed when the coefficient matrix of the linear system is hermitian positive definite. Asynchronous versions of these algorithms are considered and their convergence investigated.

Choi (2006) explained that CG, MINRES, and SYMMLQ are Krylov subspace methods for solving large symmetric systems of linear equations. According to this write CG (the conjugate-gradient method) is reliable on positive-definite systems, while MINRES and SYMMLQ are designed for indefinite systems. When these methods are applied to an inconsistent system (that is, a singular symmetric least-squares problem), CG could break down and SYMMLQ solution could explode, while MINRES would give a leastsquares solution but not necessarily the minimum-length solution (often called the pseudoinverse solution). This understanding motivates the author to design a MINRES-like algorithm to compute minimum-length solutions to singular symmetric systems. MINRES uses QR factors of the tridiagonal matrix from the Lanczos process (where R is upper-tridiagonal). Our algorithm uses a QLP decomposition (where rotations on the right reduce R to lower-tridiagonal form), and so we call it MINRES-QLP. On singular or nonsingular systems, MINRES-QLP can give more accurate solutions than MINRES or SYMMLQ. The author also derived preconditioned MINRES-QLP, new stopping rules, and better estimates of the solution and residual norms, the matrix norm and condition number. For a singular matrix of arbitrary shape, the author observe that null vectors can be obtained by solving least-squares problems involving the transpose of the matrix. For sparse rectangular matrices, this suggests an application of the iterative solver LSQR. In the square case, MINRES, MINRES-QLP, or LSQR are applicable. Results are given for solving homogeneous systems, computing the stationary probability vector for Markov Chain models, and finding null vectors for sparse systems arising in helioseismology.

Grcar (2011) explained that when modern computers (digital, electronic, and programmable) were being invented, John von Neumann and Herman Gold-

stine wrote a paper to illustrate the mathematical analyses that they believed would be needed to use the new machines effectively and to guide the development of still faster computers. Their foresight and the congruence of historical events made their work the first modern paper in numerical analysis. Von Neumann once remarked that to found a mathematical theory one had to prove the first theorem, which he and Goldstine did for the accuracy of mechanized Gaussian elimination but their paper was about more than that. Von Neumann and Goldstine described what they surmized would be the significant questions once computers became available for computational science, and they suggested enduring ways to answer them.

Li and Demmel (2004) they propose several techniques as alternatives to partial pivoting to stabilise sparse Gaussian eliminition. From numerical experiment they demonstrated that for a wide range of problems the new method is as stable as partial pivoting. The main advantage of the new method over partial pivoting is that it permits a priori determination of data structures and communication pattern for Gaussian elimination, which makes it more scalable on distributed memory achines. Based on this a priori knowledge, the authors deessif]gn higly parallel algoritms for both sparse Gaussian elimination and triangular solve and they showed that they are suitables for large scale distributes memory machines.

Parlett and Landis (2004) outlines new methods for scaling square, nonnegative matrices to doubly stochastic form are described. A generalized version of the convergence theorem of Sinkhorn and Knopp (1967) is proved and applied to show convergence for these new methods. Their tests indicate that one of the new methods has significantly better average and worst-case behavior than the Sinkhorn-Knopp method; for one of the 3 X 3 examples of Marshall and Olkin (1968), SK requires 130 times as many operations as the new algorithm to achieve row and column sums 1 ± 10^{-5}

Vecharynski (2006) in his thesis considered three crucial problems of nu-

merical linear algebra: solution of a linear system, an eigenvalue, and a singular value problem. The author focus on the solution methods which are iterative by their nature, matrix-free, preconditioned and require a fixed amount of computational work per iteration. In particular, this manuscript aims to contribute to the areas of research related to the convergence theory of the restarted Krylov subspace minimal residual methods, preconditioning for symmetric indefinite linear systems, approximation of interior eigenpairs of symmetric operators, and preconditioned singular value computations. The author first considered solving non-Hermitian linear systems with the restarted generalized minimal residual method (GMRES). The author proved that the cycleconvergence of the method applied to a system of linear equations with a normal (preconditioned) coefficient matrix is sublinear. In the general case, however, it is shown that any admissible cycle-convergence behavior is possible for the restarted GMRES at a number of initial cycles, moreover the spectrum of the coefficient matrix alone does not determine this cycle-convergence. Next we shift our attention to iterative methods for solving symmetric indefinite systems of linear equations with symmetric positive definite preconditioners. The author also described a hierarchy of such methods, from a stationary iteration to the optimal Krylov subspace preconditioned minimal residual method, and suggest a preconditioning strategy based on an approximation of the inverse of the absolute value of the coefficient matrix (absolute value preconditioners). We present an example of a simple (geometric) multigrid absolute value preconditioner for the symmetric model problem of the discretized real Helmholtz (shifted Laplacian) equation in two spatial dimensions with a relatively low wavenumber.

Poole and Neal (1991) explains the algorithm known as Gaussian elimination (GE) is fully understood in an exact-arithmetic environment. But in the finite-precision environment of computers, a full understanding of GE has been somewhat elusive. Heretofore, the analysis of this popular and important algorithm has been primarily from a numerical perspective. This paper seeks to

analyze GE from a geometric perspective, and by so doing, confirm the classical numerical analysis and demonstrate a new level of understanding through the Euclidean geometry of GE.

Poole and Neal (2002) also looked at the linear system$Ax = b$, of the ordered pair (D, F) of nonsingular diagonal matrices determine a scaling of the system through the two equations $D(AF)y = Db, y = F^{-1}x$. When scaling is implemented along with partial pivoting (PP) to solve $Ax = b$ by Gaussian elimination (GE), it is well known that certain ordered pairs (D, F) produce better computed solutions than those obtained in the absence of scaling, while others produce worse solutions. The two most common explanations of this fact are that (D, F) modifies (magnifies or reduces) the classical condition number of A, and (D, F) modifies the magnitudes of the elements of A. In the latter, if a scaling yields entries of approximately the same magnitude, it is called an equilibration. Here, the underlying hyperplane geometry of both the sweepout phase and the back-substitution phase of GE is used to achieve a new level of understanding. The authors presented what is believed to be a better explanation of how scaling or equilibration influences PP in the selection of pivot equations, a process critical to both phases of GE.

Poole and Neal (2000) based on their past work tittled Geometric Analysis of Gaussian elimination (GE), a new pivoting strategy, Rook's pivoting (RP), is introduced which encourages stability in the back-substitution phase of GE while controlling the growth of round-off error during the sweep-out. Earlier works has previously showed that RP, as with complete pivoting, cannot have exponential growth error. Empirical evidence presented in this work showed that RP produces computed solutions with consistently greater accuracy than partial pivoting. That is, Rook's pivoting is, on average, more accurate than partial pivoting, with comparable costs. Moreover, the overhead to implement Rook's pivoting in a scalar or serial environment is only about three times the overhead to implement partial pivoting. The theoretical proof establishing this fact

is presented, and is empirically confirmed in this paper and supported in Foster (1997)

Li et al. (2013) explained that the Higham matrix is a complex symmetric matrix $A = B + iC$, where both B and C are real, symmetric and positive definite and $i = \sqrt{-1}$ is the imaginary unit. According to the authors, for any Higham matrix A, Ikramov et al showed that the growth factor in Gaussian elimination is less than 3. In this paper, based on the previous results, a new bound of the growth factor is obtained by using the maximum of the condition numbers of matrices B and C for the generalized Higham matrix A, which strengthens this bound to 2 and proves the Higham conjecture.

Lipshitz et al. (2004) in their paper explained that matrix multiplication is a fundamental kernel of many high performance and scientific computing applications. The authors disclosed that most parallel implementations use classical $O(n^3)$ matrix multiplication, even though there exist Strassen - like matrix multiplication algorithms that have lower arithmetic complexity, as the classical ones perform better in practice. The authors also obtained a new parallel algorithm that is based on Strassen's fast matrix multiplication (SPAA 12) that minimizes communication: it communicates asymptotically less than all classical and all previous Strassen - based algorithms, and it attains corresponding lower bounds. It is also the first parallel-Strassen algorithm that exhibits perfect strong scaling. In this paper, the authors showed that the new algorithm is also faster in practice. The authors benchmark and compare the performance of our new algorithm to previous algorithms on Franklin (Cray XT4), Hopper (Cray XE6), and Intrepid (IBM BG/P). They also demonstrate significant speedups over previous algorithms both for large matrices and for small matrices on large numbers of processors. Moreover, the writers model and analyze the performance of the algorithm, and predict its performance on future exascale platforms.

Thorson (2001) discussed two FORTRAN routines in his paper. The FORTRAN routines can be used to solve banded linear systems. The routines

use a Gaussian elimination algorithm tailored to the specific case of a banded matrix. Instead of the $n^3/3$ multiplies required to reduce a full matrix, a banded matrix can be reduced in about $nm^2/4$ multiplies, where n is the dimension of the matrix and m is its bandwidth. Only the nonzero diagonals of the matrix need to be stored. Algorithm 2 does no pivoting. Algorithm 3 performs partial pivoting. Partial pivoting is inherently stable than no pivoting al all, though the difference in the output between the two algorithms is probably negligible for regular wave equation operators. The algorithm listings contain all relevant documentation for their use.

Dumas et al. (2013) posit that Gaussian elimination with full pivoting generates a $PLUQ$ matrix decomposition. Depending on the strategy used in the search for pivots, the permutation matrices can reveal some information about the row or the column rank profiles of the matrix. The authors proposed a new pivoting strategy that makes it possible to recover at the same time both row and column rank profiles of the input matrix and of any of its leading sub-matrices. We propose a rank-sensitive and quad-recursive algorithm that computes the latter $PLUQ$ triangular decomposition of an $m \times n$ matrix of rank r in $O(mnr^{\omega-2})$ field operations, with ω the exponent of matrix multiplication. Compared to the LEU decomposition by Malashonock, sharing a similar recursive structure, its time complexity is rank sensitive and has a lower leading constant. Over a word size finite field, this algorithm also improves the practical efficiency of previously known implementations.

Chapter 3

METHODOLOGY

3.1 Various forms of errors

For any system of linear equations, the question of the types of errors in a solution obtained by a numerical method is not readily answered. There are three main sources of errors in numerical computation: rounding, data uncertainty, and truncation.

Rounding errors are an unavoidable consequence of working in finite precision arithmetic. Numerical methods for solving systems of linear equations involve large numbers of arithmetic operations. For example, the Gauss elimination according to Atkinson (1993), involves $(n^3 + 3n^2 - n)/3$ multiplications/divisions and $(2n^3 + 3n^2 - 5n)/6$ additions/subtractions in the case of a system with n unknowns. Since round-off errors are propagated at each step of an algorithm, the growth of round-off errors can be such that, when n is large, a solution differs greatly from the true one.

Uncertainty in the data is always a possibility when we are solving practical problems. It may arise in several ways: from errors of measurement or estimation for example engineering and economical data, from errors in storing the data on the computer (rounding errors-tiny) and from the result of errors (big or small) in an earlier computation if the data is itself the solution to another problem. The effects of errors in the data are generally easier to understand than the effects of rounding errors committed during a computation, because data errors can be analyzed using perturbation theory for the problem at hand, while intermediate rounding errors require an analysis specific to the given method.

Analyzing truncation errors, or discretization errors, is one of the major tasks

of the numerical analyst. Many standard numerical methods (for example, the trapezium rule for quadrature, Euler's method for differential equations, and Newton's method for nonlinear equations) can be derived by taking finitely many terms of a Taylor series. The terms omitted constitute the truncation error, and for many methods the size of this error depends on a parameter (often called h), the step-size, whose appropriate value is a compromise between obtaining a small error and a fast computation. However, some sources of errors are indicated below:

Errors in the coefficients and constants - in many practical cases, the coefficients of the variables, and also the constants on the right-hand sides of the equations are obtained from observations of experiments or from other numerical calculations. They will have errors; and therefore, once the solution of a system has been found, it too will contain errors.

3.2 System of Equations

A system of linear equations (or linear system) is a collection of linear equations involving the same set of variables considered collectively, rather than individually and a solution to a linear system is an assignment of numbers to the variables such that all the equations are simultaneously satisfied. Matrix algebra is used to solve a system of simultaneous linear equations. For many mathematical procedures such as the solution to a set of nonlinear equations, interpolation, integration, and differential equations, the solutions reduce to a set of simultaneous linear equations.

3.2.1 Coefficient Matrix of Systems of Equations

The coefficient matrix refers to a matrix consisting of the coefficients of the variables in a set of linear equations. In general, a system with m linear equations and n unknowns can be written as:

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = c_1$$
$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = c_2$$
$$\vdots$$
$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = c_m$$

where $x_1, x_2, \cdots, x_n$ are the unknowns and the numbers $a_{11}, a_{12}, \cdots, a_{mn}$ are the coefficients of the system. The coefficient matrix is a $m \times n$ matrix with the coefficient a_{ij} as the $(ij)^{th}$ entry:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

can be rewritten in the matrix form as:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}$$

This is denoted by $[A]$, $[X]$ and $[C]$ respectively, the system of equation is $[A][X] = [C]$, where $[A]$ is called the coefficient matrix, $[C]$ is called the right hand side vector and $[X]$ is called the solution vector.

Alternatively, $[A][X] = [C]$ systems of equations are written in the augmented form as:

$$\left[A \mid C\right] = \left[\begin{array}{cccc:c} a_{11} & a_{12} & \cdots & a_{1n} & c_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & c_2 \\ \cdots & \cdots & \cdots & \cdots & \\ a_{m1} & a_{m2} & \cdots & a_{mn} & c_n \end{array}\right]$$

3.3 Consistent and Inconsistent Systems of Equations

A system of equations $[A][X] = [C]$ is consistent if there is a solution, and it is inconsistent if there is no solution. However, a consistent system of equations does not mean a unique solution, that is, a consistent system of equations may have a unique solution or infinite solutions.

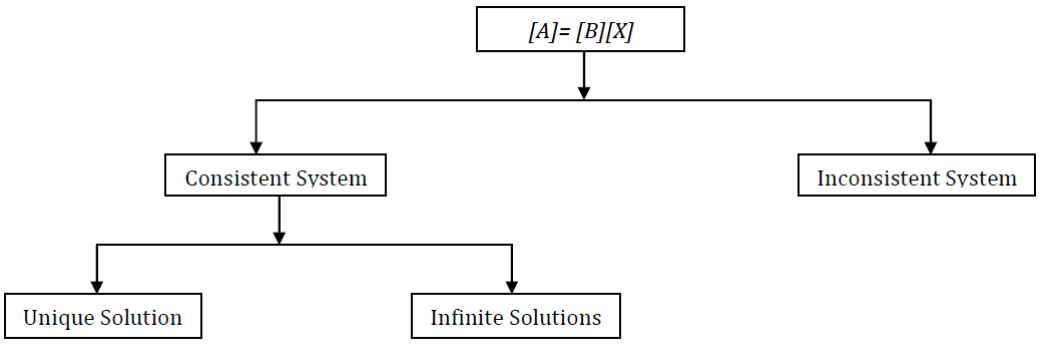

Figure 1.2 [Consistent and Inconsistent system of equations]

3.4 Rank of Matrices

The rank of a matrix is defined as the order of the largest square sub-matrix whose determinant is not zero. If A is $n \times n$ matrix and $\det(A) \neq 0$, the largest square sub-matrix possible is of order n and that is [A] itself therefore the rank of [A] is of order n conversely if $\det(A) = 0$, the rank of [A]$< n$ other square sub matrices of [A] are explored and the rank is the order of the matrix that gives $\det(A) = 0$. Also given that A is $m \times n$ matrix, the rank of [A] is at most order m since there are no square sub-matrices of order n so square sub-matrices of [A] of order m is explored; if any of these square sub-matrices have determinant not equal to zero, then the rank is m. For example, given that:

$$[\mathrm{A}]=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

[A] is a 3×3 matrix and the largest square sub-matrix is [A] itself. If $\det(A) \neq 0$, the rank of [A] is of order 3; conversely if $\det(A) = 0$, obviously the rank of [A]< 3 and if the determinant of the next square sub-matrix which is a 2×2 matrix $\neq 0$, therefore the rank is of order 2.

Also given that:

$$[\mathrm{A}]=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}$$

[A] is 3×4 matrix and since there are no square sub-matrices of order 4 as [A] is a 3×4 matrix, the rank of [A] is at most 3. So the next square sub-matrices of [A] is explored which is a 3×3; if these square sub-matrix have determinant not equal to zero, then the rank is 3.

Alternatively, the rank of a matrix can be obtained by transforming the coefficient matrix A into an echelon form either reduced form or unreduced form R by counting the number of nonzero rows or the number of pivots or leading coefficients in the echelon form in R. In fact, the pivot columns (i.e. the columns with pivots in them) are linearly independent. Elementary row operations can be use to reduce A to echelon form.

The concept of rank can be used to determine if a system is either consistent or inconsistent, a system of equations $[A][X] = [C]$ is consistent if the rank of A is equal to the rank of the augmented matrix $[A\vdots C]$ and a system of equations $[A][X] = [C]$ is inconsistent if the rank of A is less than the rank of the augmented matrix $[A\vdots C]$. For instance a system of equations $[A][X] = [C]$,

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \end{bmatrix} = \begin{bmatrix} c_{11} \\ c_{21} \\ c_{31} \end{bmatrix}$$

in which $\det(A) = \alpha \neq 0$, then rank(A)=3;

The augmented matrix is [B]= $\left[\begin{array}{cccc} a_{11} & a_{12} & a_{13} & \vdots c_{11} \\ a_{21} & a_{22} & a_{23} & \vdots c_{21} \\ a_{31} & a_{32} & a_{33} & \vdots c_{31} \end{array}\right]$ Since there are no square sub-matrices of order 4 as [B] is a 3×4 matrix, the rank of the augmented [B] is at most 3. So square sub-matrices of the augmented matrix [B] of order 3 is explored to see if any of these have determinants not equal to zero, then the rank is 3. For example, a square sub-matrix of the augmented matrix [B] is

$$[\mathrm{D}]=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \text{ has } \det(D) = \beta \neq 0$$

Hence the rank of the augmented matrix [B] is 3. Since [A]= [D] , the rank of [A] is 3. Since the rank of the augmented matrix [B] equals the rank of the coefficient matrix [A], the system of equations is consistent.

On the contrary, if all of the square sub-matrices of the augmented matrix have determinant equal to zero, other square sub-matrices of order $n-1$ is explored to find their determinants. That is if

$$[\mathrm{D}]=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \det(D) = 0,$$

$$[\mathrm{E}]=\left[\begin{array}{ccc} a_{12} & a_{13} & \vdots c_{11} \\ a_{22} & a_{23} & \vdots c_{21} \\ a_{32} & a_{33} & \vdots c_{31} \end{array}\right] \det(E)=0,$$

$$[\mathrm{F}]=\left[\begin{array}{ccc} a_{11} & a_{12} & \vdots c_{11} \\ a_{21} & a_{22} & \vdots c_{21} \\ a_{31} & a_{32} & \vdots c_{31} \end{array}\right] \det(F)=0$$

and

$$[\mathrm{G}]=\begin{bmatrix} a_{11} & a_{12} & \vdots c_{11} \\ a_{21} & a_{13} & \vdots c_{23} \\ a_{33} & a_{32} & \vdots c_{31} \end{bmatrix} \quad \det(G)=0$$

All the square sub-matrices of order 3×3 of the augmented matrix [B] have a zero determinant. So the rank of the augmented matrix [B] is obviously less than 3. The other square sub-matrices of order $n-1$ is explored to find their determinants and if the determinant of any of the 2×2 square sub-matrices of the augmented matrix [B] is not equal to zero, then the rank of the augmented matrix [B] is 2. For example some of the possible 2×2 sub-matrices of the augmented matrix [B] are $\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$, $\begin{bmatrix} b_{11} & b_{13} \\ b_{21} & b_{22} \end{bmatrix}$, $\begin{bmatrix} b_{11} & b_{12} \\ b_{31} & b_{32} \end{bmatrix}$, $\begin{bmatrix} b_{11} & b_{13} \\ b_{31} & b_{33} \end{bmatrix}$ etc. So the rank of the augmented matrix [B] is 2 and if the rank of the coefficient matrix [A] is also 2, hence, rank of the coefficient matrix [A] equals the rank of the augmented matrix [B]. So the system of equations $[A][X] = [C]$ is consistent otherwise it is inconsistent.

Furthermore, for a consistent system, $[A][X] = [C]$. If the rank of the coefficient matrix [A] is same as the number of unknowns, then the solution is unique; if the rank of the coefficient matrix [A] is less than the number of unknowns, then infinite solutions exist. If there are more equations than unknowns in $[A][X] = [C]$, does not mean the system is inconsistent it depends on the rank of the augmented matrix $[A\vdots C]$ and the rank of $[A]$ and if the rank of (A) equals the number of unknowns, the solution is not only consistent but also unique; on the contrary if the rank of $[A] <$ the number of unknowns, infinite solutions exist.

If the system has a single unique solution, the system can be basically solved with direct or iterative methods. When the system has no solution, only an approximate solution can be estimated, usually by formulating a least squares problem and when the system has infinitely many solutions this occurs for rank-deficient problems or under-determined problems. In spite of infinitely many solutions, a good approximation to the true solution can be obtained if some a *priori* knowledge about the nature of the true solution is accessible. The addi-

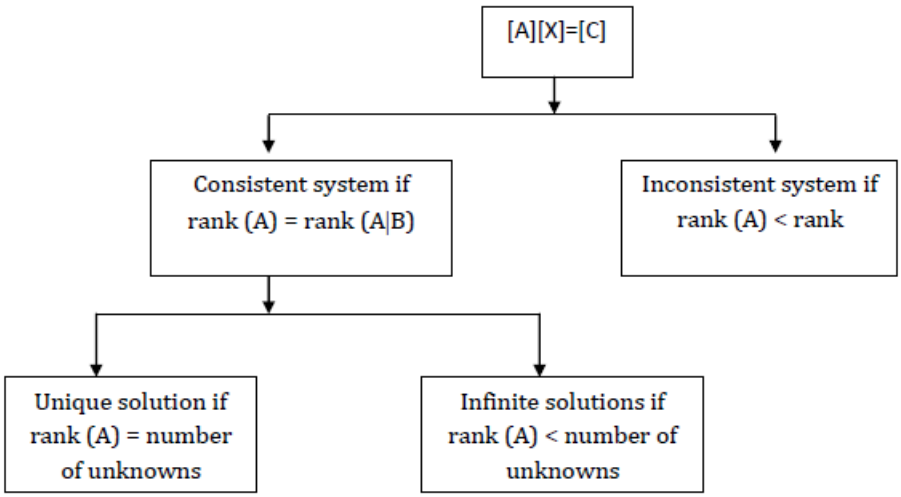

Figure 1.3 Flow chart of conditions for consistent and inconsistent system of equations

tional constraints are usually concerned with a degree of sparsity or smoothness of the true solution.

3.5 Nullity of Matrices

Suppose A is an $m \times n$ matrix and R is a reduced echelon form of A given as :

$$[\mathrm{A}]=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}$$

$$[\mathrm{R}]=\begin{bmatrix} 1 & a_{12} & a_{13} & a_{14} \\ 0 & 1 & a_{23} & a_{24} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

the number of free variables or entries of $Ax = b$ is the nullity of A. Since the number of nonzero rows is 2 i.e. there are two (2) pivots, therefore the $Rank(A) = 2$ and the $Nullity = 4 - 2 = 2$. The relationship between $Rank(A)$ and $Nullity(A)$ is given as:

$$Rank(A) + Nullity = \text{number of columns in } A$$

i.e. $n = Rank(A) + Nullity(A)$. The $Rank(A)$ counts the pivot variables, the $Nullity(A)$ counts the free variables, and the number of columns corresponds to the total number of variables for the coefficient matrix A.

Suppose A is an $n \times n$ matrix. A is an invertible and nonsingular if and only if $Rank(A) = n$

3.6 Specific Types of Linear Systems

The behavior of a linear system is determined by the relationship between the number of equations and the number of unknowns

3.6.1 Homogeneous Systems

A system of linear equations is called homogeneous if the right hand side is the zero vector. This system actually has a number of solutions, but there is one obvious one, called the trivial solution. A vector is called trivial if all its coordinates are 0, i. e. if it is the zero vector. In Linear Algebra we are not interested in only finding one solution to a system of linear equations but all possible solutions. In particular, homogeneous systems of equations are very important in that whether or not there is any non-trivial solution, i. e. whether there is any solution other than the trivial one.

A system of linear equations of the form:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= 0 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= 0 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= 0 \end{aligned}$$

is called homogeneous systems. It is always consistent satisfied by the solution $x_1 = x_2 = \cdots = x_n = 0$. This solution is called the trivial solution.

3.6.2 Underdetermined Systems

An underdetermined system is a system of linear equations in which there are more unknowns (variables, n) than constraints (equations, m) i.e. a linear system of equations that has fewer equations than variables. For example, a system with two equations and three unknowns is underdetermined. That is, an underdetermined linear system has the form:

$$A_{m\times n}x_{n\times 1} = b_{m\times 1}$$

where A is a matrix with m rows and n columns and as such $m < n$. An underdetermined system might be consistent or inconsistent and it never has a unique solution but can have infinitely many solutions this is because an underdetermined system must have at least one free variable. Therefore, an underdetermined system which is consistent must have an infinite number of solutions. The rank of underdetermine system is less than or equal to the number of unknowns n.
If $Ax = b$ with A a matrix that has fewer rows than columns, this implies that the solutions, if they exist, will not be unique. Two ways to see this:
Method 1 If $Ax = b$ is solved by reducing A into echelon form, you will find that not every column in the echelon form can have a pivot. Therefore, when you write down the general solution, there will be free variables, leading to an infinite number of solutions. **Method 2** Since A has fewer rows than columns, then A is an $m\times n$ matrix with $m < n$. Then $rank(A) \leq n$, and since $rank(A)$+nullity(A)= n , then $nullity(A) = n - rank(A) \geq n - m > 0$. Therefore the null space of A has dimension greater than 0 , so if x_p is a particular solution to the equation, then any vector of the form $x_p + h$ is also a solution for any $h \in Nul(A)$, and there are infinitely many choices for h.

3.6.3 Overdetermined Systems

A linear system of equations in which there are more constraints (equations, m) than unknowns (variables, n) i.e. any system of linear equations having more

equations than variables. For example, a system with three equations and only two variables is overdetermined. That is, an overdetermined linear system has the form:

$$A_{m\times n}x_{n\times 1} = b_{m\times 1}$$

where A is a matrix with m rows, n columns and as such $m > n$. The reason why more equations than unknowns arise in such problems is that repeated measurements are taken to minimize errors. This produces an overdetermined and often inconsistent system of linear equations. In general, overdetermined systems have no solution. In some cases, linear least squares may be used to find an approximate solution.

3.6.4 Square Systems

A linear system of equations in which the number of constraints (equations, m) is equal to the number of unknowns (variables, n) i.e. any system of linear equations having the same number of equations and variables. For example, a system with three equations and three variables is a square system. That is, a square linear system has the form:

$$A_{m\times n}x_{n\times 1} = b_{m\times 1}$$

where A is a matrix with m rows, n columns and as such $m = n$ and usually, a system with the same number of equations and unknowns has a single unique solution. An m by n consistent system of equations will have a unique solution if and only if the nullity of the coefficient matrix is zero. The set of linear equations that are considered in this thesis is said to be a square matrix. A typical and a well-known example of an ill conditioned linear matrix called the Hilbert matrix is taken into consideration.

3.7 Condition of the linear system

Condition is the technical term used to describe how sensitive the solution is to changes in the coefficient matrix or the right hand side. In practice, the input data A and b may be contaminated by error. This error may be experimental, may come from the process of discretization, and so on. In order to estimate the accuracy of the computed solution, the error in the data should be taken into account. Problems whose solutions may change drastically even with small changes in the input data are said to be ill-conditioned. Ill conditioning is independent of the algorithms used to solve the problems.

3.8 Measures of errors

Perturbations in the data change the solution of the linear system and as such we need to understand how to measure the size of vectors and of matrices. This leads to vector norms and matrix norms. Matrix and vector norms are denoted by the same symbol $\| \cdot \|$, however vector-norms and matrix-norms are computed very differently. Thus, before computing a norm we need to examine carefully whether it is applied to a vector or to a matrix. It should be clear from the context which norm, a vector-norm or a matrix-norm, is used.

3.8.1 Vector norms

A vector norm on R^n is a function

$$\| \cdot \| : R^n \longrightarrow R$$

$$x \longrightarrow \| x \|$$

which for all $x, y E R^n$ and αER satisfies

1. $\| x \| \geq 0, \| x \| = 0 \Longleftrightarrow x = 0.$

2. $\| \alpha x \| = | \alpha | \| x \|$

3. $\| x + y \| \leq \| x \| + \| y \|$, (triangle inequality)

The most frequently used vector norms on R^n are 2-norm, p-norm and ∞-norm given as $\| x \|_2 = (\sum_{i=1}^{n} x_i^2)^{1/2}$, $\| x \|_p = (\sum_{i=1}^{n} | x_i^2 |)^{1/p}$ and $\| x_\infty \| = \max | x_i |$ respectively.

3.8.2 Matrix norms

A matrix norm on $R^{m \times n}$ is a function

$$\| \cdot \| : R^{m \times n} \longrightarrow R$$

$$A \longrightarrow \| A \|$$

which for all A, B $ER^{m \times n}$ and αER satisfies

1. $\| A \| \geq 0, \| A \| = 0 \Longleftrightarrow x = 0$.(zero matrix)

2. $\| \alpha A \| = | \alpha | \| A \|$

3. $\| A + B \| \leq \| A \| + \| B \|$, (triangle inequality)

For any AE$R^{m \times n}, B$E$R^{n \times k}$ and xER^n

$$\| Ax \|_p \leq \| A \|_p \| x \|_p \text{ (compatibility of matrix and vector norms)}$$

and

$$\| AB \|_p \leq \| A \|_p \| B \|_p \text{ (sub-multiplicativity of matrix norms)}$$

The most common matrix norms are the maximum column norm, maximum row norm and the spectral norm which are given as $\| A_1 \| = \max \sum_{i=1}^{m} | a_{ij} |$, $\| A_1 \| = \max \sum_{i=1}^{m} | a_{ij} |$ and $\| A_2 \| = \sqrt{\lambda_{\max}(A^T A)}$ respectively. Where $\lambda_{\max}(A^T A)$ is the largest eigen value of $(A^T A)$.

Condition number of an invertible square matrix depends on the norm of a matrix.

The norm of a matrix is a simple unique scalar number which measures the size of errors in the coefficient matrix A of linear systems.

3.8.3 Effect of the perturbation of RHS

$$\text{Let } [A][X] = [C]$$

if $[C]$ is changed to $[C']$, $[X]$ will change to $[X']$
such that

$$[A][X'] = [C']$$

Denoting change in $[C]$ and $[X]$ matrices as $[\Delta C]$ and $[\Delta X]$, respectively

$$[\Delta C] = [C'] - [C]$$
$$[\Delta X] = [X'] - [X]$$

then

$$[A](X + \Delta X) = [C] + [\Delta C]$$

Expanding the above expression

$$[A][X] + [A][\Delta X] = [C] + [\Delta C]$$

$[A][\Delta X] = [\Delta C]$ since $[A][X] = [C]$
Applying the theorem of norms, that the norm of multiplied matrices is less than the multiplication of the individual norms of the matrices,

$$\| \Delta X \| \leq \| A^{-1} \| \| \Delta C \| \tag{3.1}$$

and

$$\| C \| \leq \| A \| \| X \| \tag{3.2}$$

Multiplying the two equations together

$$\| \Delta X \| \| C \| \leq \| A^{-1} \| \| \Delta C \| \| A \| \| X \|$$

Dividing both sides by

$$\| X \| \text{ and } \| C \|$$

$$\frac{\| \Delta X \| \| C \|}{\| X \| \| C \|} \leq \frac{\| A \| \| A^{-1} \| \| \Delta C \| \| X \|}{\| X \| \| C \|}$$

$$\therefore \frac{\| \Delta X \|}{\| X \|} \leq \| A \| \| A^{-1} \| \frac{\| \Delta C \|}{\| C \|}$$

3.8.4 Effect of the perturbation of coefficient matrix

$$\text{Let } [A][X] = [C]$$

if $[A]$ is changed to $[A']$, $[X]$ will change to $[X']$

such that

$$[A'][X'] = [C]$$

$$\Rightarrow [A][X] = [A'][X']$$

Denoting change in $[A]$ and $[X]$ matrices as $[\Delta A]$ and $[\Delta X]$, respectively

$$[\Delta A] = [A'] - [A]$$
$$[\Delta X] = [X'] - [X]$$

then

$$[A][X] = ([A] + [\Delta A])([X] + [\Delta X])$$

Expanding the above expression

$$[A][X] = [A][X] + [A][\Delta X] + [\Delta A][X] + [\Delta A][\Delta X]$$

Grouping like terms

$$[A][X] - [A][X] = [A][\Delta X] + [\Delta A]([X] + [\Delta X])$$
$$[0] = [A][\Delta X] + [\Delta A]([X] + [\Delta X])$$
$$-[A][\Delta X] = [\Delta A]([X] + [\Delta X])[\Delta X]$$
$$[\Delta X] = -[A]^{-1}[\Delta A]([X] + [\Delta X])$$

Applying the theorem of norms, that the norm of multiplied matrices is less than the multiplication of the individual norms of the matrices,

$$\| \Delta X \| \leq \| A^{-1} \| \| \Delta A \| \| X + \Delta X \|$$

Multiplying both sides by $\| A \|$

$$\| A \| \| \Delta X \| \leq \| A \| \| A^{-1} \| \| \Delta A \| \| X + \Delta X \|$$

Dividing both sides by

$$\| A \| \text{ and } \| X + \Delta X \|$$

$$\frac{\| A \| \| \Delta X \|}{\| A \| \| X + \Delta X \|} \leq \frac{\| A \| \| A^{-1} \| \| \Delta A \| \| X + \Delta X \|}{\| A \| \| X + \Delta X \|}$$

$$\therefore \frac{\| \Delta X \|}{\| X + \Delta X \|} \leq \| A \| \| A^{-1} \| \frac{\| \Delta A \|}{\| A \|}$$

3.8.5 Condition number

Condition number of a function with respect to an argument measures how much the output value of the function can change for a small change in the input argument. This is used to measure how sensitive a function is to changes or errors in the input, and how much error in the output results from an error in the input. The condition number of a square and nonsingular matrix A is defined as

$$\kappa(A) = \| A \| \| A^{-1} \|$$

Properties

- $\kappa(A) \geq 1$ for all A
- A is a well-conditioned if $\kappa(A)$ is small (close to 1): the relative error in x is not much larger than the relative error in b
- A is badly or ill-conditioned if $\kappa(A)$ is large: the relative error in x can be much larger than the relative error in b

3.8.6 Ill Condition Systems

A square matrix A is ill-conditioned if it is invertible but can become non-invertible (singular) if some of its entries are changed ever so slightly. The condition number of A is a measure of how ill-conditioned A is and can be found using A and A^{-1}. The bigger the condition number is the more ill-conditioned A is. Solving linear systems whose coefficient matrices are ill-conditioned is tricky because even a small change in the data (e.g., the right-hand side vector) can lead to radically different answers.

When the solution is highly sensitive to the values of the coefficient matrix A or the righthand side constant vector b, the equations are called to be ill-conditioned. Ill-conditioned systems pose particular problems where the coefficients or constants are estimated from experimental results or from a mathematical model. Therefore, we cannot rely on the solutions coming out of an ill-conditioned system. The problem is then how do we know when a system of linear equations is ill-conditioned. To do that we have to first define vector and matrix norms.

There may be two ways of identifying if a matrix is ill conditioned. Firstly, compute $cond(A)$. This is relatively expensive and sometimes hard to interpret because the value may be in an intermediate range. Secondly, one can introduce deliberate representation of errors by slightly perturbing one or more elements in A. Call the new matrix A', and solve $A'x' = b$. If $x' \approx x$, then there is probably no ill conditioning. The danger here is that you might be unlucky, and chose the wrong element to perturb. But if you try this several times with different elements and all the solutions are about the same, then you have confidence that the matrix is well conditioned.

3.8.7 Well Condition Systems

For a square matrix A we can measure the sensitivity of the solution of the linear algebraic system $Ax = b$ with respect to changes in vector b and A is well-conditioned if small errors in the data produce small errors in the result. The condition number of a well condition system is always greater or equal to 1. If it is close to one, the matrix is well conditioned which means its inverse can be computed with good accuracy.

3.8.8 Solution Methods

Solutions methods that are normally applied to solving for the solution of systems of linear equations are two namely: direct methods and iterative methods. In these thesis however the efficacy of some of the direct methods would be tested namely elimination method and General approach algorithm. The simplest type of elimination method; Gaussian elimination with no pivoting would be used, the process is based upon the principle that, if we convert $[A]$ to an upper triangular matrix, we can solve for $[x]$ by backwards substitution.

The general approach algorithm is also based upon the principle that if one equation of $[A]$, for example the last equation is nearly similar or parallel to any one of the other equations. We want to replace it with another one (perpendicular to it), resulting in an equivalent system, A'x=b'

3.8.9 Adequacy of linear systems

The general relationship that exists between $\frac{\| \Delta X \|}{\| X \|}$ and $\frac{\| \Delta C \|}{\| C \|}$ is given as

$$\frac{\| \Delta X \|}{\| X + \Delta X \|} \leq \| A \| \| A^{-1} \| \frac{\| \Delta C \|}{\| C \|} \tag{3.3}$$

or between $\frac{\| \Delta X \|}{\| X \|}$ and $\frac{\| \Delta A \|}{\| A \|}$ is given as

$$\frac{\| \Delta X \|}{\| X + \Delta X \|} \leq \| A \| \| A^{-1} \| \frac{\| \Delta A \|}{\| A \|} \tag{3.4}$$

Equations (3.3) and (3.4) are two inequalities showing the relative change in the norm of the right hand side vector or the coefficient matrix which can be amplified by as much as the condition number, $\| A \| \| A^{-1} \|$.

Thus if the condition number is not too large, then a small pertubation in the vector b will have little effect on the solution. On the other hand, if the condition number is large, then even a small pertubation in b might change the solution drastically.

The norm is related to the conditioning of the matrix and there is a general relationship that exists between the relative change in the norm of solution vector and the relative change in the norm of the right hand side vector. There also exist a relationship between the relative change in the norm of solution vector and the relative change in the norm of the coefficient matrix and this helps to identify well-conditioned and ill conditioned system of equations and it also tells how many significant digits we could trust in the solution of a system of simultaneous linear equations. The condition number coupled with the machine epsilon, the quantification of the accuracy of the solution of the linear system can be known i.e. by knowing how many significant digits are correct in the solution vector in order to trust the accuracy in the solution vector.

That is the relative error in a solution vector is $\leq$ Cond (A) $\times$ relative error in either the right hand side or the coefficient matrix and the possible relative error in the solution vector is $\leq$ Cond(A) $\times$ machine epsilon (ϵ_{mach}).

3.9 Gaussian elimination

One of the most popular techniques for solving simultaneous linear equations is the Gaussian elimination method also known as row reduction which is a numerical algorithm for solving systems of linear equations. The approach is designed to solve a general set of n equations and n unknowns and it consists of two steps namely forward elimination of unknowns: In this step, the unknown is eliminated in each equation starting with the first equation. This way, the equations are reduced to one equation and one unknown in each equation and secondly, back substitution: In this step, starting from the last equation, each of the unknowns is found. Gaussian elimination transforms the linear system into an upper triangular form (an upper triangular matrix is a square matrix where all elements below the diagonal are 0, and the other elements may be either zero or non-zero.), which is easier to solve. This process, in turn, is equivalent to finding the factorization $A = LU$, where L is a unit lower triangular matrix and U is an upper triangular matrix. This factorization is especially useful when solving many linear systems involving the same coefficient matrix but different right-hand sides, which occurs in various applications.

3.9.1 No pivoting

Gaussian Elimination without pivoting or Naive Gauss elimination proceeds by successively eliminating the elements below the diagonal of the matrix of the linear system until the matrix becomes triangular, when the solution of the system is very easy. There are two pitfalls of the Naive Gauss elimination method namely *division by zero* and *round-off errors*. One method of decreasing the *round-off errors* would be to use more significant digits, that is, use double or quad precision for representing the numbers. However, this would not avoid possible *division by zero* errors in the Naive Gauss elimination method. To avoid *division by zero* as well as reduce (not eliminate) round-off error, a way around this involves the use

of pivots.

Solution of the system Ax=b based on LU factorization of A.

Algorithm Steps:

for $k = 1$ to $n - 1$

Find an elementary matrix M_k such that

$$A^{(k)} = M_k A^{(k-1)}$$

has zeros below (k, k) entry of the k^{th} column.

$$\text{Where} = \begin{cases} L = (M_{n-1} M_{n-2} ... M_2 M_1)^{-1} \\ U = M_{n-1} M_{n-2} ... M_2 M_1 A \end{cases}$$

$$\text{and } A = LU$$

3.9.2 Partial pivoting

Gaussian Elimination with partial pivoting selects the pivot row to be the one with the maximum pivot entry in absolute value from those in the leading column of the reduced sub-matrix. The term *partial* in partial pivoting refers to the fact that in each pivot search only entries in the left column of the matrix or sub-matrix are considered. Two rows are interchanged to move the designated row into the pivot row position. For increased numerical stability, the largest possible pivot element is used. This requires searching in the partial column below the pivot element. Partial pivoting is usually sufficient. To avoid division by zero, swap the row having the zero pivot with one of the rows below it. To minimize the effect of roundoff, the row that puts the largest pivot element on the diagonal is always chosen. The two methods are the same, except in the beginning of each step of forward elimination, a row switching is done based on the following criterion. If there are n equations, then there are n-1 forward elimination steps.

At the beginning of the $k\ th$ step of forward elimination, one finds the maximum of

$$|a_{kk}, |a_{k+1,k}|, ...|a_{nk}|$$

then if the maximum of these values is $|a_{pk}|$ in the p_{th} row, $k \leq p \leq n$, then switch rows p and k. The other steps of forward elimination are the same as the Naive Gauss elimination method. The back substitution steps stay exactly the same as the Naive Gauss elimination method.

Solution of the system Ax=b based on LU factorization of A.

Algorithm Steps:

for $k = 1$ to $n - 1$

Scan the entries of the k^{th} column of the matrix A^{k-1} below the row $(k-1)$ identify the pivot $a_{r_k k}$, such that $\mid a_{r_k k} \mid = max \mid a_{tk} \mid$. Form the permutation matrix P_k and the elementary matrix M_k such that

$$A^{(k)} = M_k P_k A^{(k-1)}$$

has zeros below (k, k) entry of the k^{th} column.

$$\text{Where} \begin{cases} L = P(M_{n-1}P_{n-1}M_{n-2}P_{n-2}...M_2P_2M_1P_1)^{-1} \\ U = (M_{n-1}P_{n-1}M_{n-2}P_{n-2}...M_2P_2M_1P_1)^{-1}A \\ P = (P_1P_2...P_{n-2}P_{n-1})^{-1} \end{cases}$$

$$\text{and } PA = LU$$

Solution of Ax=b without Explicit factorization

Algorithm Steps:

For $k = 1, 2, ..., n = 1$

Step 1:

Find r_k such that $\mid a_{r_k,k} \mid = \max \mid a_{ik}$. If $a_{r_k,k} = 0$, stop. Otherwise, go to step 2.

Step 2:

Interchange the rows of k and r_k of A and b

step 3:

Form multipliers $a_{ik} \equiv m_{ik} = \dfrac{-a_{ik}}{a_{kk}}$

step 4: Update the entries of A: $a_{ij} = a_{ij} + m_{ik}a_{kj}$

Step 5: Update the entries of b

$b_j = b_i + m_{ik}b_k$

3.9.3 Complete pivoting

Gaussian Elimination with complete pivoting exchange both rows and columns of the matrix. Column exchange requires changing the order of the x_i. For increased numerical stability, the largest possible pivot element is used. This requires searching in the pivot row, and in all rows below the pivot row, starting the pivot column. Gaussian elimination with complete pivoting selects the pivot entry as the maximum pivot entry from all entries in the sub-matrix. (This complicates things because some of the unknowns are rearranged.) Two rows and two columns are interchanged to accomplish this. Complete pivoting is less susceptible to roundoff, but the increase in stability comes at a cost of more complex programming. Unfortunately, neither complete pivoting nor partial pivoting solves all problems of rounding error. Some systems of linear equations, called ill-conditioned systems, are extremely sensitive to numerical errors. For such systems, pivoting is not much help. A common type of system of linear equations that tends to be ill-conditioned is one for which the determinant of the coefficient matrix is nearly zero.

Solution of the system Ax=b based on LU factorization of A.

Algorithm Steps:

For $k = 1$ to $n - 1$

Scan the entries of the A^{k-1} below the row $(k-1)$ to the right of the column $(k-1)$ to identify the pivot $a_{r_k k}$, such that $\mid a_{r_k,sk} \mid = max \mid a_{tk} \mid$.

Form the permutation matrices P_k and Q_k, and the elementary matrix M_k such that

$$A^{(k)} = M_k P_k A^{(k-1)} Q_k$$

has zeros below (k,k) entry of the k^{th} column.

$$\text{Where} = \begin{cases} L = P(M_{n-1}P_{n-1}...M_{n-1}P_{n-1})^{-1} \\ U = (M_{n-1}P_{n-1}...M_{n-1}P_{n-1}AQ_{n-1}...Q_{n-1} \\ P = (P_1 P_2 ... P_{n-2} P_{n-1})^{-1} \\ Q = Q_1 Q_2 ... Q_{n-2} Q_{n-1} \end{cases}$$

$$\text{and } PAQ = LU$$

Solution of Ax=b without Explicit factorization

Algorithm Steps:

For $k = 1, 2, ..., n = 1$

Step 1:

Find r_k and s_k such that $\mid a_{r_k,k} \mid$=max$\mid a_{ik} \mid$. If $a_{r_k,s_k} = 0$, stop. Otherwise, go to step 2.

Step 2a:

Interchange the rows of k and r_k of A and b

Step 2b:

Interchange the columns of k and s_k of A and b

step 3:

Form multipliers $a_{ik} \equiv m_{ik} = \dfrac{-a_{ik}}{a_{kk}}$

step 4:

Update the entries of A:

$a_{ij} = a_{ij} + m_{ik}a_{kj}$

Step 5:

Update the entries of b

$b_j = b_i + m_{ik}b_k$

3.10 Improving the solvability of ill-conditioned system of linear equations

The difficulty of solving ill-conditioned system may be negotiated by solving different but equivalent systems which are well-conditioned; well-conditioned systems have matrices with small condition numbers. The approach put forward here constructs a new matrix and a new right-hand side that constitute an instance of an equivalent linear system to the one given which is ill-conditioned. Moreover, this new matrix has a small condition number compared to that of the matrix of the initial linear system. This means that solving this equivalent system must be better than solving the original one by virtue of the difference in the magnitude of the condition numbers of their matrices.

Considering the linear algebraic system

$$\mathrm{A}x = \mathrm{b} \tag{3.5}$$

or

$$\sum a_{ij}x_j = b_i,$$

where $i = 1, ..., n$ and $j = 1, ..., n$.

Reflecting $(a_{n1}, a_{n2}, ..., a_{n(n-1)})$, the last row of equation (3.5) about a three dimensional hyperplane into $(a'_{n1}, a'_{n2}, ..., a'_{n(n-1)})$, resulting in an equivalent system,

$$\mathrm{A}'x = \mathrm{b}' \tag{3.6}$$

of the form:

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, i = 1, ..., n-1,$$

$$\sum_{j=1}^{n} a'_{nj}x_j = b'_n$$

Therefore there is the need to find a'_{nn} and the right-hand side last entry b'_n.

3.10.1 Finding the left-hand side vector: Algorithm

Finding a suitable orthogonal row vector to $(a'_{n1}, a'_{n2}, ..., a'_{nn},)$ i.e., a vector that is orthogonal to $(a'_{n1}, a'_{n2}, ..., a'_{nn},)^T$, is done as follows:

- Reflecting the last entry of equation (3.5) about a three dimensional hyperplane $(a'_{n1}, a'_{n2}, ..., a'_{n(n-1)})$
- compute $a'_{nn} = \dfrac{\sum_{n-1}^{j=1} a_{nj} \times a'_{nj}}{a_{nn}}$

$a'_{n1}, a'_{n2}, ..., a'_{n(n-1)}$ is of the same magnitude as $a_{n1}, a_{n2}, ..., a_{nn}$.

3.10.2 Computing the right-hand side last entry b'_n

Since equations (3.5) and (3.6) are equivalent

i.e.,

$$A\mathbf{x} = \mathbf{b} \equiv A'\mathbf{x} = \mathbf{b}'$$

$$\mathbf{x} = A^{-1}\mathbf{b} \equiv \mathbf{x} = A'^{-1}\mathbf{b}'$$

$$\Rightarrow \mathbf{x} = A^{-1}\mathbf{b} = A'^{-1}\mathbf{b}',$$

and

$$\mathbf{b}' = A'A^{-1}\mathbf{b} \tag{3.7}$$

where

$$A^{-1} = \frac{adj(A)}{|A|} \text{ and } A'^{-1} = \frac{adj(A')}{|A'|}$$

with

$$adj(A) = \begin{pmatrix} A_{11} & A_{21} & . & . & . & A_{n1} \\ . & . & . & . & . & . \\ A_{1n} & A_{2n} & . & . & . & A_{nn} \end{pmatrix} \text{ and } adj(A') = \begin{pmatrix} A'_{11} & A'_{21} & . & . & . & A'_{n1} \\ . & . & . & . & . & . \\ A'_{1n} & A'_{2n} & . & . & . & A'_{nn} \end{pmatrix}$$

A_{ij} and A'_{ij} represent the cofactors of corresponding elements of A and A' respectively, for all i and j. Since only the n^{th} row of the original system has been changed, it is clear that the cofactors of $a_{n1}, a'_{n1}, a_{n2}, a'_{n2}, ..., a_{nn}, a'_{nn}$, are the same. Therefore $A_{n1} = A'_{n1}, A_{n2} = A'_{n2}, ..., A_{nn} = A'_{nn}$. As the two systems are equivalent, b' can be calculated as in (2).

More explicitly:

$$\frac{\begin{pmatrix} a_{11} & a_{21} & . & . & . & a_{n1} \\ . & . & . & . & . & . \\ a_{1n} & a_{2n} & . & . & . & a_{nn} \end{pmatrix} \begin{pmatrix} A_{11} & A_{21} & . & . & . & A_{n1} \\ . & . & . & . & . & . \\ A_{1n} & A_{2n} & . & . & . & A_{nn} \end{pmatrix}}{\Delta} = \begin{pmatrix} b_1 \\ . \\ b'_n \end{pmatrix}$$

or

$$\frac{\begin{pmatrix} \sum_{j=1}^{n} a_{1j}A_{1j} & 0 & . & . & . & 0 \\ 0 & \sum_{j=1}^{n} a_{2j}A_{2j} & 0 & . & . & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \sum_{j=1}^{n} a'_{nj}A_{1j} & \sum_{j=1}^{n} a'_{nj}A_{2j} & . & . & \sum_{j=1}^{n} a'_{nj}A_{nj} & \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ . \\ . \\ b_n \end{pmatrix}}{\Delta} = \begin{pmatrix} b_1 \\ b_2 \\ . \\ . \\ b'_n \end{pmatrix}$$

or

$$\frac{\begin{pmatrix} b_1 \\ b_2 \\ . \\ . \\ \sum_{j=1}^{n} a'_{nj} A_{1j} b_1 + \sum_{j=1}^{n} a'_{nj} A_{2j} b_2 + \cdots + \sum_{j=1}^{n} a'_{nj} A_{nj} b_n \end{pmatrix}}{\Delta} = \begin{pmatrix} b_1 \\ b_2 \\ . \\ . \\ b'_n \end{pmatrix}$$

Now, by identification

$$b'_n = \frac{\sum_{j=1}^{n} a'_{nj} A_{1j} b_1 + \sum_{j=1}^{n} a'_{nj} A_{2j} b_2 + ... \sum_{j=1}^{n} a'_{nj} A_{nj} b_n}{\Delta},$$

or

$$b'_n = \frac{\sum_{j=1}^{n} a'_{nj} (A_{1j} b_1 + A_{2j} b_2 + ... + A_{nj} b_n)}{\Delta},$$

i.e.,

$$b'_n = \frac{\sum_{j=1}^{n} a'_{nj} (\sum_{i=1}^{n} A_{ij} b_i)}{\Delta} \tag{3.8}$$

3.10.3 Solution of $A'\mathbf{x} = \mathbf{b}'$

The solution of the equation can be solved by any of the direct methods. In this thesis however Gaussian elimination with no pivoting is used to compute the solution of the linear system due to its stability for ill conditioned systems that are extremely sensitive to numerical errors.

Chapter 4

ANALYSIS

Matrix algebra is used to solve system of simultaneous linear equations for many mathematical procedures such as the solution to a set of linear and nonlinear equations, interpolation, integration, and differential equations, the solutions reduce to a set of simultaneous linear equations. MATLAB is used to run most of the analysis. In this thesis however, a well-known ill conditioned matrix called the Hilbert matrix is used for the analysis.

In linear algebra, a Hilbert matrix, introduced by Hilbert, is a square matrix $n \times n$ with entries being the unit fractions. The Hilbert matrices are canonical examples of ill-conditioned matrices, making them notoriously difficult to use in numerical computation. In this thesis however, $H_{ij}^{(4\times4)}$ is considered with a suitable right hand side vector. The linear system is

$$H^{(n)}x = b$$

4.1 The Hilbert System

The Hilbert system is made up of the Hilbert matrix and a suitable right hand side vector which are given as $H_{ij}^{(n)} = \frac{1}{i+j-1}$, and $b_i = \sum_{j=1}^{n} H_{ij}$ respectively.

if $n = 4$,

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{25}{12} \\ \frac{77}{60} \\ \frac{342}{360} \\ \frac{638}{840} \end{bmatrix}$$

where

$$H^{4\times4} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix}, \; x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \text{ and } b = \begin{bmatrix} \frac{25}{12} \\ \frac{77}{60} \\ \frac{342}{360} \\ \frac{638}{840} \end{bmatrix}$$

As $n \longrightarrow \infty$, $H_{ij}^{(n)}$ becomes more ill-conditioned.

4.1.1 Rank of Hilbert Matrix

[H] is a 4×4 matrix and the largest square sub-matrix is [H] itself. The determinant of [H], $\det(H) = 1.6534e - 007$, which is not equal to zero therefore the rank of [A] is of order 4.

4.1.2 Nullity of Hilbert Matrix

The nullity of the linear system is the difference between the number of columns and the rank of the linear system. Nullity counts the number of free variables whiles rank counts the pivot variables and the column corresponds to the total number of variables for the coefficient matrix of the linear system. The rank of the linear system is 4 and the number of columns is 4, therefore the nullity of the linear system is given as:

$$\text{Nullity} = \text{n - rank} = 4 - 4 = 0$$

H is an invertible and nonsingular if and only if $Rank(H) = n$

4.1.3 Norm of Hilbert Matrix

The row sum norm (also called the uniform-matrix norm) is used and which is the sum of the absolute value of the elements of each row of $[H]$ and it is defines as:

$$\|H\|_\infty = \max_{1 < i < m} \sum_{j=1}^{n} \mid H_{ij} \mid$$

$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix}$$

$$\begin{aligned} \|H\|_\infty &= \max_{1<i<4} \sum_{j=1}^{4} \mid H_{ij} \mid \\ &= \mathbf{max}[(\mid 1 \mid) + (\mid \frac{1}{2} \mid) + (\mid \frac{1}{3} \mid) + (\mid \frac{1}{4} \mid), (\mid \frac{1}{2} \mid) + (\mid \frac{1}{3} \mid) + (\mid \frac{1}{4}) \mid) + (\mid \frac{1}{5} \mid), \\ &+ (\mid \frac{1}{3} \mid) + (\mid \frac{1}{4} \mid) + (\mid \frac{1}{5} \mid) + (\mid \frac{1}{6}) \mid), (\mid \frac{1}{4} \mid) + (\mid \frac{1}{5} \mid) + (\mid \frac{1}{6} \mid) + (\mid \frac{1}{7}) \mid)] \\ &= \mathbf{max}[\frac{25}{12}, \frac{77}{60}, \frac{342}{360}, \frac{638}{840}] \\ &= \frac{25}{12} \approx 2.0833 \end{aligned}$$

4.1.4 Condition number of Hilbert Matrix

This is given as:

$$\kappa(H) = \| H \|_\infty \| H^{-1} \|_\infty$$

where

$$\|H\|_\infty = \frac{25}{12} \approx 2.0833,$$

$$H^{-1} = \begin{bmatrix} 16 & -120 & 240 & -140 \\ -120 & 1200 & -2700 & 1680 \\ 240 & -2700 & 6480 & -4200 \\ -140 & 1680 & -4200 & 2800 \end{bmatrix}$$

and

$$\begin{aligned}
\|H^{-1}\|_\infty &= \max_{1<i<4} \sum_{j=1}^{4} \mid H_{ij} \mid \\
&= \mathbf{max}[(\mid 16 \mid) + (\mid -120 \mid) + (\mid 240) + (\mid -140), \\
&+ (\mid -120 \mid) + (\mid 1200 \mid) + (\mid -2700 \mid) + (\mid 1680), \\
&+ (\mid 240 \mid) + (\mid -2700 \mid) + (6480 \mid) + (\mid -4200), \\
&(\mid 140) + (\mid 1680) + (\mid -4200) + (\mid 2800)] \\
&= \mathbf{max}[516, 5700, 13620, 8820] \\
&= 13620
\end{aligned}$$

Therefore

$$\begin{aligned}
\kappa(A) = \frac{25}{12} &\approx 2.0833 \times 13620 \\
&= 2.8375 \times 10^{+004}
\end{aligned}$$

Since the condition number is greater than one it suggest that $[H]$ is ill condition, i.e. the solution is not very accurate if input is rounded and then a small roundoff error can have a drastic effect on the output, and so even pivoting techniques will not be useful. However, if the matrix is well-conditioned, then the computerized solution will be quite accurate. Thus the accuracy of the solution depends on the conditioning number of the matrix.

4.1.5 Solution Methods of Hilbert System

There would be comparison of unperturbed and perturbed solutions with their respective errors. Gaussian elimination with no pivoting is used to compute the solutions which is considered to be a stable method in practice. Ill conditioned systems are extremely sensitive to numerical errors and as such pivoting is not much of a help.

Unpertubed Hilbert System

It is assumed that neither the coefficient matrix nor the right hand side vector is contaminated by error.

$$H_{ij}^{(n)} = \frac{1}{i+j-1},\ b_i = \sum_{j=1}^{n} H_{ij}$$

if $n = 4$,

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{25}{12} \\ \frac{77}{60} \\ \frac{342}{360} \\ \frac{638}{840} \end{bmatrix}$$

The matlab solution is given: as

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$\| X \|_\infty = 1$

$\| C \|_\infty = 2.0833$

Perturbing RHS of Hilbert System

In order to estimate the accuracy of the computed solution, the error in the Hilbert system is taken into account. The effect of small perturbations in the right hand side vector is considered, i.e. successively adding and subtracting 0.001 to the entries of b respectively.

$$H_{ij}^{(n)} = \frac{1}{i+j-1},\ b_i = \sum_{j=1}^{n} H_{ij}$$

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{25}{12} + 0.001 \\ \frac{77}{60} - 0.001 \\ \frac{342}{360} + 0.001 \\ \frac{638}{840} - 0.001 \end{bmatrix} = \begin{bmatrix} 2.0843 \\ 1.2823 \\ 0.951 \\ 0.7585 \end{bmatrix}$$

The matlab solution is given as:

$$\begin{bmatrix} x_1^{'} \\ x_2^{'} \\ x_3^{'} \\ x_4^{'} \end{bmatrix} = \begin{bmatrix} 1.5228 \\ -4.7760 \\ 14.8020 \\ -7.9380 \end{bmatrix}$$

Change in the solution vector:

$$\begin{aligned} [\Delta X] &= [X^{'}] - [X] \\ &= \begin{bmatrix} 1.5228 \\ -4.7760 \\ 14.8020 \\ -7.9380 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 0.5228 \\ -5.7760 \\ 13.8020 \\ -8.9382 \end{bmatrix} \end{aligned}$$

$$\therefore \| \Delta X \|_{\infty} = 13.8020$$

Change in the right hand side vector:

$$[\Delta C] = [C^{'}] - [C]$$

$$= \begin{bmatrix} 2.0843 \\ 1.2823 \\ 0.951 \\ -1.5296 \end{bmatrix} - \begin{bmatrix} \frac{25}{12} \\ \frac{77}{60} \\ \frac{342}{360} \\ -1.5286 \end{bmatrix}$$

$$= \begin{bmatrix} 0.001 \\ -0.001 \\ 0.001 \\ -0.001 \end{bmatrix}$$

$$\therefore \| \Delta C \|_{\infty} = 0.001$$

Relative change in the norm of the solution vector

$$\frac{\|\Delta X\|_{\infty}}{\|X\|_{\infty}} = \frac{13.8020}{1} = 13.8020$$

Relative change in the norm of the RHS vector:

$$\frac{\|\Delta C\|_{\infty}}{\|C\|_{\infty}} = \frac{0.001}{2.0833} = 4.8001 \times 10^{-04}$$

Conclusion: small relative change of 4.8001×10^{-04} in right hand side vector norm results in a large relative change in the solution vector norm of 13.8020. The ratio between them are $\frac{13.8020}{4.8001 \times 10^{-04}} = 2.8754 \times 10^{+04} = 28754$

4.1.6 Adequacy of solution

$$\frac{\| \Delta X \|}{\| X + \Delta X \|} \leq \| A \| \| A^{-1} \| \frac{\| \Delta C \|}{\| C \|}$$

and

$$\frac{\| \Delta X \|}{\| X + \Delta X \|} \leq \| A \| \| A^{-1} \| \frac{\| \Delta A \|}{\| A \|}$$

are theorems used to find how many significant digits can be trusted in the solution vector of linear system.

The above algorithms show that the relative error in a solution vector is $\leq$ Cond (A) $\times$ relative error in either the right hand side or the coefficient matrix.
The possible relative error in the solution vector is $\leq$ Cond(A) $\times$ machine epsilon, ϵ_{mach}. Hence Cond(A) $\times$ machine epsilon, ϵ_{mach} gives the number of significant digits (m), for which Cond(A) $\times$ machine epsilon, ϵ_{mach} is less than 0.5×10^{-m}
Considering the Hilbert square linear system:

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{25}{12} \\ \frac{77}{60} \\ \frac{342}{360} \\ \frac{638}{840} \end{bmatrix}$$

where

$$\begin{aligned} \kappa(H) &= \| H \|_\infty \| H^{-1} \|_\infty \\ &= 2.0833 \times 13620 \\ &= 2.8375 \times 10^{+4} \end{aligned}$$

Assuming single precision with 23 bits used in the mantissa for real numbers, the machine epsilon is

$$\begin{aligned} \epsilon &= 2^{-23} \\ &= 0.119209 \times 10^{-6} \\ &\Rightarrow \kappa(H) \times \epsilon_{mach} \\ &= (2.8375 \times 10^{+4}) \times (0.119209 \times 10^{-6}) \\ &= 0.0034 \\ &= 3.4 \times 10^{-3} \end{aligned}$$

The maximum positive value of m for which $\kappa(A) \times \epsilon_{mach} \leq 0.5 \times 10^{-m}$ is given as:

$$\begin{aligned}
3.4 \times 10^{-3} &\leq 0.5 \times 10^{-m} \\
6.8 \times 10^{-3} &\leq 10^{-m} \\
\log 6.8 \times 10^{-3} &\leq \log 10^{-m} \\
-2.1675 &\leq \log 10^{-m} \\
-2.1675 &\leq -m \\
\therefore m &= 2.1675 \\
&\approx 2
\end{aligned}$$

So two (2) significant digits are at least correct in the solution vector.

4.2 Transformed Hilbert System

A new and equivalent system is constructed from that of the original Hilbert system. Solving this new equivalent system must be better than solving the original Hilbert system by the virtue of the huge difference in the magnitude of the condition numbers.

Original Hilbert system:

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{25}{12} \\ \frac{77}{60} \\ \frac{342}{360} \\ \frac{638}{840} \end{bmatrix}$$

Reflecting $(h_{n1}, h_{n2}, ..., h_{n(n-1)})$ of the last row of the system about a three dimensional hyperplane into $(h'_{n1}, h'_{n2}, ..., h'_{n(n-1)})$ resulting in an equivalent system, $H'x = b'$.

Reflecting the entries of the last row $(h'_{n1}, h'_{n2}, h'_{n3})$

$$h_{41} = h'_{41} = -\tfrac{1}{4}$$

$$h_{42} = h'_{42} = -\tfrac{1}{5}$$
$$h_{43} = h'_{43} = -\tfrac{1}{6}$$

Computing h'_{nn}

$$h'_{nn} = \frac{\sum_{n-1}^{j=1} a_{nj} \times h'_{nj}}{h_{nn}}$$

$$h'_{44} = \frac{[h_{41} \times h'_{41}] + [h_{42} \times h'_{42}] + [h_{43} \times a'_{43}]}{a_{44}}$$

$$h'_{44} = \frac{[\frac{1}{4} \times -\frac{1}{4}] + [\frac{1}{5} \times -\frac{1}{5}] + [\frac{1}{6} \times -\frac{1}{6}]}{\frac{1}{7}}$$

$$h'_{44} = \frac{-\frac{469}{3600}}{\frac{1}{7}}$$

$$h'_{44} = -\frac{3283}{3600} \approx -0.91194$$

The transformed Hilbert matrix is given as:

$$H' = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ -\frac{1}{4} & -\frac{1}{5} & -\frac{1}{6} & -\frac{3283}{3600} \end{bmatrix}$$

Computing the right-hand side entry b'_n

$$H\mathbf{x} = \mathbf{b} \equiv H'\mathbf{x} = \mathbf{b}'$$
$$\mathbf{x} = H^{-1}\mathbf{b} \equiv \mathbf{x} = H'^{-1}\mathbf{b}'$$
$$\Rightarrow \mathbf{x} = H^{-1}\mathbf{b} \equiv H'^{-1}\mathbf{b}',$$

Therefore

$$\mathbf{b}' = H' H^{-1} \mathbf{b}$$

but

$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ -\frac{1}{4} & -\frac{1}{5} & -\frac{1}{6} & -\frac{3283}{3600} \end{bmatrix}, H^{-1} = \begin{bmatrix} 16 & -120 & 240 & -140 \\ -120 & 1200 & -2700 & 1680 \\ 240 & -2700 & 6480 & -4200 \\ -140 & 1680 & -4200 & 2800 \end{bmatrix} \text{ and}$$

$$b = \begin{bmatrix} \frac{25}{12} \\ \frac{77}{60} \\ \frac{342}{360} \\ \frac{638}{840} \end{bmatrix}$$

$$\mathbf{b}' = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ -\frac{1}{4} & -\frac{1}{5} & -\frac{1}{6} & -\frac{3283}{3600} \end{bmatrix} \times \begin{bmatrix} 16 & -120 & 240 & -140 \\ -120 & 1200 & -2700 & 1680 \\ 240 & -2700 & 6480 & -4200 \\ -140 & 1680 & -4200 & 2800 \end{bmatrix} \times \begin{bmatrix} \frac{25}{12} \\ \frac{77}{60} \\ \frac{342}{360} \\ \frac{638}{840} \end{bmatrix} = \begin{bmatrix} \frac{25}{12} \\ \frac{77}{60} \\ \frac{342}{360} \\ -1.5286 \end{bmatrix}$$

The new transformed Hilbert system is given as:

$$H' = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ -\frac{1}{4} & -\frac{1}{5} & -\frac{1}{6} & -\frac{3283}{3600} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{25}{12} \\ \frac{77}{60} \\ \frac{342}{360} \\ -1.5286 \end{bmatrix}$$

4.2.1 Rank of Transformed Hilbert Matrix

$[H']$ is a 4×4 matrix and the largest square sub-matrix is $[H']$ itself. The determinant of $[H']$, $\det(H') = -3.5622 \times 10^{-004}$, which is not equal to zero therefore the rank of $[H']$ is of order 4.

4.2.2 Nullity of Transformed Hilbert Matrix

The nullity of the linear system is the difference between the number of columns and the rank of the linear system. Nullity counts the number of free variables whiles rank counts the pivot variables and the column corresponds to the total number of variables for the coefficient matrix of the linear system. The rank of the linear system is 4 and the number of columns is 4, therefore the nullity of the linear system is given as:

$$\text{Nullity} = n \text{ - rank} = 4 \text{ - } 4 = 0$$

H is an invertible and nonsingular if and only if $Rank(H) = n$

4.2.3 Norm of Transformed Hilbert Matrix

The row sum norm (also called the uniform-matrix norm) is used and which is the sum of the absolute value of the elements of each row of $[H']$ and it is defines as:

$$\|H'\|_\infty = \max_{1 < i < m} \sum_{j=1}^{n} \mid H'_{ij} \mid$$

$$H' = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ -\frac{1}{4} & -\frac{1}{5} & -\frac{1}{6} & -\frac{3283}{3600} \end{bmatrix}$$

$$\begin{aligned}
\|H'\|_\infty &= \max_{1 < i < 4} \sum_{j=1}^{4} \mid H'_{ij} \mid \\
&= \mathbf{max}[(\mid 1 \mid) + (\mid \frac{1}{2} \mid) + (\mid \frac{1}{3} \mid) + (\mid \frac{1}{4} \mid), (\mid \frac{1}{2} \mid) + (\mid \frac{1}{3} \mid) + (\mid \frac{1}{4}) \mid) + (\mid \frac{1}{5} \mid), \\
&+ (\mid \frac{1}{3} \mid) + (\mid \frac{1}{4} \mid) + (\mid \frac{1}{5} \mid) + (\mid \frac{1}{6}) \mid), (\mid -\frac{1}{4} \mid) + (\mid -\frac{1}{5} \mid) + (\mid -\frac{1}{6} \mid) + (\mid -\frac{3283}{3600}) \mid)] \\
&= \mathbf{max}[\frac{25}{12}, \frac{77}{60}, \frac{342}{360}, 1.5286] \\
&= \frac{25}{12} \approx 2.0833
\end{aligned}$$

4.2.4 Condition number of Transformed Hilbert Matrix

This is given as:

$$\kappa(H') = \| H' \|_\infty \| H^{-1\prime} \|_\infty$$

where

$$\|H'\|_\infty = \frac{25}{12} \approx 2.0833,$$

$$H^{-1\prime} = \begin{bmatrix} 9.0032 & -36.0390 & 30.0975 & 0.0650 \\ -36.0390 & 192.4679 & -181.1697 & -0.7798 \\ 30.0975 & -181.1697 & 182.9242 & 1.9495 \\ -0.0650 & 0.7798 & -1.9495 & -1.2996 \end{bmatrix}$$

and

$$\begin{aligned} \|H^{-1}\|_\infty &= \max_{1 < i < 4} \sum_{j=1}^{4} | H_{ij} | \\ &= \mathbf{max}[(| 9.0032 |) + (| -36.0390 |) + (| 30.0975 |) + (| 0.0650), \\ &(| -36.0390 |) + (| 192.4679 |) + (| -181.1697 |) + (| -0.7798 |)+ \\ &+ (| 30.0975 |) + (| -181.1697 |) + (| 182.9242 |) + (| -1.9495), \\ &(| -0.0650 |) + (| 0.7798 |) + (| -1.9495 |) + (| -1.2996 |)] \\ &= \mathbf{max}[75.2047, 410.4564, 396.1409, 4.0939] \\ &= 410.4564 \end{aligned}$$

Therefore

$$\begin{aligned} \kappa(H') = \frac{25}{12} &\approx 2.0833 \times 410.4564 \\ &= 855.10 \end{aligned}$$

Though $\kappa(H')$ is significantly different from 1, it is far better than $\kappa(H)$, since this new matrix has a smaller condition number compared to that of the original matrix. This means that solving the transformed Hilbert system must be better

than the original Hilbert system by the virtue of the difference in the magnitude of the condition numbers of their matrices. Moreover the solutions of H' are quite accurate than that of H since the accuracy of the solution depends on the condition number of the matrix and the condition of H' is enhanced.

4.2.5 Solution Methods of Transformed Hilbert Systems

Here there would be comparison of unperturbed and perturbed solutions with their respective errors.

Unperturbed Transformed Hilbert Matrix

It is assumed that neither the coefficient matrix nor the right hand side vector is contaminated by error.

$$H' = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ -\frac{1}{4} & -\frac{1}{5} & -\frac{1}{6} & -\frac{3283}{3600} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{25}{12} \\ \frac{77}{60} \\ \frac{342}{360} \\ -1.5286 \end{bmatrix}$$

The matlab solution is given: as

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$\| X \|_\infty = 1$

$\| C \|_\infty = 2.0833$

Perturbed Transformed Hilbert Matrix

In order to estimate the accuracy of the computed solution, the error in the system is taken into account. The effect of small perturbations of the right hand side vector is considered, i.e. successively adding and subtracting 0.001 to the entries of b respectively.

$$H_{ij}^{(n)} = \frac{1}{i+j-1},\ b_i = \sum_{j=1}^{n} H_{ij}$$

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ -\frac{1}{4} & -\frac{1}{5} & -\frac{1}{6} & -\frac{3283}{3600} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{25}{12} + 0.001 \\ \frac{77}{60} - 0.001 \\ \frac{342}{360} + 0.001 \\ -1.5286 - 0.001 \end{bmatrix} = \begin{bmatrix} 2.0843 \\ 1.2823 \\ 0.951 \\ -1.5296 \end{bmatrix}$$

The matlab solution is given as:

$$\begin{bmatrix} x_1^{'} \\ x_2^{'} \\ x_3^{'} \\ x_4^{'} \end{bmatrix} = \begin{bmatrix} 1.0760 \\ 0.5859 \\ 1.3973 \\ 0.9985 \end{bmatrix}$$

Change in the solution vector:

$$\begin{aligned} [\Delta X] &= [X^{'}] - [X] \\ &= \begin{bmatrix} 1.0760 \\ 0.5859 \\ 1.3973 \\ 0.9985 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 0.0760 \\ -0.4141 \\ 0.3973 \\ -0.0015 \end{bmatrix} \end{aligned}$$

$$\therefore \| \Delta X \|_{\infty} = 0.4141$$

Change in the right hand side vector:

$$
\begin{aligned}
[\Delta C] &= [C^{'}] - [C] \\
&= \begin{bmatrix} 2.0843 \\ 1.2823 \\ 0.951 \\ -1.5296 \end{bmatrix} - \begin{bmatrix} \frac{25}{12} \\ \frac{77}{60} \\ \frac{342}{360} \\ -1.5286 \end{bmatrix} \\
&= \begin{bmatrix} 0.001 \\ -0.001 \\ 0.001 \\ -0.001 \end{bmatrix}
\end{aligned}
$$

$$\therefore \| \Delta C \|_{\infty} = 0.001$$

Relative change in the norm of the solution vector

$$\frac{\|\Delta X\|_{\infty}}{\|X\|_{\infty}} = \frac{0.4141}{1} = 0.4141 = 4.141 \times 10^{-1}$$

Relative change in the norm of the RHS vector:

$$\frac{\|\Delta C\|_{\infty}}{\|C\|_{\infty}} = \frac{0.001}{2.0833} = 4.8001 \times 10^{-04}$$

Conclusion: small relative change of 4.8001×10^{-04} in right hand side vector norm results in a small relative change in the solution vector norm of 4.141×10^{-1}. The ratio between them are $\frac{4.141 \times 10^{-1}}{4.8001 \times 10^{-04}} = 2.8754 \times 10^{+04} = 862.69$

4.2.6 Adequacy of solution

$$\frac{\| \Delta X \|}{\| X + \Delta X \|} \leq \| A \| \| A^{-1} \| \frac{\| \Delta C \|}{\| C \|}$$

and

$$\frac{\| \Delta X \|}{\| X + \Delta X \|} \leq \| A \| \| A^{-1} \| \frac{\| \Delta A \|}{\| A \|}$$

are theorems used to find how many significant digits can be trusted in the solution vector of linear system.

The above algorithms show that the relative error in a solution vector is $\leq$ Cond (A) $\times$ relative error in either the right hand side or the coefficient matrix.

The possible relative error in the solution vector is $\leq$ Cond(A) $\times$ machine epsilon, ϵ_{mach}. Hence Cond(A) $\times$ machine epsilon, ϵ_{mach} gives the number of significant digits (m), for which Cond(A) $\times$ machine epsilon, ϵ_{mach} is less than 0.5×10^{-m}

Considering the new transformed Hilbert system is given as:

$$H' = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ -\frac{1}{4} & -\frac{1}{5} & -\frac{1}{6} & -\frac{3283}{3600} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{25}{12} \\ \frac{77}{60} \\ \frac{342}{360} \\ -1.5286 \end{bmatrix}$$

where

$$\begin{aligned} \kappa(H') &= \| H' \|_\infty \| H^{-1} \|_\infty \\ &= 2.0833 \times 410.4564 \\ &= 855.10 \\ &= 8.5510 \times 10^2 \end{aligned}$$

Assuming single precision with 23 bits used in the mantissa for real numbers, the machine epsilon is

$$\begin{aligned} \epsilon &= 2^{-23} \\ &= 0.119209 \times 10^{-6} \\ &\Rightarrow \kappa(H) \times \epsilon_{mach} \\ &= (8.5510 \times 10^2) \times (0.119209 \times 10^{-6}) \\ &= 1.0194 \times 10^4 \end{aligned}$$

The maximum positive value of m for which $\kappa(A) \times \epsilon_{mach} \leq 0.5 \times 10^{-m}$ is given as:

$$
\begin{aligned}
1.0194 \times 10^{4} &\leq 0.5 \times 10^{-m} \\
2.0387 \times 10^{-4} &\leq 10^{-m} \\
\log 2.0387 \times 10^{-4} &\leq \log 10^{-m} \\
-3.6906 &\leq \log 10^{-m} \\
-3.6906 &\leq -m \\
\therefore m &= 3.6906 \\
&\approx 4
\end{aligned}
$$

So four (4) significant digits are at least correct in the solution vector.

4.3 Discussions of Findings

Errors in system of equations is always a possibility when solving practical problems such as economical and engineering observation, measurements, estimations and experiments. It is a desirable property that system of linear equations must be well condition in its inputs in order to have confidence in the solution, so that we would not get completely different results from slight changes in the input. However, this is not always the case some matrices are very sensitive to small changes in input data which results in a large change in the solution vector. This is the solution of ill conditioned systems which are unreliable, unstable and cannot be trusted to any degree of accuracy and as such it is necessary to convert it into a well conditioned system in order to have any degree of accuracy in their solutions.

Analysis is made between an original Hilbert system and a transformed Hilbert system. Under original Hilbert system comparison is made between unperturbed and perturbed Hilbert system and under transformed Hilbert system

comparison is made between unperturbed and perturbed transformed Hilbert system and the findings of the thesis are as follows:

1. The Hilbert system is consistent since there is a solution and the rank of the coefficient matrix is the same as the number of unknowns; this also means that the solution is also unique. The nullity of the linear system is equal to zero, since the rank and the number of columns is 4.

2. The condition numbers of the original Hilbert matrix and the transformed Hilbert matrix are $\kappa(H) = 2.8375 \times 10^{+004}$ and $\kappa(H') = 855.1175$ respectively. Though both $\kappa(H)$ and $\kappa(H')$ are greater than 1, $\kappa(H')$ is closer to 1 than $\kappa(H)$ and hence $[H']$ is improved and better conditioned than $[H]$. It also suggest that small error in the matrix may produce larger error in the solution of $[H]$ than $[H']$, i.e. the solution of $[H]$ is not very accurate if input is rounded.

3. The relative change in the norm of the solution vector and the relative change in the norm of the RHS vector of $[H]$ are 13.8020 and 4.8001 $\times 10^{-04}$ respectively whiles that of $[H']$ are 4.141×10^{-1} and 4.8001×10^{-04} respectively. It is obviously clear that for $[H']$, small relative change of 4.8001×10^{-04} in the right hand side vector norm results in a small relative change in the solution vector norm of 4.141×10^{-1} and the ratio between these two parameters are $\frac{4.141 \times 10^{-1}}{4.8001 \times 10^{-04}} = 862.69$. For $[H]$, it is also clear that the same small relative change of 4.8001×10^{-04} in the right hand side vector norm results in a larger relative change in the solution vector norm of 13.8020. The ratio between the parameters are $\frac{13.8020}{4.8001 \times 10^{-04}} = 2.8754 \times 10^{+04} = 28754$. This means that $[H]$ contained more error than $[H^{'}]$.

4. For original Hilbert system $[H]$, the unperturbed and perturbed exact so-

lutions are $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ and $\begin{bmatrix} x_1^{'} \\ x_2^{'} \\ x_3^{'} \\ x_4^{'} \end{bmatrix} = \begin{bmatrix} 1.5228 \\ -4.7760 \\ 14.8020 \\ -7.9380 \end{bmatrix}$ respectively.

It can be seen that a small change of 0.001 in the RHS resulted in greatly changes in the solution vector. This means that the solution of the perturbed original Hilbert systems is far different from that of the exact solution. For the transformed Hilbert system $[H']$, its unperturbed and perturbed solutions are

$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ and $\begin{bmatrix} x_1^{'} \\ x_2^{'} \\ x_3^{'} \\ x_4^{'} \end{bmatrix} = \begin{bmatrix} 1.0760 \\ 0.5859 \\ 1.3973 \\ 0.9985 \end{bmatrix}$ respectively. It can be seen that a small change of 0.001 in the RHS resulted in small changes in the solution vector. This means that the solution of the perturbed transformed Hilbert systems is not far different from that of the exact solution.

This suggest that the solution of $[H']$ are relatively stable and reliable than that of $[H]$ if their RHS entries are slightly perturbed.

Chapter 5

CONCLUSION

This thesis is a theoretical academic work which explores on a well known ill condition matrix called the Hilbert matrix. The dimension of Hilbert matrix used is a 4×4 with entries being the unit fractions.

The approach put forward here constructs a new matrix and a new right-hand side that constitute an instance of an equivalent and a transformed linear system to the one given which is ill-conditioned, this new matrix has a small condition number compared to that of the matrix of the initial linear system. Hence converting an ill conditioned Hilbert system to an improved system have been achieved.

The difficulty of solving ill-conditioned system is negotiated by solving different but equivalent and transformed systems which are well-conditioned. This means that solving this equivalent and transformed system which constitute the improved system must be better than solving the original one by virtue of the difference in the magnitude of the condition numbers of their matrices and as a result, the solution of the ill-conditioned Hilbert system is computed via the solution of an equivalent and improved system.

It is obviously clear that for $[H']$, small relative change of 4.8001×10^{-04} in the right hand side vector norm results in a small relative change in the solution vector norm of 4.141×10^{-1} and the ratio between these two parameters are 862.69. For $[H]$, it is also clear that the same small relative change of 4.8001×10^{-04} in the right hand side vector norm results in a larger relative change in the solution vector norm of 13.8020. The ratio between the parameters are =28754. This means that the solution of $[H']$ is relative stable and reliable than that of $[H]$.

5.1 RECOMMENDATION

This thesis is recommended for university lecturers, researches, students and policy makers. University lecturers can use the concept and approaches in the teaching of numerical errors as well as serving as a base or a reference material for researches to dive into the problem for onward suggestions. It will also aid as an introductory step for university students to develop interest in working at the topic, and moreover it can serve as a curricular material for policy makers in which recommendations can be made to enhance and broaden the horizons of teaching and learning of the research topic.

REFERENCES

Ballard, G., Demmel, J., Lipshitz, B., Schwartz, O., and Toledo, S. (2005). Communication efficient gaussian elimination with partial pivoting using a shape morphing data layout. *AMS*, 3:74–82.

Castel, J., Migallon, V., and Penades, J. (1998). Convergencce of non-stationary parallel multisplitting methods for hermitian positive definite matrices. *Mathematics Of Computation*, Volume 67, Number 221,:209–220.

Choi, S. C. T. (2006). *Iterative Methods for Singular Linear Equations and Least-square Problems*. PhD thesis, Stanford University.

Cortes, V. and Pena, J. (2006). Growth factor and expected growth factor of some pivoting strategies. *Journal of computational and applied mathematics*, pages 292–303.

Dekker, T., Hoffmann, W., and Potma, K. (1994). Parallel algorithms for solving large linear systems. *Journal of computational and applied mathematics*, 50:221–232.

Dumas, J.-G., Pernet, C., and Sultan, Z. (2013). Simultaneous computation of the row and column rank profiles. *arXiv:1301.4438*, 1.

Enright, M. (1978). Well condition systems versus ill-condition systems. *Mathematics of Compution*, 5:15–23.

Foster, L. V. (1994). Gaussian elimination with partial pivoting can fail in practice. *SIAM Journal*, 15:1354–1362.

Foster, L. V. (1997). The growth factor and efficiency of gaussian elimination with rook pivoting. *Journal of computational and applied mathematics*, 86:177–194.

Grcar, J. F. (2011). John von neumann's analysis of gaussian elimination and the origins of modern numerical analysis. *Society for Industrial and Applied Mathematics*, Vol. 53, No. 4,:607-682.

Higham, N. J. (2009). How accurate is gaussian elimination? *AMS*.

Higham, N. J. (2011). Gaussian elimination. *WIREs Computational Statistics*, 3:230–238.

Higham, N. J. and Higham, D. J. (1989). Large growth factors in gaussian elimination with pivoting. *SIAM Journal*, 10. No. 2:155–164.

Huard, J. (1979). How ill is ill-condition systems? *SIAM Journal*, 7:3–9.

Khabou, A. (2013). Dense matrix computations : communication cost and numerical stability. Master's thesis, Universite Paris-Sud.

Li, H.-B., Guo, Q.-P., and Gu, X.-M. (2013). A note on the growth factor in gaussian elimination for generalized higham matrices. *arXiv:130511*, 2:15–23.

Li, X. S. and Demmel, J. W. (2004). Making sparese gaussian elimination scalable. *SC Journal*, 8:20–36.

Lipshitz, B., Ballard, G., Demmel, J., and Schwartz, O. (2004). Communication-avoiding parallel strassen: Implementation and performance. *Computer Science and Mathematics Division*, 3:5–16.

Marshall, J. and Olkin, W. (1968). Stability of differential equations. *AMS*, 8:13–19.

Mead, J., Renaut, R. A., and Welfert, B. D. (2001). Stabilty of a pivoting strategy for parallel gaussian elimination. *BIT*, 41, No. 3:633–639.

Olson, M. (2009). Scaling and stability of ill-conditioned systems. *AMS*, 3:5–10.

Pan, V. Y. and Qian, G. (2012). More on the power of randomized matrix multiplication. *arxiv:1212.4560v2*.

Pan, V. Y., Qian, G., and Yan, X. (2013). Supporting genp with random multipliers. *arXiv:1312.3805*, 1.

Parlett, B. N. and Landis, T. L. (2004). Methods for scaling to doubly stochastic form. *SIAM Journal*, pages 200–227.

Poole, G. and Neal, L. (1991). A geometric analysis of gaussian elimination. Technical report, East Tennessee State Unicersity.

Poole, G. and Neal, L. (2000). The rook's pivoting strategy. *Journal of Computational and Applied Mathematics*, 123:353–369.

Poole, G. and Neal, L. (2002). Gaussian elimination: When is scaling beneficial? *SIAM Journal*, 8:12–22.

Sankar, A. (2004). *Smoothed Analysis of Gaussian Elimination.* PhD thesis, Massachusetts Institute Of Technology. Cambridge U.S.A.

Sinkhorn, J. and Knopp, B. (1967). Direct methods for solving gaussian elimination. *GMS Journal*, 3:15–19.

Skeel, R. D. (1980). Iterative refinement implies numerical stability for gaussian elimination. *Mathematics Of Compution*, 817-832:817–832.

Thorson, J. (2001). Gaussian elimination on a banded matrix. *SIAM Journal, Vol, No. 3:105-108.*

Trefethen, L. N. and Schreiber, R. S. (1990). Average-case stabilty of gaussian elimination. *SIAM Journal*, Vol. 11, No. 3:335–360.

Vecharynski, E. (2006). *Preconditioned Iterative Methods For Linear Systems, Eigenvalue And Singular Value Problems.* PhD thesis, Belarus State University.

Wikipedia (2014). Floating point. www.en.wikipedia.org/wiki/floating point.

Xue, X. J., Kozaczek, K. J., Kurtzl, S. K., and Kurtz, D. S. (2000). A direct algorithm for solving ill-conditioned linear algebraic systems. *International Centre for Diffraction Data 2000*, 42:629–634.

Yeung, M. (2004). Probabilistic analysis of complex gaussian elimination without pivoting. *Linear Algebra and its Applications*, 384:109–134.

Yeung, M. and Chan, T. F. (1997). Probabilistics analysis of gaussian elimination without pivoting. *SIAM Journal*, 18, No.2:499–517.