SCALING ALGORITHMS FOR MATRICES

By

CHIN-CHIEH CHIANG

Bachelor of Science in Applied Mathematics
Feng Chia University
Taichung, TAIWAN
1987

Master of Science in Mathematics
University of Connecticut
Storrs, Connecticut, USA
1994

Master of Science in Computer Science
Oklahoma State University
Stillwater, Oklahoma, USA
2002

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
December, 2007

SCALING ALGORITHMS FOR MATRICES

Dissertation Approved:

Dr. John P. Chandler

Dissertation Advisor

Dr. George E. Hedrick

Dr. Blayne E. Mayfield

Dr. David Wright

Dr. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to my major advisor, Dr. John Chandler for his intelligent supervision, constructive guidance, inspiration and friendship. My sincere appreciation extends to my other committee members Dr. Hedrick, Dr. Mayfield, and Dr. Wright whose guidance, assistance, encouragement, and friendship are also invaluable.

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

1.1 Prologue

1.1.1 Floating point arithmetic and finite precision computation

Numbers that are not integers are usually represented in computers or calculators as "floating point" numbers.

The floating point number system used in any computer has a base, a small integer. The base used in most computers is two, although the floating point base used in IBM mainframe computers (IBM 360, 370, 3090, etc.) is 16. The floating point base used in all electronic hand calculators is ten.

Every floating point number is a fraction, either proper or improper. The numerator of the fraction is the "mantissa" of the floating point number and the denominator is some integral power of the base. The mantissa contains only a limited number of digits. All commonly used floating point number systems can represent all small positive and negative integers exactly, but they can represent most fractions and all irrational numbers with only finite precision, that is, with some nonzero error.

Some operations in finite precision can introduce errors that would not occur in perfectly precise computation. For example, many results of floating point operations must be rounded to fit into a floating point result. Related to this is the fact that a small number can "fall off the end of a register" when it is added to, or subtracted from, a larger value in the register. Thus, the result of $((1.0 + X) - 1.0)$ can be equal to zero in floating point arithmetic, and often is, even when X is nonzero. When X is small enough in magnitude, the

computed result is 0.0 when the correct result is X, a relative error of 1.0 or a percentage error of 100%.

Relative error can be greatly magnified by "loss of leading significant digits due to subtractive cancellation". This is not an introduction of a new error, but can nevertheless be important. It is discussed in textbooks on numerical computation.

1.1.2 The condition of linear systems

A numerical problem is said to be "ill-conditioned" if a small relative change in the numbers that specify the problem is capable of causing a large relative change in the exact mathematical solution of the problem. Conditioning has nothing to do with the algorithm being used to solve the problem.

Solving a set of $n = 2$ linear equations in two variables is an ill-conditioned problem if the two lines representing the two equations are nearly, but not exactly, parallel. Obviously, in this case, certain kinds of changes in one of the equations could cause the corresponding line to shift in such a way as to cause the point of intersection of the two lines to move a long distance.

Conditioning is measured by a "condition number" that is essentially the factor by which an error in the specification of the problem can be magnified during the solution of the problem. The condition number that is usually used with a system $Ax = b$ of n linear equations in n variables $x[1], \cdots, x[n]$ is

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

[Wilkinson [29]], where $\|A\|$ is a norm of the coefficient matrix A. Unfortunately this definition is sensitive to the scaling of A. For example, the matrix

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 10^{20} \end{bmatrix}.$$

presents no problem in obtaining an accurate numerical solution even in finite precision floating point arithmetic. However, $cond(A) = 10^{20}$, indicating that unavoidable loss of precision could occur in the solution of any system with this matrix of coefficients, which is not true at all.

In this work we will investigate definitions of conditioning of matrices such that these definitions are not affected by scaling.

1.1.3 Pivoting

In applying Gaussian elimination to solve a system of n linear algebraic equations in n variables, it is necessary to use pivoting in order to guarantee to be able to solve any system with a nonsingular matrix of coefficients [Forsythe & Moler [10]]. Pivoting also helps to make Gaussian elimination more stable when solving a system using the finite precision floating point arithmetic available in digital computers [Forsythe & Moler [10]]. A stable algorithm is one that does not unnecessarily amplify any rounding errors that occur during the solution process.

The most common kind of pivoting used is partial pivoting. In partial pivoting, a search is carried out at each stage for the element of largest magnitude on or below the current diagonal element. Then the equation containing the largest magnitude element in this column is exchanged with the equation containing the diagonal element before proceeding with the current stage of elimination. It is now known [Foster [12], Wright [30]] that even with partial pivoting, Gaussian elimination can be an unstable algorithm in a few practical problems, but this is rare and the algorithm will continue to be widely used. Partial pivoting requires time only $O(n^2)$, and this does not increase materially the total time required for elimination, which is $O(n^3)$, on any sizable linear system.

Complete pivoting involves a search of the entire unreduced square portion of the coefficient matrix, A. Complete pivoting is known to be, in a particular technical sense, a stable algorithm in all cases [Wilkinson [29]]. To quote Wilkinson, "Complete pivoting is never a

very poor strategy." Complete pivoting requires time $O(n^3)$, and hence does increase materially the total time required for elimination. For this reason, the use of complete pivoting is not common.

1.1.4 The stability of algorithms

An algorithm that solves every well-conditioned problem in its domain of application is called "stable". A stable algorithm never causes unnecessary amplification of relative errors. An ill-conditioned problem requires amplification of relative errors and cannot be solved reliably in finite precision arithmetic by any algorithm, stable or unstable, in the sense that the exact solution can be affected greatly by tiny changes in the problem, causing the solution to be of limited use for practical applications.

1.1.5 Scaling

The choice of successive pivots is affected by the scaling of a linear problem. Also, it is a fact that the accuracy of the numerical solution of the problem is not affected at all by implicit scaling except as the scaling affects this order of pivoting.

Among the $n!$ different orders (permutations) of pivoting, it is often true that some permutations allow Gaussian elimination with partial pivoting (GEPP) to produce a stable solution, whereas other permutations do not. Furthermore, a suitable rescaling of the coefficient matrix A can cause GEPP to use any specified permutation of pivots. Therefore, the use of GEPP without some effort to scale the problem satisfactorily can make the pivoting process worthless.

For Gaussian elimination with partial pivoting, only the scaling of the equations makes a difference in the order of selection of pivots. That is, the rows of the matrix A should be properly scaled but the scaling of the columns is irrelevant. Wilkinson [29] recommended dividing each linear equation by the coefficient of largest magnitude in that equation, so that all elements $a[i, j]$ of the matrix A will be lie in the interval $[-1, +1]$ and every row

of A contains at least one element of magnitude 1.0 . Recent researchers have tended to discourage even this row scaling, calling it unnecessary. Wilkinson [29] and Forsythe and Moler [10] showed, in our opinion, that scaling is necessary.

In addition to helping GEPP to choose a better permutation of pivots, there are at least three other reasons to scale a problem before solving it. We will discuss all of these reasons in the proposed dissertation.

In this work we will investigate a complete scaling of a general matrix A, that is, we will seek to find diagonal matrices L and R such that the matrix LAR has all of its elements in the interval $[-1, +1]$, such that LAR has at least one element of magnitude 1.0 in every row and in every column, and such that the element(s) of smallest magnitude in LAR are as large in magnitude as possible.

1.1.6 The SCALGM scaling algorithm

We could achieve the first two objectives of scaling mentioned above by simply dividing every row of A by its coefficient of largest magnitude, then doing the same for every column of A. This procedure has at least three disadvantages, however. First, there is no reason why we should scale the rows first and then the columns, and it can easily be verified that scaling the columns first and then the rows can produce quite a different result. Second, scaling the rows and then the columns, or vice versa, does not make the elements of smallest magnitude in the scaled matrix as large as possible. Third, and most important, if the matrix A is symmetric, either of these two algorithms will usually destroy the symmetry of A. This is disastrous, as a linear system with a symmetric matrix can be solved in about half the time required for a general nonsymmetric system, so that for practical computation it is important to maintain any symmetry that may be present in A .

The basic operation of SCALGM consists of

1. Scale A by rows, then by columns, producing diagonal scaling matrices L_1 and R_1 .

2. Scale A (the original matrix A) by columns, then by rows, producing new diagonal scaling matrices L_2 and R_2 .

3. Compute $L = \sqrt{L_1 L_2}$ and $R = \sqrt{R_1 R_2}$. L and R are the final scaling matrices for this basic operation.

It is obvious that this operation preserves the symmetry of A, if A is in fact a symmetric matrix. It is not obvious, and in fact not true, that it achieves the other objectives, of having all scaled elements lie in $[-1, +1]$, of having at least one element of magnitude in every row and every column, and of causing the elements of smallest magnitude to be as large as possible.

Call the above basic step "scale down" because the magnitudes of the elements in the scaled matrix will be less than or equal to 1.0.

It is also possible to "scale up" by applying the above basic operation to the inverses $1/a[i, j]$ of the elements of $a[i, j]$ that are not zero. If the resulting elements are reinverted, the elements of the scaled matrix will now usually be greater than 1.0 in magnitude, and never less than 1.0.

SCALGM consists in iterating these two operations: scale up, scale down; scale up, scale down; etc. The final operation is always a "scale down". It was observed by Chandler [7] that this algorithm seemed always to converge, that it produces a scaled matrix having its small elements fairly large (and possibly as large as possible), and that it produces elements of plus or minus 1.0 in many rows and columns of the resulting scaled matrix.

This research has shown that some rows or columns may have no elements of magnitude 1.0 in them, and has amended SCALGM to remedy this drawback. In this research Chiang will prove that SCALGM does in fact make the smallest elements of the scaled matrix as large as possible. It is believed that some of the other (non-smallest) elements of LAR are also as large as possible, but this is complicated and has not yet been proved.

It is important to consider the diagonal matrices L and R, and not just the result LAR of each step, for the following reason. If A is an $m \times n$ matrix it contains mn elements whereas L and R together contain only $m+n$ nonzero elements. SCALGM is susceptible to certain methods for accelerating the convergence of the iterative process, and these acceleration methods should be applied to as few quantities as possible, that is, to the m+n nonzero elements of L and R rather than to the mn elements of LAR.

1.2 Preview of Our Study and Results

Matrix scaling or equilibration has been an important subject in the scientific computing on linear algebraic systems. A linear algebraic system of m equations in n unknowns $x_j, 1 \leq j \leq n$,

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

is often written in matrix notation as

$$Ax = b \ .$$

where $A = [a_{ij}] \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$.

The definitions of the *condition number* and *equilibration* of matrices are given as below and will be used later.

Definition 1.2.1 (Condition Number) *For any nonsingular matrix A we define the* condition number *of A, denoted by $cond(A)$ or $\kappa(A)$, to be the number $\|A\| \cdot \|A^{-1}\|$. (The condition number depends on the norm used.)*

Definition 1.2.2 (Row Equilibrated) *A matrix $A \in \mathbf{R}^{m \times n}$ is* row equilibrated *if all its rows have the same length in some norm.*

Definition 1.2.3 (Column Equilibrated) *A matrix $A \in \mathbf{R}^{m \times n}$ is* column equilibrated *if all its columns have the same length in some norm.*

Definition 1.2.4 (Equilibrated) *A matrix $A \in \mathbf{R}^{m\times n}$ is* equilibrated *if it is both row equilibrated and column equilibrated.*

The objective of scaling on matrices A is to find suitable row and column scale factors, written as diagonal matrices, D and E, respectively, such that the scaled matrices DAE satisfy the desired properties. For instances:

- Forsythe and Moler [10] present the motivation of scaling to make pivoting work well.

- Berman, Parlett and Plemmons [3] give a necessary and sufficient condition for A to be diagonally equivalent to an orthogonal matrix Q, and offer an algorithm that either it produces positive diagonal matrices D and E such that DAE is orthogonal or it fails if no such pair D, E exists. (Note: Every orthogonal matrix Q is really orthonormal, the Euclidean norm of every column is unity, and hence is fairly well scaled.)

- Bunch [5] presents an algorithm for any symmetric matrix A (with no null rows) such that the scaled matrix DAE is equilibrated in the ∞-norm.

- Curtis and Reid [8] propose an algorithm for scaling based on the assumption that the given matrix can be scaled into the required form that all scaled matrix elements are of comparable size. Thus, the usual pivotal strategies for Gaussian elimination can be applied on the scaled matrix.

- Fulkerson and Wolfe [13] present a method for finding scale factors which minimize the ratio of the matrix entry of largest absolute value to that of smallest non-zero absolute value. They say it is believed that such a number is a useful condition number.

- Rothblum, Schneider and Schneider [21] present an algorithm so that for a given nonnegative symmetric matrix A and a positive vector r, it either finds a positive

diagonal matrix D such that $B = DAD$ has row maxima prescribed by r or shows that no such D exists.

- Parlett and Reinsch [20] present an algorithm based on the work of Osborne [19] on balancing a matrix for calculation of eigenvalues and eigenvectors.

- Skeel [23] shows the effect of scaling on the stability of Gaussian elimination.

- The problem of optimal scaling of matrices with respect to the condition number κ has been extensively studied, as seen in papers presented by Bauer [2], Braatz and Morari [4], Businger [6], Forsythe and Strauss [11], Golub and Varah [14], McCarthy and Strang [18], Rump [22], and Watson [28].

We present an iterative algorithm, called SCALGM, that works on any given nonzero matrix $A \in \mathbf{R}^{m \times n}$ to produce the row and column scale factors in the form of diagonal matrices D and E, respectively, such that the scaled matrix DAE satisfies the properties listed below:

P1: the maximum magnitude of elements in DAE is 1.

P2: the nonzero rows and columns of DAE are equilibrated in ∞-norm.

P3: the ratio of the minimum magnitude of nonzero elements to the maximum magnitude of elements in DAE is maximized.

In fact, property **P3** holds for the SCALGM algorithm provided A is a nonzero matrix. From the above properties, the range of the magnitude of nonzero elements in DAE is within an interval $[m, 1]$ for some $m \in \mathbf{R}$ depending on the given matrix A, and such that m is maximized. We give numerical evidence as well as a theoretical proof for property **P3**.

CHAPTER 2

The SCALGM Algorithm

SCALGM takes the name from the geometric mean of the row and column scale factors. The original SCALGM algorithm was invented by J. P. Chandler [7] and the basic idea is described in Algorithm 1.

Algorithm 1 Original_SCALGM(A)

1: Scale up by rows and then by columns
2: Scale up (the original matrix) by column and then by rows
3: Take the geometric mean of these two results
4: Scale down by rows and then by columns
5: Scale down (the original matrix) by columns and then by rows
6: Take the geometric mean of these two results

SCALGM can take any general real $m \times n$ matrix $A \in \mathbf{R}^{m \times n}$ as its input, and then return a scaled matrix S such that the ratio of the nonzero minimum magnitude to the maximum magnitude of elements in S is maximized. However, S is not (row and column) equilibrated. Now we call the original SCALGM algorithm as phase one, and append to it another scaling method, namely phase two, which take the returned matrix from phase one as its input, such that the returned scaled matrix by phase two preserves the maximum ratio property as in phase one, and is equilibrated with ∞-norm 1 of each nonzero row and column.

Without loss of generality, we assume all elements of the matrices are nonnegative since the row and column scale factors chosen in each iteration of the SCALGM algorithm are based on the magnitudes of the elements in the involved matrices.

2.1 Big Picture

The SCALGM algorithm consists of two phases of scaling. It takes any given matrix $A \in \mathbf{R}^{m\times n}$ as its input, as shown in Figure 2.1. The first phase consists of two procedures, $ScaleUP(\ )$ and $ScaleDown(\ )$, which are performed sequentially during each iteration until some criterion is met. The second phase just consists of one procedure $ScaleDown(\ )$, which is the same one as used in the first phase and is performed repeatedly until some criterion is met.

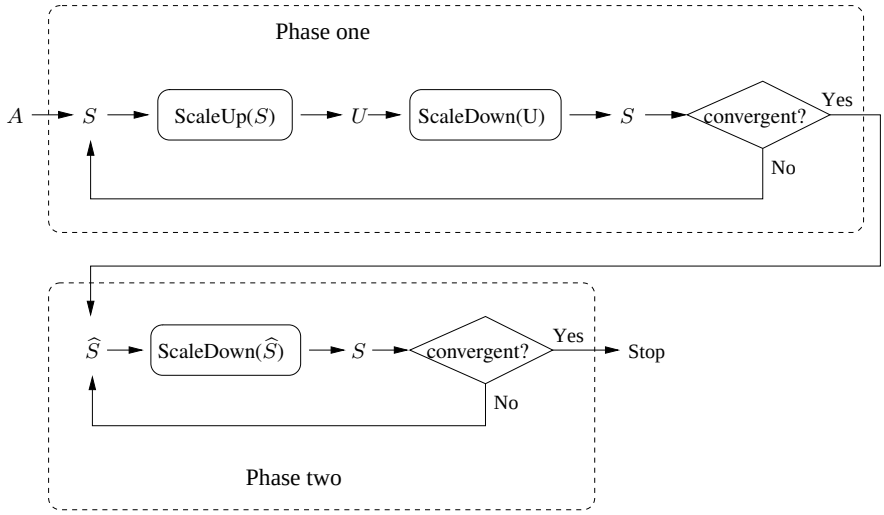

Figure 2.1: Flowchart of the SCALGM algorithm

At the beginning, the algorithm takes the given matrix A as its input, assigns it into another matrix $S^{(0)}$,

$$S^{(0)} \leftarrow A\ ,$$

and then enters phase one for looping. In the kth $(k \geq 1)$ iteration of phase one, the procedure $ScaleUp(\ )$ takes $S^{(k-1)}$ as its input to determine the row and column scale factors and uses these factors to produce the scaled matrix $U^{(k)}$,

$$U^{(k)} \leftarrow ScaleUp(S^{(k-1)})\ ,$$

and so as to determine the maximum magnitude $max^{(k)}$ among all elements in $U^{(k)}$,

$$max^{(k)} \leftarrow \max\{|u_{ij}| : U^{(k)} = [u_{ij}]\} \ .$$

Next, the procedure $ScaleDown(\ )$ is invoked by taking $U^{(k)}$ as its input to determine the row and column scale factors and use them to produce the scaled matrix $S^{(k)}$,

$$S^{(k)} \leftarrow ScaleDown(U^{(k)}) \ ,$$

and thus to determine the minimum magnitude $min^{(k)}$ among all nonzero elements in $S^{(k)}$,

$$min^{(k)} \leftarrow \min\{|s_{ij}| > 0 : S^{(k)} = [s_{ij}]\} \ .$$

Phase one repeats the process of calling procedures

$$ScaleUp(S^{(k-1)}) \text{ and } ScaleDown(U^{(k)})$$

alternately until

$$\lim_{k\to\infty} (max^{(k)} \cdot min^{(k)}) = 1 \ ,$$

which is used as the criterion of convergence for phase one. That is, a tolerance is set for phase one, which stops when

$$|max^{(k)} \cdot min^{(k)} - 1| < tolerance \ ,$$

for some sufficiently large iteration k.

The procedure $PhaseOne(\ )$ is summarized in Algorithm 2.

In phase two, the procedure $ScaleDown(\ )$ is called repeatedly. Phase two takes the

Algorithm 2 PhaseOne(A)

$S \leftarrow A$
repeat
  $U \leftarrow$ ScaleUp(S)
  $\max U \leftarrow \max\{u_{ij} : U = [u_{ij}]\}$
  $S \leftarrow$ ScaleDown(U)
  $\min S \leftarrow \min\{s_{ij} : S = [s_{ij}]\}$
until $|\max U \cdot \min S - 1.0| <$tolerance
return S

scaled matrix $\widehat{S}$ which is returned from phase one as its input,

$$S^{(0)} \leftarrow \widehat{S}\,,$$

and repeatedly computes the scaled matrices as below,

$$S^{(k)} \leftarrow ScaleDown(S^{(k-1)})\,,$$

for $k = 1, 2, \ldots,$ until

$$S^{(k)} \rightarrow S \quad entrywise \quad as \quad k \rightarrow \infty\,.$$

Again a tolerance is set for phase two , which terminates when

$$\max\{|\triangle s_{ij}^{(k)}| : S^{(k)} - S^{(k-1)} = [\triangle s_{ij}^{(k)}]\} < tolerance\,,$$

for some sufficiently large iteration k.

Algorithm 3 is the procedure of $PhaseTwo(\ )$.

Algorithm 3 PhaseTwo(S)

repeat
  $\widehat{S} \leftarrow S$
  $S \leftarrow$ ScaleDown($\widehat{S}$)
until $\max|S - \widehat{S}| <$ tolerance
return S

The overview of our scaling method SCALGM is given in Algorithm 4. The detailed scenario of procedures $ScaleUp(\ )$ and $ScaleDown(\ )$ will be introduced in the following sections.

Algorithm 4 SCALGM(A)

// Phase One
$S \leftarrow A$
repeat
  $U \leftarrow$ ScaleUp(S)
  $\max U \leftarrow \max\{u_{ij} : U = [u_{ij}]\}$
  $S \leftarrow$ ScaleDown(U)
  $\min S \leftarrow \min\{s_{ij} : S = [s_{ij}]\}$
until $|\min S \cdot \max U - 1.0| <$tolerance
// Phase Two
repeat
  $\widehat{S} \leftarrow S$
  $S \leftarrow$ ScaleDown($\widehat{S}$)
until $\max |S - \widehat{S}| <$ tolerance
return S

2.2 Scale Up Procedure

The purpose of the procedure $ScaleUp(\ )$ is to determine the row and column scale factors depending on the input matrix and then scale the input matrix such that the magnitudes of the nonzero scaled elements are greater than or equal to 1. Let $S = [s_{ij}] \in \mathbf{R}^{m\times n}$ be the input matrix of the procedure $ScaleUp(\ )$ and let $U = [u_{ij}]$ be the output matrix as below:

$$U \leftarrow ScaleUp(S)$$

Thus, the maximum magnitude of elements in the scaled matrix, $\max |U|$, is obtained and will be used later to check for the convergence criterion. For each row and column of S, there are two row and two column scale factors, as shown in Figure 2.2.

Let α_i be the minimum magnitude of nonzero elements in the ith row of S, where $1 \leq i \leq m$, and let β_j be the minimum magnitude of nonzero elements in the jth column of S,

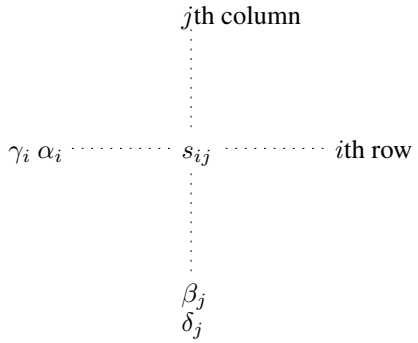

Figure 2.2: Scale up factors

where $1 \leq j \leq n$. Thus,

$$\begin{aligned} \alpha_i &\leftarrow \min\{|s_{ij}| > 0 : 1 \leq j \leq n\}, for\ 1 \leq i \leq m, and \\ \beta_j &\leftarrow \min\{|s_{ij}| > 0 : 1 \leq i \leq m\}, for\ 1 \leq j \leq n. \end{aligned}$$

Next, based on the $\alpha_{i's}$ and $\beta_{i's}$ information, let γ_i be the minimum magnitude of nonzero elements divided by their respective $\beta_{i's}$ in the ith row of S, where $1 \leq i \leq m$, and let δ_j be the minimum magnitude of nonzero elements divided by their respective $\alpha_{i's}$ in the jth column of S, where $1 \leq j \leq n$. Thus,

$$\begin{aligned} \gamma_i &\leftarrow \min\{\frac{|s_{ij}|}{\beta_j} > 0 : 1 \leq j \leq n\}, for\ 1 \leq i \leq m, and \\ \delta_j &\leftarrow \min\{\frac{|s_{ij}|}{\alpha_i} > 0 : 1 \leq i \leq m\}, for\ 1 \leq j \leq n. \end{aligned}$$

Note that if there exists zero row or column vectors in S, we simply take their scale factors for the row or the columns to be $1's$. That is,

$$\alpha_i, \gamma_i \leftarrow 1, if\ the\ ith\ row\ of\ S\ is\ a\ zero\ vector, and$$

$$\beta_j, \delta_j \leftarrow 1, if\ the\ jth\ column\ of\ S\ is\ a\ zero\ vector$$

Algorithm 5 for $ScaleUp(S)$ is under the assumption that S does not contain any null rows or columns, but these zero rows or columns are handled easily.

Algorithm 5 ScaleUp(S)

for row $i \leftarrow 1$ to m **do**
  $\alpha_i \leftarrow \min\{|s_{ij}| > 0 : 1 \leq j \leq n\}$
end for
for column $j \leftarrow 1$ to n **do**
  $\beta_j \leftarrow \min\{|s_{ij}| > 0 : 1 \leq i \leq m\}$
end for
for row $i \leftarrow 1$ to m **do**
  $\gamma_i \leftarrow \{|s_{ij}|/\beta_j > 0 : 1 \leq j \leq n\}$
end for
for column $j \leftarrow 1$ to n **do**
  $\delta_j \leftarrow \min\{|s_{ij}|/\alpha_i > 0 : 1 \leq i \leq m\}$
end for
for $i \leftarrow 1$ to m **do**
  for $j \leftarrow 1$ to n **do**
    $u_{ij} \leftarrow s_{ij}/\sqrt{\alpha_i \gamma_i \beta_j \delta_j}$
  end for
end for

Without the requirements of no null rows or columns in matrix S, we use the algorithm below to find the row or column scale factor min. That is, if $\mathbf{s} = [s_1, s_2, \cdots, s_n]$ is any row or column vector of S, then $min = 1$ provided $\mathbf{s} = \mathbf{0}$, and otherwise min is the minimum nonzero magnitude of elements in $\mathbf{s}$.

2.3 Scale Down Procedure

The purpose of the procedure $ScaleDown(\ )$ is to find the row and column scale factors depending on the input matrix, then scale the input matrix such that the magnitudes of the scaled elements are less than or equal to 1. Let $U = [u_{ij}] \in \mathbf{R}^{m \times n}$ be the input matrix of

Algorithm 6 FindNonzeroMin(s $= [s_i]$)

```
1: for i ← 1 to n do
2:    if s[i] ≠ 0 then
3:       min ← |s[i]|
4:       start ← i
5:       break
6:    end if
7: end for
8: if i = n + 1 then
9:    min ← 1
10: else
11:    for i ← start + 1 to n do
12:       if |s[i]| > 0 and |s[i]| < min then
13:          min ← |s[i]|
14:       end if
15:    end for
16: end if
17: return min
```

the procedure $ScaleDown(\ )$ and let $S = [s_{ij}]$ be the output as below:

$$S \leftarrow ScaleUp(U)$$

Thus, the minimum magnitude of the nonzero elements in the scaled matrix, $\min|S|$, is obtained and used with $\max|U|$ obtained from $ScaleUp(\ )$ to check for the convergence criterion. For each row and column of U, there are two row and two column scale factors, as shown in Figure 2.3.

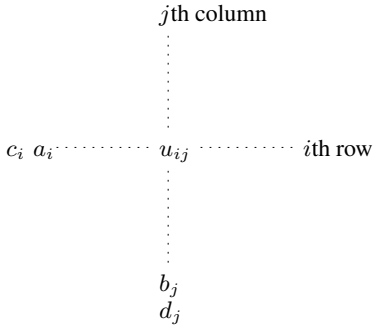

Figure 2.3: Scale down factors

Let a_i be the maximum magnitude of elements in the ith row of U, where $1 \leq i \leq m$, and

let b_j be the maximum magnitude of elements in the jth column of U, where $1 \leq j \leq n$. Thus,

$$\begin{aligned} a_i &\leftarrow \max\{|u_{ij}| : 1 \leq j \leq n\}, for\ 1 \leq i \leq m, and \\ b_j &\leftarrow \max\{|u_{ij}| : 1 \leq i \leq m\}, for\ 1 \leq j \leq n. \end{aligned}$$

Next, based on the $a_{i's}$ and $b_{i's}$ information, let c_i be the maximum magnitude of elements divided by their respective $b_{i's}$ in the ith row of U, where $1 \leq i \leq m$, and let d_j be the maximum magnitude of elements divided by their respective $a_{i's}$ in the jth column of U, where $1 \leq j \leq n$. Thus,

$$\begin{aligned} c_i &\leftarrow \max\{\frac{|u_{ij}|}{b_j} : 1 \leq j \leq n\}, for\ 1 \leq i \leq m, and \\ d_j &\leftarrow \max\{\frac{|u_{ij}|}{a_i} : 1 \leq i \leq m\}, for\ 1 \leq j \leq n. \end{aligned}$$

Note that if there exists zero row or column vectors in U, we simply take their scale factors for the row or the columns to be $1's$. That is,

$$\begin{aligned} a_i, c_i &\leftarrow 1, if\ ith\ row\ of\ U\ is\ a\ zero\ vector, and \\ b_j, d_j &\leftarrow 1, if\ jth\ column\ of\ U\ is\ a\ zero\ vector \end{aligned}$$

Algorithm 7 for $ScaleDown(U)$ is under the assumption that U does not contain any null rows or columns.

It is crucial to note that both $ScaleUp$ and $ScaleDown$ preserve the symmetry if the input matrix is symmetric. Linear systems with symmetric coefficient matrices can be

Algorithm 7 ScaleDown(U)

for row $i \leftarrow 1$ to m **do**
  $a_i \leftarrow \max\{|u_{ij}| : 1 \le j \le n\}$
end for
for column $j \leftarrow 1$ to n **do**
  $b_j \leftarrow \max\{|u_{ij}| : 1 \le i \le m\}$
end for
for row $i \leftarrow 1$ to m **do**
  $c_i \leftarrow \max\{|u_{ij}|/b_j : 1 \le j \le n\}$
end for
for column $j \leftarrow 1$ to n **do**
  $d_j \leftarrow \max\{|u_{ij}|/a_i : 1 \le i \le m\}$
end for
for $i \leftarrow 1$ to m **do**
  for $j \leftarrow 1$ to n **do**
    $s_{ij} \leftarrow u_{ij}/\sqrt{a_i c_i b_j d_j}$
  end for
end for

solved twice as fast as general linear systems, so preservation of symmetry is a requirement of any good scaling algorithm.

2.4 Remarks on Our Implementation

In our implementation, the given matrix $A \in \mathbf{R}^{m\times n}$ is never copied in SCALGM: the auxiliary storage required by SCALGM is $\mathcal{O}(m+n)$, not $\mathcal{O}(mn)$! The scaled-up matrices U and the scaled-down matrices S can be computed as needed but are never stored.

CHAPTER 3

Convergence of the SCALGM Algorithm

Phase one and phase two are convergent in the SCALGM algorithm, as proved in Theorem 3.1.4 and Theorem 3.2.1, respectively. Phase one is the procedure of calling $ScaleUp()$ and $ScaleDown(\ )$ alternatively until the stopping criterion is met, and at this point the magnitudes of all nonzero scaled elements are within an interval $[m, 1]$ for some $m > 0$ and such that m is maximized. Phase two is the procedure of calling $ScaleDown(\ )$ iteratively until the scaled matrix converges entrywise. The final scaled matrix is row and column equilibrated with ∞-norm 1, which means each row and each column of the scaled matrix must contain at least one element of unit magnitude.

Without loss of generality as considering the magnitudes of matrix elements, we may assume matrix elements are nonnegative. Moreover, since the scaling of zero elements is unchanged, we may assume all matrix elements are positive.

3.1 Phase One

Consider the diagram below of phase one in which A is the input matrix, $\widehat{S}$ is the returned scaled matrix, and $U^{(k)}$, $S^{(k)}$ are intermediate matrices in the kth iteration, called the scaled up, scaled down matrix, respectively.

$$A = S^{(0)} \underbrace{\to U^{(1)} \to S^{(1)}}_{\text{1st iteration}} \underbrace{\to U^{(2)} \to S^{(2)}}_{\text{2nd iteration}} \to \cdots \to \widehat{U} \to \widehat{S}$$

For each iteration $k \geq 1$, all the elements in the scaled up matrix $U^{(k)}$ are bounded below by 1 and all the elements in the scaled down matrix $S^{(k)}$ are bounded above by 1.

Lemma 3.1.1 *Let* $U^{(k)} = [u_{ij}^{(k)}]$, $S^{(k)} = [s_{ij}^{(k)}] \in \mathbf{R}_+^{m\times n}$ *be the scaled up, scaled down matrix, respectively, in the* k*th iteration of phase one. Then* $u_{ij}^{(k)} \geq 1$ *and* $s_{ij}^{(k)} \leq 1$*, for* $k = 1, 2, \cdots$.

Proof. Let iteration $k \geq 1$ be given and let $U^{(k)} = [u_{ij}^{(k)}]$, $S^{(k)} = [s_{ij}^{(k)}]$ be the scaled up and scaled down matrices, respectively, in the kth iteration. Then, by Algorithm 7 of the procedure $ScaleDown(\ )$ on $U^{(k)}$, for each $u_{ij}^{(k)}$, we have its row scale down factors $a_i^{(k)}$, $c_i^{(k)}$ and column scale down factors $b_j^{(k)}$, $d_j^{(k)}$ that satisfy the following inequalities:

$$\begin{aligned} a_i^{(k)} &\geq u_{ij}^{(k)}, \\ b_j^{(k)} &\geq u_{ij}^{(k)}, \\ c_i^{(k)} &\geq \frac{u_{ij}^{(k)}}{b_j^{(k)}}, \ and \\ d_j^{(k)} &\geq \frac{u_{ij}^{(k)}}{a_i^{(k)}}. \end{aligned}$$

Thus,

$$\begin{aligned} c_i^{(k)} b_j^{(k)} &\geq u_{ij}^{(k)}, \ and \\ a_i^{(k)} d_j^{(k)} &\geq u_{ij}^{(k)}. \end{aligned}$$

Hence,

$$s_{ij}^{(k)} \equiv \frac{u_{ij}^{(k)}}{\sqrt{a_i^{(k)} c_i^{(k)} b_j^{(k)} d_j^{(k)}}} \leq 1$$

Similarly, it is straightforward to prove $u_{ij}^{(k)} \geq 1$ by considering Algorithm 5 of the procedure $ScaleUp(\ )$ on $S^{(k-1)}$, where $S^{(k-1)} = [s_{ij}^{(k-1)}]$ is the scaled down matrix in the $(k-1)$th iteration. Thus, for each $s_{ij}^{(k-1)}$, we have its row scale up factors $\alpha_i^{(k-1)}$, $\gamma_i^{(k-1)}$

and column scale up factors $\beta_j^{(k-1)}$, $\delta_j^{(k-1)}$ that satisfy the following inequalities:

$$\begin{aligned}
\alpha_i^{(k-1)} &\leq s_{ij}^{(k-1)}, \\
\beta_j^{(k-1)} &\leq s_{ij}^{(k-1)}, \\
\gamma_i^{(k-1)} &\leq \frac{s_{ij}^{(k-1)}}{\beta_j^{(k-1)}}, \ and \\
\delta_j^{(k-1)} &\leq \frac{s_{ij}^{(k-1)}}{\alpha_i^{(k-1)}}.
\end{aligned}$$

Thus,

$$\begin{aligned}
\gamma_i^{(k-1)}\beta_j^{(k-1)} &\leq s_{ij}^{(k-1)}, \ and \\
\alpha_i^{(k-1)}\delta_j^{(k-1)} &\leq s_{ij}^{(k-1)}.
\end{aligned}$$

Hence,

$$u_{ij}^{(k)} \equiv \frac{s_{ij}^{(k-1)}}{\sqrt{\alpha_i^{(k-1)}\gamma_i^{(k-1)}\beta_j^{(k-1)}\delta_j^{(k-1)}}} \geq 1 \ .$$

This completes the proof. ■

Lemma 3.1.2 *Let $k \geq 2$ and let $U^{(k)} = [u_{ij}^{(k)}]$ be the scaled up matrix in the kth iteration of phase one. Then $u_{ij}^{(k)} \leq \frac{1}{min^{(k-1)}}$, for all i,j , where $min^{(k-1)} = \min\{s_{ij}^{(k-1)} : all\ i,j\}$ and $S^{(k-1)} = [s_{ij}^{(k-1)}]$ is the scaled down matrix in the $(k-1)$th iteration.*

Proof. Let $k \geq 2$, $S^{(k-1)} = [s_{ij}^{(k-1)}]$ be the scaled down matrix in the $(k-1)$th iteration of phase one, and $min^{(k-1)} = \min\{s_{ij}^{(k-1)} : all\ i,j\}$. Then, by Algorithm 5, for each $s_{ij}^{(k-1)}$, the row scale up factors $\alpha_i^{(k-1)}$, $\gamma_i^{(k-1)}$ and the column scale up factors $\beta_j^{(k-1)}$, $\delta_j^{(k-1)}$ satisfy

the following inequalities:

$$\begin{aligned} \alpha_i^{(k-1)} &\geq min^{(k-1)} , \\ \gamma_i^{(k-1)} &\geq 1 , \\ \beta_j^{(k-1)} &\geq min^{(k-1)}, \ and \\ \delta_j^{(k-1)} &\geq 1 . \end{aligned}$$

Also, by Lemma 3.1.1, we have

$$s_{ij}^{(k-1)} \leq 1 .$$

Thus,

$$u_{ij}^{(k)} \equiv \frac{s_{ij}^{(k-1)}}{\sqrt{\alpha_i^{(k-1)}\gamma_i^{(k-1)}\beta_j^{(k-1)}\delta_j^{(k-1)}}} \leq \frac{1}{min^{(k-1)}} .$$

This completes the proof. ■

Lemma 3.1.3 *Let $k \geq 1$ and let $S^{(k)} = [s_{ij}^{(k)}]$ be the scaled down matrix in the kth iteration of phase one. Then $s_{ij}^{(k)} \geq \frac{1}{max^{(k)}}$, for all i,j , where $max^{(k)} = \max\{u_{ij}^{(k)} : all\ i,j\}$ and $U^{(k)} = [u_{ij}^{(k)}]$ is the scaled up matrix in the kth iteration.*

Proof. Let $k \geq 1$, $U^{(k)} = [s_{ij}^{(k)}]$ be the scaled up matrix in the kth iteration of phase one, and $max^{(k)} = \max\{u_{ij}^{(k)} : all\ i,j\}$. Then, by Algorithm 7, for each $u_{ij}^{(k)}$, the row scale down factors $a_i^{(k)}$, $c_i^{(k)}$ and the column scale down factors $b_j^{(k)}$, $d_j^{(k)}$ satisfy the following inequalities:

$$\begin{aligned} a_i^{(k)} &\leq max^{(k)} , \\ c_i^{(k)} &\leq 1 , \\ b_j^{(k)} &\leq max^{(k)}, \ and \\ d_j^{(k)} &\leq 1 . \end{aligned}$$

Also, by Lemma 3.1.1, we have

$$u_{ij}^{(k)} \geq 1 \ .$$

Thus,

$$s_{ij}^{(k)} \equiv \frac{u_{ij}^{(k)}}{\sqrt{a_i^{(k)} c_i^{(k)} b_j^{(k)} d_j^{(k)}}} \geq \frac{1}{max^{(k)}} \ .$$

This completes the proof. ■

Theorem 3.1.4 (Convergence of Phase One) *Let $U^{(k)}$ and $S^{(k)}$ be the scaled up and scaled down matrices, respectively, in the kth iteration of phase one, and let*

$$\begin{aligned} min^{(k)} &= \min\{s_{ij}^{(k)} : S^{(k)} = [s_{ij}^{(k)}]\} \ \ and \\ max^{(k)} &= \max\{u_{ij}^{(k)} : U^{(k)} = [u_{ij}^{(k)}]\} \end{aligned}$$

Then $\lim_{k\to\infty}(min^{(k)} \cdot max^{(k)}) = 1$.

Proof. By Lemma 3.1.1 $\sim$ 3.1.3, we have:

$$\begin{aligned} 1 \ \leq u_{ij}^{(k)} \leq \ & \frac{1}{min^{(k-1)}} \ , \ for \ k \geq 2, \ and \\ 1 \ \geq s_{ij}^{(k)} \geq \ & \frac{1}{max^{(k)}} \ , \ for \ k \geq 1. \end{aligned}$$

That is, $\frac{1}{min^{(k-1)}}$ is an upper bound for every elements of $U^{(k)}$, for $k \geq 2$, and $\frac{1}{max^{(k)}}$ is a lower bound for every elements of $S^{(k)}$, for $k \geq 1$. Thus, we have

$$\begin{aligned} max^{(k)} \ &\leq \ \frac{1}{min^{(k-1)}} \ , \ for \ k \geq 2 \ , \ and \\ min^{(k)} \ &\geq \ \frac{1}{max^{(k)}} \ , \ for \ k \geq 1 \ . \end{aligned}$$

Combining above inequalities, we have

$$min^{(k-1)} \leq \frac{1}{max^{(k)}} \leq min^{(k)} \ , \ for \ k \geq 2 \ ,$$

which implies

$$min^{(1)} \leq \frac{1}{max^{(2)}} \leq min^{(2)} \leq \frac{1}{max^{(3)}} \leq \cdots .$$

Clearly, $\{min^{(k)} : k = 1, 2, 3, \cdots\}$ is a nondecresing sequence and is bounded above by 1, and $\{max^{(k)} : k = 2, 3, 4, \cdots\}$ is a nonincresing sequence and is bounded below by 1, then there exists m and $M \in \mathbf{R}$ such that

$$\begin{aligned} \lim_{k\to\infty} min^{(k)} &= m \ , \ and \\ \lim_{k\to\infty} max^{(k)} &= M \ . \end{aligned}$$

Moreover, by the Sandwich Theorem, we have

$$\lim_{k\to\infty} (min^{(k)} max^{(k)}) = 1 \ .$$

This completes the proof. ■

Example 3.1.5 (Convergence of Phase One) *Consider $A \in \mathbf{R}^{4\times 4}$ as follows:*

$$A = \begin{bmatrix} 0.12012 & 3.52725 & 0.01031 & 1.37877 \\ 0.97429 & 0.01533 & 0.27907 & 42.55476 \\ 24.79849 & 17.58030 & 0.82502 & 0.11058 \\ 40.54019 & 13.88618 & 1.49386 & 0.61834 \end{bmatrix} .$$

Let $\widehat{U}$ and $\widehat{S}$ be the scaled up and scaled down matrices, respectively, in the last iteration of phase one. Then we have:

$$\widehat{U} = \begin{bmatrix} 1.00000 & 198.96417 & 1.00000 & 18.61240 \\ 9.37977 & 1.00000 & 31.30212 & \boxed{664.32079} \\ 138.30021 & \boxed{664.32079} & 53.60666 & 1.00000 \\ 40.43269 & 93.83905 & 17.35855 & 1.00000 \end{bmatrix}$$

and

$$\widehat{S} = \begin{bmatrix} 0.01678 & 1.00000 & 0.04114 & 0.09355 \\ 0.04714 & \boxed{0.00151} & 0.38567 & 1.00000 \\ 0.69510 & 1.00000 & 0.66047 & \boxed{0.00151} \\ 0.95018 & 0.66047 & 1.00000 & 0.00704 \end{bmatrix}$$

Note that $max\widehat{U} = 664.32079$ *and* $min\widehat{S} = 0.00151$ *with* $max\widehat{U} \cdot min\widehat{S} = 1.0$.

Example 3.1.6 (Equilibration of Phase Two) *Consider* $A \in \mathbf{R}^{4\times4}$ *as follows:*

$$A = \begin{bmatrix} 0.12012 & 3.52725 & 0.01031 & 1.37877 \\ 0.97429 & 0.01533 & 0.27907 & 42.55476 \\ 24.79849 & 17.58030 & 0.82502 & 0.11058 \\ 40.54019 & 13.88618 & 1.49386 & 0.61834 \end{bmatrix} .$$

Let $\widehat{S} = PhaseOne(A)$. *Then:*

$$\widehat{S} = \begin{bmatrix} 0.01678 & \boxed{1.00000} & 0.04114 & 0.09355 \\ 0.04714 & 0.00151 & 0.38567 & \boxed{1.00000} \\ 0.69510 & \boxed{1.00000} & 0.66047 & 0.00151 \\ 0.95018 & 0.66047 & \boxed{1.00000} & 0.00704 \end{bmatrix}$$

is not equilibrated in ∞*-norm* 1.00000 *due to its* 1*st column. Let* $S = PhaseTwo(\widehat{S})$. *Then:*

$$S = \begin{bmatrix} 0.01766 & \boxed{1.00000} & 0.04114 & 0.09355 \\ 0.04961 & 0.00151 & 0.38567 & \boxed{1.00000} \\ 0.73154 & \boxed{1.00000} & 0.66047 & 0.00151 \\ \boxed{1.00000} & 0.66047 & \boxed{1.00000} & 0.00704 \end{bmatrix}$$

is equilibrated. Note that the minimum magnitude of elements in $\widehat{S}$ *and* S *is the same as* 0.00151.

3.2 Phase Two

Consider the diagram below of phase two in which $\widehat{S}$ is the input matrix returned from phase one, and $S^{(k)}$ is the resultant matrix after the kth iteration, the scaled down matrix performed by the procedure $ScaleDown(\ )$ on $S^{(k-1)}$.

$$\widehat{S} = S^{(0)} \underbrace{\longrightarrow S^{(1)}}_{\text{1st iteration}} \underbrace{\longrightarrow S^{(2)}}_{\text{2nd iteration}} \longrightarrow \cdots \longrightarrow S$$

Theorem 3.2.1 (Convergence of Phase Two) *Let $\widehat{S}$ be the scaled matrix obtained from the Phase One. Then the sequence of matrices*

$$\{S^{(k)} : S^{(0)} = \widehat{S},\ S^{(k)} = \textit{ScaleDown}(S^{(k-1)})\ for\ k \geq 1\}$$

converges entrywise.

Proof. Since $S^{(0)} = [s_{ij}^{(0)}]$ is the scaled down matrix returned from the phase one, by Lemma 3.1.1, we have $s_{ij}^{(0)} \leq 1$ for all i, j . Thus, for each $s_{ij}^{(0)}$, by Algorithm 7, the corresponding row scale factors $a_i^{(0)}, c_i^{(0)}$ and column scale factors $b_j^{(0)}, d_j^{(0)}$ satisfy the following inequalities:

$$\begin{aligned}
s_{ij}^{(0)} &\leq a_i^{(0)} \leq 1\ , \\
s_{ij}^{(0)} &\leq b_j^{(0)} \leq 1\ , \\
\frac{s_{ij}^{(0)}}{b_j^{(0)}} &\leq c_i^{(0)} \leq 1\ ,\ and \\
\frac{s_{ij}^{(0)}}{a_i^{(0)}} &\leq d_j^{(0)} \leq 1\ .
\end{aligned} \tag{3.1}$$

Thus,

$$s_{ij}^{(1)} \equiv \frac{s_{ij}^{(0)}}{\sqrt{a_i^{(0)} c_i^{(0)} b_j^{(0)} d_j^{(0)}}} \leq 1\ .$$

Moreover,

$$s_{ij}^{(1)} \geq s_{ij}^{(0)} \; since \; \sqrt{a_i^{(0)} c_i^{(0)} b_j^{(0)} d_j^{(0)}} \leq 1 \; .$$

Continuing the same argument on the elements $s_{ij}^{(k)}$ in $S^{(k)}$, for $k \geq 1$, we can conclude that

$$s_{ij}^{(k)} \leq s_{ij}^{(k+1)} \leq 1, \; for \; k \geq 0 \; .$$

Therefore, $\{s_{ij}^{(k)} : k \geq 0\}$ is monotone and bounded, and hence convergent. This completes the proof. ■

By the proof of Theorem 3.2.1, we have the following Corollary immediately:

Corollary 3.2.2 *Let* $\{S^{(k)} = [s_{ij}^{(k)}] : S^{(0)} = \widehat{S}, \; S^{(k)} = ScaleDown(S^{(k-1)}) \; for \; k \geq 1\}$ *be the sequence of matrices in Phase Two. Then for each* (i, j) *entry,* $\{s_{ij}^{(k)} : k \geq 0\}$ *is nondecreasing.*

Theorem 3.2.3 (Row and Column Equilibrated in ∞-norm) *Let* $S = SCALGM(A)$. *Then every row and column of* S *contains at least one* 1.0 .

Proof. By the proof in Theorem 3.2.1, we have the derivation of the (i, j) elements in matrices from $S^{(k)} = [s_{ij}^{(k)}]$ to $S^{(k+1)} = [s_{ij}^{(k+1)}]$ in phase two that

$$s_{ij}^{(k+1)} \equiv \frac{s_{ij}^{(k)}}{\sqrt{a_i^{(k)} c_i^{(k)} b_j^{(k)} d_j^{(k)}}} \leq 1 \; , \; for \; k \geq 0 \; ,$$

where scale factors $a_i^{(k)}, b_j^{(k)}, c_i^{(k)}, d_j^{(k)} \leq 1$. Since $\{s_{ij}^{(k)}\}$ converges, we have

$$\lim_{k \to \infty} \sqrt{a_i^{(k)} c_i^{(k)} b_j^{(k)} d_j^{(k)}} = 1 \; .$$

In particular,

$$\lim_{k \to \infty} a_i^{(k)} = 1 = \lim_{k \to \infty} b_j^{(k)} \; ,$$

true for each i and j. By the choice of row scale factors $a_{i's}^{(k)}$ and column scale factors $b_{j's}^{(k)}$,

this implies that each row and column of the convergent matrix S contains at least one 1.0 . This completes the proof. ■

Theorem 3.2.4 *Algorithm SCALGM preserves symmetry. That is, S = SCALGM(A) is symmetric whenever A is symmetric.*

Proof. Let $A = [a_{ij}] \in \mathbf{R}^{n\times n}$ be symmetric with $a_{ij} = a_{ji}$ for all i and j. By the choice of scale factors on A, it is easy to see that the row [resp. column] scale up factors of a_{ij} are the column [resp. row] scale up factors of a_{ji}. Thus, the geometric mean of row and column scale up factors of a_{ij} is the same as that of a_{ji} and hence $ScaleUp(A)$ is symmetric. This proves the initial step. Next, let $S^{(k)} = [s_{ij}^{(k)}]$ be the intermediate matrix derived from A by the SCALGM algorithm. Suppose $S^{(k)}$ is symmetric by the induction hypothesis. Then, by the choice of scale factors on $S^{(k)}$, no matter whether in procedure $ScaleUp(\ )$ or $ScaleDown(\ )$, the row [resp. column] scale factors of $s_{ij}^{(k)}$ are the column [resp. row] scale factors of $s_{ji}^{(k)}$. Thus, the geometric mean of row and column scale factors of $s_{ij}^{(k)}$ is the same as that of $s_{ji}^{(k)}$ and hence $S^{(k+1)} = Scale(S^{(k)})$ is symmetric. This completes the proof by induction. ■

Example 3.2.5 *Consider the following symmetric matrix $A \in \mathbf{R}^{5\times 5}$:*

$$A = \begin{bmatrix} 4.5203873 & 2.9885969 & 1.9780225 & 0.0548313 & 64.2249838 \\ 2.9885969 & 0.0608930 & 19.3557172 & 0.1784661 & 2.9473931 \\ 1.9780225 & 19.3557172 & 1.3747245 & 0.1195760 & 0.0459842 \\ 0.0548313 & 0.1784661 & 0.1195760 & 0.1617764 & 4.7173173 \\ 64.2249838 & 2.9473931 & 0.0459842 & 4.7173173 & 25.9124215 \end{bmatrix}.$$

Let $S = SCALGM(A)$. Then:

$$S = \begin{bmatrix} 0.0283972 & 0.0467028 & 0.0410815 & 0.0046896 & 1.0000000 \\ 0.0467028 & 0.0023671 & 1.0000000 & 0.0379700 & 0.1141589 \\ 0.0410815 & 1.0000000 & 0.0943942 & 0.0338118 & 0.0023671 \\ 0.0046896 & 0.0379700 & 0.0338118 & 0.1883792 & 1.0000000 \\ 1.0000000 & 0.1141589 & 0.0023671 & 1.0000000 & 1.0000000 \end{bmatrix}$$

is symmetric, too.

3.3 Maximum Ratio Property of SCALGM

We will prove SCALGM is capable of maximizing the ratio of the minimum nonzero magnitude to the maximum magnitude of the elements in the resultant scaled matrix.

Notation 3.3.1 *Let $A = [a_{ij}] \in \mathbf{R}^{m \times n}$ be a nonzero matrix. We denote the ratio of the minimum nonzero magnitude to the maximum magnitude of the elements in A to be ρ_A. That is,*

$$\rho_A \equiv \frac{\min\{|a_{ij}| > 0 : all\ i, j\}}{\max\{|a_{ij}| : all\ i, j\}} .$$

Clearly, $0 < \rho_A \leq 1$ by its own definition. The ratio number we defined here is reciprocal to the ratio number defined in the paper of Fulkerson and Wolfe [13] as they present a method for finding scale factors which minimize the ratio of the matrix entry of largest absolute value to that of smallest non-zero absolute value. They say it is believed that such a ratio number is a useful condition number.

Our numerical experiments on maximizing the ratio number ρ_A by scaling on A have been brought to our attention that the numerical results agree with the results obtained from the SCALGM algorithm.

Claim 3.3.2 (Maximum Ratio Property of SCALGM) *Let $A \in \mathbf{R}^{m \times n}$ be a nonzero matrix and let $S = SCALGM(A)$. Then $\rho_S \geq \rho_{DAE}$ for all diagonal matrices D and E.*

Numerical experiments as well as the theoretical proof of the maximum ratio property of SCALGM will be discussed in the next sections.

CHAPTER 4

Numerical Experiments on the Maximum Ratio Property

Given any nonzero matrix $A \in \mathbf{R}^{m\times n}$, there is a corresponding ratio number ρ_A as defined in Notation 3.3.1. Our goal is to maximize the ratio number, from ρ_A to ρ_{DAE}, by finding such diagonal matrices D and E, and then compare this experimental result ρ_{DAE} with ρ_S, where $S = \text{SCALGM}(A)$. Our numerical experiment is based on a heuristic approach and the numerical results, obtained from Adaptive Simulated Annealing (ASA) [17], have supported our assertion.

The purpose of these experiments was to support the conjecture that SCALGM maximizes the minimum element of the scaled matrix. Although we later proved this conjecture, we present the experiments here for insights we believe they provide.

4.1 Objective Function

Let nonzero matrix $A \in \mathbf{R}^{m\times n}$ be given. Define the objective function $f : \mathbf{R}^{m+n} \to \mathbf{R}$ as:

$$f(r_1, \cdots, r_m, c_1, \cdots, c_n) = \frac{\min\{a_{ij}/(r_i c_j) : 1 \le i \le m, 1 \le j \le n\}}{\max\{a_{ij}/(r_i c_j) : 1 \le i \le m, 1 \le j \le n\}}$$

where $r_1, \cdots, r_m$ and $c_1, \cdots, c_n$ are row and column scaling factors of A, respectively. Thus,

$$f(r_1, \cdots, r_m, c_1, \cdots, c_n) = \rho_{DAE},$$

where $D = diag(1/r_1, \cdots, 1/r_m)$ and $E = diag(1/c_1, \cdots, 1/c_n)$. Our goal is to maximize the objective function f over the domain $\mathbf{R}^{m+n}$, so we use the method of Adaptive Simulated Annealing to approach the maximum value of f and then compare the results

with that of the SCALGM algorithm. Since the method of Adaptive Simulated Annealing (ASA) is used to minimize the objective function, we simply let

$$g \leftarrow -f \ ,$$

and use g as the objective function in ASA. Then the negative of the returned value, $-g$, is the maximum value from the ASA search.

We choose

$$(r_1, \cdots, r_m, c_1, \cdots, c_n) \leftarrow \mathbf{1} \in \mathbf{R}^{m+n}$$

as the starting point in the domain for the ASA search.

4.2 Numerical Results

Example 4.2.1 *Consider $A \in \mathbf{R}^{5\times 4}$ as follows:*

$$A = \begin{bmatrix} 0.0600657 & 0.0314488 & 0.0505425 & 0.3580263 \\ 31.3616505 & 1.7508878 & 0.3605699 & 4.7460253 \\ 0.0465615 & 0.2379094 & 0.0466107 & 50.8265669 \\ 0.3988647 & 2.4928637 & 0.2774323 & 0.9406760 \\ 7.2345614 & 34.9655153 & 6.2236013 & 47.1342023 \end{bmatrix} .$$

Let $S = SCALGM(A)$ and $T = ASA(A)$, then we have:

$$S = \begin{bmatrix} 0.2303080 & 0.1251810 & 1.0000000 & 0.1068071 \\ 1.0000000 & 0.0579578 & 0.0593269 & \boxed{0.0117743} \\ \boxed{0.0117743} & 0.0624555 & 0.0608210 & 1.0000000 \\ 0.1541256 & 1.0000000 & 0.5531810 & \underline{0.0282808} \\ 0.1993058 & 1.0000000 & 0.8847299 & 0.1010291 \end{bmatrix}$$

and

$$T = \begin{bmatrix} 0.2355454 & 0.0325978 & 0.2038492 & 0.1092359 \\ 1.0000000 & 0.0147569 & \underline{0.0118248} & \boxed{0.0117743} \\ \boxed{0.0117743} & 0.0159021 & 0.0121226 & 1.0000000 \\ 0.1482661 & 0.2449352 & 0.1060663 & 0.0272056 \\ 0.2278995 & 0.2911438 & 0.2016402 & 0.1155234 \end{bmatrix}$$

We can see that $\rho_S = minS = 0.0117743 = minT = \rho_T$ *and* S *is equilibrated, but* T *is not. The number of iterations in the SCALGM algorithm is* 5 *and the number of evaluations on the objective function in ASA is* 100262 *. Also note that the second-smallest magnitude of* S *is* 0.0282808*, which is larger than that of* T*, which is* 0.0118248*.*

Example 4.2.2 *Consider a symmetric* $A \in \mathbf{R}^{5\times 5}$ *as follows:*

$$A = \begin{bmatrix} 0.0746825 & 1.0919649 & 53.7545583 & 10.3215426 & 2.4708322 \\ 1.0919649 & 42.0772536 & 0.0515623 & 15.9470032 & 1.3133666 \\ 53.7545583 & 0.0515623 & 74.3663876 & 92.8061326 & 0.1247173 \\ 10.3215426 & 15.9470032 & 92.8061326 & 0.9240242 & 0.7549703 \\ 2.4708322 & 1.3133666 & 0.1247173 & 0.7549703 & 12.1175272 \end{bmatrix}.$$

Let $S =$ *SCALGM*(A) *and* $T =$ *ASA*(A)*, then we have:*

$$S = \begin{bmatrix} \underline{0.0019221} & 0.0270059 & 1.0000000 & 0.1538613 & 0.1138700 \\ 0.0270059 & 1.0000000 & \boxed{0.0009218} & 0.2284373 & 0.0581641 \\ 1.0000000 & \boxed{0.0009218} & 1.0000000 & 1.0000000 & 0.0041546 \\ 0.1538613 & 0.2284373 & 1.0000000 & 0.0079782 & 0.0201528 \\ 0.1138700 & 0.0581641 & 0.0041546 & 0.0201528 & 1.0000000 \end{bmatrix}$$

and

$$T = \begin{bmatrix} \underline{0.0011224} & 0.0171436 & 0.6348116 & 0.0542032 & 0.0320459 \\ 0.0248423 & 1.0000000 & \boxed{0.0009218} & 0.1267705 & 0.0257854 \\ 0.9198860 & \boxed{0.0009218} & 0.9999999 & 0.5549465 & 0.0018418 \\ 0.1076754 & 0.1737882 & 0.7607699 & 0.0033683 & 0.0067968 \\ 0.0293013 & 0.0162704 & 0.0011622 & 0.0031284 & 0.1240114 \end{bmatrix}$$

We can see that $\rho_S = minS = 0.0009218 = minT = \rho_T$ *and* S *is equilibrated, but* T *is not. The number of iterations in the SCALGM algorithm is* 3 *and the number of evaluations on the objective function in ASA is* 100326 *. Also note that* S *is symmetric but* T *is not and the second-smallest magnitude of* S *is* 0.0019221*, which is larger than that of* T*, which is* 0.0011224*.*

Example 4.2.3 *Consider* $A \in \mathbf{R}^{5\times 5}$ *as follows:*

$$A = \begin{bmatrix} 43.1439813 & 0.8087632 & 81.4332735 & 95.6318987 & 15.6605996 \\ 0.8087632 & 0.0968742 & 0.0801011 & 0.7368188 & 24.4376276 \\ 81.4332735 & 0.0801011 & 0.8900663 & 0.0379431 & 0.2939808 \\ 95.6318987 & 0.7368188 & 0.0379431 & 0.7466436 & 0.0219261 \\ 15.6605996 & 24.4376276 & 0.2939808 & 0.0219261 & 0.5210018 \end{bmatrix}.$$

Let $S =$ *SCALGM*(A) *and* $T =$ *ASA*(A)*, then we have:*

$$S = \begin{bmatrix} 0.0176014 & \boxed{0.0027546} & 1.0000000 & 1.0000000 & 0.0767602 \\ \boxed{0.0027546} & \boxed{0.0027546} & \underline{0.0082120} & 0.0643237 & 1.0000000 \\ 1.0000000 & \underline{0.0082120} & 0.3289976 & 0.0119427 & 0.0433729 \\ 1.0000000 & 0.0643237 & 0.0119427 & 0.2001161 & \boxed{0.0027546} \\ 0.0767602 & 1.0000000 & 0.0433729 & \boxed{0.0027546} & 0.0306809 \end{bmatrix}$$

and

$$T = \begin{bmatrix} 0.1682435 & 0.0059742 & 0.9999731 & 0.9999970 & 0.3470527 \\ 0.0057901 & 0.0013137 & 0.0018058 & 0.0141450 & 0.9942430 \\ 0.9999878 & 0.0018632 & 0.0344178 & \boxed{0.0012494} & 0.0205154 \\ 1.0000000 & 0.0145948 & 0.0012494 & 0.0209358 & 0.0013029 \\ 0.3381898 & 0.9996554 & 0.0199912 & \underline{0.0012697} & 0.0639382 \end{bmatrix}$$

We can see that $\rho_S = minS = 0.0027546 > minT = \rho_T = 0.0012494$ *and* S *is equilibrated, but* T *is not. The number of iterations in the SCALGM algorithm is* 39 *and the number of evaluations on the objective function in ASA is* 100359 *. Also note that* S *is symmetric but* T *is not and the second-smallest magnitude of* S *is* 0.0082120*, which is larger than that of* T*, which is* 0.0012697*. SCALGM has produced much better results in this case (*$minS > minT$*) than ASA has.*

Example 4.2.4 *Consider* $A \in \mathbf{R}^{15\times 6}$ *as follows:*

$$A = \begin{bmatrix} 39.1825004 & 65.4978133 & 2.0482634 & 57.5320599 & 0.8727561 & 0.0866749 \\ 0.0174157 & 0.1809213 & 4.8632161 & 5.6575103 & 0.0149143 & 0.2662696 \\ 0.1763694 & 11.0195404 & 0.7433910 & 67.0325689 & 0.0502083 & 0.0525480 \\ 99.7785982 & 7.0789378 & 40.9894436 & 0.3173076 & 97.3849539 & 60.3235253 \\ 12.4345575 & 21.2371584 & 0.0182033 & 1.3924923 & 0.0513201 & 27.5509926 \\ 0.0320970 & 0.0201085 & 18.0452977 & 6.5743054 & 0.0115688 & 0.1574914 \\ 56.9826977 & 0.0201479 & 2.8493551 & 2.7711917 & 11.3986891 & 4.2496192 \\ 73.7884126 & 0.0201038 & 0.4682885 & 0.5485364 & 0.0134761 & 2.3511967 \\ 2.8824512 & 0.0134463 & 0.1664398 & 1.1815007 & 0.4266600 & 0.1620873 \\ 0.7127229 & 0.0530533 & 0.0344227 & 1.2973887 & 7.3876290 & 0.1766576 \\ 0.3574435 & 23.7120526 & 0.3552317 & 0.0645017 & 1.5589028 & 0.4109612 \\ 1.0158471 & 0.8883048 & 0.8280001 & 0.0289451 & 0.0246166 & 0.0943812 \\ 12.3005614 & 0.0181642 & 0.1897420 & 0.0576021 & 0.9963737 & 0.2556982 \\ 13.5433903 & 0.0287200 & 0.3438184 & 0.0225416 & 3.3932683 & 14.6693574 \\ 0.3653704 & 0.0241846 & 77.8257591 & 1.2577048 & 3.1376826 & 5.7494783 \end{bmatrix}.$$

Let $S = \mathit{SCALGM}(A)$ and $T = \mathit{ASA}(A)$, then we have:

$$S = \begin{bmatrix}
0.2022233 & 1.0000000 & 0.0188295 & 0.5032417 & 0.0244160 & 0.0008312 \\
0.0018163 & 0.0558176 & 0.9034115 & 1.0000000 & 0.0084313 & 0.0515963 \\
0.0015524 & 0.2869354 & 0.0116552 & 1.0000000 & 0.0023955 & 0.0008594 \\
0.1890178 & 0.0396705 & 0.1383097 & 0.0010188 & 1.0000000 & 0.2123255 \\
0.1979246 & 1.0000000 & \boxed{0.0005161} & 0.0375656 & 0.0044279 & 0.8148107 \\
0.0009986 & 0.0018507 & 1.0000000 & 0.3466558 & 0.0019510 & 0.0091039 \\
0.9222430 & 0.0009646 & 0.0821418 & 0.0760145 & 1.0000000 & 0.1277916 \\
1.0000000 & \underline{0.0008060} & 0.0113042 & 0.0125993 & 0.0009900 & 0.0592040 \\
1.0000000 & 0.0137999 & 0.1028514 & 0.6947038 & 0.8023478 & 0.1044809 \\
0.0177981 & 0.0039192 & 0.0015311 & 0.0549098 & 1.0000000 & 0.0081966 \\
0.0050957 & 1.0000000 & 0.0090204 & 0.0015585 & 0.1204643 & 0.0108855 \\
0.3865736 & 1.0000000 & 0.5612409 & 0.0186684 & 0.0507779 & 0.0667328 \\
1.0000000 & 0.0043684 & 0.0274760 & 0.0079367 & 0.4390763 & 0.0386236 \\
0.4968968 & 0.0031171 & 0.0224690 & 0.0014017 & 0.6748384 & 1.0000000 \\
0.0026357 & \boxed{0.0005161} & 1.0000000 & 0.0153769 & 0.1226912 & 0.0770620
\end{bmatrix}$$

and

$$T = \begin{bmatrix} 0.1045794 & 0.6743718 & 0.0126958 & 0.2892120 & 0.0070063 & 0.0005793 \\ 0.0013276 & 0.0532030 & 0.8609357 & 0.8122788 & 0.0034196 & 0.0508254 \\ 0.0008192 & 0.1974358 & 0.0080183 & 0.5863837 & 0.0007014 & 0.0006111 \\ 0.2850180 & 0.0780049 & 0.2719110 & 0.0017071 & 0.8367017 & 0.4314725 \\ 0.1517802 & 1.0000000 & \boxed{0.0005160} & 0.0320133 & 0.0018842 & 0.8420805 \\ 0.0005304 & 0.0012818 & 0.6924670 & 0.2046056 & 0.0005750 & 0.0065163 \\ 0.4231624 & 0.0005772 & 0.0491395 & 0.0387599 & 0.2546026 & 0.0790216 \\ 0.9435393 & 0.0009917 & 0.0139061 & 0.0132108 & \underline{0.0005183} & 0.0752822 \\ 0.2282907 & 0.0041082 & 0.0306128 & 0.1762434 & 0.1016371 & 0.0321445 \\ 0.0047214 & 0.0013558 & 0.0005296 & 0.0161872 & 0.1471967 & 0.0029303 \\ 0.0020291 & 0.5192682 & 0.0046831 & 0.0006896 & 0.0266175 & 0.0058417 \\ 0.0633027 & 0.2135379 & 0.1198242 & 0.0033972 & 0.0046139 & 0.0147269 \\ 0.4425750 & 0.0025211 & 0.0158542 & 0.0039035 & 0.1078271 & 0.0230367 \\ 0.2209158 & 0.0018072 & 0.0130241 & 0.0006925 & 0.1664798 & 0.5991590 \\ 0.0020208 & \boxed{0.0005160} & 0.9996329 & 0.0131017 & 0.0521976 & 0.0796264 \end{bmatrix}$$

We can see that $\rho_S = minS = 0.0005161 \approx minT = \rho_T = 0.0005160$ *and* S *is equilibrated, but* T *is not. The number of iterations in the SCALGM algorithm is* 5 *and the number of evaluations on the objective function in ASA is* 100898 *. Note that the second-smallest magnitude of* S *is* 0.0008060*, which is larger than that of* T*, which is* 0.0005183*.*

CHAPTER 5

Theoretical Proof of the Maximum Ratio Property

5.1 Equivalence Relation

Define a relation "$\sim$" in $\mathbb{R}^{m\times n}$ as follows:

$$A \sim B \; in \; \mathbb{R}^{m\times n}, \; if \; B = DAE$$

for some positive diagonal matrices D and E. Clearly, "$\sim$" is an equivalence relation in $\mathbb{R}^{m\times n}$ as it satisfies the properties of reflexivity, symmetry, and transitivity. Thus, the equivalence classes of $\sim$ form a partition (a disjoint collection of non-empty subsets whose union is the whole set) of $\mathbb{R}^{m\times n}$. Denote the equivalence class, for instance, as

$$[A] = \{B \in \mathbb{R}^{m\times n} : B \sim A\},$$

or

$$[A] = \{DAE \in \mathbb{R}^{m\times n} : D, E, \; positive \; diagonal \; matrices\}$$

We can view the equivalence class $[A]$ as the collection of all scaled matrices S resultant from A, obtained by all two-sided diagonal scaling methods, in particular, including all one-sided diagonal scaling methods in the case of either $D = I_m$ or $E = I_n$.

Theorem 5.1.1 is our Claim 3.3.2 and its proof will be given in next section.

Theorem 5.1.1 (Maximum Ratio Property of SCALGM) *Let $A \in \mathbb{R}^{m\times n}$.*
If $S =$ SCALGM(A), then $\rho_S \geq \rho_B$ for all $B \in [A]$, where the ratio number ρ is defined in Notation 3.3.1.

5.2 Proof of Maximum Ratio Property

Notation 5.2.1 *Let A be a matrix in $\mathbf{R}^{m\times n}$. We denote $pre(A)$ as the immediately precedent matrix of A such that A is obtained from $pre(A)$ by one step, either scaleUp or scaleDown procedure, of the SCALGM algorithm.*

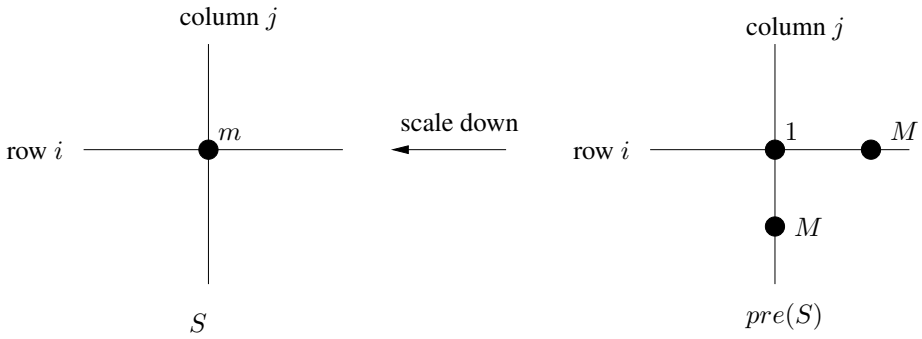

Figure 5.1: S and $pre(S)$

Lemma 5.2.2 (Refer to Figure 5.1) *Let $A \in \mathbf{R}^{m\times n}$, $S =$ SCALGM(A), and $m = \min S$ with $m \cdot M = 1$. Then m is the (i,j) element of S if and only if 1 is the (i,j) element of $pre(S)$ and there exists M's in the ith row and the jth column of $pre(S)$.*

Proof. ($\Rightarrow$) Suppose m is the (i,j) element of S. Let u_{ij} be the (i,j) element of $pre(S)$ and let a_i, c_i the ith row scale factors and b_j, d_j the jth column scale factors of $pre(S)$, as determined in the SCALGM algorithm. Then we have:

$$1 \leq u_{ij}, a_i, b_j \leq M,$$

and

$$\frac{1}{M} \leq c_i, d_j \leq 1\ .$$

Thus,

$$\frac{1}{M} \leq \sqrt{a_i c_i b_j d_j} \leq M,$$

and hence

$$\frac{1}{M} \leq \frac{u_{ij}}{\sqrt{a_i c_i b_j d_j}} = m \; (\; = \frac{1}{M} \;)$$

implies

$$\frac{u_{ij}}{\sqrt{a_i c_i b_j d_j}} = \frac{1}{M} \; .$$

Thus,

$$1 \leq u_{ij} = \frac{1}{M}\sqrt{a_i c_i b_j d_j} \leq \frac{1}{M} M = 1$$

implies $u_{ij} = 1$ and $\sqrt{a_i c_i b_j d_j} = M$. Finally we have $u_{ij} = 1$, $a_i = b_j = M$, and $c_i = d_j = 1$. The results follow.

($\Leftarrow$) Suppose 1 is the (i, j) element of $pre(S)$ and there exists M's in the ith row and in the jth column of $pre(S)$. Then, by the SCALGM algorithm, the scale down factors $a_i = b_j = M$ and $c_i = d_j = 1$. Thus, the (i, j) element of S is:

$$s_{ij} = \frac{u_{ij}}{\sqrt{a_i c_i b_j d_j}} = \frac{1}{M} = m.$$

The result follows. ■

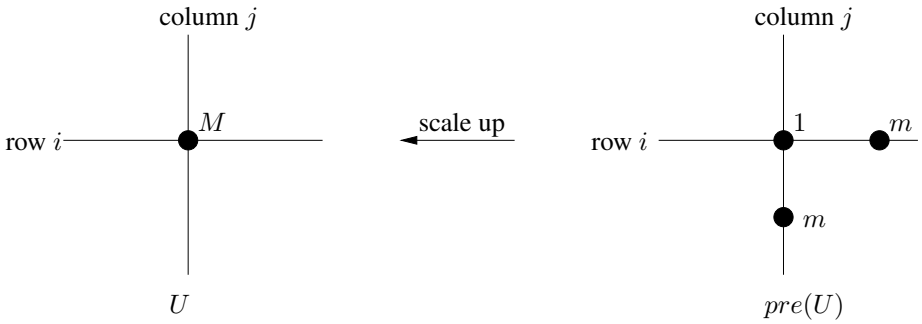

Figure 5.2: U and $pre(U)$

Lemma 5.2.3 (Refer to Figure 5.2) *Let A be a matrix in $\mathbf{R}^{m \times n}$, $S =$ SCALGM(A), and $U = pre(S)$. Then M is the (i, j) element of U if and only if 1 is the (i, j) element of $pre(U)$ and there exists m's in the ith row and the jth column of $pre(U)$.*

Proof. ($\Rightarrow$) Suppose M is the (i, j) element of U. Let s_{ij} be the (i, j) element of $pre(U)$ and let α_i, γ_i the ith row scale factors and β_j, δ_j the jth column scale factors of $pre(U)$, as determined in the SCALGM algorithm. Then we have:

$$m \leq s_{ij}, \alpha_i, \beta_j \leq 1,$$

and

$$1 \leq \gamma_i, \delta_j \leq \frac{1}{m} .$$

Thus,

$$(\frac{1}{m} =) M = \frac{s_{ij}}{\sqrt{\alpha_i \gamma_i \beta_j \delta_j}} \leq \frac{1}{m}$$

implies

$$\frac{1}{m} = \frac{s_{ij}}{\sqrt{\alpha_i \gamma_i \beta_j \delta_j}} .$$

Thus,

$$s_{ij} = \frac{1}{m} \sqrt{\alpha_i \gamma_i \beta_j \delta_j} \leq 1,$$

and

$$s_{ij} = \frac{1}{m} \sqrt{\alpha_i \gamma_i \beta_j \delta_j} \geq \frac{1}{m} m = 1$$

Thus $s_{ij} = 1$ and $\sqrt{\alpha_i \gamma_i \beta_j \delta_j} = m$. Finally, we have $s_{ij} = 1$, $\alpha_i = \beta_j = m$ and $\gamma_i = \delta_j = 1$. The results follow.

($\Leftarrow$) Suppose 1 is the (i, j) element of $pre(U)$ and there exists m's in the ith row and the jth column of $pre(U)$. Then, by the SCALGM algorithm, the scale down factors $\alpha_i = \beta_j = m$ and $\gamma_i = \delta_j = 1$. Thus, the (i, j) element of U is:

$$u_{ij} = \frac{s_{ij}}{\sqrt{\alpha_i \gamma_i \beta_j \delta_j}} = \frac{1}{m} = M.$$

■

Lemma 5.2.4 *If m is the (i, j) element of S, then there are at least one 1's in its corresponding ith row and jth column.*

Proof. The result immediately follows from Lemma 5.2.2 and the fact that the SCALGM algorithm maps M onto 1, as illustrated in Figure 5.3. ■

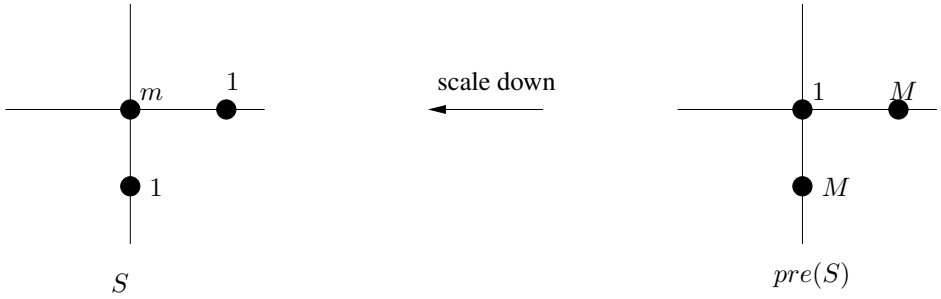

Figure 5.3: S and $pre(S)$

Lemma 5.2.5 *If M is the (i, j) element of U, then there are at least one 1's in its corresponding ith row and jth column.*

Proof. The result immediately follows from Lemma 5.2.3 and the fact that the SCALGM maps m onto 1, as illustrated in Figure 5.4. ■

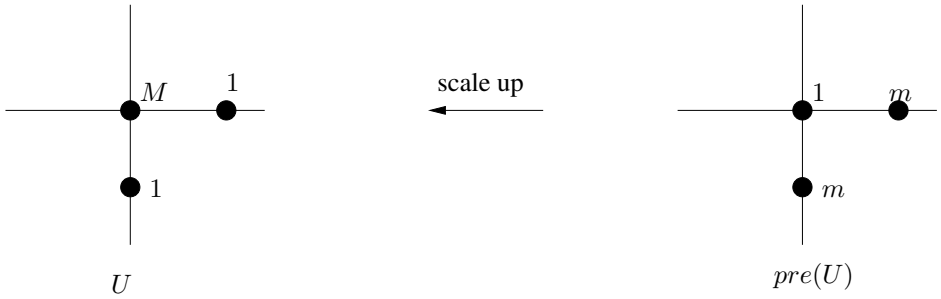

Figure 5.4: U and $pre(U)$

Theorem 5.2.6 *Let $A \in \mathbf{R}^{m \times n}$, $\widehat{S} = PhaseOne(A)$, and $S = PhaseTwo(\widehat{S})$. Then the minimum of nonzero magnitude of elements in $\widehat{S}$ is the same as that in S.*

Proof. Let m be the minimum of nonzero magnitude of elements in $\widehat{S}$ and $m = |\widehat{s}_{ij}|$ for some (i, j) element in $\widehat{S}$. Then, by Lemma 5.2.4, Lemma 3.1.1, and Algorithm 7, we have its row scale down factors a_i, c_i and column scale down factors b_j, d_j that satisfy $a_i = c_i = b_j = d_j = 1$. Thus, m is not changed after scaling. The same argument is applied to the rest iterations of $ScaleDown(\ )$ in phase two. Also, by Corollary 3.2.2, no any other element of magnitude will be scaled less than m. This completes the proof. ■

Definition 5.2.7 (CPEV) *Let $A = [a_{ij}] \in \mathbf{R}^{m\times n}$ with extreme absolute values:*

$$\begin{aligned} min &= \min\{|a_{ij}| > 0 : 1 \le i \le m, 1 \le j \le n\},\ and \\ max &= \max\{|a_{ij}| : 1 \le i \le m, 1 \le j \le n\} \end{aligned}$$

We say A has a closed path for its extreme absolute values min and max, or simply say A has the CPEV property, if there exists distinct row indices $i_1, \cdots, i_k$ and distinct column indices $j_1, \cdots, j_k$, for some $k \ge 2$, such that

$$\begin{aligned} |a_{i_p j_p}| &= min, for\ 1 \le p \le k, \\ |a_{i_p j_{p-1}}| &= max, for\ 1 \le p \le k, and \\ |a_{i_1 j_k}| &= max. \end{aligned}$$

Example 5.2.8 *Consider the matrices, as illustrated in the Figure 5.5 and Figure 5.6, are the ones satisfying the $CPEV$ property with extreme absolute values min and max.*

Let A be a matrix with the $CPEV$ property with extreme absolute values $\{min, max\}$, then under the geometric view of Definition 5.2.7, we can pick any element of A, say $A[i_1][j_1]$, with extreme absolute value, say min, as the starting point, then move along its corresponding row or column, say $column$, to reach the element, say $A[i_2][j_1]$, with the extreme absolute value max, as the second point. Next, from $A[i_2][j_1]$, move along its corresponding row to reach the point, say $A[i_2][j_2]$, with the extreme absolute value

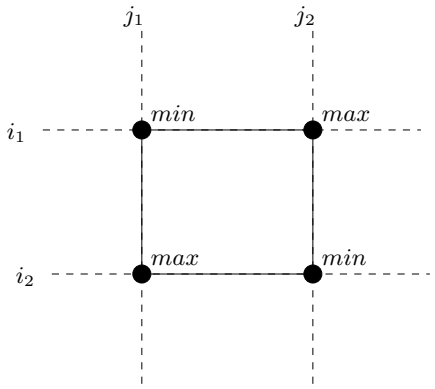

Figure 5.5: Closed Path for $k = 2$

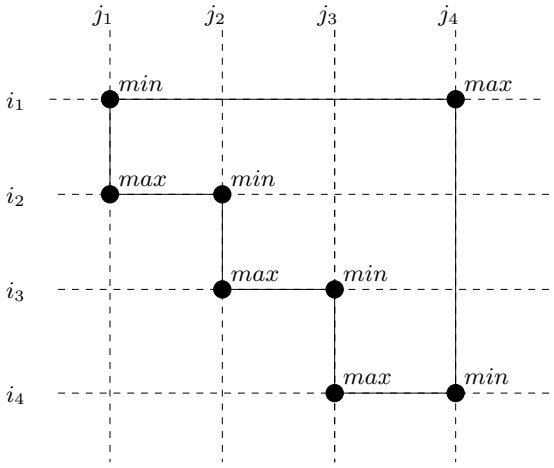

Figure 5.6: Closed path for $k = 4$

min. Continue this procedure of choosing points sequentially and alternately based on the following paths of $\{row, column\}$ and extreme absolute values of $\{min, max\}$ to get back to the starting point.

Example 5.2.9 *Consider the following matrix $A \in \mathbf{R}^{10\times 4}$ that satisfies the $CPEV$ property with extreme values $min = 0.00076$ and $max = 1$:*

$$A = \begin{bmatrix} 1.00000 & 0.99852 & 0.93902 & 0.11498 \\ 0.00118 & 1.00000 & 0.00096 & 0.01452 \\ \boxed{1.00000} & 0.00604 & 0.06724 & \boxed{0.00076} \\ 0.00309 & 1.00000 & 0.34584 & 0.00208 \\ 0.00187 & 0.02913 & \boxed{0.00076} & \boxed{1.00000} \\ 1.00000 & 0.01727 & 0.00544 & 0.00326 \\ 0.02198 & 0.01647 & 1.00000 & 0.00081 \\ 0.52561 & 1.00000 & 0.80994 & 0.00225 \\ 1.00000 & 0.00120 & 0.00225 & 0.00125 \\ \boxed{0.00076} & 0.21480 & \boxed{1.00000} & 0.74804 \end{bmatrix}$$

The closed path is:

$$A[10][1] \rightarrow A[3][1] \rightarrow A[3][4] \rightarrow A[5][4] \rightarrow A[5][3] \rightarrow A[10][3] \rightarrow A[10][1]$$

with row indices $i_1 = 10$, $i_2 = 3$ and $i_3 = 5$, and column indices $j_1 = 1$, $j_2 = 4$ and $j_3 = 3$.

Example 5.2.10 *Consider the following matrix $A \in \mathbf{R}^{5\times 5}$:*

$$A = \begin{bmatrix} 4.5203873 & 2.9885969 & 1.9780225 & 0.0548313 & 64.2249838 \\ 2.9885969 & 0.0608930 & 19.3557172 & 0.1784661 & 2.9473931 \\ 1.9780225 & 19.3557172 & 1.3747245 & 0.1195760 & 0.0459842 \\ 0.0548313 & 0.1784661 & 0.1195760 & 0.1617764 & 4.7173173 \\ 64.2249838 & 2.9473931 & 0.0459842 & 4.7173173 & 25.9124215 \end{bmatrix}.$$

Let $S = SCALGM(A)$. Then:

$$S = \begin{bmatrix} 0.0283972 & 0.0467028 & 0.0410815 & 0.0046896 & 1.0000000 \\ 0.0467028 & \boxed{0.0023671} & \boxed{1.0000000} & 0.0379700 & 0.1141589 \\ 0.0410815 & \boxed{1.0000000} & 0.0943942 & 0.0338118 & \boxed{0.0023671} \\ 0.0046896 & 0.0379700 & 0.0338118 & 0.1883792 & 1.0000000 \\ 1.0000000 & 0.1141589 & \boxed{0.0023671} & 1.0000000 & \boxed{1.0000000} \end{bmatrix}$$

satisfies the $CPEV$ property with extreme values $min = 0.0023671$ and $max = 1$. The closed path is:

$$S[2][2] \rightarrow S[3][2] \rightarrow S[3][5] \rightarrow S[5][5] \rightarrow S[5][3] \rightarrow S[2][3] \rightarrow S[2][2]$$

with row indices $i_1 = 2$, $i_2 = 3$ and $i_3 = 5$, and column indices $j_1 = 2$, $j_2 = 3$ and $j_3 = 5$.

Definition 5.2.11 *Let $A = [a_{ij}] \in \mathbf{R}^{m \times n}$ with extreme absolute values:*

$$\begin{aligned} min &= \min\{|a_{ij}| > 0 : 1 \leq i \leq m, 1 \leq j \leq n\}, \text{ and} \\ max &= \max\{|a_{ij}| : 1 \leq i \leq m, 1 \leq j \leq n\} \end{aligned}$$

We use the notation $\rho_A = \frac{min}{max}$ to denote the ratio of min to max of A.

Lemma 5.2.12 *Let $A = [a_{ij}] \in \mathbf{R}^{m \times n}$ with extreme absolute values min and max.*
If $x, y \in \{|a_{ij}| > 0 : 1 \leq i \leq m, 1 \leq n \leq n\}$, then $\frac{x}{y}, \frac{y}{x} \geq \rho_A$

Proof. Result follows by Definition 5.2.11 that $\rho_A = \frac{min}{max}$ and that $min \leq x, y \leq max$. ■

Theorem 5.2.13 *Let $A = [a_{ij}] \in \mathbf{R}^{m \times n}$ with extreme absolute values min and max.*
If A has the $CPEV$ property, then the ratio of min to max of A is greater than or equal to the ratio of min to max of RAC for all positive diagonal matrices R and C. That is, $\rho_A \geq \rho_{RAC}$ for all positive diagonal matrices R and C.

Proof. Suppose not. That is, there exists positive diagonal matrices $R = diag(r_i > 0 : 1 \leq i \leq m)$ and $C = diag(c_j > 0 : 1 \leq j \leq n)$ such that $\rho_A < \rho_B$, where $B = [b_{ij}] = RAC = [r_i a_{ij} c_j]$. Since A has the $CPEV$ property, there exists distinct row indices $i_1, \cdots, i_k$ and distinct column indices $j_1, \cdots, j_k$ that satisfy the conditions of Definition 5.2.7 and the closed path formed by these indices is illustrated in Figure 5.7.

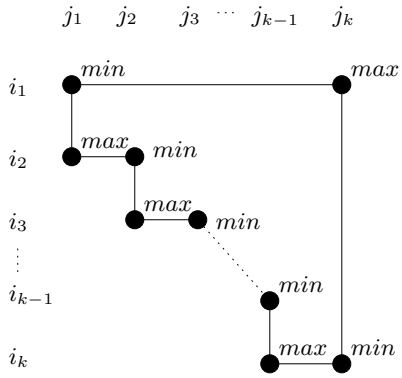

Figure 5.7: Closed path of A

Now, consider the corresponding $column$ path in B from (i_1, j_1) to (i_2, j_1).

By Lemma 5.2.12, we have

$$\frac{|b_{i_1 j_1}|}{|b_{i_2 j_1}|} \geq \rho_B,$$

which is equivalent to

$$\frac{|r_{i_1} a_{i_1 j_1} c_{j_1}|}{|r_{i_2} a_{i_2 j_1} c_{j_1}|} \geq \rho_B > \rho_A = \frac{min}{max}.$$

Thus, we have

$$|r_{i_1}| > |r_{i_2}|,$$

since $|a_{i_1 j_1}| = min$ and $|a_{i_2 j_1}| = max$. Apply the same argument to the remaining $k-1$ $column$ paths in B, we have

$$|r_{i_2}| > |r_{i_3}| > \cdots > |r_{i_k}| > |r_{i_1}|.$$

Then we get a contradiction that $|r_{i_1}| > |r_{i_1}|$. This completes the proof. ■

Theorem 5.2.14 *Let $A = [a_{ij}] \in \mathbf{R}^{m\times n}$. If $S =$ SCALGM(A), then S has the $CPEV$ property.*

Proof. Let $S = [s_{ij}]$ and let $m = \min\{|s_{ij}| > 0 : 1 \le i \le m, 1 \le j \le n\} = |s_{i_1 j_1}|$ for some i_1 and j_1. Then, by Lemma 5.2.2, there exist $i_2 \neq i_1$ and $j_2 \neq j_1$ such that M is the magnitude of the (i_2, j_1) and (i_1, j_2) elements in $pre(S)$, as shown in Figure 5.8.

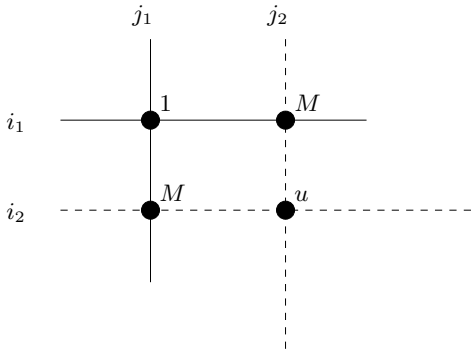

Figure 5.8: $pre(S)$

Now, let u be the magnitude of the (i_2, j_2) element in $pre(S)$. If $u = 1$, then $|s_{i_2 j_2}| = m$, as shown in Figure 5.9, and hence S has the $CPEV$ property and the proof is done. If

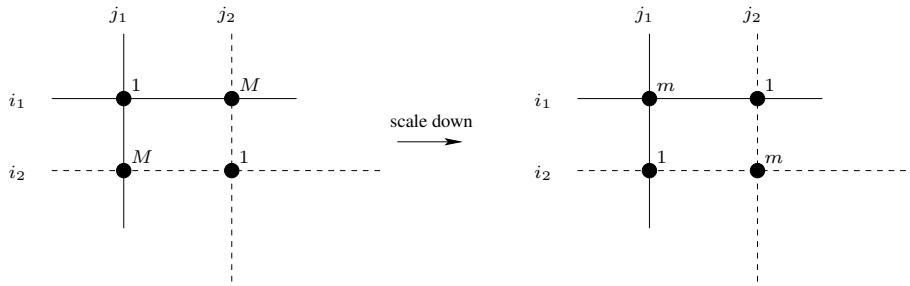

Figure 5.9: $pre(S)$ and S

$u \neq 1$, then, by Lemma 5.2.5, there exists $i_3 \notin \{i_1, i_2\}$ and $j_3 \notin \{j_1, j_2\}$ such that 1 is the magnitude of the (i_3, j_2) and (i_2, j_3) elements in $pre(S)$, as shown in Figure 5.10.

Now, let u be the magnitude of the (i_3, j_3) element in $pre(S)$, as shown in Figure 5.11. If $u = M$, then the magnitude of the (i_3, j_3) element of S is 1, as shown in Figure 5.12, and hence S has the $CPEV$ property and the proof is done. If $u \neq M$ in $pre(S)$, then we consider $pre(pre(S))$, the immediately precedent matrix of $pre(S)$. By Lemma 5.2.5, the

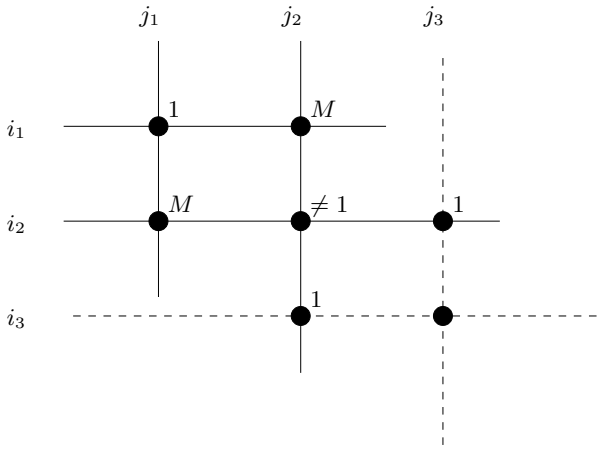

Figure 5.10: $pre(S)$

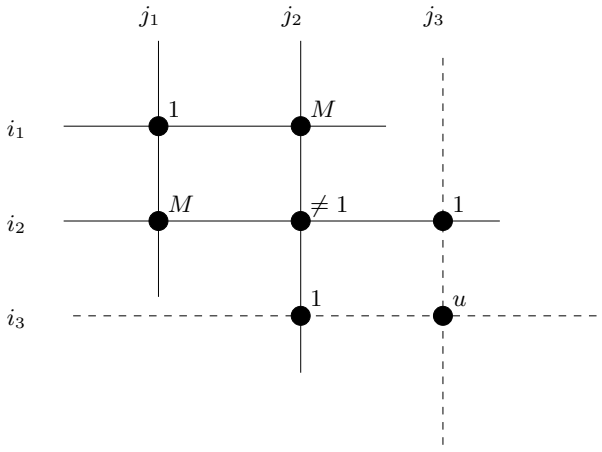

Figure 5.11: $pre(S)$

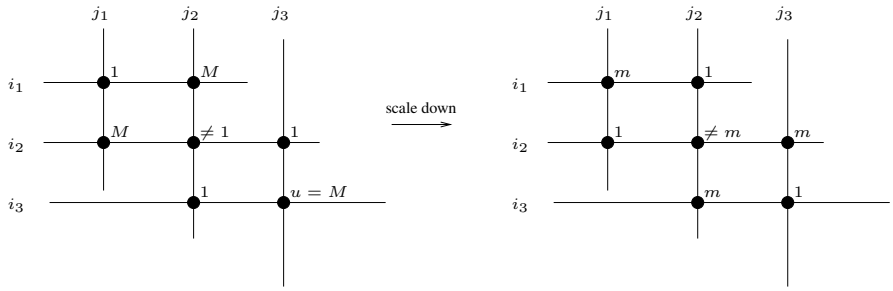

Figure 5.12: $pre(S)$ and S

magnitude of the (i_3, j_3) element in the $pre(pre(S))$ cannot be 1, as shown in Figure 5.13.

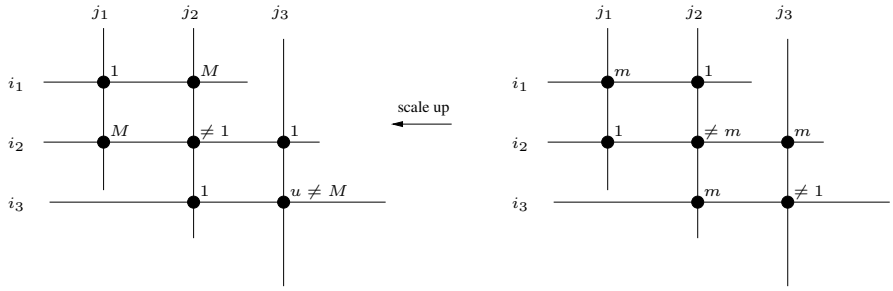

Figure 5.13: $pre(S)$ and $pre(pre(S))$

Moreover, by observing the (i_3, j_2) and (i_2, j_3) elements in $pre(pre(S))$ of magnitude m. Lemma 5.2.4 implies there exists row index $i_4 \notin \{i_2, i_3\}$ and column index $j_4 \notin \{j_2, j_3\}$ such that 1 is the magnitude of the (i_4, j_3) and (i_3, j_4) elements in $pre(pre(S))$, as shown in Figure 5.14. In fact, we can assume $i_4 \neq i_1$ and $j_4 \neq j_1$, or we get a closed path in $pre(pre(S))$ and the proof is done.

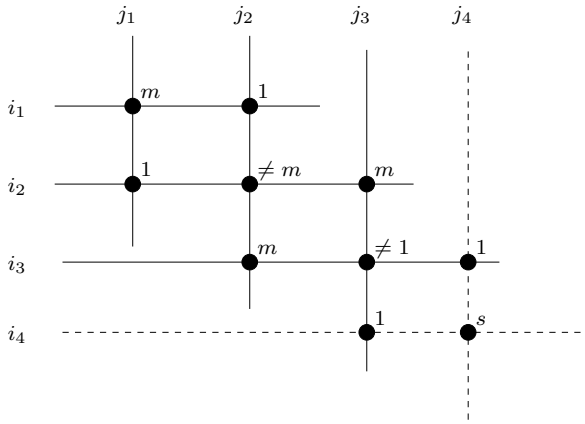

Figure 5.14: $pre(pre(S))$

Now, let s be the magnitude of the (i_4, j_4) element in $pre(pre(S))$. If $s = m$, then there is a closed path in $pre(pre(S))$ and thus the proof is done. If $s \neq m$, consider the (i_3, j_2) and (i_2, j_3) elements of the magnitude m in $pre(pre(S))$, then, by Lemma 5.2.2, we have its immediately precedent matrix, as shown in Figure 5.15 that there exists M's as the magnitude of the (i_4, j_3) and (i_3, j_4) elements in $pre(pre(pre(S)))$.

By our assumption that the magnitude of the (i_4, j_4) element in $pre(pre(S))$ is $s \neq m$,

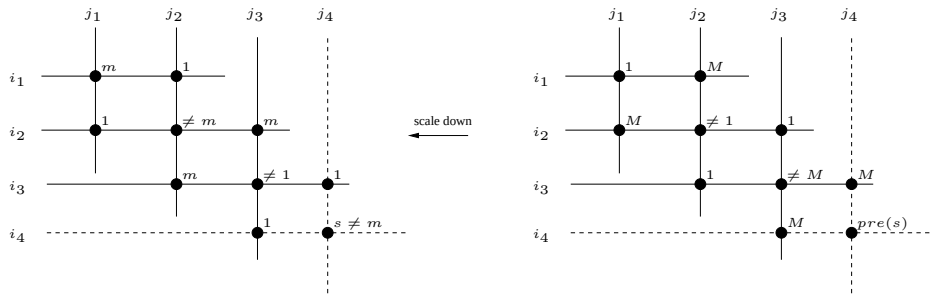

Figure 5.15: $pre(pre(S))$ and $pre(pre(pre(S)))$

the magnitude $pre(s)$ of the (i_4, j_4) element in $pre(pre(pre(S)))$ cannot be 1, and thus, by Lemma 5.2.5, there exists $i_5 \notin \{i_3, i_4\}$ and $j_5 \notin \{j_3, j_4\}$ such that 1 is the magnitude of the (i_5, j_4) and (i_4, j_5) elements in $pre(pre(pre(S))$ as shown in Figure 5.16.

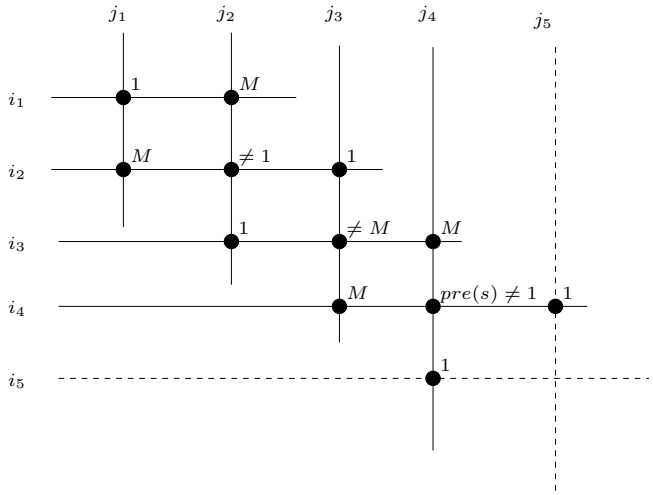

Figure 5.16: $pre(pre(pre(S)))$

In fact, we can assume $i_5 \notin \{i_1, i_2, i_3, i_4\}$ and $j_5 \notin \{j_1, j_2, j_3, j_4\}$, otherwise if $i_5 \in \{i_1, i_2\}$ or $j_5 \in \{j_1, j_2\}$, there there exists a closed path in $pre(pre(pre(S))))$ and hence in $pre(pre(S))$.

Now, consider the graph formed by row i_2 and column j_2 and the graph formed by row i_4 and column j_4, as shown in Figure 5.17. These two graphs have the same type of $\{1, M\}$ information, which is due to the magnitude of the (i_3, j_3) element for not being equal to M.

Thus, we can continue this argument on the (i_5, j_5) element. Since the dimension of the

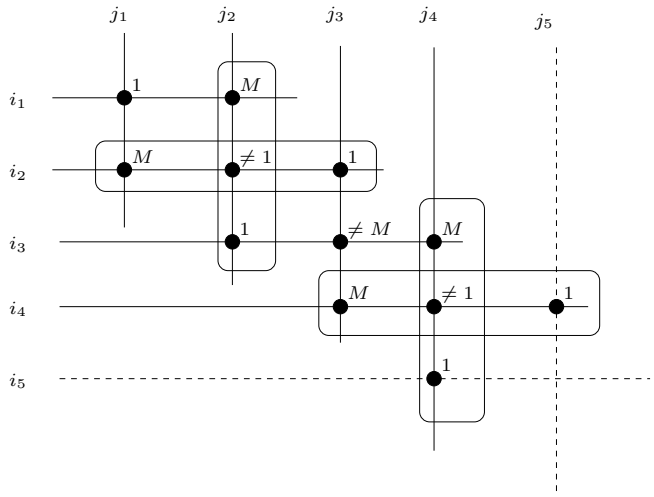

Figure 5.17: $pre(pre(pre(S)))$

given matrix A is finite, there exists some k such that the magnitude of the (i_k, j_k) element is M. Therefore, a closed path exists in some precedent matrix of S, and hence in S. This completes the proof.

■

Corollary 5.2.15 (Claim 3.3.2) *Let* $A = [a_{ij}] \in \mathbf{R}^{m \times n}$ *and* $S =$ *SCALGM*(A). *Then* $\rho_S \geq \rho_{RAC}$ *for all positive diagonal matrices* R *and* C.

Proof. It follows immediately from Theorem 5.2.13 and Theorem 5.2.14. ■

CHAPTER 6

Condition Numbers

Conventionally the condition number of $A \in \mathbf{R}^{m\times m}$ is defined to be

$$\kappa_p(A) = \|A\|_p \cdot \|A^{-1}\|_p \ ,$$

where $1 \leq p \leq \infty$ and

$$\|A\|_p = \max_{x\neq 0} \frac{\|Ax\|_p}{\|x\|_p} \ .$$

It can be shown that $\kappa_p(A) \geq 1$ for any value of p. If $\kappa_p(A)$ is small, A is said to be well-conditioned; if $\kappa_p(A)$ is large, A is ill-conditioned. If A is singular, it is customary to write $\kappa_p(A) = \infty$.

Throughout this chapter, unless specified, we simply denote $\kappa(A) = \kappa_2(A)$ for the 2-norm condition numbers of matrices A and denote $\|\cdot\| = \|\cdot\|_2$ for the 2-norm of vectors or matrices.

Let $A \in \mathbf{R}^{m\times n}$. The transpose of an $m \times n$ matrix A, written A^T, is the $n \times m$ matrix whose (i, j) entry is the (j, i) entry of A. .

If $Q \in \mathbf{R}^{m\times m}$ is an orthogonal matrix, then $\|Q\| = 1$ since

$$\begin{aligned} \|Qx\|^2 &= (Qx)^T(Qx) \\ &= x^T Q^T Q x \\ &= x^T x = \|x\|^2 \ , \end{aligned}$$

for all x. Thus, by the definition, we have $\|Q\| = 1$. Similarly, $\|Q^{-1}\| = 1$ since $Q^{-1} =$

Q^T. Therefore, $\kappa(Q) = 1$.

6.1 Measuring Linear Dependence by Angles

Let

$$A = [a_1|\cdots|a_n] \in \mathbf{R}^{m\times n} ,$$

where a_j is the jth column vector of A with

$$a_j = \begin{bmatrix} a_{1j} \\ \vdots \\ a_{nj} \end{bmatrix} .$$

Let $x = [x_1, \cdots, x_n]^T \in \mathbf{R}^{n\times 1}$, we may view $Ax \in range(A)$ as

$$\sum_{j=1}^{n} x_j a_j \in \mathcal{L}(a_1, \cdots, a_n)$$

, where $\mathcal{L}(a_1, \cdots, a_n)$ is the linear span of the column vectors a_j of A, or the linear space spanned by the column vectors a_j of A. In fact, $range(A) = \mathcal{L}(a_1, \cdots, a_n)$.

In this section, we will investigate this problem: Let $A = [a_1, \cdots, a_n] \in \mathbf{R}^{m\times n}$ be given. Denote the linear subspace

$$\mathcal{S}_j = \mathcal{L}(a_1, \cdots, a_{j-1}, a_{j+1}, \cdots, a_n) ,$$

and let

$$\theta_j(A) = \angle(a_j, \mathcal{S}_j)$$

be the angle between a_j and $\mathcal{S}_j$, as illustrated in Figure 6.1

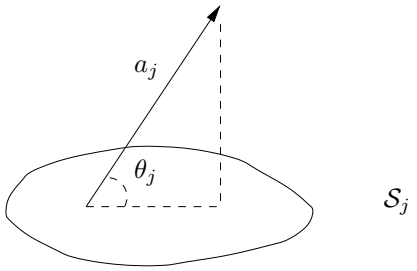

Figure 6.1: Angle θ_j between a_j and $\mathcal{S}_j$

Our goal is to find the minimum angle

$$\theta_{\min}(A) = \min\{\theta_j(A) : 1 \leq j \leq n\},$$

and we believe the minimum angle $\theta_{\min}(A)$ contains the essential information pertaining to the condition number $\kappa(A)$ of any well scaled matrix A. If $\theta_{\min} = 90^\circ$, the matrix is optimally well-conditioned; as $\theta_{\min} \to 0^\circ$ the "true condition" increases without limit. In the next section, we will discuss our approach for determining the orthogonal projector P that projects a_j onto the linear subspace $\mathcal{S}_j$ so as to find the angle θ_j by using the property of the inner product:

$$a_j^T(Pa_j) = \|a_j\|\|Pa_j\|\cos(\theta_j)$$

6.1.1 Orthogonal Projectors

Now consider the least squares problem: Given a matrix $A = [a_1|\cdots|a_n] \in \mathbf{R}^{m\times n}$ with $m \geq n$ and a column vector $b \in \mathbf{R}^{m\times 1}$, find $x \in \mathbf{R}^{n\times 1}$ such that $\|Ax - b\|$ is minimized. Note that $Ax = b$ might not have a solution for x as b might not in the range of A, which is illustrated in Figure 6.2.

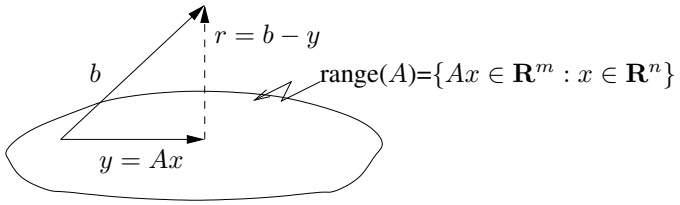

Figure 6.2: Solving x for minimizing $\|Ax - b\|$

In geometric view, our goal is to find $x \in \mathbf{R}^n$ such that $Ax \in range(A) \subset \mathbf{R}^m$ is the closest point to b, and thus $(b - Ax) \perp Ax$. Therefore, we would like to find an orthogonal projector $P \in \mathbf{R}^{m \times m}$ that maps each given $b \in \mathbf{R}^m$ such that

$$\begin{aligned} Pb &\in range(A) \quad \text{and} \\ b - Pb &\perp range(A) \, . \end{aligned}$$

If $A \in \mathbf{R}^{m \times n}$ with $m \geq n$ and has full rank $(= n)$, then the set $\{a_1, \cdots, a_n\}$ of column vectors a_j of A is linearly independent and we have

$$a_j \perp (b - Pb), \quad for\ all\ j \in \{1, \cdots, n\} \, .$$

Thus, the inner products, ${a_j}^T(b - Pb) = 0$ for all $j \in \{1, \cdots, n\}$, imply that

$$A^T(b - Pb) = \mathbf{0} \in \mathbf{R}^{n \times 1},$$

which is equivalent to

$$\begin{aligned} A^T b &= A^T P b \\ &= A^T A x \, , \end{aligned}$$

for some $x \in \mathbf{R}^{n \times 1}$ since $Pb \in range(A)$. Since A is of full rank by our assumption, $A^T A$ is nonsingular, and hence we can rewrite above equation as

$$x = (A^T A)^{-1} A^T b,$$

where $(A^T A)^{-1} A^T$ is called the pseudoinverse of A, and we denote

$$A^\dagger = (A^T A)^{-1} A^T \, .$$

Therefore,

$$y = Ax = A[(A^T A)^{-1} A^T b] = AA^{\dagger} b,$$

where

$$P = AA^{\dagger} \in \mathbf{R}^{m \times m}$$

is the orthogonal projector which projects every vector $b \in \mathbf{R}^{m \times 1}$ onto the subspace $range(A)$. Note that orthogonal projectors are not orthogonal matrices, but are always symmetric matrices. In fact, every symmetric matrix is an orthogonal projector.

Another way of finding the orthogonal projector P can be based on the procedure below. First, consider the simple case: Given a column vector $b \in \mathbf{R}^m$ and $q \in \mathbf{R}^m$ with $\|q\| = 1$, as illustrated in Figure 6.3.

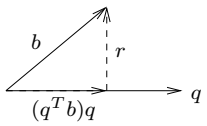

Figure 6.3: $Pb = (q^T b)q$

Then, by the property of inner product of q and b, we have

$$\begin{aligned} q^T b &= \|q\| \|b\| \cos(\theta) \\ &= \|b\| \cos(\theta) \\ &= \text{the magnitude of the component of } b \text{ along } q \end{aligned}$$

where θ is the angle between q and b. If P is the orthogonal projector that maps vectors onto the linear space $\mathcal{L}(q)$, then, in particular,

$$Pb = (q^T b)q = q(q^T b) = (qq^T)b,$$

and hence the orthogonal projector

$$P = qq^T \in \mathbf{R}^{m\times m} .$$

Also, let

$$r = b - Pb = (I_m - P)b,$$

which implies that $(I_m - P)$ is the orthogonal projector that maps every vector $b \in \mathbf{R}^m$ onto the space orthogonal to $\mathcal{L}(q)$.

Next, generalize the above case by considering given a column vector $b \in \mathbf{R}^m$ and a set of orthonormal vectors $\{q_1, \cdots, q_n\} \subset \mathbf{R}^m$, which means $q_i \perp q_j$ for all $i \neq j$ and $\|q_j\| = 1$ for all j. Let $Q = [q_1|\cdots|q_n]$. Then $Q \in \mathbf{R}^{m\times n}$ is an orthogonal matrix with $range(Q) = \mathcal{L}(q_1, \cdots, q_n) \subset \mathbf{R}^m$. From the given b and q_j for $j \in \{1, \cdots, n\}$, construct

$$v = \sum_{j=1}^{n} (q_j^T b) q_j \in \mathcal{L}(q_1, \cdots, q_n) ,$$

and then let

$$r = b - v ,$$

as illustrated in Figure 6.4:

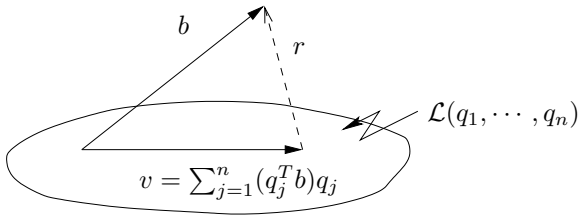

Figure 6.4: $r = b - v$, Is $r \perp v$?

Then,

$$r \perp q_j, \text{ for all } j$$

since the inner product

$$\begin{aligned} q_j^T r = q_j^T (b - v) &= q_j^T b - q_j^T v \\ &= q_j^T b - q_j^T \sum_{k=1}^{n} (q_k^T b_k) q_k \\ &= q_j^T b - \sum_{k=1}^{n} (q_k^T b_k) q_j^T q_k \\ &= q_j^T b - (q_j^T b_j) q_j^T q_j \\ &= 0 \, . \end{aligned}$$

Thus,

$$r \perp v$$

since the inner product

$$\begin{aligned} v^T r &= \left[\sum_{j=1}^{n} (q_j^T b) q_j \right]^T r \\ &= \sum_{j=1}^{n} \left[(q_j^T b) \underbrace{q_j^T r}_{0} \right] \\ &= 0 \, . \end{aligned}$$

Therefore, v is the image of b via an orthogonal projector. By the expression of v, we can write

$$\begin{aligned} v &= \sum_{j=1}^{n} (q_j^T b) q_j = \sum_{j=1}^{n} q_j (q_j^T b) \\ &= \sum_{j=1}^{n} (q_j q_j^T) b = \left[\sum_{j=1}^{n} (q_j q_j^T) \right] b \\ &= (QQ^T) b \, , \end{aligned}$$

where $Q = [q_1 | \cdots | q_n] \in \mathbf{R}^{m \times n}$ is an orthogonal matrix. Therefore $P = QQ^T \in \mathbf{R}^{m \times m}$ is

the orthogonal projector that maps each $b \in \mathbf{R}^m$ onto the linear space $\mathcal{L}(q_1, \cdots, q_n) \subset \mathbf{R}^m$ spanned by $q_1, \cdots, q_n$.

By the above discussion, we conclude the following properties:

Property 6.1.1 *If $A = [a_1|\cdots|a_n] \in \mathbf{R}^{m\times n}$ has full rank $n \leq m$, then the orthogonal projector P that projects onto $range(A) = \mathcal{L}(a_1.\cdots, a_n) \subset \mathbf{R}^m$ is*

$$P = AA^{\dagger} \in \mathbf{R}^{m\times m} ,$$

where $A^{\dagger} = (A^T A)^{-1} A^T \in \mathbf{R}^{m\times m}$ is the pseudoinverse of A.

Property 6.1.2 *If $Q = [q_1|\cdots|q_n] \in \mathbf{R}^{m\times n}$ is an orthogonal matrix, then the orthogonal projector P that projects onto $range(Q) = \mathcal{L}(q_1, \cdots, q_n) \subset \mathbf{R}^m$ is*

$$P = QQ^T .$$

It is well known that every matrix $A \in \mathbf{R}^{m\times n}(m \geq n)$, regardless of the rank of A, has a full QR decomposition $A = QR$, and hence a reduced QR decomposition $A = \widehat{Q}\widehat{R}$, where $Q \in \mathbf{R}^{m\times m}$, $\widehat{Q} \in \mathbf{R}^{m\times n}$ are orthogonal matrices and $R \in \mathbf{R}^{m\times n}$, $\widehat{R} \in \mathbf{R}^{n\times n}$ are upper-triangular matrices. Moreover, if A is of full rank $= n$, then A has a unique reduced QR decomposition $A = \widehat{Q}\widehat{R}$ with $r_{jj} > 0$. Therefore, we have the following property:

Property 6.1.3 *If A is of full rank and let $A = \widehat{Q}\widehat{R}$ be the reduced QR decomposition, then:*

(i). $range(A) = range(\widehat{Q})$, and

(ii). the orthogonal projector P that projects onto $range(A)$ is

$$P = \widehat{Q}\widehat{Q}^T .$$

What if the given matrix $A \in \mathbf{R}^{m\times n}$ is not of full rank, say $rank(A) = k < n \leq m$. The reduced QR decomposition still is our choice for the rank-deficient matrix as to how to find an orthogonal matrix $\widehat{Q} \in \mathbf{R}^{m\times k}$ and an upper-triangular matrix $\widehat{R} \in \mathbf{R}^{k\times k}$ such that $A = \widehat{Q}\widehat{R}$ and thus

$$range(A) = range(\widehat{Q}) \ .$$

By Property 6.1.3, the orthogonal projector P that projects vectors onto $range(A)$ is

$$\widehat{Q}\widehat{Q}^T \in \mathbf{R}^{m\times m} \ .$$

6.1.2 QR Decomposition

The QR decomposition is very important in numerical linear algebra and useful in its applications. Two well-known stable algorithms are Modified Gram-Schmidt Orthogonalization (MGS) and Householder Triangularization as in Algorithm 8 and Algorithm 10, respectively.

Algorithm 8 ReducedQRdecomposition(A) via Modified Gram-Schmidt Method

Require: Input $A = [a_1|\cdots|a_n] \in \mathbf{R}^{m\times n}$ has full rank $n \leq m$

for column $j \leftarrow 1$ to n **do**
  $v_j \leftarrow a_j$
end for
for row $i \leftarrow 1$ to m **do**
  $r_{ii} \leftarrow \|v_i\|$
  $q_i \leftarrow v_i/r_{ii}$
  for column $j \leftarrow i+1$ to n **do**
    $r_{ij} \leftarrow q_i^T v_j$
    $v_j \leftarrow v_j - r_{ij}q_i$
  end for
end for
return $Q = [q_j]$ and $R = [r_j]$

Ensure: $A = QR$ with $Q \in \mathbf{R}^{m\times n}$ orthogonal and $R \in \mathbf{R}^{n\times n}$ upper-triangular

In fact, we can modify Algorithm 8 as in Algorithm 9 to make it work for any input $A \in \mathbf{R}^{m\times n}$ with $m \geq n$, regardless of the rank of A.

The important property of the QR decomposition is that for every matrix $A \in \mathbf{R}^{m\times n}$

Algorithm 9 ReducedQRdecomposition(A) via Modified Gram-Schmidt Method

Require: Input $A = [a_1|\cdots|a_n] \in \mathbf{R}^{m\times n}$ has rank $k \leq n \leq m$

for column $j \leftarrow 1$ to n **do**
  $v_j \leftarrow a_j$
end for
for row $i \leftarrow 1$ to m **do**
  $r_{ii} \leftarrow \|v_i\|$
  if $r_{ii} = 0$ **then**
    $q_i \leftarrow \mathbf{0}$
  else
    $q_i \leftarrow v_i / r_{ii}$
  end if
  for column $j \leftarrow i + 1$ to n **do**
    $r_{ij} \leftarrow q_i^T v_j$
    $v_j \leftarrow v_j - r_{ij} q_i$
  end for
end for
$Q \leftarrow [q_j] \in \mathbf{R}^{m\times n}$, $R \leftarrow [r_j] \in \mathbf{R}^{n\times n}$
$\widehat{Q} \leftarrow Q$ by removing zero columns of Q
$\widehat{R} \leftarrow R$ by removing zero rows of R
return $\widehat{Q}$ and $\widehat{R}$

Ensure: $A = \widehat{Q}\widehat{R}$ with $\widehat{Q} \in \mathbf{R}^{m\times k}$ orthogonal and $\widehat{R} \in \mathbf{R}^{k\times n}$ upper-triangular

Algorithm 10 QRdecomposition(A) via Householder Triangularization

Require: Input $A = [a_1|\cdots|a_n] \in \mathbf{R}^{m\times n}$

$R \leftarrow A$
$Q \leftarrow I \in \mathbf{R}^{m\times m}$
for column $k \leftarrow 1$ to n **do**
  $x \leftarrow R_{k:m,k}$
  if $\|x\| = 0$ **then**
    continue
  end if
  if $x_1 < 0$ **then**
    $w \leftarrow x - \|x\| e_1$
  else
    $w \leftarrow x + \|x\| e_1$
  end if
  $w \leftarrow w / \|w\|$
  $R_{k:m,k:n} \leftarrow R_{k:m,k:n} - 2w(w^T R_{k:m,k:n})$
  $Q_{1:m,k:m} \leftarrow Q_{1:m,k:m} - 2(Q_{1:m,k:m} w) w^T$
end for
return Q, R

with $m \geq n$, not necessary to be of full rank, has the QR decomposition, written as

$$A = QR,$$

satisfying that $Q \in \mathbf{R}^{m \times m}$ is an orthogonal matrix and $R \in \mathbf{R}^{m \times n}$ is an upper-triangular matrix.

In particular, if $A \in \mathbf{R}^{n \times n}$ is nonsingular, then by QR decomposition, we have $A = QR$ with $Q \in \mathbf{R}^{n \times n}$ orthogonal and $R \in \mathbf{R}^{n \times n}$ upper-triangular. Thus, let $B = A^{-1}$ and since $QQ^T = I = Q^T Q$, we have:

$$\begin{aligned} & AB = I \\ \Rightarrow \quad & QRB = I \\ \Rightarrow \quad & RB = Q^{-1} = Q^T \end{aligned}$$

Since R is upper-triangular, we can solve for B by back substitution easily by observing

$$\begin{aligned} (RB)[i][j] &= (Q^T)[i][j] \\ &= Q[j][i] \ . \end{aligned}$$

i.e.,

$$\begin{aligned} & \sum_{k=1}^{n} r_{ik} b_{kj} = q_{ji} \\ \Rightarrow \quad & 0 + \sum_{k=i}^{n} r_{ik} b_{kj} = q_{ji} \\ \Rightarrow \quad & r_{ii} b_{ij} + \sum_{k=i+1}^{n} r_{ik} b_{kj} = q_{ji} \\ \Rightarrow \quad & b_{ij} = (q_{ji} - \sum_{k=i+1}^{n} r_{ik} b_{kj}) / r_{ii} \end{aligned}$$

This implies that for each column j of the matrix $B = [b_{ij}]$, b_{ij} is ready to be computed if $b_{i+1,j}, b_{i+2,j}, \cdots, b_{nj}$ are known. The method of computing A^{-1} via QR decomposition and back substitution is described in Algorithm 11.

Algorithm 11 Inverse(A) via QR Decomposition

Require: Input $A \in \mathbf{R}^{n \times n}$ is nonsingular
1: $Q, R \leftarrow$ QRdecomposition(A) // Algorithm 8, 9 or 10
2: **for** column $j \leftarrow 1$ to n **do**
3: **for** row $i \leftarrow n$ to 1 step -1 **do**
4: sum $\leftarrow 0$
5: **for** $k \leftarrow i+1$ to n **do**
6: sum $\leftarrow$ sum + $R[i][k] \cdot B[k][j]$
7: **end for**
8: $B[i][j] \leftarrow (Q[j][i] -$ sum$)/R[i][i]$ // Solve B for $RB = Q^T$ by back substitution
9: **end for**
10: **end for**
11: **return** B

If $A \in \mathbf{R}^{m \times n}$ ($m \geq n$) has full rank n, then $A^T A \in \mathbf{R}^{n \times n}$ is nonsingular and hence by Algorithm 11, the pseudoinverse $A^\dagger$ of A can be computed via

$$A^\dagger = (A^T A)^{-1} A^T$$

as in Algorithm 12.

Algorithm 12 PseudoInverse(A)

Require: Input $A \in \mathbf{R}^{m \times n}$ has full rank $n \leq m$
1: $C \leftarrow A^T A$
2: $B \leftarrow$ Inverse(C) // Algorithm 11
3: $A^\dagger \leftarrow B A^T$
4: **return** $A^\dagger$

If $A \in \mathbf{R}^{m \times n}$ has full rank $n \leq m$, then the orthogonal projector $P \in \mathbf{R}^{m \times m}$ that maps onto $range(A)$ can be obtained by Algorithm 13.

Let $A = [a_1, \cdots, a_n] \in \mathbf{R}^{m \times n}$ be given with $m \geq n$ and let $rank(A) = k$. Matrix A could have full rank if $k = n$, or be rank-deficient if $k < n$. No matter the former or the

Algorithm 13 Projector(A)

Require: Input $A \in \mathbf{R}^{m \times n}$ has full rank $n \leq m$

1: $A^{\dagger} \leftarrow$ PseudoInverse(A) // Algorithm 12
2: $P \leftarrow AA^{\dagger}$
3: **return** P

latter case, A always can be factored, via the reduced QR decomposition, into the form:

$$A = \widehat{Q}\widehat{R},$$

where $\widehat{Q} = [q_1, \cdots, q_k] \in \mathbf{R}^{m \times k}$ with $\widehat{Q}^T\widehat{Q} = I_k$ and $\widehat{R} \in \mathbf{R}^{k \times k}$ upper triangular. Since

$$range(A) = \mathcal{L}(a_1, \cdots, a_n) = \mathcal{L}(q_1, \cdots, q_k) \ ,$$

the orthogonal projector $P \in \mathbf{R}^{m \times m}$ that projects onto $range(A)$ is, by Property 6.1.2,

$$\widehat{Q}\widehat{Q}^T \ ,$$

and the method is given in Algorithm 14.

Algorithm 14 Projector(A)

Require: Input $A \in \mathbf{R}^{m \times n}$, $m \geq n$

1: $\widehat{Q}, \widehat{R} \leftarrow$ ReducedQRdecomposition(A) // Algorithm 9
2: $P \leftarrow \widehat{Q}\widehat{Q}^T$
3: **return** P

6.1.3 Computing the Minimum Angle $\theta_{\min}$

Let $A = [a_1|\cdots|a_n] \in \mathbf{R}^{m \times n}$ be given with $m \geq n$ and A does not have to be full rank. From the column vectors of A, we compute the angle θ_j between a_j and

$\mathcal{L}(a_1, \cdots, a_{j-1}, a_{j+1}, \cdots, a_n)$, as illustrated in Figure 6.5 .

Note that $[a_1|\cdots|a_{j-1}|a_{j+1}|\cdots|a_n] \in \mathbf{R}^{m \times (n-1)}$, denoted by $\widehat{A}_j$, is a submatrix of A

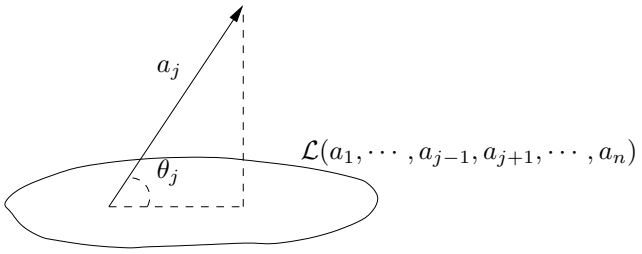

Figure 6.5: Computing the angle θ_j

by removing the jth column vector from A, and thus

$$range(\widehat{A}_j) = \mathcal{L}(a_1, \cdots, a_{j-1}, a_{j+1}, \cdots, a_n) \; .$$

Now we are ready to use Algorithm 14 to find the orthogonal projector $P_j \in \mathbf{R}^{m \times m}$ that projects any vector onto $range(\widehat{A}_j)$. In particular, for $a_j \in \mathbf{R}^m, 1 \leq j \leq n$, we have

$$(a_j - P_j a_j) \perp Pa_j \; .$$

Let $\theta_j = \angle(a_j, P_j a_j)$. Then $0 \leq \theta_j \leq \pi/2$ and by the property of inner product, we have:

$$a_j^T (P_j a_j) = \|a_j\| \|P_j a_j\| \cos(\theta_j) \; .$$

Solve for each θ_j and let $\theta_{\min} = \min\{\theta_j : 1 \leq j \leq n\}$, then $0 \leq \theta_{\min} \leq \pi/2$. In Algorithm 15, the returned $\theta_{\min}$ is measured in degrees, rather than radians, for easy to read in human sense.

Consider $A = [a_1 | \cdots | a_n] \in \mathbf{R}^{n \times n}$:

If $\theta_{\min}(A) = 0$, then clearly, for some column vector a_j of A, $a_j \in range(\widehat{A}_j)$, that is, a_j is a linear combination of other columns of A, and hence A is singular with $\kappa(A) = \infty$.

If $\theta_{\min}(A) = \pi/2$, then $a_i \perp a_j$ for all $i \neq j$. Thus, we have:

$$A^T A = diag(\|a_1\|^2, \cdots, \|a_n\|^2),$$

Algorithm 15 MinimumAngle(A)

Require: Input $A = [a_1|\cdots|a_n] \in \mathbf{R}^{m\times n}$ with $m \geq n$

$\pi \leftarrow 4.0 \cdot \arctan(1.0)$
$\theta_{\min} \leftarrow \pi/2$
for column $j \leftarrow 1$ to n **do**
  $\widehat{A} \leftarrow [a_1|\cdots|a_{j-1}|a_{j+1}|\cdots|a_n]$
  $P \leftarrow Projector(\widehat{A})$ // Algorithm 14
  $\theta \leftarrow \arccos(a_j^T(Pa_j)/(\|a_j\|\|Pa_j\|))$
  if $\theta < \theta_{\min}$ **then**
    $\theta_{\min} \leftarrow \theta$
  end if
end for
return $\theta_{\min} \cdot 180/\pi$ // convert from radians to degrees

and hence, by Properties 6.2.2 and 6.2.4,

$$\kappa(A) = \frac{\max\{\|a_1\|, \cdots, \|a_n\|\}}{\min\{\|a_1\|, \cdots, \|a_n\|\}} .$$

Next, consider $A \in \mathbf{R}^{n\times n}$ and let $S = [s_1|\cdots|s_n] = \text{SCALGM}(A)$:

If $\theta_{\min}(S) = 0$, then $\kappa(S) = \infty$.

If $\theta_{\min}(S) = \pi/2$, then, by Theorem 3.2.3 [Row and Column Equilibrated in ∞-norm], we have

$$1 \leq \min\{\|s_1\|, \cdots, \|s_n\|\} \leq \max\{\|s_1\|, \cdots, \|s_n\|\} \leq \sqrt{n} ,$$

since $0 \leq |s_{ij}| \leq 1$, the upper bound $\sqrt{n}$ for 2-norm of columns $\|s_j\|$ is attained if $s_j = [1, \cdots, 1]^T$, and the lower bound 1 for 2-norm of columns $\|s_j\|$ is attained if $s_j =$ some unit vector. Thus,

$$\kappa(S) = \frac{\max\{\|s_1\|, \cdots, \|s_n\|\}}{\min\{\|s_1\|, \cdots, \|s_n\|\}} \leq \sqrt{n}$$

The condition numbers $\kappa(A)$ of badly scaled matrices A can be very large even when $\theta_{\min}(A) = \pi/2$. For instance,

$$A = \begin{bmatrix} 10^5 & 0 \\ 0 & 10^{-5} \end{bmatrix} ,$$

we have $\theta_{\min}(A) = \pi/2$ and $\kappa(A) = 10^{10}$, but if $S = \text{SCALGM}(A)$, then S is the identity I_2 with $\theta_{\min}(S) = \pi/2$ and $\kappa(S) = 1$.

In our experiments, if $S = \text{SCALGM}(A)$, then there is a linear equation that describes at least roughly the relation between the condition number $\kappa(S)$ and the minimum angle $\theta_{\min}(S)$ for all well-scaled matrices S.

6.2 Computing the 2-norm Condition Numbers

Notation 6.2.1 *Let $A \in \mathbf{R}^{m\times n}$ with $m \geq n$. We denote $\Sigma(A)$ the set of all singular values σ of A and $\Lambda(A)$ the set of all eigenvalues λ of A.*

Our approach for computing the 2-norm condition numbers κ of the given matrix $A \in \mathbf{R}^{m\times m}$ is based on the following well known properties:

Property 6.2.2 *Let $A \in \mathbf{R}^{m\times n}$. Then*

(i). $\Lambda(A^TA) = \Lambda(AA^T) \subset \mathbf{R}$.

(ii). A^TA *is orthogonally diagonalizable.*

(iii). $\Sigma(A) = \Lambda^{1/2}(A^TA)$*, i.e., the singular values of A are the square roots of the eigenvalues of A^TA.*

Property 6.2.3 *Let $A \in \mathbf{R}^{n\times n}$. If $A = A^T$, then $\Sigma(A) = |\Lambda|(A)$, i.e., the singular values of A are the absolute values of the eigenvalues of A.*

Property 6.2.4 *Let $A \in \mathbf{R}^{m\times n}$ with $m \geq n$ and let $\Sigma(A) = \{\sigma_1, \sigma_2, \cdots, \sigma_n\}$ with $\sigma_1 \geq \sigma 2 \geq \cdots \geq \sigma_n \geq 0$. Then*

(i). $\|A\| = \sigma_1$.

(ii). If A is nonsingular, then $\|A^{-1}\| = 1/\sigma_n$, and hence $\kappa(A) = \sigma_1/\sigma_n$.

(iii). If A is of full rank $= n < m$, then the pseudoinverse $A^\dagger$ of A is $(A^TA)^{-1}A^T$ with $\|A^\dagger\| = 1/\sigma_n$, and hence $\kappa(A) = \sigma_1/\sigma_n$.

Algorithm 16 is the outline of computing the 2-norm condition number of any given matrix $A \in \mathbf{R}^{m\times n}$ with $m \geq n$.

Algorithm 16 ConditionNumber(A)

Require: Input $A \in \mathbf{R}^{m\times n}$ with $m \geq n$

1: $B \leftarrow A^T A$
2: $\Lambda \leftarrow Q^T BQ$ // Property 6.2.2(ii)
3: $\Sigma \leftarrow \Lambda^{1/2}$ // Property 6.2.2(iii)
4: $\sigma_1 \leftarrow \max \Sigma$
5: $\sigma_n \leftarrow \min(\Sigma \setminus \{0\})$
6: **return** σ_1/σ_n

Definition 6.2.5 *An upper-Hessenberg matrix $H \in \mathbf{R}^{m\times n}$ is a matrix with zeros below the first subdiagonal.*

To implement line 2 of Algorithm 16, we apply Algorithm 17 on B to obtain

$$H_1 \leftarrow Hessenberg(B) \ ,$$

where $H_1 \in \mathbf{R}^{n\times n}$ is upper-Hessenberg, and thus we have

$$H_1 = Q_0^T BQ_0,$$

for some orthogonal matrix $Q_0 \in \mathbf{R}^{n\times n}$, which can be obtained in Algorithm 17. Note that $B = A^T A$ is symmetric, and hence so is H_1. Therefore, the symmetric upper-Hessenberg matrix H_1 is tridiagonal.

Next, we perform QR decomposition on H_1 such that

$$H_1 = Q_1 R_1,$$

where Q_1 is orthogonal and R_1 is upper-triangular. Thus, we can obtain $H_2 \in \mathbf{R}^{n\times n}$ as

Algorithm 17 Hessenberg(A) via Householder Reduction

Require: Input $A \in \mathbf{R}^{m \times m}$

for column $k \leftarrow 1$ to $m-2$ **do**
  $x \leftarrow A_{k+1:m,k}$
  if $x_1 < 0$ **then**
    $w \leftarrow x - \|x\| e_1$
  else
    $w \leftarrow x + \|x\| e_1$
  end if
  $w \leftarrow w/\|w\|$
  $A_{k+1:m,k:m} \leftarrow A_{k+1:m,k:m} - 2w(w^T A_{k+1:m,k:m})$
  $A_{1:m,k+1:m} \leftarrow A_{1:m,k+1:m} - 2(A_{1:m,k+1:m} w) w^T$
end for
return A

follows:

$$\begin{aligned} H_2 = Q_1^T H_1 Q_1 &= (Q_1^T H_1) Q_1 \\ &= R_1 Q_1 \end{aligned}$$

This process,

$$H_{i+1} = Q_i^T H_i R_i = R_i Q_i,$$

is repeated for $i = 1, 2, \cdots$ until $\{H_i\}$ converges to H. Let $Q = Q_0 Q_1 Q_2 \cdots$. Then we have $H = Q^T B Q$ and H is diagonal. By this orthogonal transformation on B, we have $\Lambda(H) = \Lambda(B)$. Algorithm 18 is the detailed one for Algorithm 16:

6.3 Best Estimated Interval

Notation 6.3.1 *Denote $\mathcal{D}_n \subseteq \mathbf{R}^{n \times n}$ the set of $n \times n$ diagonal matrices with positive diagonal elements, and denote $\varrho(A)$ the spectral radius of A, i.e., the largest absolute value $|\lambda|$ of an eigenvalue λ of A:*

$$\varrho(A) = \max\{|\lambda| : \lambda \in \Lambda(A)\}$$

The following theorem, due to Rump [22], gives us an estimated interval for the condi-

Algorithm 18 ConditionNumber(A)

Require: Input $A \in \mathbf{R}^{m \times n}$ with $m \geq n$

$B \leftarrow A^T A$
$H \leftarrow Hessenberg(B)$ // Algorithm 17
repeat
  $Q, R \leftarrow QRdecomposition(H)$ // Algorithm 10
  $H \leftarrow RQ$
until H converges
$\Lambda \leftarrow diag(H)$
$\Sigma \leftarrow \Lambda^{1/2}$ // Property 6.2.2(iii)
$\sigma_1 \leftarrow \max \Sigma$
$\sigma_n \leftarrow \min(\Sigma \setminus \{0\})$
return σ_1/σ_n

tion number of the optimal two-sided scaled matrix $D_1 A D_2$.

Theorem 6.3.2 *Let nonsingular $A \in \mathbf{R}^{n \times n}$ be given and let $1 \leq p \leq \infty$. Define*

$$\mu_p := \inf_{D_1, D_1 \in \mathcal{D}_n} \kappa_p(D_1 A D_2),$$

Then

$$\mu_p \leq \varrho(|A^{-1}|\,|A|) \leq \alpha_p^2 \cdot \mu_p$$

with $\alpha_p := n^{\min\{1/p,\, 1-1/p\}}$. For $p \in \{1, \infty\}$ both inequalities are equalities. The left bound is sharp for all p . For $p = 2$ the right inequality is sharp at least for the infinitely many values of n where an $n \times n$ Hadamard matrix exists.

In particular, for $p = 2$, we have $\alpha = n^{1/2}$ and hence

$$\begin{aligned} & \mu \leq \varrho(|A^{-1}|\,|A|) \leq n \cdot \mu \\ \Rightarrow \quad & \varrho(|A^{-1}|\,|A|)/n \leq \mu \leq \varrho(|A^{-1}|\,|A|) \,. \end{aligned}$$

In order to compute $\varrho(|A^{-1}|\,|A|)$, Algorithm 19 presented by Hall and Porsching [15]

is used in our experiments. Then we can compare this interval

$$[\varrho(|A^{-1}|\ |A|)/n\ ,\ \ \varrho(|A^{-1}|\ |A|)]$$

with the condition number $\kappa(S)$ of the scaled matrix S, performed by our SCALGM algorithm.

Now, let S^* be the optimal two-sided scaled matrix of A with $\kappa(S^*) = \mu$ as described in Theorem 6.3.2, and let S be the scaled matrix of A, performed by SCALGM algorithm. Then there exist D_1^*, D_2^*, D_1 and $D_2 \in \mathcal{D}_n$ such that

$$\begin{aligned} S^* &= D_1^* A D_2^* \,, \\ S &= D_1 A D_2 \,. \end{aligned}$$

Thus, $S^* \sim S$ are in the same equivalence class $[A]$, as "$\sim$" is defined in Section 5.1 . Instead of computing $\varrho(|A^{-1}|\ |A|)$ to estimate $\mu = \kappa(S^*)$, we compute $\varrho(|S^{-1}|\ |S|)$ to estimate μ once S is obtained from SCALGM algorithm on A , since $\mu = \mu(A) = \mu(S) = \mu(S^*)$. In fact, the μ values are equal for all matrices in the same equivalence class. The method we use to compute the estimated interval for the condition number of the optimal two-sided scaled matrix is shown in Algorithm 20 .

Finally, in order to approach the condition number $\mu = \mu(A)$ of the optimal two-sided scaled matrix $D_1^* A D_2^*$ for some $D_1^*, D_2^* \in \mathcal{D}_n$, we construct an objective function and apply the pattern search method [9] [24] and its C code [7] on it to find the optimal scale factors and so as to minimize the condition number. The experiment results are sampled in the next section.

Algorithm 19 MaxEigenvalue(A)

Require: $A = [a_{ij}] \in \mathbf{R}^{n \times n}$ is nonnegative and irreducible.

// Computing the maximal eigenvalue λ and eigenvector x

$x \leftarrow [1, \cdots, 1]^T \in \mathbf{R}^{n \times 1}$
repeat
  for $i \leftarrow 1$ to n **do**
    the ith row sum $R_i \leftarrow \sum_{j=1}^{n} a_{ij}$
  end for
  max row sum $R \leftarrow \max\{R_i : i = 1, \cdots, n\}$
  min row sum $r \leftarrow \min\{R_i : i = 1, \cdots, n\}$
  index set $J \leftarrow \{i : R_i = r\}$
  for $i \leftarrow 1$ to n **do**
    $b_i \leftarrow \sum_{j \in J} a_{ij}$
  end for
  choose μ such that $R_\mu = R$
  choose ν such that $b_\nu = \min\{b_i : i \in J\}$
  $a \leftarrow 4b_\mu$
  $b \leftarrow 2R - 6b_\mu - 2b_\nu$
  $c \leftarrow R + r - 2b_\mu - 2b_\nu$
  if $a = 0$ **then**
    $d \leftarrow \frac{c}{b}$
  else
    $d \leftarrow \frac{-b + \sqrt{b^2 + 4ac}}{2a}$
  end if
  for $i \leftarrow 1$ to n **do**
    if $i \in J$ **then**
      $d_i \leftarrow d$
    else
      $d_i \leftarrow 1$
    end if
  end for
  $D \leftarrow \text{diag}[d_1, \cdots, d_n]$
  $D \leftarrow D / \|D\|_\infty$
  $x \leftarrow Dx$
  $A \leftarrow D^{-1} A D$
until R_i converges to λ for all $i = 1, \cdots, n$
return eigenvalue λ and eigenvector x

Algorithm 20 OptimalInterval(A)

Require: Input $A \in \mathbf{R}^{m\times n}$ is positive and has full rank $n \leq m$

1: $S \leftarrow$ SCALGM(A) // Algorithm 4
2: **if** $m = n$ **then**
3: $\quad S^{\dagger} \leftarrow Inverse(S)$ // Algorithm 11
4: **else**
5: $\quad S^{\dagger} \leftarrow PseudoInverse(S)$ // Algorithm 12
6: **end if**
7: $P \leftarrow S \cdot S^{\dagger}$
8: $\lambda \leftarrow MaxEigenvalue(\mathrm{P})$ // Algorithm 19
9: **return** interval $[\lambda/m,\ \lambda]$

6.4 Experimental Results

Let $A = [a_1|\cdots|a_n] \in \mathbf{R}^{m\times n}$ be given and let $\theta_j = \theta_j(A)$ be the angle, measured in degrees, between the jth column vector a_j of A and the linear subspace $\mathcal{L}(a_1, \cdots, a_{j-1}, a_{j+1}, \cdots, a_n)$ spanned by other columns of A. Denote $\theta_{\min} = \theta_{\min}(A) = \min\{\theta_1, \cdots, \theta_n\}$. In this section we will compare the condition number of some matrices that have been scaled using SCALGM to an empirical formula

$$f(A) = 1.4(90/\theta_{\min}(A)) - 0.4$$

that we have found to be a reasonable, if rough, approximation for $\kappa(A)$ in terms of the minimum angle $\theta_{\min}(A)$ discussed above. All angles will be measured in degrees.

6.4.1 Some Examples

Example 6.4.1 *Consider $A \in \mathbf{R}^{4\times 4}$ as follows:*

$$A = \begin{bmatrix} 0.1480770 & 7.2100848 & 0.1305800 & 15.4921243 \\ 9.9878992 & 57.2956969 & 1.4177203 & 0.0343257 \\ 0.0175972 & 7.9973998 & 22.2709231 & 1.2756261 \\ 0.3718530 & 0.1517864 & 80.5274769 & 0.5036384 \end{bmatrix}$$

Then, we have $\theta_1 = 7.6290198, \theta_2 = 7.5595667, \theta_3 = 85.4822349$, and $\theta_4 = 47.8432547$. Thus, $\theta_{\min}(A) = 7.5595667$. The 2-norm condition number $\kappa(A) = 64.1742100$, and $1.4(90/\theta_{\min}(A)) - 0.4 = 16.2676219$.

Now, let $S = \text{SCALGM}(A)$, then we have

$$S = \begin{bmatrix} 0.0155002 & 0.1315657 & 0.0021193 & 1.0000000 \\ 1.0000000 & 1.0000000 & 0.0220076 & 0.0021193 \\ 0.0050962 & 0.4037436 & 1.0000000 & 0.2278069 \\ 0.0297831 & 0.0021193 & 1.0000000 & 0.0248746 \end{bmatrix},$$

and we have $\theta_1 = 15.8723160, \theta_2 = 15.0347404, \theta_3 = 49.5444269$, and $\theta_4 = 59.2475015$. Thus, $\theta_{\min}(S) = 15.0347404$. The 2-norm condition number $\kappa(S) = 8.0578672$, and observe $1.4(90/\theta_{\min}(S)) - 0.4 = 7.9805903$, which is very close to $\kappa(S)$. The ratio $\rho(S)$ of min/max in magnitude is 0.0021193 . Also, observe that the closed path of $\min$ and $\max$ in S is

$$S = \begin{bmatrix} \times & \times & \min & \max \\ \times & \max & \times & \min \\ \times & \times & \times & \times \\ \times & \min & \max & \times \end{bmatrix},$$

with $S[1][3] \to S[1][4] \to S[2][4] \to S[2][2] \to S[4][2] \to S[4][3] \to S[1][3]$.

The estimated interval for the condition number μ of the optimal two-sided scaled matrix is $[0.6306089, 2.5224355]$.

Next. let $T = PatterSearch(S)$, then we have

$$T = \begin{bmatrix} 0.1577278 & 0.5316732 & 0.0011684 & 2.1062209 \\ 1.8848862 & 0.7485433 & 0.0022474 & 0.0008268 \\ 0.0618953 & 1.9473666 & 0.6580056 & 0.5726785 \\ 0.9153895 & 0.0258673 & 1.6651569 & 0.1582436 \end{bmatrix},$$

and we have $\theta_1 = 61.1521800, \theta_2 = 53.8934825, \theta_3 = 62.2106164$, and $\theta_4 = 61.2947051$. Thus, $\theta_{\min}(S) = 53.8934825$. The 2-norm condition number $\kappa(T) = 2.0020228$, and observe $1.4(90/\theta_{\min}(T)) - 0.4 = 1.9379450$, which is very close to $\kappa(T)$ and within the above estimated interval. The ratio $\rho(T)$ of min/max in magnitude is $0.00082680/2.10622093 = 0.00039255$. Table 6.1 shows the summary from above results.

	A	S	T
θ_1	7.6290198	15.8723160	61.1521800
θ_2	7.5595667	15.0347404	53.8934825
θ_3	85.4822349	49.5444269	62.2106164
θ_4	47.8432547	59.2475015	61.2947051
$\theta_{\min}$	7.5595667	15.0347404	53.8934825
ρ	-	0.0021193	0.00039255
κ	64.1742100	8.0578672	2.0020228
$f(\theta_{\min})$	-	7.9805903	1.9379450

Table 6.1: Interval for μ:$[0.6306089, 2.5224355]$

Example 6.4.2 *Consider $A \in \mathbf{R}^{4\times 4}$ as follows:*

$$A = \begin{bmatrix} 0.0926612 & 17.0784926 & 0.3127063 & 12.7526810 \\ 1.7811361 & 54.0213344 & 1.4953060 & 14.7655003 \\ 0.3460217 & 0.0680433 & 0.2626770 & 0.0227214 \\ 1.3745248 & 45.1500312 & 0.0505958 & 1.4314422 \end{bmatrix}$$

Now, let $S = \text{SCALGM}(A)$, then we have

$$S = \begin{bmatrix} 0.0243606 & 0.3890154 & 0.1082947 & 1.0000000 \\ 0.3805432 & 1.0000000 & 0.4208410 & 0.9409447 \\ 1.0000000 & 0.0170377 & 1.0000000 & 0.0195857 \\ 0.3513716 & 1.0000000 & 0.0170377 & 0.1091433 \end{bmatrix},$$

Next. let $T = PatterSearch(S)$, then we have

$$T = \begin{bmatrix} 0.1171181 & 0.6907233 & 0.4922486 & 1.4434180 \\ 1.0922594 & 1.0600407 & 1.1420381 & 0.8108517 \\ 1.1754749 & 0.0073965 & 1.1113581 & 0.0069121 \\ 0.9256365 & 0.9729141 & 0.0424350 & 0.0863229 \end{bmatrix},$$

The results for matrices A, S, and T regarding the angles θ_j, $\theta_{\min}$, the ratio ρ of $\min / \max$ in magnitude, the 2-norm condition numbers κ, the values of the fitting function $f(\theta_{\min})$, as well as the estimated interval for the optimal condition number μ are summarized in Table 6.2.

	A	S	T
θ_1	4.8352583	6.3541657	10.8802034
θ_2	4.9049531	14.7800779	14.5441070
θ_3	10.4000159	6.6880465	12.9141023
θ_4	10.0054682	17.3776228	16.0505446
$\theta_{\min}$	4.8352583	6.3541657	10.8802034
ρ	-	0.01703767	0.00039255
κ	460.2705191	23.9129780	14.4856257
$f(\theta_{\min})$	-	19.4295112	11.1806658

Table 6.2: Interval for μ:$[3.6213799, 14.4855194]$

Example 6.4.3 *Consider $A \in \mathbf{R}^{6\times 3}$ as follows:*

$$A = \begin{bmatrix} 4.2436341 & 0.0269694 & 0.0622383 \\ 0.0149600 & 10.8894167 & 49.9185153 \\ 2.3807229 & 0.0453816 & 0.4947489 \\ 0.0910166 & 0.0291091 & 0.0240973 \\ 0.3740146 & 0.2130833 & 0.2439024 \\ 0.0650553 & 0.1287199 & 0.0238598 \end{bmatrix}$$

Now, let $S = \text{SCALGM}(A)$, then we have

$$S = \begin{bmatrix} 1.0000000 & 0.0036574 & 0.0020965 \\ 0.0020965 & 0.8782185 & 1.0000000 \\ 1.0000000 & 0.0109700 & 0.0297065 \\ 1.0000000 & 0.1840528 & 0.0378462 \\ 1.0000000 & 0.3278651 & 0.0932186 \\ 0.8782185 & 1.0000000 & 0.0460428 \end{bmatrix},$$

Next. let $T = PatterSearch(S)$, then we have

$$T = \begin{bmatrix} 1.0638298 & 0.0166244 & 0.0209650 \\ 0.0005098 & 0.9124348 & 2.2857143 \\ 1.7021277 & 0.0797817 & 0.4753033 \\ 0.0000696 & 0.0000547 & 0.0000248 \\ 0.0000696 & 0.0000975 & 0.0000610 \\ 0.5338714 & 2.5974026 & 0.2631015 \end{bmatrix},$$

The results for matrices A, S, and T regarding the angles θ_j, $\theta_{\min}$, the ratio ρ of $\min / \max$ in magnitude, the 2-norm condition numbers κ, the values of the fitting function $f(\theta_{\min})$, as well as the estimated interval for the optimal condition number μ are summarized in Table 6.3.

	A	S	T
θ_1	88.6713931	55.7560570	73.4727064
θ_2	1.1215613	36.8741891	61.8206507
θ_3	1.1215571	42.4600963	63.7554493
$\theta_{\min}$	1.1215571	36.8741891	61.8206507
ρ	-	0.00209650	0.00000953
κ	245.2917162	4.1852132	1.6924102
$f(\theta_{\min})$	-	3.0170243	1.6381539

Table 6.3: Interval for μ:$[0.3607154, 2.1642922]$

6.4.2 An approximate equation for condition numbers of well-scaled matrices

Let $A \in \mathbf{R}^{n \times n}$ and let $S = \text{SCALGM}(A)$. For some column j of S, let $\theta_j(S)$ is the angle between the jth column of S and the linear subspace spanned by the remaining columns of S, and thus let $\theta_{\min}(S)$ is the minimum angle among them.

Below is an approximate equation we have found from our experiments for the relation between the condition numbers $\kappa(S)$ and the minimum angles $\theta_{\min}(S)$ for well-scaled matrices S obtained from our scaling method, namely SCALGM algorithm.

$$\kappa(S) \approx 1.4 \cdot \frac{90}{\theta_{\min}(S)} - 0.4 \,,$$

where $\theta_{\min}(S)$ is measured in degrees.

CHAPTER 7

Summary and Conclusions

We present an iterative algorithm, called SCALGM, that consists of two phases. The convergence is proved in Theorem 3.1.4 and Theorem 3.2.1, respectively. SCALGM works on any given nonzero matrix $A \in \mathbf{R}^{m\times n}$ to produce the row and column scale factors in the form of diagonal matrices D and E, respectively, such that the scaled matrix DAE satisfies the properties listed below:

P1: the maximum magnitude of elements in DAE is 1. (Lemma 3.1.1)

P2: the nonzero rows and columns of DAE are equilibrated in the ∞-norm. (Theorem 3.2.3)

P3: the minimum magnitude of nonzero elements in DAE is maximized. (Corollary 5.2.15)

Property **P2** holds for any scaled matrix, in contrast to the algorithm presented by Bunch [5] that works on any symmetric matrix A (with no null rows) such that the scaled matrix DAE is equilibrated in the ∞-norm. Property **P3** holds for the SCALGM algorithm provided A is a nonzero matrix. From the above properties, the range of the magnitude of nonzero elements in DAE is within an interval $[m, 1]$ for some $m \in \mathbf{R}$ depending on the given matrix A, and m is maximized.

Our numerical results show SCALGM can make the condition number reasonably small for badly scaled matrices. Moreover, for such m, we have:

P4: Algorithm SCALGM has the $CPEV$ property. (Definition 5.2.7 and Theorem 5.2.14) In particular, the minimum magnitude m of nonzero elements in DAE must occur in a pair or more.

Also, by SCALGM, DAE is symmetric whenever A is symmetric, thus we have:

P5: Algorithm SCALGM preserves symmetry. (Theorem 3.2.4)

Finally, an approximate equation is presented from our experiments for the relation between the condition numbers $\kappa(DAE)$ and the minimum angles $\theta_{\min}(DAE)$ for well-scaled matrices DAE obtained from our scaling method SCALGM with $S = DAE$:

P6:

$$\kappa(S) \approx 1.4 \cdot \frac{90}{\theta_{\min}(S)} - 0.4 \,,$$

where $\theta_{\min}(S)$ is measured in degrees.

There is every reason to believe that application of SCALGM to a matrix before beginning to apply an elimination method will:

- result in a good sequence of pivots,
- allow the diagnosis of any instability in the elimination process,
- help to prevent underflow or overflow, and
- give a smaller condition number that more realistically estimates the true condition of the matrix.

BIBLIOGRAPHY

[1] J. Albrecht. Minimal norms of nonnegative irreducible matrices. *Linear Algebra and its Applications*, 249(1–3):255–258, December 1996.

[2] Bauer, F. L. Optimally scaled matrices. *Numer. Math.*, 5:73–87, 1963.

[3] A. Berman, B. N. Parlett, and R. J. Plemmons. Diagonal scaling to an orthogonal matrix. *SIAM Journal on Algebraic and Discrete Methods*, 2(1):57–65, March 1981.

[4] Richard D. Braatz and Manfred Morari. Minimizing the Euclidean condition number. *SIAM Journal on Control and Optimization*, 32(6):1763–1768, November 1994.

[5] J. R. Bunch. Equilibration of symmetric matrices in the max-norm. *J. Assoc. Comput. Mach.*, 18:566–572, 1971.

[6] P. A. Businger. Matrices which can be optimally scaled. *Numer. Math.*, 12:346–348, 1968.

[7] J. P. Chandler. - private communication.

[8] A. R. Curtis and J. K. Reid. On the automatic scaling of matrices for Gaussian elimination. *J. Inst. Maths. Applics.*, 10:118–124, 1972.

[9] René de Vogelaere. Remark on algorithm 178: Direct search. *Commun. ACM*, 11(7):498, 1968.

[10] G. Forsythe and C. Moler. *Computer Solution of Linear Algebraic Systems*. Prentice–Hall, Englewood Cliffs, 1967.

[11] G. E. Forsythe and E. G. Strauss. On best conditioned matrices. *Proc.Amer.Math.Soc.*, 6:340–345, 1955.

[12] Leslie V. Foster. Gaussian elimination with partial pivoting can fail in practice. *SIAM J. Matrix Anal. Appl.*, 15(4):1354–1362, 1994.

[13] D. R. Fulkerson and P. Wolfe. An algorithm for scaling matrices. *SIAM Review*, 1962:142–146, 1962.

[14] G. H. Golub and J. M. Varah. On a characterization of the best L_2-scaling of a matrix. *SIAM J. Numer. Anal.*, 11:472–479, 1974.

[15] C. A. Hall and T. A. Porsching. Computing the maximal eigenvalue and eigenvector of a nonnegative irreducible matrix. *SIAM Journal on Numerical Analysis*, 5(3):470–474, September 1968.

[16] Nicholas J. Higham. Optimization by direct search in matrix computations. *SIAM J. Matrix Anal. Appl.*, 14(2):317–333, April 1993.

[17] L. Ingber. Adaptive simulated annealing (ASA). Technical report, Lester Ingber Research, McLean, VA, 1993.

[18] C. McCarthy and G. Strang. Optimal conditioning of matrices. *SIAM J. Numer. Anal.*, 10:370–388, 1973.

[19] E. E. Osborne. On preconditioning of matrices. *J. Assoc. Comput. Mach.*, 7:338–345, 1960.

[20] B. N. Parlett and C. Reinsch. Balancing a matrix for calculation of eigenvalues and eigenvectors. *Numer. Math.*, 13:292–304, 1969.

[21] Uriel G. Rothblum, Hans Schneider, and Michael H. Schneider. Scaling matrices to prescribed row and column maxima. *SIAM J. Matrix Anal. Appl.*, 15(1):1–14, 1994.

[22] Siegfried M. Rump. Optimal scaling for P-norms and componentwise distance to singularity. *IMA J. Numer. Anal.*, 2002. To appear.

[23] R. D. Skeel. Scaling for numerical stability in Gaussian elimination. *J. Assoc. Comput. Mach.*, 26:494–526, 1979.

[24] Lyle B. Smith. Remark on Algorithm 178 [E4]: Direct search. *Communications of the ACM*, 12(11):638–638, November 1969.

[25] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, Philadelphia, PA, 1997.

[26] A. van der Sluis. Condition numbers and equilibration matrices. *Numer. Math.*, 14:14–23, 1969.

[27] A. van der Sluis. Condition, equilibration, and pivoting in linear algebraic systems. *Numer. Math.*, 15:74–86, 1970.

[28] G. A. Watson. An algorithm for optimal ℓ_2 scaling of matrices. *IMA J. Numer. Anal.*, 11:481–492, 1991.

[29] J. H. Wilkinson. *Rounding Errors in Algebraic Processes*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1963.

[30] Stephen J. Wright. A collection of problems for which Gaussian elimination with partial pivoting is unstable. *SIAM J. Sci. Statist. Comput.*, 14(1):231–238, 1993.

APPENDIX A

Listing of Source Code

Our numerical results are obtained from running the following source code written in C/C++ by Chin-Chieh Chiang, Chandler [7] and Ingber [17].

For the results in Chapter 4, we use:

- `asa_app.h`
- `asa.h`
- `scalgm_asa.cpp`
- `scalgm.cpp`
- `scalgm.h`
- `asa.c`
- `asa_usr_asa.h`

For the results in Chapter 6, we use:

- `lindep.cpp`
- `maxeigen.cpp`
- `projection.h`
- `scalgm.cpp`
- `lindep.h`

- `maxeigen.h`
- `qrdecomp.cpp`
- `scalgm.h`
- `main.cpp`
- `projection.cpp`
- `qrdecomp.h`

To ensure the accuracy of our source code, all the experimental results obtained from our scource code agree with that from Maple 10. The source code is available upon request at chiangc@cs.okstate.edu .

VITA

Chin-Chieh Chiang

Candidate for the Degree of

Doctor of Philosophy

Dissertation: SCALING ALGORITHMS FOR MATRICES

Major Field: Computer Science

Biographical:

Personal Data: Born in Kaohsiung, TAIWAN on January 23, 1964.

Education:

Received the B.S. degree from Feng Chia University,Taichung, TAIWAN, 1987, in Applied Mathematics.
Received the M.S. degree from University of Connecticut, Storrs, Connecticut, USA, 1994, in Mathematics.
Received the M.S. degree from Oklahoma State University, Stillwater, Oklahoma, USA, 2002, in Mathematics.
Completed the requirements for the degree of Doctor of Philosophy with a major in Computer Science Oklahoma State University in December, 2007.

Experience:

Math Teacher, Li-Chi Senior High School, Kaohsiung, TAIWAN **1989 - 1992**

- Taught Math and served as a supervisor for students' Math research projects.

Software Developer, Upperspace Corporation, Pryor, Oklahoma, USA **2000 - 2001**

- Worked on *DesignCAD* and *ScanPro* software to develop drawing, image editing and processing tools by using Math background in 2D/3D equations and programming skills in Visual C++ and MFC.
- Developed software for generating product serial numbers and number verification.
- Developed heuristic algorithms for implementation and solving problems in software.

Name: Chin-Chieh Chiang Date of Degree: December, 2007

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: SCALING ALGORITHMS FOR MATRICES

Pages in Study: 87 Candidate for the Degree of Doctor of Philosophy

Major Field: Computer Science

We present an iterative algorithm, called SCALGM, which asymptotically scales both rows and columns of any given matrix such that each element of the scaled matrix is in the interval [-1, 1] and the elements of minimum magnitude are maximized. The object is to make the condition number reasonably small, thus causing the pivoting process in Gaussian elimination to work well, and to diagnose any instability in the elimination process. Numerical evidence is presented showing the effectiveness of the algorithm.

ADVISOR'S APPROVAL: _____