SURVEY AND COMPARISON OF MATRIX SCALING METHODS

By

CHUNG-CHUAN WANG

Bachelor of Science

National Chen-Kung University

Taiwan, R. O. C.

1983

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 1993

OKLAHOMA STATE UNIVERSITY

SURVEY AND COMPARISON OF MATRIX SCALING METHODS

Thesis Approved:

J P Chandler
Thesis Adviser

Hongbo Lu

Thomas C. Collins
Dean of the Graduate College

PREFACE

This paper presents a new matrix scaling method by Chandler for improving the stability of pivoting algorithms and making it easier to estimate the condition of a problem and the accuracy of the solution.

The algorithm has been implemented using FORTRAN, and compared with Hamming's matrix scaling method with respect to their performance prior to Gaussian elimination with the partial pivoting method. Numerical results on 36100 matrices tests indicate that the new method is efficient as well as robust.

I wish to express my sincere appreciation to my advisor, Dr. John P. Chandler, who introduced me to such an interesting topic to study and gave me intelligent advisement and patient guidance to solve problems through my graduate study in OSU. I extend sincere thanks to committee members Dr. K. M. George and Dr. H. Lu for their helpful advisement and suggestions.

My special thanks are due to my wife, Ru-Sheow Hwang and to my daughter, Angela, for their moral support and patience. My deepest appreciation is extended to my parents whose encouragement and understanding were precious throughout the study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION AND LITERATURE REVIEW

This thesis studies matrix scaling methods in numerical analysis. Matrix scaling methods are used to improve the stability of pivoting algorithms and make it easier to estimate the condition of a problem and the accuracy of the solution. These methods are useful in solving linear equation systems.

What is matrix scaling? It is the method of scaling a matrix by rows and columns (or the reverse) to make the largest element in any row or column equal to, or close to 1. How to scale a matrix? A matrix is scaled by multiplying any row or column by a nonzero number or substituting any unknown with a new one which is a multiple of the old one.

Forsythe and Moler (1967) state that "the need for proper scaling of a matrix is very compelling if we are to devise a program to solve as many linear equation systems as possible," but "it is quite unclear to us how to program a reasonable scaling of a general matrix." Since the lack of equality in the sizes of the elements of a matrix is responsible for the question, it is reasonable to try to scale the rows and columns of a matrix to balance it.

For a linear system of equations $Ax=b$, the various components x_i frequently have different physical measure units. To unify the system, we always need to scale the matrix before solving the linear system of equations. If we multiply two nonzero diagonal matrices D_1 to b and D_2 to x, the original equations are changed to $AD_2x = D_1b$ which is the same as $D_1^{-1}AD_2x = b$. Therefore, we can look at $D_1^{-1}AD_2$ as a scaled equivalent of A where $D_1^{-1}A$ is a row-scaled equivalent of A and AD_2 is a column-scaled equivalent of A. In general for scaling, it is not necessary to select the elements of D_1^{-1} and D_2 very precisely. In floating-point computation with base β, we could choose the elements of D_1^{-1} and D_2 as integer powers of the base β so that only the exponents of the float-point number elements of A are changed but the fractional parts or mantissas of them are unchanged (Forsythe and Moler, 1967, p.23). Therefore, there are no rounding errors in such scaling. If the elements of D_1^{-1} and D_2 are integer powers of the base β and A' $=D_1^{-1}AD_2$, A' is β-scaled equivalent to A. Forsythe and Moler (1967) state that "Let floating-point matrices A and A' $(=D_1^{-1}AD_2)$ be β-scaled equivalent. Suppose that $b=D_1b'$. Then, if the indices of the pivot elements and their order of selection have been fixed in advance, the solution by Gaussian elimination in floating-point arithmetic of the systems $Ax=b$ and $A'x'=b'$ will produce precisely the same significands in all answers and all intermediate numbers (unless there is an exponent overflow or underflow)". "The only possible effect that scaling a matrix by integer powers of β can have on elimination is to alter the choice of pivot elements". Therefore, improper scaling method may make a general pivoting method inadequate.

Let $Ax=b$ be a linear system with non-singular matrix and n unknowns. To figure out the effect of uncertainty of A and b on the solution x, we set the original linear system $Ax=b$ as $(A + \delta A)(x + \delta x)=b$ and $A(x + \delta x)=(b + \delta b)$.

For $(A + \delta A)(x + \delta x)=b$, we get that (Forsythe and Moler, 1967, p.23)

$$\frac{\|\delta x\|}{\|x+\delta x\|} \leq \mathrm{cond}(A)\frac{\|\delta A\|}{\|A\|} \quad \text{where } \mathrm{cond}(A)=\|A\|\cdot\|A^{-1}\| .$$

For $A(x + \delta x)=(b + \delta b)$, assuming $b\neq 0$ we get that (Forsythe and Moler, p.20, 1967)

$$\frac{\|\delta x\|}{\|x\|} \leq \mathrm{cond}(A)\frac{\|\delta b\|}{\|b\|} \quad \text{where } \mathrm{cond}(A)=\|A\|\cdot\|A^{-1}\| .$$

If the condition number is small, then small perturbations in A or b lead to small perturbations in x. The inequality cond(A)≥ 1 is always true (Kincaid and Cheney, 1991, p.166), because cond(A)=$\|A\|\cdot\|A^{-1}\|\geq\|AA^{-1}\|=1$. If c is any scalar, we can get cond(cA)=cond(A). If A is a symmetric matrix, we have cond(A)=$|\lambda_1|/|\lambda_2|$ where λ_1 and λ_2 are the eigenvalues of largest and smallest magnitude (Wilkinson, 1965, p.191). Defining the relative error of x by $\rho_x=\dfrac{\|\delta x\|}{\|x\|}$ and the corresponding relative residual by $\rho_r=\dfrac{\|\delta b\|}{\|b\|}$ or $\rho_r=\dfrac{\|\delta A\|}{\|A\|}$, we get

$$\frac{1}{cond(A)}\leq\frac{\rho_x}{\rho_r}\leq cond(A)$$ (Forsythe and Moler, 1967, p.54).

A matrix with a large condition number is said to be ill-conditioned. For an ill-conditioned matrix A, there may be small changes in A or b that can generate big changes in x (Hamming, 1971, p.117). If the condition number is of moderate size and the results are not sensitive to small changes in the coefficients, the matrix is said to be well-conditioned (Noble, 1969, p.231). In general, we decide whether the condition of a matrix is good or poor by the condition number. The bigger the condition number is , the worse the condition is, usually. Conversely, the smaller the value is, the better the condition is. However, poor scaling may inflate the condition number without degrading the accuracy of x.

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

We get $A^{-1}=\begin{bmatrix}1 & 0\\ 0 & 10^{10}\end{bmatrix}$ and the condition number $k_\infty(A)\approx 10^{10}$, but the numerical solution of a system with matrix A has no subtraction and is very accurate.

Since the condition number is an upper bound of the true condition, sometimes we can use it to as reference index. The condition number is defined with respect to a chosen norms. The natural norm associated with the 1, 2, and ∞ vector norms are

$$\|A\|_1 = \max_j \sum_{i=1}^{n} |a_{ij}| \text{ (maximum absolute column sum),}$$

$$\|A\|_2 = \{\text{maximum eigenvalue of } A^H A\}^{1/2} \text{ (spectral or Euclidean norm).}$$

$$\|A\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}| \text{ (maximum absolute row sum)} \quad \text{(Noble, 1969, p.429).}$$

For using pivots to get optimum solutions, we must avoid pivots elements which are small in absolute value because cancellation could cause incorrect results. Therefore, we need to choose the largest element in every column with row interchange. This method is the well-known partial pivoting method (Forsythe and Moler, 1967, p.35).

For equilibration, we scale the rows or columns of the matrix A such that the rows or columns attain more or less equal norms for some vector norm. In general we know that equilibration tends to reduce the condition number of matrix A (Van Der Sluis,1970, p.75). Therefore, how to design a proper scaling method is very important.

The following examples (Forsythe and Moler, 1967, p.34 and p.40) show the cases of (1) well-scaled problem without pivoting, (2) well-scaled problem with partial pivoting, and (3) badly-scaled problem with partial pivoting.

Example. Assuming three-decimal floating arithmetic, we need to solve the system

$$0.000100\, x_1 + 1.00\, x_2 = 1.00$$

$$1.00\, x_1 + 1.00\, x_2 = 2.00$$

the true solutions are x_1=1.00010≈1.00 and x_2=0.99990≈1.00

(1) well-scaled without pivoting:

The solution is by Gaussian elimination without interchange. We have

$$0.000100\, x_1 + 1.00\, x_2 = 1.00$$

$$-10{,}000\, x_2 = -10{,}000.$$

x_2=1.00 and x_1=0.00 (wrong due to subtractive cancellation of leading significant digits).

(2) well-scaled with partial pivoting :

The solutions are by Gaussian elimination with interchange. We have

$$1.00\ x_1 + 1.00\ x_2 = 2.00$$

$$1.00\ x_2 = 1.00$$

x_1=1.00 and x_2=1.00 (correct).

(3) badly-scaled with partial pivoting:

We multiply the first original equation by 10^5:

$$10.0\ x_1 + 100{,}000\ x_2 = 100{,}000$$

$$1.00\ x_1 + \quad 1.00 x_2 = \quad 2.00.$$

Because 10.0 > 1.00, after elimination we get

$$10.0\ x_1 + 100{,}000\ x_2 = 100{,}000$$

$$-10{,}000 x_2 = -10{,}000.$$

The results x_2=0.00 and x_1=1.00 (wrong) are incorrect also.

From the above examples we know (1) Solving a well-scaled problem without pivoting, we may get bad results. (2) Solving a badly scaled problem with pivoting, we may also get bad results. (3) Solving a well-scaled problem with pivoting, we get good results. Lack of pivoting can produce very poor results, but pivoting without scaling can also produce very poor results. For partial pivoting, row equilibration is sufficient. For complete pivoting, we need to scale both rows and columns.

A new method, to attempt to solve the problem of matrix scaling, was designed by Chandler. I will test this method with different sizes (from 2 to 20) of dimension to understand the condition of convergence after scaling the matrix each time. Similarly, I will test other methods in the same condition and compare the results.

This thesis is organized as follows:

In Chapter II , I will explain how to generate an exponentially random matrix, how to scale a symmetric positive definite matrix, how to scale a general matrix, how to judge whether the scaled matrix is converging exactly or linearly, the condition number k_{pp}, and Hamming's method.

In Chapter III, I will study the convergence of the scaled matrix, then I collect results and analyze data for any kind m*n (m,n>=2 and m,n<=20) dimension scaled matrix, each dimension with 100 different random matrices and give an example of problem of Gaussian elimination with the partial pivoting method and how to use Chandler's and Hamming's methods to solve linear equation systems.

In Chapter IV, I will give conclusions based on the previous chapters.

Finally, a source program which implements these different matrix scaling methods and output data will be put into appendices.

CHAPTER II

METHODOLOGY

In this chapter, I mainly describe Chandler's matrix scaling method (Chandler, 1992) and Hamming's matrix scaling method (Hamming, 1971, p.115). Therefore, this chapter will introduce (1) how to generate an exponentially random matrix, (2) how to scale a symmetric positive definite matrix, (3) how to scale a general matrix by Chandler's method, (4) how to judge whether the scaled matrix is converging exactly or linearly, (5) the condition number k_{pp}, and (6) Hamming's method.

How to Generate an Exponentially Random Matrix

A random matrix of our method is generated by a pseudo random number generator (Cheney and Kincaid, 1980, p.203). Using the pseudo random number generator with a seed (initial value), we can get a fixed sequence of floating-point numbers uniformly distributed between 0 and 1. Namely, if we set different seeds in initially, the random number generator generates different fixed sequences of floating-point numbers. The desirable purpose is easily to debug a program and get a new sequence of uniformly distributed numbers as the seed is changed. The following is the expression of the random number generator

$$l_i = (7^5 l_{i-1}) \bmod (2^{31}-1) \qquad (1)$$

$$x_i = \frac{l_i}{2^{31}-1} \quad \text{where } i \geq 1 \qquad (2).$$

The initial value l_0 is called the seed. After initializing the seed l_0 from formula (1), the sequence of l_1, l_2, ... is generated between 1 and $2^{31}-1$. From formula (2), we get a sequence of x_1, x_2, ... between 0 and 1. Besides the elements of the matrix including tiny numbers, we also want huge magnitude elements included in our random matrix so that we multiply the sequence of x_1, x_2, ... by 30.0 as exponents of the decimal base. We get all the elements of a random matrix in the range of $10^0 \sim 10^{30}$. We can set different numbers of rows and columns and use for-loops to call the generator with different seeds to get any dimension and different random matrices.

How to Scale a Symmetric Positive Definite Matrix

If A is a symmetric positive definite matrix, then A=$B^T B$. We could freely choose a matrix B, then scale B as || col. of B || = 1 by normalizing each column of B to unit length. From A=$B^T B$, we would then get the symmetric positive definite matrix A where a_{jj}=1, $|a_{jk}| \leq 1$ when j≠k (as large as possible). Similarly, we scale a symmetric positive definite matrix A by dividing a_{jk} by $\sqrt{a_{jj}a_{kk}}$. We also get the same matrix a_{jj}=1, $|a_{jk}| \leq 1$ when j≠k.

Example:

Set matrix $B = \begin{bmatrix} 8 & 2 & 2 \\ 1 & 2 & 6 \\ 4 & 1 & 3 \end{bmatrix}$.

(1) Scale B by || col. of B || = 1, we get $B = \begin{bmatrix} 8/9 & 2/3 & 2/7 \\ 1/9 & 2/3 & 6/7 \\ 4/9 & 1/3 & 3/7 \end{bmatrix}$. After $B^T B$, we get the matrix

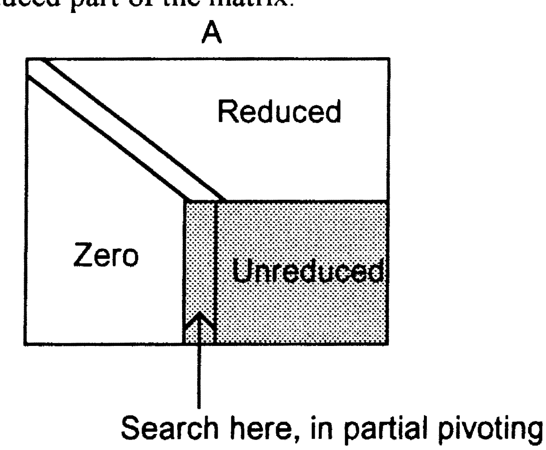
$$A = \begin{bmatrix} 1 & 22/27 & 34/63 \\ 22/27 & 1 & 19/21 \\ 34/63 & 19/21 & 1 \end{bmatrix}.$$

(2) From $A = B^T B$, we get $A = \begin{bmatrix} 81 & 22 & 34 \\ 22 & 9 & 19 \\ 34 & 19 & 49 \end{bmatrix}$. After dividing a_{jk} by $\sqrt{a_{jj}a_{kk}}$, we get the

same result $A = \begin{bmatrix} 1 & 22/27 & 34/63 \\ 22/27 & 1 & 19/21 \\ 34/63 & 19/21 & 1 \end{bmatrix}$.

How to Scale a General Matrix by Chandler's Method

If we perform Gaussian elimination on a general matrix using floating point arithmetic, we must use some kind of pivoting in order to have a stable algorithm that gives accurate results (Wilkinson, 1965, p.212; Forsythe and Moler, 1967, p.35). In partial pivoting we search on and below the main diagonal of the first column of the unreduced part of the matrix.



We use the element of largest magnitude as the next pivot.

If the matrix A is scaled arbitrarily, partial pivoting can give bad results even on a well-conditioned matrix A (Forsythe and Moler, 1967, p.40). To prevent this we should scale the matrix performing Gaussian elimination. For partial pivoting it is sufficient to scale only the rows of A. This operation, often called row equilibration (Wilkinson,1965, p.213), is carried out by dividing each equation by the coefficient of largest magnitude in that equation. In partial pivoting there is no reason to scale the columns of A, as this could not affect the choice of pivot.

An algorithm that is even more stable than partial pivoting is complete pivoting, in which the entire unreduced square portion of the matrix is searched at each stage. Again, the element of largest magnitude becomes the next pivot element. For complete pivoting it is necessary to scale columns as well as rows. One might think that you could just "scale rows, then scale columns", in each case dividing by the coefficient of largest magnitude. Unfortunately, this method often gives different results than "scale columns, then scale rows", and there is no rationale for choosing one method over the other. Also, either method can leave some elements with a smaller magnitude than necessary, allowing greater error than necessary in the numerical solution of the linear system.

We wish to adapt the simple "scale rows then scale columns" algorithm to one that will still scale any matrix but which, if applied to a symmetric positive definite matrix, will produce the diagonal scaling shown above in one iteration. One method that will do that is

(1) Scale rows by dividing by the square root of the largest magnitude element,

(2) Scale columns,

(3) Scale rows.

Expressed in the mathematical formula, this method of "scale down" is:

(1) $DL(J) = \frac{DL(J)}{\sqrt{\max|DL(J)*A(J,K)*DR(K)|}}$ where J= 1 to number of rows,

(2) $DR(k) = \frac{DR(K)}{\max|DL(J)*A(J,K)*DR(K)|}$ where K= 1 to number of columns,

(3) $DL(J) = \frac{DL(J)}{\max|DL(J)*A(J,K)*DR(K)|}$ where J= 1 to number of rows.

This algorithm scales down large elements. It has at least three disadvantages, however, (1) The small elements are often smaller than necessary. (2) The condition number is often larger than necessary. (3) The results of this rows-columns-rows algorithm is different from the corresponding columns-rows-columns algorithm. The small elements need to be scaled up. Perhaps the simplest way to scale elements up is to invert each nonzero element, scale the matrix down and invert the elements back again. So we alternate the "scale up" and scale down" procedures iteratively, finishing with "scale down" to produce the desired elements of unit magnitude in the final scaled matrix. Expressed in mathematical formulas, this method of "scaling up" is:

(1) $DL(J) = \frac{DL(J)}{\sqrt{\min|DL(J)*A(J,K)*DR(K)|}}$ where J=1 to number of rows,

(2) $DR(K) = \frac{DR(K)}{\min|DL(J)*A(J,K)*DR(K)|}$ where K=1 to number of columns,

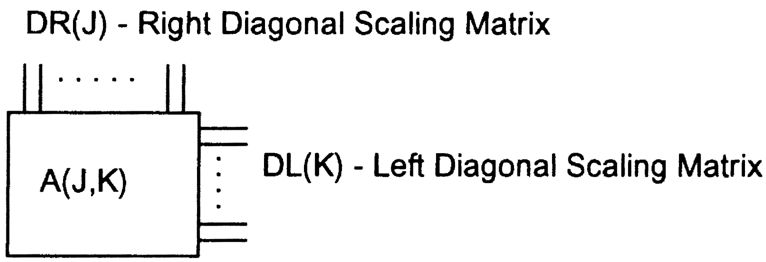
(3) $DL(J) = \frac{DL(J)}{\min|DL(J)*A(J,K)*DR(K)|}$ where J=1 to number of rows.

In this procedure two switches are used, one called INV to control scaling up (INV=1) or down (INV=0), the other is called MRC to decide whether to scale rows (MRC=1) or columns (MRC=0) first. We set two arrays as the left and right diagonal scaling matrices to keep the magnitude of left (DL(*)) and right (DR(*)) diagonal scaling matrices after every scaling down or up. These two arrays are set to 1.0 in all elements initially. The usage of this subroutine is to call SCALA with INV=1 and then with INV=0. Repeat this pair of calls until the DL and the DR diagonal matrices all converge, but compare their values only after the calls with INV=0, so that the final operation is a "scale down".

Note:

After scaling, the original system $Ax=b$ is changed to $DL(J)A(J,K)DR(K)x' = DL(J)b'$

$$\text{where } DL = \begin{bmatrix} d_1^{(l)} & & & \\ & \bullet & & \\ & & \bullet & \\ & & & d_n^{(l)} \end{bmatrix} \quad DR = \begin{bmatrix} d_1^{(r)} & & & \\ & \bullet & & \\ & & \bullet & \\ & & & d_n^{(r)} \end{bmatrix}.$$



How to Judge Whether the Scaled Matrix Is Converging Exactly Or Linearly

This method has the extremely unusual feature that some elements of DL and DR may converge exactly to their final values in a finite number of iterations while other elements converge at only a linear rate toward their limits. The iteration has never failed to converge, although convergence has not been proved. Repeatedly scale up and down the matrix till the scaled matrix converges. The ratio of magnitude of the change of elements of the diagonal scaling matrices to the elements themselves is used to judge whether the matrix is converging exactly or linearly. In general, if the ratio is close to 0 ($\leq 10^{-13}$) and the number of iterations is less than (number of rows + number of columns + 10), the matrix has converged exactly. On the other hand, if the ratio is converging slowly or the number of iterations is greater than (number of rows + number of columns + 10), the matrix is converging linearly.

Expressed in a mathematical formula,

$$\text{ratio}=\frac{|\Delta DL(J)|}{\max(|DL(J)|,|DL\_SAVE\ (J)|)} \text{ or } \frac{|\Delta DR(K)|}{\max(|DR(K)|,|DR\_SAVE\ (K)|)}$$

where $DL\_SAVE(J)$=previous value of $DL(J)$, $\Delta DL(J) = DL(J) - DL\_SAVE\ (J)$

$DR\_SAVE(K)$=previous value of $DR(K)$, $\Delta DR(K) = DR(K) - DR\_SAVE\ (K)$.

Exactly : ratio $\le 10^{-13}$ and Num_Iter. < (number of rows + number of columns + 10)

Linearly : not exactly

This method has always produced the same scaling whether MRC=0 or MRC=1, and the small elements of the scaled matrix always seem to be as large as possible, in some sense.

The Condition Number

In general we know how to use the condition numbers k_∞ (row max) and k_1 (column max) to measure the condition of a matrix, but these reference indices relate to the absolute error, not the relative error. For a more accurate measure of the condition of a matrix, we need to use a reference index of relative error. k_{pp}, the condition number of Gaussian elimination of partial pivoting, is a reference index of relative error.

$$k_{pp}=\frac{largest\_|element|\_in\_scaled\_matrix}{smallest\_|pivot|\_in\_Gaussian\_elim.with\_p.p.on\_the\_scaled\_matrix}$$

Example:

The matrix is $\begin{bmatrix} 6 & 12 & 24 \\ 3 & 4 & 5 \\ 2 & 8 & 10 \end{bmatrix}$. The largest element of absolute value in the matrix is 24.

Using Gaussian elimination with the partial pivoting method, we get the matrix

$\begin{bmatrix} 6 & 12 & 24 \\ 3 & 4 & 5 \\ 2 & 8 & 10 \end{bmatrix} \Rightarrow \begin{bmatrix} 6 & 12 & 24 \\ 0 & 2 & -5 \\ 0 & 4 & 2 \end{bmatrix} \Rightarrow \begin{bmatrix} 6 & 12 & 24 \\ 0 & 4 & 2 \\ 0 & 0 & -6 \end{bmatrix}$. The smallest of absolute value of pivots (6, 4, -6) is 4. Therefore, we get the condition number k_{pp}= 24/4=6.

Hamming's Method

This method is a method of simultaneous row and column scaling. At first the rows are multiplied by $2^{r_i} (i = 1,\dots,m)$ and the columns are multiplied by $2^{c_j} (j = 1,\dots,n)$ and all of the m*n elements are multiplied by 2^M. If the matrix $A = \begin{bmatrix} a_{1,1} \cdots a_{1,n} \\ \dots\dots\dots\dots \\ a_{m,1} \cdots a_{m,n} \end{bmatrix}$ and $|a_{i,j}| = 2^{b_{ij}}$ where i=1,...,m and j=1,...,n then the new exponent is $b_{i,j} + M + r_i + c_j$.

To minimize the sum of the squares of the *logs* of the absolute values of the nonzero elements in the scaled matrix,

we let $m = \sum_{i=1}^{n} \sum_{j=1}^{n+1} (b_{i,j} + M + r_i + c_j)^2$. Then we differentiate with r_i and c_j to get

$$\frac{\partial m}{\partial r_i} = 2\sum_{j=1}^{n+1} (b_{i,j} + M + r_i + c_j) = 0 \qquad (i=1,\dots,m)$$

$$\frac{\partial m}{\partial c_j} = 2\sum_{i=1}^{n} (b_{i,j} + M + r_i + c_j) = 0 \qquad (j=1,\dots,n)$$

and we set the negative of the average of all the $b_{i,j}$ as

$$M = -\frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} b_{i,j} \text{ =-(average matrix element)}$$

Therefore, we get

$$r_i = \frac{-1}{n}\sum_{j=1}^{n}(b_{i,j} + M) = \text{-(average row element} + M\text{) and}$$

$$c_j = \frac{-1}{m}\sum_{i=1}^{m}(b_{i,j} + M) = \text{-(average column element} + M\text{).}$$

The new exponent

$b_{i,j} + M + r_i + c_j$

= $b_{i,j}$ - M - (average row element) - (average column element)

= $b_{i,j}$ + (- $M/2$ - (average row element)) + (-$M/2$ - (average column element))

= $b_{i,j}$ + ((half average matrix element) - (average row element))

+ ((half average matrix element) - (average column element))

Therefore, we can design Hamming's scaling method as the following six steps:

(1) accumulate the logarithm of the absolute value of the elements in each row, then get the average (ROW_AV) of each row.

(2) accumulate the logarithm of the absolute value of the elements in each column, then get the average (COL_AV) of each column.

(3) accumulate the logarithm of the absolute value of the elements in the matrix, then get the half average (HALF_AV) of the matrix.

(4) DL(J)=EXP(HALF_AV-ROW_AV(J)).

(5) DR(K)=EXP(HALF_AV-COL_AV(K)).

(6) A'(J,K)=DL(J)*A(J,K)*DR(K).

CHAPTER III

RESULTS

In this chapter, I mainly (1) make an illustration of the convergence of the scaled matrix, (2) collect results and data analysis, including iteration and percentage of exact convergence and reduction of the condition number by Chandler's and Hamming's methods, for several sizes m*n (m,n>=2 and m,n<=20) dimension scaled matrix, each dimension with 100 different random matrices, (3) give an example of a problem of Gaussian elimination with the partial pivoting method and how to use Chandler's and Hamming's methods to solve linear equation systems.

Illustration of Using Chandler's Method: the Convergence of the Scaled Matrix

Example of Linear Convergence of Chandler's Method

The matrix to be scaled is $\begin{bmatrix} 2.814985E+06 & 2.213478E+06 & 4.762788E+21 \\ 6.519498E+19 & 3.142927E+17 & 4.973675E+17 \\ 1.139945E+08 & 1.114884E+22 & 6.154004E+17 \end{bmatrix}$.

The condition numbers are k_∞=1.7104E+02 and k_1=1.7101E+02. The inital left (DL(*)) and right (DR(*)) diagonal scaling matrices are $DL = DR = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. The convergence of the DL and DR matrices is shown in TABLE I.

TABLE I

THE CONVERGENCE OF DL AND DR MATRICES

Iteration		DL	$^{\Delta DL}/_{DL}$	Ratio	DR	$^{\Delta DR}/_{DR}$	Ratio
1	(1,1)	1.118E-08	-1.000E+00		6.741E-08	-1.000E+00	
	(2,2)	2.275E-13	-1.000E+00		6.026E-11	-1.000E+00	
	(3,3)	1.488E-12	-1.000E+00		1.876E-14	-1.000E+00	
2	(1,1)	6.871E-08	8.372E-01		2.488E-07	7.291E-01	0.529
	(2,2)	6.163E-14	-7.291E-01		1.002E-10	3.989E-01	0.340
	(3,3)	8.945E-13	-3.989E-01		3.055E-15	-8.372E-01	0.071
3	(1,1)	1.081E-07	3.648E-01	0.685	3.449E-07	2.786E-01	0.305
	(2,2)	4.446E-14	-2.786E-01	0.103	1.138E-10	1.195E-01	0.271
	(3,3)	7.876E-13	-1.195E-01	0.180	1.940E-15	-3.648E-01	0.187
4	(1,1)	1.211E-07	1.072E-01	0.329	3.742E-07	7.838E-02	0.263
	(2,2)	4.098E-14	-7.838E-02	0.203	1.175E-10	3.132E-02	0.255
	(3,3)	7.629E-13	-3.132E-02	0.231	1.732E-15	-1.072E-01	0.232
5	(1,1)	1.246E-07	2.796E-02	0.268	3.819E-07	2.020E-02	0.253
	(2,2)	4.015E-14	-2.020E-02	0.237	1.185E-10	7.923E-03	0.251
	(3,3)	7.569E-13	-7.923E-03	0.245	1.684E-15	-2.796E-02	0.246
6	(1,1)	1.255E-07	7.065E-03	0.254	3.839E-07	5.088E-03	0.251
	(2,2)	3.995E-14	-5.088E-03	0.247	1.187E-10	1.987E-03	0.250
	(3,3)	7.554E-13	-1.987E-03	0.249	1.672E-15	-7.065E-03	0.249
7	(1,1)	1.257E-07	1.771E-03	0.251	3.844E-07	1.275E-03	0.250
	(2,2)	3.990E-14	-1.275E-03	0.249	1.187E-10	4.970E-04	0.250
	(3,3)	7.550E-13	-4.970E-04	0.249	1.669E-15	-1.771E-03	0.250
8	(1,1)	1.258E-07	4.430E-04	0.250	3.845E-07	3.188E-04	0.250
	(2,2)	3.988E-14	-3.188E-04	0.250	1.188E-10	1.243E-04	0.250
	(3,3)	7.549E-13	-1.243E-04	0.250	1.668E-15	-4.430E-04	0.250
9	(1,1)	1.258E-07	1.108E-04	0.250	3.845E-07	7.971E-05	0.250
	(2,2)	3.988E-14	-7.971E-05	0.250	1.188E-10	3.107E-05	0.250
	(3,3)	7.549E-13	-3.107E-05	0.250	1.668E-15	-1.108E-04	0.250
10	(1,1)	1.258E-07	2.770E-05	0.250	3.845E-07	1.993E-05	
	(2,2)	3.988E-14	-1.993E-05	0.250	1.188E-10	7.768E-06	
	(3,3)	7.549E-13	-7.768E-06	0.250	1.668E-15	-2.770E-05	

After scaling, the matrix is changed to

$$\begin{bmatrix} 1.3623885E-07 & 3.3096120E-11 & 1.0000000E+00 \\ 1.0000000E+00 & 1.4893515E-06 & 3.3096124E-11 \\ 3.3096121E-11 & 1.0000000E+00 & 7.7511279E-10 \end{bmatrix}.$$

From the above table, the relative differences ${}^{\Delta DL}/_{DL}$ and ${}^{\Delta DR}/_{DR}$ are getting smaller and smaller in what we call linear convergence. The ratios, the quotients of any two successive ΔDL or ΔDR, finally all converge to 0.250. As for condition numbers, they are changed from the original k_∞=171.04 and k_1=171.01 to k_∞=1.00 and k_1=1.00 which are optimal. Moreover, the condition number k_{pp} is changed from original 171.01 to 1.00. Because the minimum value of a condition number is 1, the matrix is well scaled.

Using Hamming's scaling method, the matrix is changed to

$$\begin{bmatrix} 6.056871E-02 & 6.629677E-06 & 2.490344E+06 \\ 2.003158E+05 & 1.344251E-01 & 3.713679E-05 \\ 8.242073E-05 & 1.122089E+06 & 1.081275E-02 \end{bmatrix}.$$

The condition numbers of Hamming's method are changed from k_∞=171.04 and k_1=171.01 to k_∞=12.432 and k_1=12.432. The condition number k_{pp} is changed from 171.008 to 12.43 also. Therefore, using Hamming's method in this example, we also get good results for the scaled matrix, but not optimal.

Example of Exact Convergence of Chandler's Method

The matrix to be scaled is $\begin{bmatrix} 7.915819E+02 & 1.046740E+25 & 2.698205E+28 \\ 1.192224E+01 & 2.220116E+00 & 3.424969E+01 \\ 9.410550E+22 & 3.530955E+17 & 2.654004E+27 \end{bmatrix}$. The condition numbers are k_∞=2.2264E+27 and k_1=2.6679E+27. The inital left (DL(*)) and right (DR(*)) diagonal scaling matrices are $DL = DR = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. The convergence of the DL and DR matrices is shown in TABLE II.

TABLE II

THE CONVERGENCE OF DL AND DR MATRICES

Iteration		DL	$\Delta DL/DL$	Ratio	DR	$\Delta DR/DR$	Ratio
1	(1,1)	2.934E-15	-1.000E+00		5.558E-02	-9.444E-01	
	(2,2)	1.508E+00	3.373E-01		3.255E-11	-1.000E+00	
	(3,3)	1.911E-22	-1.000E+00		1.262E-14	-1.000E+00	
2	(1,1)	1.059E-15	-6.388E-01		2.007E-02	-6.388E-01	
	(2,2)	4.177E+00	6.388E-01		9.013E-11	6.388E-01	
	(3,3)	5.292E-22	6.388E-01		3.496E-14	6.388E-01	
3	(1,1)	1.059E-15	0.000E+00	0.000	2.007E-02	0.000E+00	0.000
	(2,2)	4.177E+00	0.000E+00	0.000	9.013E-11	-1.434E-16	0.000
	(3,3)	5.292E-22	0.000E+00	0.000	3.496E-14	0.000E+00	0.000

After scaling, the matrix is changed to

$$\begin{bmatrix} 1.6844848E-14 & 1.0000000E+00 & 1.0000000E+00 \\ 1.0000000E+00 & 8.3600273E-10 & 5.0032545E-12 \\ 1.0000000E+00 & 1.6844848E-14 & 4.9117968E-08 \end{bmatrix}$$

. The condition numbers are changed to k_∞=8.0082E+07 and k_1=8.0082E+07 and the condition number k_{pp} is changed from the original 7.8912E+22 to 2.00204E+07. Therefore, the matrix using Chandler's method is scaled well in this case of exact convergence. Using Hamming's method, the scaled matrix is changed to

$$\begin{bmatrix} 1.869264E-11 & 1.177970E+06 & 4.541454E+04 \\ 1.765520E+05 & 1.566791E-01 & 3.615068E-05 \\ 3.030099E+05 & 5.418195E-06 & 6.090999E-01 \end{bmatrix}$$

. The condition numbers are changed to k_∞=5.3649E+06 and k_1=3.3898E+06. The condition number k_{pp} is changd from the original 7.89123E+22 to 3.26395E+06. Therefore, using Hamming's method to scale this matrix we also get good results in this case, in fact better than for Chandler's method, as judged by these three condition numbers.

Results Colection And Data Analysis

For more details on Chandler's method, I tested m*n matrices where m,n <=20 and m,n >=2. Each dimenson includes 100 different matrices generated from 100 different random number seeds. After scaling all these 36100 matrices, I collected all results, mainly divided into three parts which are the average number of iterations of exact convergence, the percentage of elements of DL, DR that converged exactly, and the reduction of the condition number k_∞ and k_{pp} by Chandler's and Hamming's methods Furthermore, I draw the comparison Figure 1 to Figure 7 from the results of (1) n*n (2) 2*n (3) 3*n (4) 5*n (5) 10*n (6) 15*n (7) 20*n matrices where n>=2 and n<=20.

The Average Iteration of Exact Convergence

We define the average iteration of exact convergence as

$$Avg.Exact\_Conv.Iter. = \frac{\sum Exact\_Conv.Iter.}{Num.of.Matrices}.$$

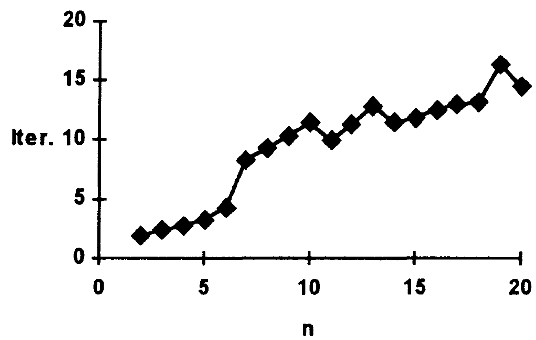


Figure 1. Iter. of Exac Conv. of n*n matrices

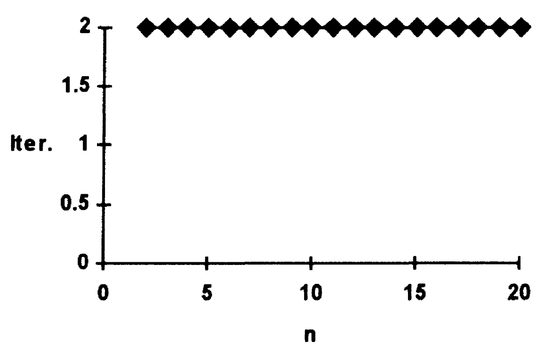


Figure 2. Iter. of Exac Conv. of 2*n matrices

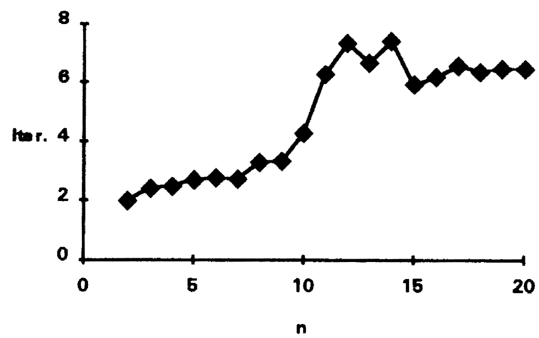


Figure 3. Iter. of Exac Conv. of 3*n matrices

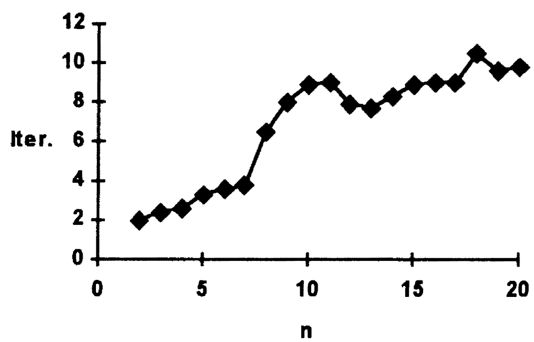


Figure 4. Iter. of Exac Conv. of 5*n matrices

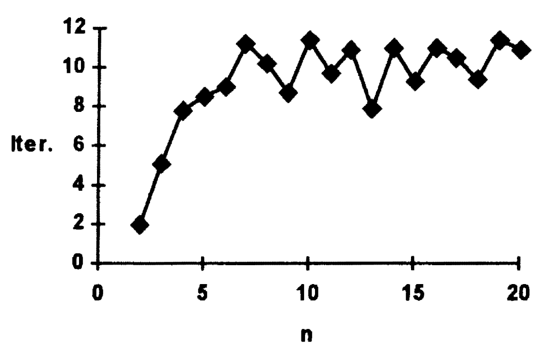


Figure 5. Iter. of Exac Conv. of 10*n matrices

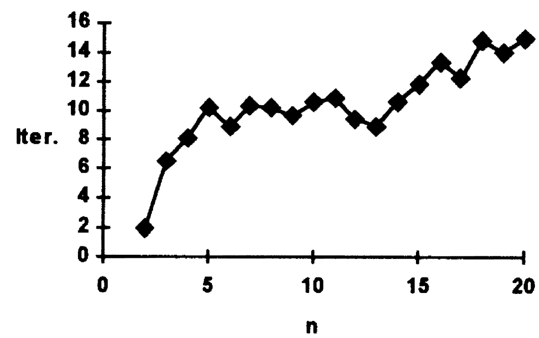


Figure 6. Iter. of Exac Conv. of 15*n matrices

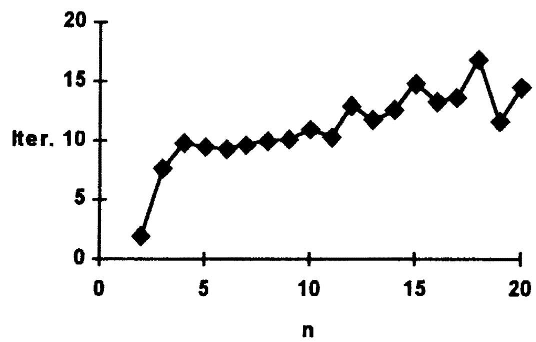


Figure 7. Iter. of Exac Conv. of 20*n matrices

For the above graphs, using Chandler's method, in general we get the following conclusions :

(1) For n*n matrices, the iteration of exact convergence tends to increase with increasing n.

(2) For any kind of 2*n matrix, it converges exactly on the second iteration.

(3) For an m*n matrix, the smallest m=3 that gives linear convergence.

(4) For m*n matrices, if m is fixed, in general, the larger n is, the more iterations exact convergence requires.

Percentage of Elements of DL, DR That Converged Exactly

The following Figures from Figure 8 to Figure 14 are the percentage of elements that converge exactly.

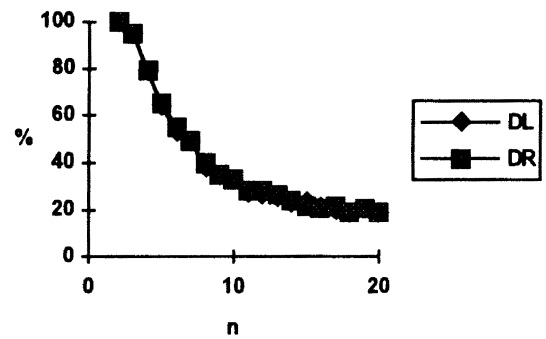


Figure 8. Percentage of Elements with Exact Conv. of n*n matrices

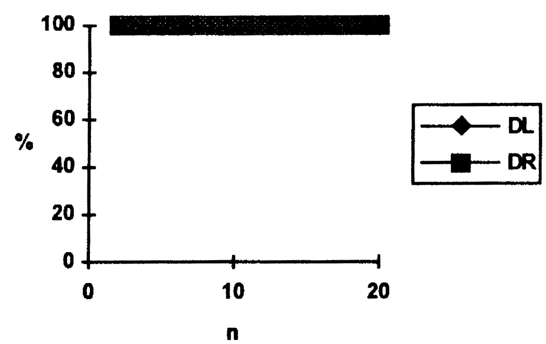


Figure 9. Percentage of Elements with Exact Conv. of 2*n matrices

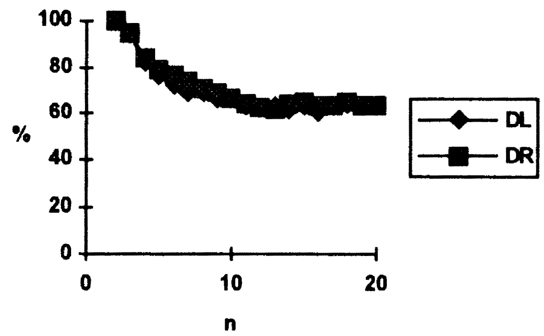


Figure 10. Percentage of Elements with Exact Conv. of 3*n matrices

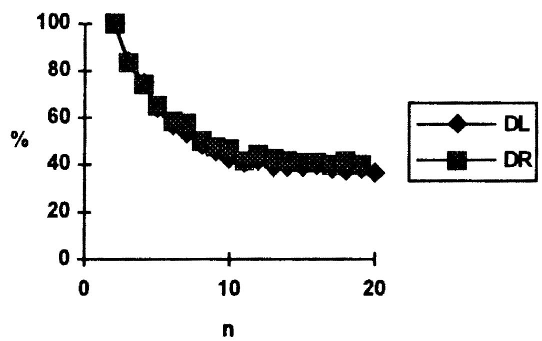


Figure 11. Percentage of Elements with Exact Conv. of 5*n matrices

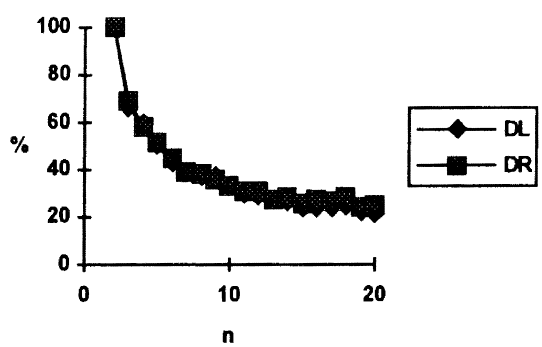


Figure 12. Percentage of Elements with Exact Conv. of 10*n matrices

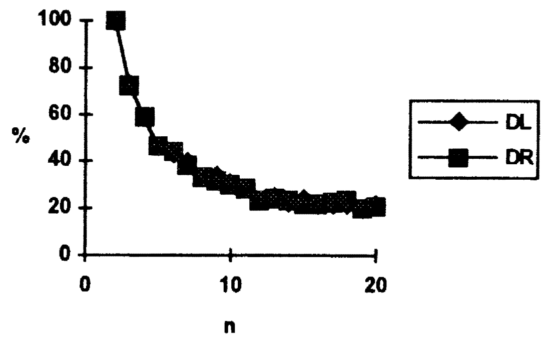


Figure 13. Percentage of Elements with Exact Conv. of 15*n matrices

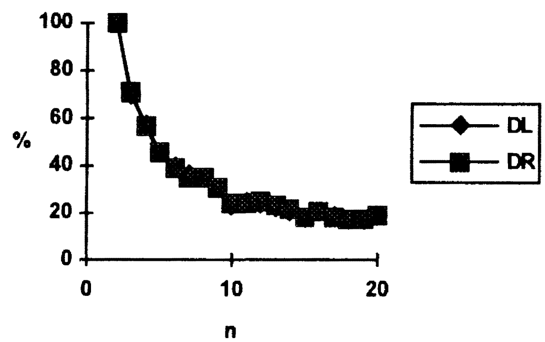


Figure 14. Percentage of Elements with Exact Conv. of 20*n matrices

From the above graphs, we get the following conclusions about the percentage of elements of DL, DR that converged exactly:

(1) For 2*n matrices, 100% of them give exact convergence.

(2) The curves of DL and DR are very close.

(3) For m*n matrices where n>10, the curve tends to a horizontal asymptote. The curve of 2*n matrices is 100%, 3*n matrices tend to about 60% , 5*n matrices tend to about 40% , 10*n, 15*n, and 20*n matrices tend to about 20%.

Reduction of the Condition Number

I calculated the condition number of n*n matrices where 2<=m<=20. To point out the improvement due to the scaling methods, I accumulated the condition number k_∞ and k_{pp} of Chandler's and Hamming's methods. Then, I get the reduction of both method's condition numbers from the original matrix by the following formulas

$$REDU\_K_{chan} = \log_{10}\frac{\sum K_{chan}}{\sum K_{orig}} \quad \text{and} \quad REDU\_K_{ham} = \log_{10}\frac{\sum K_{ham}}{\sum K_{orig}}$$

The following Figures from Figure 15 to Figure 18 show the reduction of the condition number k_∞ and k_{pp}.

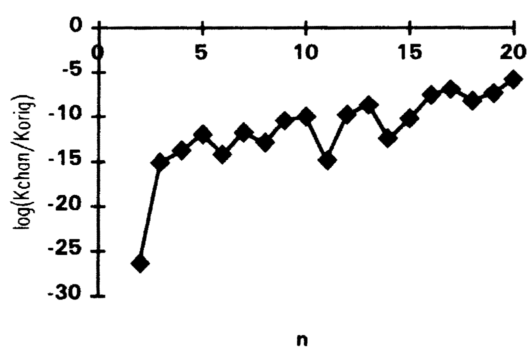


Figure 15. REDU_Kchan of n*n matrices

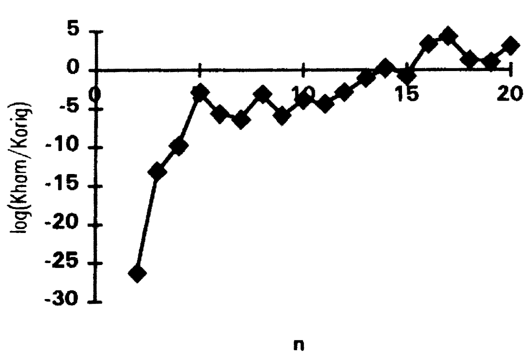


Figure 16. REDU_Kham of n*n matrices

$$REDU\_Kpp_{chan} = \log_{10} \frac{\sum Kpp_{chan}}{\sum Kpp_{orig}} \text{ and } REDU\_Kpp_{ham} = \log_{10} \frac{\sum Kpp_{ham}}{\sum Kpp_{orig}}$$

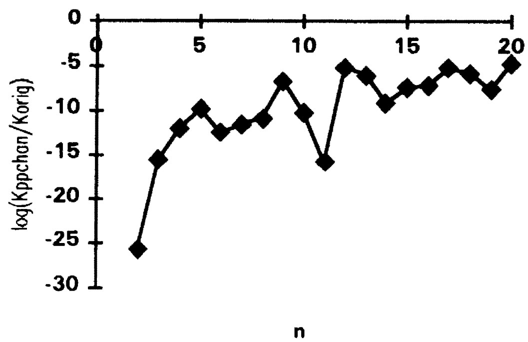


Figure 17. REDU_Kppchan of n*n matrices

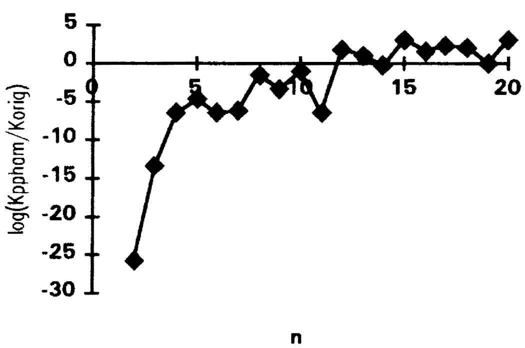


Figure 18. REDU_Kppham of n*n matrices

Note that if the REDU_K is less than zero, the condition number is reduced. On the other hand, if the REDU_K is greater than zero, the condition number is increased.

From the above graphs, we get the following conclusions about the reducing rate of REDU_Kchan, REDU_Kham, REDU_Kppchan, and REDU_Kppham:

(1) All the reduction of REDU_Kchan and REDU_Kppchan are less than zero. In general, the more n value is, the less reducing is.

(2) For the curves of REDU_Kham and REDU_Kppham, we can roughly draw two lines from graphs . The first line where n is between 2 and 5 is ascending steeply (slope around 5/3) and straight. The second line where n > 5 has its slope is around 1. From the figures, the reduction of the condition number is decreased drastically when dimension n of matrices is between 2 and 5 and it is decreased slowly when dimension n of matrices is between 6 and 11 but when the dimension n>15 it is increased.

Example of problem of Gaussian elimination with partial pivoting method and how to use Chandler's and Hamming's scaling method to solve linear equation system

For a linear system of equations $Ax=b$,Gaussian elimination with the partial pivoting method is that we select the pivot to be the largest-sized absolute coefficient in the next column and use the corresponding equation as the basis for the elimination process. In the following examples, we use three-decimal floating-point arithmetic and partial pivoting.

Example :

$$x_1+2x_2+3x_3=6$$

$$x_1- x_2+ x_3=1$$

$$2\cdot 10^{-4}x_1+ x_2+ x_3=2.$$

The rounded solution is x_1=1.0010, x_2=1.0004, x_3=0.9994.

If we use Gaussian elimination with the partial pivoting method , after the forward elimination we get

$$x_1+2x_2+3x_3=6$$

$$- 3x_2-2x_3=-5$$

$$0.332x_3=.333.$$

Back solving, we get x_1=1.00 , x_2=1.00, x_3=1.00 (the order of pivoting was 1, 2, 3).

The other method is to scale the matrix first with $A'=D_1^{-1}AD_2$

To form $Ax=b$, we get $A=\begin{bmatrix} 1 & 2 & 3 \\ 1 & -1 & 1 \\ 2\cdot 10^{-4} & 1 & 1 \end{bmatrix}$ and $b=\begin{bmatrix} 6 \\ 1 \\ 2 \end{bmatrix}$.

Let $D_1^{-1}=\begin{bmatrix} 1 & & \\ & 1 & \\ & & 10^4 \end{bmatrix}$ and $D_2=\begin{bmatrix} 1 & & \\ & 10^{-4} & \\ & & 10^{-4} \end{bmatrix}$.

After scaling (Forsythe and Moler, p41).,

$$A'=\begin{bmatrix} 1 & 2\cdot 10^{-4} & 3\cdot 10^{-4} \\ 1 & -10^{-4} & 10^{-4} \\ 2 & 1 & 1 \end{bmatrix}$$

and the equations become

$$x_1+2\cdot 10^{-4}x_2+3\cdot 10^{-4}x_3=6$$
$$x_1-\ 10^{-4}x_2+\ 10^{-4}x_3=1$$
$$2x_1+\ x_2+\ x_3=2\cdot 10^4.$$

Using Gaussian elimination with partial pivoting, we select 2 as our first pivot and eliminate x_1. The equations become

$$2x_1+\ x_2+\ x_3=2\cdot 10^4$$
$$-0.500x_2-0.500x_3=-10^4$$
$$-0.500x_2-0.500x_3=-10^4.$$

Eliminating x_2, the equations become

$$2x_1+\ x_2+\ x_3=2\cdot 10^4$$
$$-0.500x_2-0.500x_3=-10^4$$
$$0=0.$$

which are singular. The order of pivoting was 3, 2, 1. A general scaling makes the normal pivoting strategies incorrect.

We can express this example with $Ax=b$ where $A=\begin{bmatrix} 1 & 2 & 3 \\ 1 & -1 & 1 \\ 2\cdot 10^{-4} & 1 & 1 \end{bmatrix}$ and $b=\begin{bmatrix} 6 \\ 1 \\ 2 \end{bmatrix}$.

Substituting x with $D_R x'$ and b with $D_1 b'$, the system is changed to $A D_R x' = D_1 b'$. Multiplying D_1^{-1} to both sides of the equal sign, the system could be rewritten as $D_1^{-1} A D_R x' = b'$. Let $D_L = D_1^{-1}$, we get the system $A'x' = b'$ where $A' = D_L A D_R$.

Using Chandler's scaling method, we get $D_L = \begin{bmatrix} 0.119 & 0 & 0 \\ 0 & 0.119 & 0 \\ 0 & 0 & 8.41 \end{bmatrix}$ and

$D_R = \begin{bmatrix} 8.41 & 0 & 0 \\ 0 & 0.119 & 0 \\ 0 & 0 & 0.119 \end{bmatrix}$. $A' = D_L A D_R = \begin{bmatrix} 1 & 0.0283 & 0.0425 \\ 1 & -0.0142 & 0.0142 \\ 0.0141 & 1 & 1 \end{bmatrix}$.

$b' = D_1^{-1} b = D_L b = \begin{bmatrix} 0.714 \\ 0.119 \\ 16.8 \end{bmatrix}$. After scaling, we get the order of pivoting 1, 3, 2. Eliminating

x_1, the equations become

$$\begin{aligned} x_1 + \quad 2x_2 + \quad 3x_3 &= 6 \\ x_2 + \quad 0.999x_3 &= 2 \\ -3x_2 - \quad 2x_3 &= -5. \end{aligned}$$

Eliminating x_2, the equations become

$$\begin{aligned} x_1 + \quad 2x_2 + \quad 3x_3 &= 6 \\ x_2 + \quad 0.999\,x_3 &= 2 \\ 0.997x_3 &= 1. \end{aligned}$$

Solving, $x_1 = 1.00$, $x_2 = 1.00$, and $x_3 = 1.00$. The result is the same as the exact solution.

Note that we use this scaled system to select the pivots, but we eliminate in the original system, not in this scaled system (Forsythe and Moler, 1967, p39).

Using Hamming's scaling method, we get $D_{LH}=\begin{bmatrix}0.379 & 0 & 0\\ 0 & 0.688 & 0\\ 0 & 0 & 11.8\end{bmatrix}$ and

$D_{RH}=\begin{bmatrix}11.8 & 0 & 0\\ 0 & 0.546 & 0\\ 0 & 0 & 0.477\end{bmatrix}$. $A'=D_{LH}AD_{RH}=\begin{bmatrix}4.47 & 0.414 & 0.542\\ 8.12 & -0.376 & 0.328\\ 0.0278 & 6.44 & 5.63\end{bmatrix}$.

$b'=D_{LH}b=\begin{bmatrix}2.27\\ 0.688\\ 23.6\end{bmatrix}$. After scaling, we get the order of pivoting was 2, 3, 1. Eliminating x_1, the equations become

$$\begin{aligned} x_1 - \quad & x_2 + \quad x_3 = 1\\ & x_2 + \quad x_3 = 2\\ & 3x_2 + \quad 2x_3 = 5. \end{aligned}$$

Eliminating x_2, the equations become

$$\begin{aligned} x_1 - \quad & x_2 + \quad x_3 = 1\\ & x_2 + \quad x_3 = 2\\ & \quad -x_3 = -1. \end{aligned}$$

Solving, $x_1 = 1.00$, $x_2 = 1.00$, and $x_3 = 1.00$. The result is also the same as the accurate answer. Both results using Chandler's and Hamming's method are totally the same. Check the condition numbers k_∞, k_{pp} of the original matrix and the other scaled matrices. For k_∞, we get k_{orig}=2.22638E+27, k_{chan}=291.148, and k_{ham}=110.111. For k_{pp}, we have k_{ppor}=7.89123E+22, k_{ppchan}=70.7531, and k_{ppham}=44.8679. The condition number of Hamming's scaled matrix is better than Chandler's but not by a great difference. We think both methods are good methods for solving linear system equations and if there is no big difference of the condition numbers, it is hard to say which one is better because the condition number is an upper bound on the true condition. However, Hamming's method has the advantage that it does not require iteration. For larger systems, as noted previously, Hamming's method may actually degrade the condition number and perhaps should be avoided for that reason.

CHAPTER IV

CONCLUSIONS

This study provide a good method for matrix scaling. This method of Chandler's is from concepts of scaling a symmetric positive definite matrix, is an iteration which has never failed to converge, and is derived from the basic scaling definition which is to make the largest or smallest element in any row and any column to be equal to 1. The results of numerical tests from 36100 different exponentially random matrices show that Chandler's method is very competitive because of the following four reasons:

(1) The condition number either k_∞ or k_{pp} descends drastically from the original matrix to the scaled matrix by Chandler's method. This means that the upper bound on the true condition decreases.

(2) The ratio of the largest and the smallest magnitude in matrix is decreased.

(3) Even in large matrices of dimension >=15, the condition number is also decreased, however, the condition number of Hamming's method is not.

(4) This method could reduce the problem of cancellation in using Gaussian elimination in solving linear equation system.

Future work might be done in the following aspects:

(1) Measure the cancellation that actually occurs in Gaussian elimination. This is the truest measure of condition.

(2) Investigate the effects of scaling on the solution of linear least squares problems and on eigenproblems.

(3) Prove that the smallest elements in the scaled matrix are as large as possible.

(4) Prove results about convergence and independence of the results with respect to MRC=0 or 1.

A SELECTED BIBLIOGRAPHY

1. Bauer, F.L., Optimally Scaled Matrices, Numerische Mathematik, Vol 5, pp. 73-87,1963.
2. Businger, P.A., Matrices Which Can Be Optimally scaled, Numerische Mathematik, Vol 12, pp. 346-48, 1968.
3. Chandler, J. P., Private Communication, Oklahoma State University, 1993.
4. Cheney, W. and Kincaid, D., Numerical Mathematics and Computing, Brooks/Cole Pub. Co., 1980.
5. Curtis, A. R. and Reid, J.K., On the Automatic Scaling of Matrices for Gaussian Elimination, Journal Inst. Mathematics Applics, Vol. 10, pp. 118-124, 1972
6. Golub, G. H. Matrix Computations, Addison-Wesley Pub. Co., 1977.
7. Fenner, T. and Loizou, G., Some New Bounds on The Condition Numbers of Optimally Scaled Matrices, Journal of ACM, Vol. 21, pp. 514-524, 1974.
8. Forsythe, G. and Moler, C.B., Computer Solution of Linear Algebraic Systems, Prentice-Hall, Englewood Cliffs, NJ, 1967.
9. Golub, G. H. and Varah, J. M., On a Characterization of the Best *l*2-scaling of a Matrix, SIAM Journal on Numerical Analysis.,Vol. 11, pp.472-79, 1974.
10. Hamming, R.W., Introduction to Applied Numerical Analysis, McGraw-hill Book Company, 1971.
11. Kahan,W., Numerical Linear Algebra, Canadian Mathematics Bulletin, Vol. 9, pp.757-801, 1966.
12. Kincaid, D. and Cheney, W., Numerical Analysis, Brooks/Cole Pub. Co., 1991.

13. McCarthy, C. and Strang, G., Optimal Conditioning of Matrices, SIAM Journal Numerical Anal. Vol.10,pp. 370-88, 1973.

14. Noble, B., Applied Linear Algebra, Prentice-Hall, Englewood Cliffs, NJ, 1969.

15. Skeel, R., Scaling for Numerical Stability in Gaussian Elimination, Journal of ACM, Vol. 26, pp.494-526, 1979.

16. Stoer ,J. and Witzgall, C., Transformations by Diagonal Matrices in a Normed Space, Numerische Mathematik, Vol.4, pp.158-171,1902.

17. Van der Sluis, A., Condition Numbers and Equilibration Matrices, Numerische Mathematik, Vol.14, pp.14-23, 1969.

18. Van der Sluis, A., Stability of Solutions of Linear Algebraic Systems, Numerische Mathematik, Vol 14, pp.246-251, 1970.

19. Wilkinson, J.H., Rounding Errors in Algebraic Processes, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1963.

20. Wilkinson, J.H., The Algebraic Eigenvalue Problem, Oxford, Clarendon Press, 1965.

21. Wilkinson, J.H., Error Analysis of Direct Methods of Matrix Inversion, Association for Computing Machinery, Vol. 8, pp.281-330, March, 1961.

APPENDIX A

PROGRAM LISTING

```
C
C SCALTEST 1.1 -- TEST SUBROUTINE SCALA         11/87
C
C J. P. CHANDLER, COMPUTER SCIENCE DEPARTMENT,
C   OKLAHOMA STATE UNIVERSITY
C
      INTEGER IN,ITER,J,K,KNPREV,KNVRG,KOUNT,LADIM,LP,M,
     *  MAXIT,MRC,N,NL,NONES,NPOWRT,NPR,NR,Q,TOTEXACT,
     *  DLNUMEXA,DRNUMEXA,FMDL,FMDR,DLMAT,DRMAT
C
      DOUBLE PRECISION A,DL,DR,SCRAT,B,C,DSEED,DRANDM,X,BASE,
     *  UNITR,DLH,DRH,ROW,DLSAV,DRSAV,DLOLD,DROLD,NUMMAT,
     *  RZERO,DENOM,TOL,HUGE,TEMP,AJKMIN,SUMLOG,SEEDBASE,
     *  DABS,DLOG,DMAX1,NEW,TOTITER,SEEDSAV,CONDPP,KPPOR,
     *  KPPCHAN,DLOG10,KPPHAM,TOTKPPOR,TOTKPPCHAN,TOTKPPHAM,
     *  KORIG,KCHAN,KHAM,TOTKORIG,TOTKCHAN,TOTKHAM,CONDNO,
     *
LOGKCHAN,LOGKHAM,LOGKPPCHAN,LOGKPPHAM,PERDLEXA,PERDREXA
C
      DIMENSION A(20,20),NEW(20,20),DL(20),DR(20),SCRAT(20),
     *  NL(20),NR(20),DLH(20),DRH(20),ROW(20),DLSAV(20),
     *  DRSAV(20),DLOLD(20),DROLD(20)
C
      IN=5
      LP=6
C     NPR=12
C     M=3
C     N=3
      LADIM=20
      B=0.0D0
      C=30.0D0
      BASE=10.0D0
      DSEED=27469.0D0
      RZERO=0.0D0
      UNITR=1.0D0
```

```
      TOL=1.0D-13
      HUGE=1.0D70
      SEEDBASE=1.0D5
C
C  GENERATE AN EXPONENTIALLY RANDOM MATRIX.
C
      DO 390 M=2,20
       DO 380 N=2,20
        WRITE(LP,30)
   30   FORMAT('======================================================
     *==============================')
        TOTITER=0.0
        TOTEXACT=0
        TOTKORIG=0.0
        TOTKCHAN=0.0
        TOTKHAM=0.0
        TOTKPPOR=0.0
        TOTKPPCHAN=0.0
        TOTKPPHAM=0.0
        PERDLEXA=0.0
        PERDREXA=0.0
        DLMAT=0
        DRMAT=0
        NUMMAT=0.0
      DO 350 Q=1,100
        DSEED=SEEDBASE*DRANDM(DSEED)
        SEEDSAV=DSEED
      DO 20 J=1,M
        DO 10 K=1,N
          X=B+(C-B)*DRANDM(DSEED)
          A(J,K)=BASE**X
          NEW(J,K)=A(J,K)
   10     CONTINUE
   20   CONTINUE
C
C     WRITE(LP,30)M,N
C  30 FORMAT(//' M =',I3,5X,'N =',I3//
C    *   ' THE MATRIX TO BE SCALED IS ....'/' ')
C
C     DO 50 J=1,M
C       WRITE(LP,40)J,(A(J,K),K=1,N)
C  40   FORMAT(' ROW',I3/(5X,1PE14.6,4E14.6))
C  50   CONTINUE
      KPPOR=CONDPP(NEW,M,N)
      KORIG=CONDNO(NEW,M,N)
```

```
      TOTKORIG=TOTKORIG+KORIG
      TOTKPPOR=TOTKPPOR+KPPOR
C
C     MRC=2
      MRC=1
C
C  INITIALIZE DA(*) AND DB(*).
C
      DO 60 J=1,M
        DLOLD(J)=UNITR
        DLSAV(J)=UNITR
        DL(J)=UNITR
   60   CONTINUE
C
      DO 70 K=1,N
        DROLD(J)=UNITR
        DRSAV(K)=UNITR
        DR(K)=UNITR
   70   CONTINUE
C
C  ITERATE THE SCALING.
C
      MAXIT=M+N+10
C     NPOWRT=2
C     KOUNT=0
      KNVRG=0
      FMDL=0
      FMDR=0
C
      DO 300 ITER=1,MAXIT
        CALL SCALA (A,M,N,LADIM,DL,DR,1,MRC,SCRAT)
        CALL SCALA (A,M,N,LADIM,DL,DR,0,MRC,SCRAT)
        DO 275 J=1,M
          DO 275 K=1,N
            NEW(J,K)=A(J,K)*DL(J)*DR(K)
  275   CONTINUE
C     CALL CONDNO(NEW,M,N)
C
C  PRINT DETAILED RESULTS AT EACH ITERATION NUMBER THAT
C  IS EQUAL TO A POWER OF TWO.
C
C       KOUNT=KOUNT+1
C
C       IF(KOUNT.GE.NPOWRT) THEN
C         NPOWRT=NPOWRT*2
```

```
C        KOUNT=0
C      ENDIF
C
C      KNPREV=KNVRG
C
      DLNUMEXA=0
      DO 90 J=1,M
        TEMP=DABS(DL(J)-DLSAV(J))/
     *        DMAX1(DABS(DL(J)),DABS(DLSAV(J)))
        IF(TEMP.LE.TOL) THEN
         KNVRG=1
         FMDL=1
         DLNUMEXA=DLNUMEXA+1
        ENDIF
 90     CONTINUE
C
      DRNUMEXA=0
      DO 100 K=1,N
        TEMP=DABS(DR(K)-DRSAV(K))/
     *        DMAX1(DABS(DR(K)),DABS(DRSAV(K)))
        IF(TEMP.LE.TOL) THEN
         KNVRG=1
         FMDR=1
         DRNUMEXA=DRNUMEXA+1
        ENDIF
 100    CONTINUE
C
C  PRINT THE RESULTS OF THIS ITERATION.
C
      IF((KNVRG.NE.1).AND.(ITER.LT.MAXIT)) GO TO 295
      KCHAN=CONDNO(NEW,M,N)
      KPPCHAN=CONDPP(NEW,M,N)
      TOTKCHAN=TOTKCHAN+KCHAN
      TOTKPPCHAN=TOTKPPCHAN+KPPCHAN
      IF(KNVRG.NE.1) GO TO 295
      PERDLEXA=PERDLEXA+DLNUMEXA*1.0
      PERDREXA=PERDREXA+DRNUMEXA*1.0
      IF(FMDL.EQ.1) DLMAT=DLMAT+1
      IF(FMDR.EQ.1) DRMAT=DRMAT+1
      NUMMAT=NUMMAT+1
      TOTITER=TOTITER+ITER
      TOTEXACT=TOTEXACT+1
      GO TO 305
C      WRITE(LP,110)MRC,ITER,KOUNT,KNVRG
C 110    FORMAT(/' MRC =',I2,5X,'ITER =',I3,5X,'KOUNT =',I3,
```

```
C   *     5X,'KNVRG =',I2)
C
C      AJKMIN=HUGE
C      SUMLOG=RZERO
C      DO 130 J=1,M
C        DO 120 K=1,N
C          TEMP=DABS(DL(J)*A(J,K)*DR(K))
C          IF(TEMP.LE.RZERO) GO TO 120
C          IF(TEMP.LT.AJKMIN) AJKMIN=TEMP
C          SUMLOG=SUMLOG+DLOG(TEMP)
C 120        CONTINUE
C 130      CONTINUE
C
C      WRITE(LP,140)AJKMIN,SUMLOG
C 140   FORMAT(' AJKMIN =',1PE15.7,9X,'SUMLOG =',E15.7)
C      IF(KNPREV.NE.0 .AND. (KOUNT.NE.0 .OR. ITER.LE.NPR))
C   *     GO TO 260
C
C  PRINT DETAILED RESULTS
C    UNTIL EXACT CONVERGENCE OF SOME SCALE FACTOR OCCURS,
C    AT EVERY ITERATION NUMBER EQUAL TO A POWER OF TWO, AND
C    FOR THE FIRST NPR ITERATIONS.
C
C      WRITE(LP,150)(DL(J),J=1,M)
C 150   FORMAT(/' DL =',1PE14.6,4E14.6/(5X,4E14.6))
C      WRITE(LP,160)(DR(K),K=1,N)
C 160   FORMAT(/' DR =',1PE14.6,4E14.6/(5X,4E14.6))
C
C      DO 170 J=1,M
C        ROW(J)=(DL(J)-DLSAV(J))/
C   *       DMAX1(DABS(DL(J)),DABS(DLSAV(J)))
C 170      CONTINUE
C      WRITE(LP,180)(ROW(J),J=1,M)
C 180   FORMAT(/' RELDIF =',1PE14.3,4E14.3/(9X,4E14.3))
C
C      DO 190 K=1,N
C        ROW(K)=(DR(K)-DRSAV(K))/
C   *       DMAX1(DABS(DR(K)),DABS(DRSAV(K)))
C 190      CONTINUE
C      WRITE(LP,180)(ROW(K),K=1,N)
C      IF(ITER.LT.3) GO TO 260
C
C      DO 200 J=1,M
C        DENOM=DLSAV(J)-DLOLD(J)
C        ROW(J)=RZERO
```

```
C        IF(DENOM.NE.RZERO) ROW(J)=(DL(J)-DLSAV(J))/DENOM
C 200      CONTINUE
C      WRITE(LP,210)(ROW(J),J=1,M)
C 210   FORMAT(/' RATIO =',5F14.7/(8X,5F14.7))
C
C      DO 220 K=1,N
C        DENOM=DRSAV(K)-DROLD(K)
C        ROW(K)=RZERO
C        IF(DENOM.NE.RZERO) ROW(K)=(DR(K)-DRSAV(K))/DENOM
C 220      CONTINUE
C      WRITE(LP,210)(ROW(K),K=1,N)
C
C 260   IF((KNVRG.NE.1 .OR. KNPREV.NE.0) .AND.
C   *     (KOUNT.NE.0 .OR. ITER.LE.NPR)) GO TO 295
C
C  PRINT THE SCALED MATRIX.
C
C      DO 290 J=1,M
C        NONES=0
C        DO 270 K=1,N
C          ROW(K)=DL(J)*A(J,K)*DR(K)
C          IF(DABS(DABS(ROW(K))-UNITR).LE.TOL)
C   *         NONES=NONES+1
C 270        CONTINUE
C        WRITE(LP,280)J,NONES,(ROW(K),K=1,N)
C 280      FORMAT(' ROW',I3,5X,I3,' ONE(S)'/
C   *       (5X,1PE14.7,4E14.7))
C 290      CONTINUE
C
 295    DO 240 J=1,M
        DLOLD(J)=DLSAV(J)
        DLSAV(J)=DL(J)
 240      CONTINUE
C
      DO 250 K=1,N
        DROLD(K)=DRSAV(K)
        DRSAV(K)=DR(K)
 250      CONTINUE
C
 300   CONTINUE
C
C  SCALE USING HAMMING'S METHOD, TO COMPARE.
C
 305 CALL SCALH (A,M,N,LADIM,NL,NR,DLH,DRH)
C    WRITE(LP,310)(DLH(J),J=1,M)
```

```
C 310 FORMAT(/' HAMMING METHOD...'/' DL =',1PE14.6,4E14.6/
C    *  (5X,4E14.6))
C     WRITE(LP,160)(DRH(K),K=1,N)
C
      DO 330 J=1,M
        DO 320 K=1,N
C           ROW(K)=DLH(J)*A(J,K)*DRH(K)
          NEW(J,K)=DLH(J)*A(J,K)*DRH(K)
 320      CONTINUE
C       WRITE(LP,40)J,(ROW(K),K=1,N)
 330    CONTINUE
      KPPHAM=CONDPP(NEW,M,N)
      KHAM=CONDNO(NEW,M,N)
      TOTKPPHAM=TOTKPPHAM+KPPHAM
      TOTKHAM=TOTKHAM+KHAM
 350 CONTINUE
      TOTITER=TOTITER/NUMMAT
      PERDLEXA=PERDLEXA/(M*DLMAT*1.0)*100.0
      PERDREXA=PERDREXA/(N*DRMAT*1.0)*100.0
      IF(M.EQ.N)THEN
       LOGKCHAN=DLOG10(TOTKCHAN/TOTKORIG)
       LOGKHAM=DLOG10(TOTKHAM/TOTKORIG)
       LOGKPPCHAN=DLOG10(TOTKPPCHAN/TOTKPPOR)
       LOGKPPHAM=DLOG10(TOTKPPHAM/TOTKPPOR)

WRITE(6,360)M,N,TOTITER,TOTEXACT,TOTKORIG,TOTKCHAN,TOTKHAM,
     *
TOTKPPOR,TOTKPPCHAN,TOTKPPHAM,LOGKCHAN,LOGKHAM,LOGKPPCHA
N,
     * LOGKPPHAM,PERDLEXA,PERDREXA
 360   FORMAT('M=',I3,3X,'N='I3,3X,'AVG EXACT ITER=',F6.3,3X,
     * 'EXACT No.=',I5,/'TOTAL OF: KINFORIG=',1PE10.3,2X,
     * 'KINFCHAN=',E10.3,2X,'KINFHAM=',E10.3/10X,'KPPORIG =',E10.3,2X,
     * 'KPPCHAN =',E10.3,2X,'KPPHAM =',E10.3/'LOG TOTAL: ',
     * 'KINFCHAN/KINFORIG=',E10.3,2X,'KINFHAM/KINFORIG=',E10.3/
     * 11X,'KPPCHAN/KPPORIG  =',E10.3,2X,'KPPHAM/KPPORIG  =',E10.3/
     * 'PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY:',1X,
     * 'DL=',1PE10.3,2X,'DR=',E10.3)
      ELSE
       WRITE(6,370)M,N,TOTITER,TOTEXACT,PERDLEXA,PERDREXA
 370   FORMAT('M=',I3,3X,'N='I3,3X,'AVG EXACT ITER=',F6.3,3X,
     * 'EXACT No.=',I5/
     * 'PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY:',1X,
     * 'DL=',1PE10.3,2X,'DR=',E10.3)
      ENDIF
```

```
 380 CONTINUE
 390 CONTINUE
C
    STOP
C
C  END SCALTEST
C
    END
C
    SUBROUTINE SCALA (A,M,N,LADIM,DL,DR,INV,MRC,SCRAT)
C
C  SCALA 1.1      APRIL 1992
C
C  SCALES THE MATRIX A.
C
C  J. P. CHANDLER, COMPUTER SCIENCE DEPARTMENT,
C     OKLAHOMA STATE UNIVERSITY
C
C*******************************
C
C  INPUT QUANTITIES..... A(*,*),M,N,LADIM,DL(*),DR(*),INV,MRC
C  OUTPUT QUANTITIES.... DL(*),DR(*)
C  SCRATCH ARRAY........ SCRAT(*)
C
C    A(*,*)  -- THE MATRIX TO BE SCALED
C    M       -- NUMBER OF ROWS IN THE MATRIX A
C    N       -- NUMBER OF COLUMNS IN THE MATRIX A
C    LADIM   -- THE FIRST DIMENSION OF THE ARRAY A
C               (M.LE.LADIM)
C    DL(*)   -- LEFT DIAGONAL SCALING MATRIX
C    DR(*)   -- RIGHT DIAGONAL SCALING MATRIX
C    INV     -- =1 TO SCALE UP SMALL ELEMENTS OF A,
C              =0 TO SCALE DOWN LARGE ELEMENTS OF A
C                (THE FINAL CALL SHOULD BE MADE WITH
C                 INV=0)
C    MRC     -- =1 TO SCALE ROWS FIRST,
C              =2 TO SCALE COLUMNS FIRST
C                (EITHER VALUE SHOULD WORK.  USE MRC=2 .)
C    SCRAT(*) -- SCRATCH ARRAY OF AT LEAST M LOCATIONS
C
C  METHOD...
C    TO SCALE DOWN LARGE ELEMENTS (INV=0), SUBROUTINE SCALA
C    CARRIES OUT THE FOLLOWING STEPS...
C      1. DIVIDE EACH COLUMN (OR ROW, IF MRC=1) BY THE
C         SQUARE ROOT OF THE LARGEST MAGNITUDE OF ANY
```

```
C         ELEMENT IN THAT COLUMN (ROW).
C      2. DIVIDE EACH ROW (COLUMN) BY THE LARGEST MAGNITUDE
C         OF ANY ELEMENT IN THAT ROW (COLUMN).
C      3. DIVIDE EACH COLUMN (ROW) BY THE LARGEST MAGNITUDE
C         OF ANY ELEMENT IN THAT COLUMN (ROW).
C
C   TO SCALE UP SMALL ELEMENTS (INV=1), SCALA INVERTS EACH
C   NONZERO ELEMENT (IMPLICITLY), THEN SCALES DOWN LARGE
C   ELEMENTS AS DESCRIBED ABOVE, AND THEN INVERTS EACH
C   ELEMENT BACK AGAIN.
C
C USAGE...
C   CALL SCALA WITH INV=1 AND THEN WITH INV=0.
C   REPEAT THIS PAIR OF CALLS UNTIL THE DL(*) AND THE DR(*)
C   ALL CONVERGE (COMPARE THEIR VALUES ONLY AFTER THE CALLS
C   WITH INV=0).
C
C   FOR A SYMMETRIC POSITIVE DEFINITE MATRIX, CALL SCALA
C   ONCE ONLY, WITH INV=0.
C********************************
C
      INTEGER INV,J,JJ,K,LADIM,M,MRC,N
      DOUBLE PRECISION A,DL,DR,SCRAT,ZSQRT,ARG,ZABS,
     *  RZERO,UNITR,COLMAX,DRK,TEMP,DABS,DSQRT
      DIMENSION A(LADIM,N),DL(M),DR(N),SCRAT(M)
C
      ZSQRT(ARG)=DSQRT(ARG)
      ZABS(ARG)=DABS(ARG)
C
      RZERO=0.0D0
      UNITR=1.0D0
C
      IF(M.LT.1 .OR. M.GT.LADIM .OR. N.LT.1) STOP
C
C LOOP OVER THE THREE (SIC) PASSES.
C
      DO 80 JJ=1,2
        IF(MRC.EQ.1 .AND. JJ.EQ.1) GO TO 30
C
C SCALE THE COLUMNS.
C
        DO 20 K=1,N
          COLMAX=RZERO
          DRK=DR(K)
C
```

```
        DO 10 J=1,M
          TEMP=ZABS(DL(J)*A(J,K)*DRK)
          IF(TEMP.EQ.RZERO) GO TO 10
          IF(INV.EQ.1) TEMP=UNITR/TEMP
          IF(TEMP.GT.COLMAX) COLMAX=TEMP
 10       CONTINUE
C
        IF(COLMAX.EQ.RZERO) GO TO 20
        IF(INV.EQ.1) COLMAX=UNITR/COLMAX
        IF(MRC.EQ.2 .AND. JJ.EQ.1) COLMAX=ZSQRT(COLMAX)
        DR(K)=DR(K)/COLMAX
 20     CONTINUE
      IF(MRC.EQ.2 .AND. JJ.EQ.2) RETURN
 30   DO 40 J=1,M
        SCRAT(J)=RZERO
 40     CONTINUE
      DO 60 K=1,N
        DRK=DR(K)
        DO 50 J=1,M
          TEMP=ZABS(DL(J)*A(J,K)*DRK)
          IF(TEMP.EQ.RZERO) GO TO 50
          IF(INV.EQ.1) TEMP=UNITR/TEMP
          IF(TEMP.GT.SCRAT(J)) SCRAT(J)=TEMP
 50       CONTINUE
 60     CONTINUE
      DO 70 J=1,M
        TEMP=SCRAT(J)
        IF(TEMP.EQ.RZERO) GO TO 70
        IF(INV.EQ.1) TEMP=UNITR/TEMP
        IF(MRC.EQ.1 .AND. JJ.EQ.1) TEMP=ZSQRT(TEMP)
        DL(J)=DL(J)/TEMP
 70     CONTINUE
C
 80   CONTINUE
C
    RETURN
C
C  END SCALA
C
    END
C
    DOUBLE PRECISION FUNCTION DRANDM(DL)
C
C  SIMPLE PORTABLE PSEUDORANDOM NUMBER GENERATOR.
C
```

```
C DRANDM RETURNS FUNCTION VALUES THAT ARE PSEUDORANDOM
C NUMBERS UNIFORMLY DISTRIBUTED ON THE INTERVAL (0,1).
C
C 'NUMERICAL MATHEMATICS AND COMPUTING' BY WARD CHENEY AND
C DAVID KINCAID, BROOKS/COLE PUBLISHING COMPANY
C (FIRST EDITION, 1980), PAGE 203
C
C AT THE BEGINNING OF EXECUTION, OR WHENEVER A NEW SEQUENCE IS
C TO BE INITIATED, SET DL EQUAL TO AN INTEGER VALUE BETWEEN
C 1.0D0 AND 2147483647.0D0, INCLUSIVE.  DO THIS ONLY ONCE.
C THEREAFTER, DO NOT SET OR ALTER DL IN ANY WAY.
C FUNCTION DRANDM WILL MODIFY DL FOR ITS OWN PURPOSES.
C
C DRANDM USES A MULTIPLICATIVE CONGRUENTIAL METHOD.
C THE NUMBERS GENERATED BY DRANDM SUFFER FROM THE PARALLEL
C PLANES DEFECT DISCOVERED BY G. MARSAGLIA, AND SHOULD NOT BE
C USED WHEN HIGH-QUALITY RANDOMNESS IS REQUIRED.  IN THAT
C CASE, USE A "SHUFFLING" METHOD.
C
      DOUBLE PRECISION DL
C
   10 DL=DMOD(16807.0D0*DL,2147483647.0D0)
      DRANDM=DL/2147483648.0D0
      IF(DRANDM.LE.0.0D0 .OR. DRANDM.GE.1.0D0) GO TO 10
      RETURN
      END
C
      SUBROUTINE SCALH (A,M,N,LADIM,NL,NR,DL,DR)
C
C SCALH 1.3      APRIL 1992
C
C SCALES THE MATRIX A(*,*) USING HAMMING'S METHOD.
C RICHARD W. HAMMING, "INTRODUCTION TO APPLIED NUMERICAL
C   ANALYSIS", (MCGRAW-HILL 1971), PAGES 115-117
C
C J. P. CHANDLER, COMPUTER SCIENCE DEPARTMENT,
C   OKLAHOMA STATE UNIVERSITY
C
C*********************************
C
C INPUT QUANTITIES..... A(*,*),M,N,LADIM
C OUTPUT QUANTITIES.... DL(*),DR(*)
C SCRATCH STORAGE...... NL(*),NR(*)
C
C   A(*,*) -- THE MATRIX TO BE SCALED
```

```
C    M     -- NUMBER OF ROWS IN THE MATRIX A
C    N     -- NUMBER OF COLUMNS IN THE MATRIX A
C    LADIM -- THE FIRST DIMENSION OF THE ARRAY A
C             (LADIM.GE.M)
C    DL(*) -- LEFT DIAGONAL SCALING MATRIX
C    DR(*) -- RIGHT DIAGONAL SCALING MATRIX
C*******************************
C
      INTEGER M,N,LADIM,NL(M),NR(N),J,K,JKSUM,KSUM
      DOUBLE PRECISION A(LADIM,N),DL(M),DR(N),
     *  ARG,ZABS,ZLOG,ZEXP,  RZERO,SUM,SUMK,TEMP,HALFAV,
     *  DABS,DLOG,DEXP
C
      ZABS(ARG)=DABS(ARG)
      ZLOG(ARG)=DLOG(ARG)
      ZEXP(ARG)=DEXP(ARG)
C
      RZERO=0.0D0
C
      IF(M.LT.1 .OR. M.GT.LADIM .OR. N.LT.1) STOP
C
C  INITIALIZE.
C
      DO 10 J=1,M
        DL(J)=RZERO
        NL(J)=0
   10   CONTINUE
C
      DO 20 K=1,N
        DR(K)=RZERO
        NR(K)=0
   20   CONTINUE
C
      SUM=RZERO
      JKSUM=0
C
C  ACCUMULATE ALL SUMS.  PROCESS A(*,*) BY COLUMNS.
C
      DO 40 K=1,N
        SUMK=RZERO
        KSUM=0
C
        DO 30 J=1,M
          TEMP=ZABS(A(J,K))
          IF(TEMP.EQ.RZERO) GO TO 30
```

```
        TEMP=ZLOG(TEMP)
        DL(J)=DL(J)+TEMP
        SUMK=SUMK+TEMP
        SUM=SUM+TEMP
        NL(J)=NL(J)+1
        KSUM=KSUM+1
        JKSUM=JKSUM+1
  30     CONTINUE
C
      DR(K)=SUMK
      NR(K)=KSUM
  40   CONTINUE
C
C  COMPUTE CL(*) AND DR(*).
C
     IF(JKSUM.EQ.0) GO TO 70
     TEMP=JKSUM+JKSUM
     HALFAV=SUM/TEMP
C
     DO 50 J=1,M
       IF(NL(J).NE.0) DL(J)=HALFAV-DL(J)/NL(J)
  50   CONTINUE
C
     DO 60 K=1,N
       IF(NR(K).NE.0) DR(K)=HALFAV-DR(K)/NR(K)
  60   CONTINUE
C
C  TAKE ANTILOGS.
C
  70 DO 80 J=1,M
       DL(J)=ZEXP(DL(J))
  80   CONTINUE
C
     DO 90 K=1,N
       DR(K)=ZEXP(DR(K))
  90   CONTINUE
C
     RETURN
C
C  END SCALH
C
     END

     DOUBLE PRECISION FUNCTION CONDNO(NEW,M,N)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
C  CONDNO 1.0      JULY  1993
C
C  CONDITION NUMBER OF INFINITY AND ONE BY CHUNG-CHUAN
C  WANG,COMPUTER SCIENCE DEPARTMENT,OKLAHOMA STATE
UNIVERSITY
C********************************
C
C  INPUT QUANTITIES..... NEW,M,N
C  OUTPUT QUANTITIES.... CONDINF,CONDONE
C
C    NEW(*,*) -- THE MATRIX TO BE COUNTED CONDITION No.
C    M -- NUMBER OF ROWS IN THE MATRIX A
C    N -- NUMBER OF COLUMNS IN THE MATRIX A
C    CONDINF  -- CONDITION NUMBER OF INFINITY
C    CONDONE  -- CONDITION NUMBER OF ONE
C********************************
      INTEGER N,J,K,M,RROW,COL,XSAV,YSAV,XCOR,YCOR
      DOUBLE PRECISION NEW,CONDINF,CONDONE,UL,R,DX
      DOUBLE PRECISION INV,TEMP,SCALES,IPS,B,X
      DOUBLE PRECISION MAXROW,MAXCOL,IMAXROW,IMAXCOL
      DIMENSION INV(20,30),NEW(20,25),UL(20,40),
     *  R(20),DX(20),SCALES(20),IPS(20),B(20),X(20)

      IF(M.EQ.N) GO TO 400
      GO TO 530
C
C SELECT THE LARGEST SUM OF ABSOLUTE VALUSE OF ELEMENTS OF
ROW
C
 400 MAXROW=0.0
      DO 420 J=1,M
        TEMP=0.0
        DO 410 K=1,N
          TEMP=TEMP+DABS(NEW(J,K))
 410      CONTINUE
        IF(MAXROW.LT.TEMP) MAXROW=TEMP
 420  CONTINUE
C
C SELECT THE LARGEST SUM OF ABSOLUTE VALUSE OF ELEMENTS OF
COLUMN
C
      MAXCOL=0.0
      DO 440 K=1,N
        TEMP=0.0
        DO 430 J=1,M
```

```
        TEMP=TEMP+DABS(NEW(J,K))
 430    CONTINUE
      IF(MAXCOL.LT.TEMP) MAXCOL=TEMP
 440  CONTINUE
C
C GET THE INVERT MATRIX NEW
C
    CALL INVERT(M,NEW,20,INV,UL,20,B,X,SCALES,IPS,R,DX)
C
C GET THE LARGEST SUM OF ABSOLUTE ELEMENTS OF ROWS OF INVERT
MATRIX
C
    IMAXROW=0.0
    DO 490 J=1,M
      TEMP=0.0
      DO 480 K=1,N
        TEMP=TEMP+DABS(INV(J,K))
 480    CONTINUE
      IF(IMAXROW.LT.TEMP) IMAXROW=TEMP
 490  CONTINUE
C
C GET THE LARGEST SUM OF ABSOLUTE ELEMENTS OF COLUMNS OF
INVERT MATRIX
C
    IMAXCOL=0.0
    DO 510 K=1,N
      TEMP=0.0
      DO 500 J=1,M
        TEMP=TEMP+DABS(INV(J,K))
 500    CONTINUE
        IF(IMAXCOL.LT.TEMP) IMAXCOL=TEMP
 510  CONTINUE
C
C GET THE CONDITION NUMBER OF INFINITY AND ONE
C
    CONDINF=IMAXROW*MAXROW
    CONDONE=IMAXCOL*MAXCOL
    CONDNO=CONDINF
    RETURN
 530 END
    DOUBLE PRECISION FUNCTION CONDPP(NEW,M,N)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C  CONDNO 1.0      JULY  1993
C
C  CONDITION NUMBER OF  GAUSSIAN ELIMINATION WITH
```

```
C  PARTIAL PIVOTING METHOD, CHUNG-CHUAN
C  WANG,COMPUTER SCIENCE DEPARTMENT,OKLAHOMA STATE
UNIVERSITY
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C  INPUT QUANTITIES..... NEW,M,N
C  OUTPUT QUANTITIES.... CONDPP
C
C    NEW(*,*) -- THE MATRIX TO BE COUNTED CONDITION No.
C    M -- NUMBER OF ROWS IN THE MATRIX A
C    N -- NUMBER OF COLUMNS IN THE MATRIX A
C            LARGEST ABSOLUTE VALUSE OF ELEMENT IN
C            SCALED MATRIX
C    CONDPP  =  -----------------------------------
C            SMALLEST ABSOLUTE VALUE OF PIVOT IN
C            GAUSSIAN ELIMINATION WITH PARTIAL
C            PIVOTING ON THE SCALED MATRIX
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      INTEGER J,K,M,N
      DOUBLE PRECISION NEW,CONDPP,MAXELE,MINPIVOT,MINPIV
      DIMENSION NEW(20,20)

      IF(M.NE.N) GO TO 710
C
C GET THE LARGEST ABSOLUTE VALUE OF ELEMENT IN SCALED MATRIX
C
      MAXELE=0.0
      DO 700 J=1,M
        DO 700 K=1,N
          IF(MAXELE.LT.DABS(NEW(J,K))) MAXELE=DABS(NEW(J,K))
  700 CONTINUE
C
C GET SMALLEST ABSOLUTE VALUE OF PIVOT IN GAUSSIAN ELEMINATION
C WITH PARTIAL PIVOTINGON THE SCALED MATRIX
C
      MINPIVOT=MINPIV(NEW,M,N)
      CONDPP=MAXELE/MINPIVOT
  710 END

      DOUBLE PRECISION FUNCTION MINPIV(NEW,M,N)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C  CONDNO 1.0      JULY  1993
C
C  MINIMUM PIVOT OF GAUSSIAN ELIMINATION WITH
C  PARTIAL PIVOTING METHOD, CHUNG-CHUAN WANG,
C  COMPUTER SCIENCE DEPARTMENT,OKLAHOMA STATE UNIVERSITY
```

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C  INPUT QUANTITIES..... NEW,M,N
C  OUTPUT QUANTITIES.... MINPIV
C
C    NEW(*,*) -- THE MATRIX TO BE COUNTED CONDITION No.
C    M -- NUMBER OF ROWS IN THE MATRIX A
C    N -- NUMBER OF COLUMNS IN THE MATRIX A
C    MINPIV - ABSOLUTE VALUE OF MINIMUM PIVOT OF
C          GAUSSIAN ELIMINATION WITH PARTIAL
C          PIVOTING ON THE SCALED MATRIX
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      INTEGER I,J,K,M,N,MAXROW
      DOUBLE PRECISION NEW,MINPIV,GAU,MAXPIV,TEMP
      DIMENSION NEW(20,20),GAU(20,20),TEMP(20)

      DO 800 J=1,M
        DO 800 K=1,N
          GAU(J,K)=NEW(J,K)
  800 CONTINUE
C
C SELECT THE LARGEST PIVOT IN NEXT COLUMN
C
      MINPIV=1.0D70
      DO 860 I=1,M-1
        MAXPIV=0.0
        DO 810 J=I,M
          IF(MAXPIV.LT.DABS(GAU(J,I))) THEN
            MAXPIV=DABS(GAU(J,I))
            MAXROW=J
          ENDIF
  810   CONTINUE
      IF(MINPIV.GT.MAXPIV) MINPIV=MAXPIV

C
C SWAP WITH THE Ith ROW, IF MAXROW NOT EQUAL TO I
C
      IF(MAXROW.EQ.I) GO TO 830
        DO 820 K=1,N
          TEMP(K)=GAU(I,K)
          GAU(I,K)=GAU(MAXROW,K)
          GAU(MAXROW,K)=TEMP(K)
  820   CONTINUE
C
C ELIMINATE I+1 TO M ROWS' THE Ith COLUMN AND COEFFICIENT CHANGE
C
```

```
 830   DO 850 J=I+1,M
       DO 840 K=I+1,N
         GAU(J,K)=GAU(J,K)-GAU(J,I)/GAU(I,I)*GAU(I,K)
 840     CONTINUE
       GAU(J,I)=0.0D0
 850   CONTINUE
 860 CONTINUE
     IF(MINPIV.GT.DABS(GAU(M,M))) MINPIV=DABS(GAU(M,M))
     END

C
C  FMNEW.CNTL     LINEAR SYSTEM SOFTWARE FROM FORSYTHE AND
MOLER
C
     SUBROUTINE DECOMP(N,NEW,LA,UL,LUL,SCALES,IPS,LP)
C
C  DECOMPOSES A INTO THE PRODUCT A=L*U, WHERE L IS A MONIC
LOWER
C  TRIANGULAR MATRIX AND U IS UPPER TRIANGULAR.  STORES L-I AND U
IN
C  THE ARRAY UL.
C  DECOMP PERFORMS ABOUT N**3/3 MULTIPLICATIONS.
C
C  G. E. FORSYTHE AND C. B. MOLER, -COMPUTER SOLUTION OF LINEAR
C      ALGEBRAIC SYSTEMS- (PRENTICE-HALL, 1967)
C
C  J. P. CHANDLER, COMPUTER SCIENCE DEPT., OKLAHOMA STATE
UNIVERSITY
C
C    IMPLICIT REAL*8 (A-H,O-Z)
C
     DOUBLE PRECISION NEW,UL,SCALES,IPS
     DIMENSION NEW(LA,N),UL(LUL,N),SCALES(N),IPS(N)
C
     ZERO=0.0D0
     UNITY=1.0D0
C
C  INITIALIZE IPS, UL, AND SCALES.
C
     DO 20 I=1,N
       IPS(I)=I
       ROWNRM=ZERO
       DO 10 J=1,N
         UL(I,J)=NEW(I,J)
C                  ROWNRM=AMAX1(ROWNRM,ABS(UL(I,J)))
```

```
        ABSUL=UL(I,J)
        IF(ABSUL.LT.ZERO) ABSUL=-ABSUL
        IF(ABSUL.GT.ROWNRM) ROWNRM=ABSUL
 10     CONTINUE
      IF(ROWNRM.GT.ZERO) THEN
        SCALES(I)=UNITY/ROWNRM
      ELSE
        SCALES(I)=ZERO
      ENDIF
      SCALES(I)=UNITY/ROWNRM
 20   CONTINUE
C
C  PERFORM GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING.
C
    NM1=N-1
    DO 60 K=1,NM1
      BIG=ZERO
      DO 30 I=K,N
        IP=IPS(I)
C                   SIZE=ABS(UL(IP,K))*SCALES(IP)
        ABSUL=UL(IP,K)
        IF(ABSUL.LT.ZERO) ABSUL=-ABSUL
        SIZE=ABSUL*SCALES(IP)
        IF(SIZE.GT.BIG) THEN
          BIG=SIZE
          IDXPIV=I
        ENDIF
 30     CONTINUE
        IF(IDXPIV.NE.K) THEN
          J=IPS(K)
          IPS(K)=IPS(IDXPIV)
          IPS(IDXPIV)=J
        ENDIF
        KP=IPS(K)
        PIVOT=UL(KP,K)
        KP1=K+1
        DO 50 I=KP1,N
          IP=IPS(I)
          EM=-UL(IP,K)/PIVOT
          UL(IP,K)=-EM
          IF(EM.NE.ZERO) THEN
C
            DO 40 J=KP1,N
 40           UL(IP,J)=UL(IP,J)+EM*UL(KP,J)
C
```

```
C  INNER LOOP.  USE MACHINE LANGUAGE CODING IF COMPILER DOES
NOT
C  PRODUCE EFFICIENT CODE.
C
          ENDIF
  50        CONTINUE
  60    CONTINUE
    KP=IPS(N)
    RETURN
    END
    SUBROUTINE SOLVE(N,UL,LUL,B,X,IPS)
C
C  SOLVES A*X=B USING UL FROM DECOMP.
C  SOLVE PERFORMS ONLY ABOUT N**2 MULTIPLICATIONS.
C
C  G. E. FORSYTHE AND C. B. MOLER, -COMPUTER SOLUTION OF LINEAR
C      ALGEBRAIC SYSTEMS- (PRENTICE-HALL, 1967)
C
C    IMPLICIT REAL*8 (A-H,O-Z)
C
    DOUBLE PRECISION UL,B,X,IPS
    DIMENSION UL(LUL,N),B(N),X(N),IPS(N)
C
    NP1=N+1
C                   PERFORM FORWARD SUBSTITUTION.
    IP=IPS(1)
    X(1)=B(IP)
    DO 20 I=2,N
      IP=IPS(I)
      IM1=I-1
      SUM=B(IP)
      DO 10 J=1,IM1
  10      SUM=SUM-UL(IP,J)*X(J)
  20    X(I)=SUM
C                   PERFORM BACK SUBSTITUTION.
    IP=IPS(N)
    X(N)=X(N)/UL(IP,N)
    DO 40 IBACK=2,N
      I=NP1-IBACK
      IP=IPS(I)
      IP1=I+1
      SUM=X(I)
      DO 30 J=IP1,N
  30      SUM=SUM-UL(IP,J)*X(J)
  40    X(I)=SUM/UL(IP,I)
```

```
      RETURN
      END
      SUBROUTINE IMPRUV(N,NEW,LA,UL,LUL,B,X,DIGITS,IPS,R,DX,LP)
C
C  A IS THE ORIGINAL MATRIX, UL IS FROM DECOMP, B IS THE RIGHTHAND
SIDE,
C  AND X IS THE APPROXIMATE SOLUTION FROM SOLVE.
C  IMPRUV IMPROVES X TO MACHINE ACCURACY AND SETS DIGITS TO THE
NUMBER
C  OF CORRECT DIGITS IN THE FIRST ITERATE OF X.
C  IMPRUV PERFORMS ABOUT N**2 DOUBLE PRECISION MULTIPLICATIONS
PER
C  ITERATION.
C  THE USE OF AN ARRAY DUBLX WOULD SAVE TIME AT THE EXPENSE OF
STORAGE.
C
C  G. E. FORSYTHE AND C. B. MOLER, -COMPUTER SOLUTION OF LINEAR
C      ALGEBRAIC SYSTEMS- (PRENTICE-HALL, 1967)
C
C     IMPLICIT REAL*8 (A-H,O-Z)
C     REAL*16 SUM,DUBLA,DUBLX
      DOUBLE PRECISION SUM,DUBLA,DUBLX
      DOUBLE PRECISION NEW,UL,B,X,IPS,R,DX
C
      DIMENSION NEW(LA,N),UL(LUL,N),B(N),X(N),R(N),DX(N),IPS(N)
C
      ZLOG(ARG)=ALOG(ARG)
C     ZLOG(ARG)=DLOG(ARG)
C
C  EPS AND ITMAX ARE MACHINE DEPENDENT.
C  EPS IS THE RELATIVE ACCURACY OF THE FLOATING POINT MANTISSA.
C
      ZERO=0.0D0
      ONE=1.0D0
      TWO=2.0D0
C                 COMPUTE MACHINE EPSILON.
      XX=ONE
   10 XX=XX/TWO
      TEMP=ONE+XX
      IF(TEMP.GT.ONE) GO TO 10
      EPS=XX+XX
C                 ALNTN IS THE NATURAL LOG OF TEN.
      XX=10.0D0
      ALNTN=ZLOG(XX)
C
```

```
C  ITMAX IS TWICE THE NUMBER OF SIGNIFICANT DECIMAL DIGITS IN
C  A FLOATING POINT MANTISSA.  THIS IS SOMEWHAT ARBITRARY.
C
      ITMAX=ONE-TWO*ZLOG(EPS)/ALNTN
C
      XNORM=ZERO
      DO 20 I=1,N
C                 XNORM=AMAX1(XNORM,ABS(X(I)))
        ABSX=X(I)
        IF(ABSX.LT.ZERO) ABSX=-ABSX
        IF(ABSX.GT.XNORM) XNORM=ABSX
   20   CONTINUE
      IF(XNORM.LE.ZERO) THEN
        DIGITS=-ZLOG(EPS)/ALNTN
        RETURN
      ENDIF
C
      DO 60 ITER=1,ITMAX
        DO 40 I=1,N
          SUM=ZERO
          DO 30 J=1,N
            DUBLA=NEW(I,J)
            DUBLX=X(J)
   30       SUM=SUM+DUBLA*DUBLX
          DUBLA=B(I)
   40     R(I)=DUBLA-SUM
C
C  IT IS ESSENTIAL THAT A(I,J)*X(J) YIELD A DOUBLE PRECISION RESULT
C  AND THAT THE ABOVE + AND - BE DOUBLE PRECISION.
C
        CALL SOLVE(N,UL,LUL,R,DX,IPS)
        DXNORM=ZERO
        DO 50 I=1,N
          T=X(I)
          X(I)=X(I)+DX(I)
C                 DXNORM=AMAX1(DXNORM,ABS(X(I)-T))
          ABSXM=X(I)-T
          IF(ABSXM.LT.ZERO) ABSXM=-ABSXM
          IF(ABSXM.GT.DXNORM) DXNORM=ABSXM
   50     CONTINUE
        IF(ITER.EQ.1) THEN
C
C                 DIGITS=-ALOG10(AMAX1(DXNORM/XNORM,EPS))
          DIG=DXNORM/XNORM
          IF(EPS.GT.DIG) DIG=EPS
```

```
        DIGITS=-ZLOG(DIG)/ALNTN
      ENDIF
      IF(DXNORM.LE.EPS*XNORM) RETURN
   60 CONTINUE
C
    RETURN
    END
    SUBROUTINE INVERT(N,NEW,LA,INV,UL,LUL,B,X,SCALES,IPS,R,DX)
C
C  INVERTS THE MATRIX A.
C  FOR ILL-CONDITIONED MATRICES THIS ROUTINE IS MUCH MORE
ACCURATE THAN,
C  E.G., MATINV, BUT IT IS ALSO MUCH SLOWER.
C
C  G. E. FORSYTHE AND C. B. MOLER, -COMPUTER SOLUTION OF LINEAR
C     ALGEBRAIC SYSTEMS- (PRENTICE-HALL, 1967)
C
C  N IS THE ORDER OF A.
C  N MUST NOT EXCEED THE DIMENSION OF B OR X.
C
C  LA IS THE FIRST DIMENSION OF THE ARRAYS A AND AINV.
C  UL RETURNS THE TRIANGULAR DECOMPOSITION MATRICES.  UL MUST
BE
C  DIMENSIONED AT LEAST N BY N.
C  LUL IS THE FIRST DIMENSION OF THE ARRAY UL.
C
C  EXAMPLE (INVERSION OF A 6 BY 6 MATRIX OF RANDOM NUMBERS)....
C
C    DIMENSION NEW(20,25),INV(20,30),UL(15,35),B(20),X(20),
C  IN MOST CASES, A, AINV, AND UL WOULD PROBABLY BE SQUARE
ARRAYS.
C
C  J. P. CHANDLER, COMPUTER SCIENCE DEPT., OKLAHOMA STATE
UNIVERSITY
C
    DOUBLE PRECISION NEW,INV,UL,B,X,SCALES,IPS,R,DX
    DIMENSION NEW(LA,N),INV(LA,N),UL(LUL,N),B(N),X(N),
   *  SCALES(N),IPS(N),R(N),DX(N)
C
    CALL DECOMP(N,NEW,LA,UL,LUL,SCALES,IPS,LP)
    DO 30 J=1,N
      DO 10 K=1,N
  10     B(K)=0.0D0
      B(J)=1.0D0
      CALL SOLVE(N,UL,LUL,B,X,IPS)
```

```
      CALL IMPRUV(N,NEW,LA,UL,LUL,B,X,DIGITS,IPS,R,DX,LP)
C                    -DIGITS- IS IGNORED.
      DO 20 K=1,N
   20    INV(K,J)=X(K)
   30 CONTINUE
      RETURN
      END
```

APPENDIX B

OUTPUT DATA

```
==========================================================================
M= 2  N= 2  AVG EXACT ITER= 2.000  EXACT No.= 100
TOTAL OF: KINFORIG= 2.054E+28  KINFCHAN= 1.183E+02  KINFHAM= 1.183E+02
       KPPORIG = 4.143E+27  KPPCHAN = 1.033E+02  KPPHAM = 1.033E+02
LOG TOTAL: KINFCHAN/KINFORIG=-2.624E+01  KINFHAM/KINFORIG=-2.624E+01
       KPPCHAN/KPPORIG =-2.560E+01  KPPHAM/KPPORIG =-2.560E+01
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02  DR= 1.000E+02
==========================================================================
M= 2  N= 3  AVG EXACT ITER= 2.000  EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02  DR= 1.000E+02
==========================================================================
M= 2  N= 4  AVG EXACT ITER= 2.000  EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02  DR= 1.000E+02
==========================================================================
M= 2  N= 5  AVG EXACT ITER= 2.000  EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02  DR= 1.000E+02
==========================================================================
M= 2  N= 6  AVG EXACT ITER= 2.000  EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02  DR= 1.000E+02
==========================================================================
M= 2  N= 7  AVG EXACT ITER= 2.000  EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02  DR= 1.000E+02
==========================================================================
M= 2  N= 8  AVG EXACT ITER= 2.000  EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02  DR= 1.000E+02
==========================================================================
M= 2  N= 9  AVG EXACT ITER= 2.000  EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02  DR= 1.000E+02
==========================================================================
M= 2  N= 10  AVG EXACT ITER= 2.000  EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02  DR= 1.000E+02
==========================================================================
M= 2  N= 11  AVG EXACT ITER= 2.000  EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02  DR= 1.000E+02
==========================================================================
M= 2  N= 12  AVG EXACT ITER= 2.000  EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02  DR= 1.000E+02
==========================================================================
M= 2  N= 13  AVG EXACT ITER= 2.000  EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02  DR= 1.000E+02
==========================================================================
M= 2  N= 14  AVG EXACT ITER= 2.000  EXACT No.= 100
```

PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02
==
M= 2 N= 15 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02
==
M= 2 N= 16 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02
==
M= 2 N= 17 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02
==
M= 2 N= 18 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02
==
M= 2 N= 19 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02
==
M= 2 N= 20 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02
==
M= 3 N= 2 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02
==
M= 3 N= 3 AVG EXACT ITER= 2.419 EXACT No.= 93
TOTAL OF: KINFORIG= 7.445E+26 KINFCHAN= 6.372E+11 KINFHAM= 5.458E+13
KPPORIG = 5.459E+26 KPPCHAN = 1.593E+11 KPPHAM = 2.005E+13
LOG TOTAL: KINFCHAN/KINFORIG=-1.507E+01 KINFHAM/KINFORIG=-1.313E+01
KPPCHAN/KPPORIG =-1.553E+01 KPPHAM/KPPORIG =-1.344E+01
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 9.534E+01 DR= 9.534E+01
==
M= 3 N= 4 AVG EXACT ITER= 2.494 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 8.277E+01 DR= 8.427E+01
==
M= 3 N= 5 AVG EXACT ITER= 2.695 EXACT No.= 82
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 7.683E+01 DR= 7.927E+01
==
M= 3 N= 6 AVG EXACT ITER= 2.774 EXACT No.= 84
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 7.262E+01 DR= 7.698E+01
==
M= 3 N= 7 AVG EXACT ITER= 2.738 EXACT No.= 84
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.944E+01 DR= 7.398E+01
==
M= 3 N= 8 AVG EXACT ITER= 3.305 EXACT No.= 82
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.911E+01 DR= 7.088E+01
==
M= 3 N= 9 AVG EXACT ITER= 3.330 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.705E+01 DR= 6.894E+01
==
M= 3 N= 10 AVG EXACT ITER= 4.297 EXACT No.= 91
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.667E+01 DR= 6.692E+01
==
M= 3 N= 11 AVG EXACT ITER= 6.300 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.333E+01 DR= 6.436E+01
==

M= 3 N= 12 AVG EXACT ITER= 7.350 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.267E+01 DR= 6.250E+01

M= 3 N= 13 AVG EXACT ITER= 6.680 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.300E+01 DR= 6.169E+01

M= 3 N= 14 AVG EXACT ITER= 7.420 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.200E+01 DR= 6.443E+01

M= 3 N= 15 AVG EXACT ITER= 5.930 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.300E+01 DR= 6.533E+01

M= 3 N= 16 AVG EXACT ITER= 6.170 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.133E+01 DR= 6.350E+01

M= 3 N= 17 AVG EXACT ITER= 6.570 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.333E+01 DR= 6.341E+01

M= 3 N= 18 AVG EXACT ITER= 6.380 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.400E+01 DR= 6.472E+01

M= 3 N= 19 AVG EXACT ITER= 6.470 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.367E+01 DR= 6.332E+01

M= 3 N= 20 AVG EXACT ITER= 6.470 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.300E+01 DR= 6.340E+01

M= 4 N= 2 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02

M= 4 N= 3 AVG EXACT ITER= 2.500 EXACT No.= 94
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 9.202E+01 DR= 9.113E+01

M= 4 N= 4 AVG EXACT ITER= 2.776 EXACT No.= 76
TOTAL OF: KINFORIG= 1.741E+28 KINFCHAN= 2.658E+14 KINFHAM= 3.892E+18
KPPORIG = 3.405E+24 KPPCHAN = 4.188E+12 KPPHAM = 1.045E+18
LOG TOTAL: KINFCHAN/KINFORIG=-1.382E+01 KINFHAM/KINFORIG=-9.651E+00
KPPCHAN/KPPORIG =-1.191E+01 KPPHAM/KPPORIG =-6.513E+00
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 7.961E+01 DR= 7.895E+01

M= 4 N= 5 AVG EXACT ITER= 2.878 EXACT No.= 82
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.585E+01 DR= 7.122E+01

M= 4 N= 6 AVG EXACT ITER= 3.260 EXACT No.= 77
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.721E+01 DR= 6.991E+01

M= 4 N= 7 AVG EXACT ITER= 3.366 EXACT No.= 82
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.945E+01 DR= 6.394E+01

M= 4 N= 8 AVG EXACT ITER= 3.636 EXACT No.= 66
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.833E+01 DR= 6.136E+01

M= 4 N= 9 AVG EXACT ITER= 6.384 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.971E+01 DR= 5.323E+01

==

M= 4 N= 10 AVG EXACT ITER= 6.687 EXACT No.= 99
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.975E+01 DR= 5.141E+01

==

M= 4 N= 11 AVG EXACT ITER= 7.621 EXACT No.= 95
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.816E+01 DR= 5.187E+01

==

M= 4 N= 12 AVG EXACT ITER= 7.263 EXACT No.= 99
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.823E+01 DR= 5.093E+01

==

M= 4 N= 13 AVG EXACT ITER= 7.366 EXACT No.= 93
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.946E+01 DR= 5.153E+01

==

M= 4 N= 14 AVG EXACT ITER= 7.838 EXACT No.= 99
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.798E+01 DR= 5.058E+01

==

M= 4 N= 15 AVG EXACT ITER= 7.188 EXACT No.= 96
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.844E+01 DR= 5.104E+01

==

M= 4 N= 16 AVG EXACT ITER= 8.708 EXACT No.= 96
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.635E+01 DR= 4.798E+01

==

M= 4 N= 17 AVG EXACT ITER= 8.250 EXACT No.= 92
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.864E+01 DR= 5.115E+01

==

M= 4 N= 18 AVG EXACT ITER= 7.946 EXACT No.= 92
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.538E+01 DR= 4.698E+01

==

M= 4 N= 19 AVG EXACT ITER= 8.745 EXACT No.= 94
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.840E+01 DR= 4.916E+01

==

M= 4 N= 20 AVG EXACT ITER= 8.957 EXACT No.= 94
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.574E+01 DR= 4.654E+01

==

M= 5 N= 2 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02

==

M= 5 N= 3 AVG EXACT ITER= 2.438 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 8.427E+01 DR= 8.352E+01

==

M= 5 N= 4 AVG EXACT ITER= 2.592 EXACT No.= 76
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 7.526E+01 DR= 7.467E+01

==

M= 5 N= 5 AVG EXACT ITER= 3.320 EXACT No.= 75
TOTAL OF: KINFORIG= 1.679E+27 KINFCHAN= 2.031E+15 KINFHAM= 1.836E+24
KPPORIG = 4.348E+23 KPPCHAN = 6.029E+13 KPPHAM = 7.045E+18
LOG TOTAL: KINFCHAN/KINFORIG=-1.192E+01 KINFHAM/KINFORIG=-2.961E+00
KPPCHAN/KPPORIG =-9.858E+00 KPPHAM/KPPORIG =-4.790E+00
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.400E+01 DR= 6.533E+01

==

M= 5 N= 6 AVG EXACT ITER= 3.563 EXACT No.= 71
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.662E+01 DR= 5.822E+01

==

M= 5 N= 7 AVG EXACT ITER= 3.753 EXACT No.= 81

PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.333E+01 DR= 5.750E+01

M= 5 N= 8 AVG EXACT ITER= 6.494 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.828E+01 DR= 5.029E+01

M= 5 N= 9 AVG EXACT ITER= 8.033 EXACT No.= 92
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.565E+01 DR= 4.746E+01

M= 5 N= 10 AVG EXACT ITER= 8.902 EXACT No.= 92
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.261E+01 DR= 4.696E+01

M= 5 N= 11 AVG EXACT ITER= 9.052 EXACT No.= 97
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.062E+01 DR= 4.133E+01

M= 5 N= 12 AVG EXACT ITER= 7.935 EXACT No.= 93
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.172E+01 DR= 4.453E+01

M= 5 N= 13 AVG EXACT ITER= 7.720 EXACT No.= 93
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.935E+01 DR= 4.243E+01

M= 5 N= 14 AVG EXACT ITER= 8.341 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.886E+01 DR= 4.164E+01

M= 5 N= 15 AVG EXACT ITER= 8.921 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.910E+01 DR= 4.120E+01

M= 5 N= 16 AVG EXACT ITER= 9.052 EXACT No.= 96
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.000E+01 DR= 4.102E+01

M= 5 N= 17 AVG EXACT ITER= 9.042 EXACT No.= 96
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.854E+01 DR= 3.989E+01

M= 5 N= 18 AVG EXACT ITER=10.484 EXACT No.= 91
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.780E+01 DR= 4.145E+01

M= 5 N= 19 AVG EXACT ITER= 9.606 EXACT No.= 94
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.809E+01 DR= 4.037E+01

M= 5 N= 20 AVG EXACT ITER= 9.857 EXACT No.= 91
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.670E+01 DR= 3.890E+01

M= 6 N= 2 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02

M= 6 N= 3 AVG EXACT ITER= 2.763 EXACT No.= 93
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 8.441E+01 DR= 8.459E+01

M= 6 N= 4 AVG EXACT ITER= 2.921 EXACT No.= 76
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 7.303E+01 DR= 7.072E+01

M= 6 N= 5 AVG EXACT ITER= 3.355 EXACT No.= 76
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.754E+01 DR= 6.737E+01

M= 6 N= 6 AVG EXACT ITER= 4.343 EXACT No.= 70

TOTAL OF: KINFORIG= 2.665E+26 KINFCHAN= 1.444E+12 KINFHAM= 4.456E+20
KPPORIG = 1.267E+24 KPPCHAN = 3.308E+11 KPPHAM = 3.994E+17
LOG TOTAL: KINFCHAN/KINFORIG=-1.427E+01 KINFHAM/KINFORIG=-5.777E+00
KPPCHAN/KPPORIG =-1.258E+01 KPPHAM/KPPORIG =-6.501E+00
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.333E+01 DR= 5.500E+01

M= 6 N= 7 AVG EXACT ITER= 5.798 EXACT No.= 84
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.603E+01 DR= 4.847E+01

M= 6 N= 8 AVG EXACT ITER= 6.333 EXACT No.= 81
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.383E+01 DR= 4.506E+01

M= 6 N= 9 AVG EXACT ITER= 8.269 EXACT No.= 93
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.444E+01 DR= 4.492E+01

M= 6 N= 10 AVG EXACT ITER=10.516 EXACT No.= 91
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.938E+01 DR= 4.330E+01

M= 6 N= 11 AVG EXACT ITER= 8.516 EXACT No.= 95
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.825E+01 DR= 4.144E+01

M= 6 N= 12 AVG EXACT ITER=11.596 EXACT No.= 94
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.617E+01 DR= 4.051E+01

M= 6 N= 13 AVG EXACT ITER=10.784 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.447E+01 DR= 3.925E+01

M= 6 N= 14 AVG EXACT ITER= 9.467 EXACT No.= 90
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.481E+01 DR= 3.762E+01

M= 6 N= 15 AVG EXACT ITER= 9.784 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.542E+01 DR= 3.735E+01

M= 6 N= 16 AVG EXACT ITER= 8.558 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.469E+01 DR= 3.866E+01

M= 6 N= 17 AVG EXACT ITER= 8.967 EXACT No.= 92
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.352E+01 DR= 3.370E+01

M= 6 N= 18 AVG EXACT ITER= 9.920 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.295E+01 DR= 3.563E+01

M= 6 N= 19 AVG EXACT ITER=11.744 EXACT No.= 90
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.037E+01 DR= 3.287E+01

M= 6 N= 20 AVG EXACT ITER= 9.553 EXACT No.= 85
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.157E+01 DR= 3.571E+01

M= 7 N= 2 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02

M= 7 N= 3 AVG EXACT ITER= 2.679 EXACT No.= 81
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 8.289E+01 DR= 8.354E+01

M= 7 N= 4 AVG EXACT ITER= 3.263 EXACT No.= 80
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 7.125E+01 DR= 6.938E+01

==

M= 7 N= 5 AVG EXACT ITER= 4.378 EXACT No.= 74
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.502E+01 DR= 5.514E+01

==

M= 7 N= 6 AVG EXACT ITER= 5.385 EXACT No.= 78
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.165E+01 DR= 5.534E+01

==

M= 7 N= 7 AVG EXACT ITER= 8.398 EXACT No.= 88
TOTAL OF: KINFORIG= 3.012E+24 KINFCHAN= 4.295E+12 KINFHAM= 1.215E+18
KPPORIG = 5.157E+22 KPPCHAN = 1.522E+11 KPPHAM = 3.167E+16
LOG TOTAL: KINFCHAN/KINFORIG=-1.185E+01 KINFHAM/KINFORIG=-6.394E+00
KPPCHAN/KPPORIG =-1.153E+01 KPPHAM/KPPORIG =-6.212E+00
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.951E+01 DR= 4.886E+01

==

M= 7 N= 8 AVG EXACT ITER= 9.000 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.483E+01 DR= 4.598E+01

==

M= 7 N= 9 AVG EXACT ITER=10.635 EXACT No.= 85
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.429E+01 DR= 3.569E+01

==

M= 7 N= 10 AVG EXACT ITER= 8.784 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.734E+01 DR= 3.795E+01

==

M= 7 N= 11 AVG EXACT ITER= 9.292 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.547E+01 DR= 3.902E+01

==

M= 7 N= 12 AVG EXACT ITER= 9.864 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.312E+01 DR= 3.627E+01

==

M= 7 N= 13 AVG EXACT ITER= 9.604 EXACT No.= 91
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.046E+01 DR= 3.373E+01

==

M= 7 N= 14 AVG EXACT ITER=10.067 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.066E+01 DR= 3.339E+01

==

M= 7 N= 15 AVG EXACT ITER= 9.012 EXACT No.= 81
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.316E+01 DR= 3.638E+01

==

M= 7 N= 16 AVG EXACT ITER= 9.091 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.019E+01 DR= 3.395E+01

==

M= 7 N= 17 AVG EXACT ITER= 7.977 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.003E+01 DR= 3.135E+01

==

M= 7 N= 18 AVG EXACT ITER=12.119 EXACT No.= 84
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.857E+01 DR= 3.168E+01

==

M= 7 N= 19 AVG EXACT ITER=10.322 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.989E+01 DR= 3.279E+01

==

M= 7 N= 20 AVG EXACT ITER=11.235 EXACT No.= 81
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.875E+01 DR= 3.185E+01

==

M= 8 N= 2 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02

==

M= 8 N= 3 AVG EXACT ITER= 2.902 EXACT No.= 82
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 8.521E+01 DR= 8.374E+01

==

M= 8 N= 4 AVG EXACT ITER= 4.513 EXACT No.= 78
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.763E+01 DR= 6.474E+01

==

M= 8 N= 5 AVG EXACT ITER= 7.179 EXACT No.= 95
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.605E+01 DR= 5.453E+01

==

M= 8 N= 6 AVG EXACT ITER= 7.386 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.730E+01 DR= 4.811E+01

==

M= 8 N= 7 AVG EXACT ITER= 8.897 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.124E+01 DR= 4.122E+01

==

M= 8 N= 8 AVG EXACT ITER= 9.277 EXACT No.= 94
TOTAL OF: KINFORIG= 1.879E+24 KINFCHAN= 2.073E+11 KINFHAM= 1.833E+21
KPPORIG = 2.875E+19 KPPCHAN = 3.899E+08 KPPHAM = 7.613E+17
LOG TOTAL: KINFCHAN/KINFORIG=-1.296E+01 KINFHAM/KINFORIG=-3.011E+00
KPPCHAN/KPPORIG =-1.087E+01 KPPHAM/KPPORIG =-1.577E+00
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.816E+01 DR= 3.989E+01

==

M= 8 N= 9 AVG EXACT ITER=10.565 EXACT No.= 85
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.735E+01 DR= 3.725E+01

==

M= 8 N= 10 AVG EXACT ITER= 8.977 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.608E+01 DR= 3.636E+01

==

M= 8 N= 11 AVG EXACT ITER= 9.180 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.455E+01 DR= 3.463E+01

==

M= 8 N= 12 AVG EXACT ITER= 9.218 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.175E+01 DR= 3.218E+01

==

M= 8 N= 13 AVG EXACT ITER=11.682 EXACT No.= 85
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.250E+01 DR= 3.629E+01

==

M= 8 N= 14 AVG EXACT ITER=11.190 EXACT No.= 84
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.589E+01 DR= 2.806E+01

==

M= 8 N= 15 AVG EXACT ITER= 8.736 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.830E+01 DR= 2.835E+01

==

M= 8 N= 16 AVG EXACT ITER=10.705 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.770E+01 DR= 2.905E+01

==

M= 8 N= 17 AVG EXACT ITER=11.231 EXACT No.= 91
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.747E+01 DR= 3.083E+01

==

M= 8 N= 18 AVG EXACT ITER= 9.694 EXACT No.= 85

PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.824E+01 DR= 3.059E+01

==

M= 8 N= 19 AVG EXACT ITER= 9.930 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.427E+01 DR= 2.931E+01

==

M= 8 N= 20 AVG EXACT ITER=10.793 EXACT No.= 82
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.561E+01 DR= 2.744E+01

==

M= 9 N= 2 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02

==

M= 9 N= 3 AVG EXACT ITER= 3.214 EXACT No.= 84
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 7.712E+01 DR= 7.817E+01

==

M= 9 N= 4 AVG EXACT ITER= 5.278 EXACT No.= 90
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.444E+01 DR= 6.639E+01

==

M= 9 N= 5 AVG EXACT ITER= 9.625 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.202E+01 DR= 5.364E+01

==

M= 9 N= 6 AVG EXACT ITER= 9.521 EXACT No.= 94
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.047E+01 DR= 5.053E+01

==

M= 9 N= 7 AVG EXACT ITER= 9.910 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.444E+01 DR= 4.350E+01

==

M= 9 N= 8 AVG EXACT ITER=10.614 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.028E+01 DR= 3.963E+01

==

M= 9 N= 9 AVG EXACT ITER=10.395 EXACT No.= 86
TOTAL OF: KINFORIG= 1.874E+24 KINFCHAN= 7.245E+13 KINFHAM= 2.284E+18
KPPORIG = 5.802E+19 KPPCHAN = 1.166E+13 KPPHAM = 2.380E+16
LOG TOTAL: KINFCHAN/KINFORIG=-1.041E+01 KINFHAM/KINFORIG=-5.914E+00
KPPCHAN/KPPORIG =-6.697E+00 KPPHAM/KPPORIG =-3.387E+00
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.501E+01 DR= 3.514E+01

==

M= 9 N= 10 AVG EXACT ITER= 7.805 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.231E+01 DR= 3.368E+01

==

M= 9 N= 11 AVG EXACT ITER= 9.416 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.433E+01 DR= 3.544E+01

==

M= 9 N= 12 AVG EXACT ITER=12.444 EXACT No.= 90
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.000E+01 DR= 3.056E+01

==

M= 9 N= 13 AVG EXACT ITER= 9.385 EXACT No.= 91
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.808E+01 DR= 3.043E+01

==

M= 9 N= 14 AVG EXACT ITER=11.037 EXACT No.= 82
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.008E+01 DR= 3.267E+01

==

M= 9 N= 15 AVG EXACT ITER=10.860 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.545E+01 DR= 2.907E+01

==

M= 9 N= 16 AVG EXACT ITER=10.494 EXACT No.= 85
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.641E+01 DR= 2.809E+01
==

M= 9 N= 17 AVG EXACT ITER= 8.429 EXACT No.= 84
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.341E+01 DR= 2.626E+01
==

M= 9 N= 18 AVG EXACT ITER= 9.310 EXACT No.= 84
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.474E+01 DR= 2.560E+01
==

M= 9 N= 19 AVG EXACT ITER=11.648 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.323E+01 DR= 2.709E+01
==

M= 9 N= 20 AVG EXACT ITER=11.897 EXACT No.= 78
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.450E+01 DR= 2.718E+01
==

M= 10 N= 2 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02
==

M= 10 N= 3 AVG EXACT ITER= 5.125 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.716E+01 DR= 6.932E+01
==

M= 10 N= 4 AVG EXACT ITER= 7.854 EXACT No.= 96
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.990E+01 DR= 5.885E+01
==

M= 10 N= 5 AVG EXACT ITER= 8.505 EXACT No.= 97
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.093E+01 DR= 5.175E+01
==

M= 10 N= 6 AVG EXACT ITER= 9.011 EXACT No.= 94
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.362E+01 DR= 4.486E+01
==

M= 10 N= 7 AVG EXACT ITER=11.244 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.907E+01 DR= 3.953E+01
==

M= 10 N= 8 AVG EXACT ITER=10.218 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.747E+01 DR= 3.822E+01
==

M= 10 N= 9 AVG EXACT ITER= 8.727 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.727E+01 DR= 3.598E+01
==

M= 10 N= 10 AVG EXACT ITER=11.435 EXACT No.= 92
TOTAL OF: KINFORIG= 1.007E+23 KINFCHAN= 8.476E+12 KINFHAM= 1.425E+19
KPPORIG = 5.657E+19 KPPCHAN = 3.078E+09 KPPHAM = 5.467E+18
LOG TOTAL: KINFCHAN/KINFORIG=-1.007E+01 KINFHAM/KINFORIG=-3.849E+00
KPPCHAN/KPPORIG =-1.026E+01 KPPHAM/KPPORIG =-1.015E+00
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.293E+01 DR= 3.315E+01
==

M= 10 N= 11 AVG EXACT ITER= 9.744 EXACT No.= 90
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.022E+01 DR= 3.061E+01
==

M= 10 N= 12 AVG EXACT ITER=10.943 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.885E+01 DR= 3.046E+01
==

M= 10 N= 13 AVG EXACT ITER= 7.876 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.719E+01 DR= 2.740E+01

===

M= 10 N= 14 AVG EXACT ITER=11.000 EXACT No.= 90
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.644E+01 DR= 2.825E+01

===

M= 10 N= 15 AVG EXACT ITER= 9.341 EXACT No.= 91
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.440E+01 DR= 2.557E+01

===

M= 10 N= 16 AVG EXACT ITER=11.069 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.391E+01 DR= 2.766E+01

===

M= 10 N= 17 AVG EXACT ITER=10.477 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.442E+01 DR= 2.640E+01

===

M= 10 N= 18 AVG EXACT ITER= 9.382 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.461E+01 DR= 2.790E+01

===

M= 10 N= 19 AVG EXACT ITER=11.447 EXACT No.= 85
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.271E+01 DR= 2.458E+01

===

M= 10 N= 20 AVG EXACT ITER=10.931 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.195E+01 DR= 2.534E+01

===

M= 11 N= 2 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02

===

M= 11 N= 3 AVG EXACT ITER= 7.840 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 7.009E+01 DR= 6.900E+01

===

M= 11 N= 4 AVG EXACT ITER= 7.032 EXACT No.= 94
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.393E+01 DR= 6.344E+01

===

M= 11 N= 5 AVG EXACT ITER= 8.587 EXACT No.= 92
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.079E+01 DR= 5.065E+01

===

M= 11 N= 6 AVG EXACT ITER=10.351 EXACT No.= 94
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.700E+01 DR= 4.681E+01

===

M= 11 N= 7 AVG EXACT ITER= 7.032 EXACT No.= 95
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.943E+01 DR= 3.925E+01

===

M= 11 N= 8 AVG EXACT ITER= 8.818 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.626E+01 DR= 3.665E+01

===

M= 11 N= 9 AVG EXACT ITER=11.024 EXACT No.= 84
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.117E+01 DR= 3.280E+01

===

M= 11 N= 10 AVG EXACT ITER=10.449 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.921E+01 DR= 3.135E+01

===

M= 11 N= 11 AVG EXACT ITER= 9.977 EXACT No.= 86
TOTAL OF: KINFORIG= 6.240E+25 KINFCHAN= 9.848E+10 KINFHAM= 2.697E+21
KPPORIG = 6.240E+25 KPPCHAN = 9.517E+09 KPPHAM = 2.068E+19
LOG TOTAL: KINFCHAN/KINFORIG=-1.480E+01 KINFHAM/KINFORIG=-4.364E+00
KPPCHAN/KPPORIG =-1.582E+01 KPPHAM/KPPORIG =-6.480E+00

PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.780E+01 DR= 2.791E+01

==

M= 11 N= 12 AVG EXACT ITER=10.033 EXACT No.= 91
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.897E+01 DR= 2.949E+01

==

M= 11 N= 13 AVG EXACT ITER=10.446 EXACT No.= 92
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.470E+01 DR= 2.742E+01

==

M= 11 N= 14 AVG EXACT ITER=11.640 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.537E+01 DR= 2.799E+01

==

M= 11 N= 15 AVG EXACT ITER=10.516 EXACT No.= 91
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.547E+01 DR= 2.645E+01

==

M= 11 N= 16 AVG EXACT ITER=10.765 EXACT No.= 81
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.559E+01 DR= 2.755E+01

==

M= 11 N= 17 AVG EXACT ITER=10.229 EXACT No.= 83
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.322E+01 DR= 2.424E+01

==

M= 11 N= 18 AVG EXACT ITER=11.375 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.283E+01 DR= 2.551E+01

==

M= 11 N= 19 AVG EXACT ITER=10.704 EXACT No.= 81
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.099E+01 DR= 2.391E+01

==

M= 11 N= 20 AVG EXACT ITER=11.897 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.059E+01 DR= 2.276E+01

==

M= 12 N= 2 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02

==

M= 12 N= 3 AVG EXACT ITER= 6.240 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 7.417E+01 DR= 7.300E+01

==

M= 12 N= 4 AVG EXACT ITER= 7.526 EXACT No.= 95
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.167E+01 DR= 6.132E+01

==

M= 12 N= 5 AVG EXACT ITER= 8.842 EXACT No.= 95
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.061E+01 DR= 5.179E+01

==

M= 12 N= 6 AVG EXACT ITER= 8.104 EXACT No.= 96
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.731E+01 DR= 4.688E+01

==

M= 12 N= 7 AVG EXACT ITER= 9.293 EXACT No.= 92
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.732E+01 DR= 3.773E+01

==

M= 12 N= 8 AVG EXACT ITER= 8.412 EXACT No.= 97
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.720E+01 DR= 3.698E+01

==

M= 12 N= 9 AVG EXACT ITER= 9.719 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.202E+01 DR= 3.321E+01

==

M= 12 N= 10 AVG EXACT ITER=10.076 EXACT No.= 79

PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.101E+01 DR= 3.000E+01

==

M= 12 N= 11 AVG EXACT ITER=11.233 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.703E+01 DR= 2.939E+01

==

M= 12 N= 12 AVG EXACT ITER=11.306 EXACT No.= 85
TOTAL OF: KINFORIG= 1.138E+22 KINFCHAN= 1.468E+12 KINFHAM= 1.377E+19
KPPORIG = 1.002E+16 KPPCHAN = 7.221E+10 KPPHAM = 8.782E+17
LOG TOTAL: KINFCHAN/KINFORIG=-9.889E+00 KINFHAM/KINFORIG=-2.917E+00
KPPCHAN/KPPORIG =-5.142E+00 KPPHAM/KPPORIG = 1.943E+00
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.696E+01 DR= 2.853E+01

==

M= 12 N= 13 AVG EXACT ITER=10.402 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.586E+01 DR= 2.626E+01

==

M= 12 N= 14 AVG EXACT ITER=11.134 EXACT No.= 82
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.571E+01 DR= 2.587E+01

==

M= 12 N= 15 AVG EXACT ITER=12.198 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.316E+01 DR= 2.411E+01

==

M= 12 N= 16 AVG EXACT ITER=10.701 EXACT No.= 77
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.262E+01 DR= 2.240E+01

==

M= 12 N= 17 AVG EXACT ITER=12.131 EXACT No.= 84
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.282E+01 DR= 2.381E+01

==

M= 12 N= 18 AVG EXACT ITER= 9.966 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.238E+01 DR= 2.341E+01

==

M= 12 N= 19 AVG EXACT ITER=11.560 EXACT No.= 84
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.163E+01 DR= 2.393E+01

==

M= 12 N= 20 AVG EXACT ITER=11.609 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.117E+01 DR= 2.397E+01

==

M= 13 N= 2 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02

==

M= 13 N= 3 AVG EXACT ITER= 6.620 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 7.215E+01 DR= 7.333E+01

==

M= 13 N= 4 AVG EXACT ITER= 9.073 EXACT No.= 96
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.018E+01 DR= 6.068E+01

==

M= 13 N= 5 AVG EXACT ITER= 9.525 EXACT No.= 99
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.120E+01 DR= 5.212E+01

==

M= 13 N= 6 AVG EXACT ITER=10.622 EXACT No.= 90
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.085E+01 DR= 4.167E+01

==

M= 13 N= 7 AVG EXACT ITER= 9.589 EXACT No.= 90
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.094E+01 DR= 4.127E+01

==

M= 13 N= 8 AVG EXACT ITER=11.432 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.330E+01 DR= 3.295E+01

==

M= 13 N= 9 AVG EXACT ITER= 8.929 EXACT No.= 85
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.430E+01 DR= 3.176E+01

==

M= 13 N= 10 AVG EXACT ITER=10.881 EXACT No.= 84
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.967E+01 DR= 3.000E+01

==

M= 13 N= 11 AVG EXACT ITER=10.540 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.732E+01 DR= 2.884E+01

==

M= 13 N= 12 AVG EXACT ITER= 9.778 EXACT No.= 81
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.735E+01 DR= 2.809E+01

==

M= 13 N= 13 AVG EXACT ITER=12.885 EXACT No.= 78
TOTAL OF: KINFORIG= 9.963E+19 KINFCHAN= 2.107E+11 KINFHAM= 7.019E+18
KPPORIG = 2.324E+15 KPPCHAN = 2.493E+09 KPPHAM = 3.500E+16
LOG TOTAL: KINFCHAN/KINFORIG=-8.675E+00 KINFHAM/KINFORIG=-1.152E+00
KPPCHAN/KPPORIG =-5.970E+00 KPPHAM/KPPORIG = 1.178E+00
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.544E+01 DR= 2.673E+01

==

M= 13 N= 14 AVG EXACT ITER=11.365 EXACT No.= 85
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.416E+01 DR= 2.521E+01

==

M= 13 N= 15 AVG EXACT ITER=10.012 EXACT No.= 80
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.423E+01 DR= 2.608E+01

==

M= 13 N= 16 AVG EXACT ITER= 8.667 EXACT No.= 84
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.225E+01 DR= 2.463E+01

==

M= 13 N= 17 AVG EXACT ITER= 8.662 EXACT No.= 80
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.096E+01 DR= 2.022E+01

==

M= 13 N= 18 AVG EXACT ITER=10.352 EXACT No.= 91
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.944E+01 DR= 2.088E+01

==

M= 13 N= 19 AVG EXACT ITER=13.118 EXACT No.= 85
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.000E+01 DR= 2.155E+01

==

M= 13 N= 20 AVG EXACT ITER=11.575 EXACT No.= 80
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.115E+01 DR= 2.244E+01

==

M= 14 N= 2 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02

==

M= 14 N= 3 AVG EXACT ITER= 5.840 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.929E+01 DR= 6.833E+01

==

M= 14 N= 4 AVG EXACT ITER= 8.796 EXACT No.= 98
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.736E+01 DR= 5.714E+01

==

M= 14 N= 5 AVG EXACT ITER=10.054 EXACT No.= 92
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.930E+01 DR= 4.848E+01

M= 14 N= 6 AVG EXACT ITER= 8.930 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.103E+01 DR= 4.147E+01

M= 14 N= 7 AVG EXACT ITER=11.556 EXACT No.= 90
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.167E+01 DR= 4.016E+01

M= 14 N= 8 AVG EXACT ITER=11.875 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.661E+01 DR= 3.622E+01

M= 14 N= 9 AVG EXACT ITER= 8.729 EXACT No.= 85
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.294E+01 DR= 3.085E+01

M= 14 N= 10 AVG EXACT ITER=11.071 EXACT No.= 85
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.126E+01 DR= 3.012E+01

M= 14 N= 11 AVG EXACT ITER=11.550 EXACT No.= 80
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.991E+01 DR= 3.000E+01

M= 14 N= 12 AVG EXACT ITER=11.205 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.800E+01 DR= 2.708E+01

M= 14 N= 13 AVG EXACT ITER=10.556 EXACT No.= 81
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.593E+01 DR= 2.659E+01

M= 14 N= 14 AVG EXACT ITER=11.449 EXACT No.= 78
TOTAL OF: KINFORIG= 2.083E+22 KINFCHAN= 6.333E+09 KINFHAM= 4.800E+22
KPPORIG = 1.830E+17 KPPCHAN = 1.276E+08 KPPHAM = 9.619E+16
LOG TOTAL: KINFCHAN/KINFORIG=-1.252E+01 KINFHAM/KINFORIG= 3.626E-01
KPPCHAN/KPPORIG =-9.156E+00 KPPHAM/KPPORIG =-2.793E-01
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.353E+01 DR= 2.408E+01

M= 14 N= 15 AVG EXACT ITER=11.500 EXACT No.= 82
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.378E+01 DR= 2.463E+01

M= 14 N= 16 AVG EXACT ITER=12.214 EXACT No.= 84
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.262E+01 DR= 2.426E+01

M= 14 N= 17 AVG EXACT ITER=13.414 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.135E+01 DR= 2.197E+01

M= 14 N= 18 AVG EXACT ITER=11.322 EXACT No.= 90
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.135E+01 DR= 2.265E+01

M= 14 N= 19 AVG EXACT ITER=12.860 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.135E+01 DR= 2.130E+01

M= 14 N= 20 AVG EXACT ITER=14.604 EXACT No.= 91
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.080E+01 DR= 2.104E+01

M= 15 N= 2 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02

M= 15 N= 3 AVG EXACT ITER= 6.580 EXACT No.= 100

PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 7.380E+01 DR= 7.300E+01

M= 15 N= 4 AVG EXACT ITER= 8.143 EXACT No.= 98
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.912E+01 DR= 5.893E+01

M= 15 N= 5 AVG EXACT ITER=10.286 EXACT No.= 91
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.703E+01 DR= 4.637E+01

M= 15 N= 6 AVG EXACT ITER= 8.944 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.345E+01 DR= 4.438E+01

M= 15 N= 7 AVG EXACT ITER=10.424 EXACT No.= 92
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.014E+01 DR= 3.804E+01

M= 15 N= 8 AVG EXACT ITER=10.213 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.296E+01 DR= 3.371E+01

M= 15 N= 9 AVG EXACT ITER= 9.709 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.395E+01 DR= 3.204E+01

M= 15 N= 10 AVG EXACT ITER=10.622 EXACT No.= 82
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.114E+01 DR= 2.988E+01

M= 15 N= 11 AVG EXACT ITER=10.952 EXACT No.= 83
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.835E+01 DR= 2.815E+01

M= 15 N= 12 AVG EXACT ITER= 9.506 EXACT No.= 85
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.306E+01 DR= 2.343E+01

M= 15 N= 13 AVG EXACT ITER= 8.965 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.535E+01 DR= 2.416E+01

M= 15 N= 14 AVG EXACT ITER=10.721 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.287E+01 DR= 2.367E+01

M= 15 N= 15 AVG EXACT ITER=11.846 EXACT No.= 78
TOTAL OF: KINFORIG= 1.037E+19 KINFCHAN= 4.391E+08 KINFHAM= 2.372E+18
KPPORIG = 1.446E+14 KPPCHAN = 4.834E+06 KPPHAM = 2.522E+17
LOG TOTAL: KINFCHAN/KINFORIG=-1.037E+01 KINFHAM/KINFORIG=-6.407E-01
KPPCHAN/KPPORIG =-7.476E+00 KPPHAM/KPPORIG = 3.241E+00
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.376E+01 DR= 2.182E+01

M= 15 N= 16 AVG EXACT ITER=13.405 EXACT No.= 79
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.169E+01 DR= 2.191E+01

M= 15 N= 17 AVG EXACT ITER=12.253 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.169E+01 DR= 2.279E+01

M= 15 N= 18 AVG EXACT ITER=14.780 EXACT No.= 82
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.187E+01 DR= 2.310E+01

M= 15 N= 19 AVG EXACT ITER=14.034 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.969E+01 DR= 2.021E+01

M= 15 N= 20 AVG EXACT ITER=14.913 EXACT No.= 92
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.138E+01 DR= 2.114E+01

==

M= 16 N= 2 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02

==

M= 16 N= 3 AVG EXACT ITER= 5.890 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.813E+01 DR= 6.867E+01

==

M= 16 N= 4 AVG EXACT ITER= 8.030 EXACT No.= 99
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.486E+01 DR= 5.530E+01

==

M= 16 N= 5 AVG EXACT ITER= 7.495 EXACT No.= 93
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.289E+01 DR= 5.161E+01

==

M= 16 N= 6 AVG EXACT ITER= 8.527 EXACT No.= 93
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.052E+01 DR= 3.889E+01

==

M= 16 N= 7 AVG EXACT ITER= 9.484 EXACT No.= 91
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.393E+01 DR= 3.375E+01

==

M= 16 N= 8 AVG EXACT ITER= 9.112 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.462E+01 DR= 3.427E+01

==

M= 16 N= 9 AVG EXACT ITER=10.458 EXACT No.= 83
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.065E+01 DR= 2.972E+01

==

M= 16 N= 10 AVG EXACT ITER= 7.975 EXACT No.= 81
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.901E+01 DR= 2.790E+01

==

M= 16 N= 11 AVG EXACT ITER=10.889 EXACT No.= 81
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.654E+01 DR= 2.570E+01

==

M= 16 N= 12 AVG EXACT ITER= 8.474 EXACT No.= 78
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.500E+01 DR= 2.756E+01

==

M= 16 N= 13 AVG EXACT ITER=11.012 EXACT No.= 83
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.508E+01 DR= 2.475E+01

==

M= 16 N= 14 AVG EXACT ITER= 8.600 EXACT No.= 90
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.319E+01 DR= 2.389E+01

==

M= 16 N= 15 AVG EXACT ITER=12.513 EXACT No.= 76
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.072E+01 DR= 2.070E+01

==

M= 16 N= 16 AVG EXACT ITER=12.565 EXACT No.= 85
TOTAL OF: KINFORIG= 3.318E+16 KINFCHAN= 7.086E+08 KINFHAM= 1.313E+20
KPPORIG = 1.810E+14 KPPCHAN = 1.280E+07 KPPHAM = 1.032E+16
LOG TOTAL: KINFCHAN/KINFORIG=-7.670E+00 KINFHAM/KINFORIG= 3.598E+00
KPPCHAN/KPPORIG =-7.150E+00 KPPHAM/KPPORIG = 1.756E+00
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.132E+01 DR= 2.081E+01

==

M= 16 N= 17 AVG EXACT ITER=11.181 EXACT No.= 83
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.229E+01 DR= 2.232E+01

===

M= 16 N= 18 AVG EXACT ITER=12.709 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.071E+01 DR= 2.300E+01

===

M= 16 N= 19 AVG EXACT ITER=13.303 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.728E+01 DR= 1.881E+01

===

M= 16 N= 20 AVG EXACT ITER=15.511 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.861E+01 DR= 2.017E+01

===

M= 17 N= 2 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02

===

M= 17 N= 3 AVG EXACT ITER= 6.110 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.906E+01 DR= 6.900E+01

===

M= 17 N= 4 AVG EXACT ITER= 6.896 EXACT No.= 96
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.797E+01 DR= 5.859E+01

===

M= 17 N= 5 AVG EXACT ITER= 8.851 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.692E+01 DR= 4.805E+01

===

M= 17 N= 6 AVG EXACT ITER= 8.208 EXACT No.= 96
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.063E+01 DR= 4.028E+01

===

M= 17 N= 7 AVG EXACT ITER=11.326 EXACT No.= 92
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.574E+01 DR= 3.494E+01

===

M= 17 N= 8 AVG EXACT ITER= 9.728 EXACT No.= 92
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.376E+01 DR= 3.356E+01

===

M= 17 N= 9 AVG EXACT ITER=10.299 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.198E+01 DR= 3.052E+01

===

M= 17 N= 10 AVG EXACT ITER=10.293 EXACT No.= 82
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.877E+01 DR= 3.037E+01

===

M= 17 N= 11 AVG EXACT ITER= 9.212 EXACT No.= 85
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.886E+01 DR= 2.674E+01

===

M= 17 N= 12 AVG EXACT ITER=11.725 EXACT No.= 80
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.346E+01 DR= 2.313E+01

===

M= 17 N= 13 AVG EXACT ITER=12.292 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.254E+01 DR= 2.195E+01

===

M= 17 N= 14 AVG EXACT ITER=12.070 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.216E+01 DR= 2.326E+01

===

M= 17 N= 15 AVG EXACT ITER=10.205 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.126E+01 DR= 2.174E+01

===

M= 17 N= 16 AVG EXACT ITER=10.966 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.961E+01 DR= 2.040E+01

==

M= 17 N= 17 AVG EXACT ITER=13.000 EXACT No.= 90
TOTAL OF: KINFORIG= 1.230E+17 KINFCHAN= 1.720E+10 KINFHAM= 2.801E+21
KPPORIG = 4.579E+13 KPPCHAN = 2.411E+08 KPPHAM = 1.101E+16
LOG TOTAL: KINFCHAN/KINFORIG=-6.854E+00 KINFHAM/KINFORIG= 4.357E+00
KPPCHAN/KPPORIG =-5.279E+00 KPPHAM/KPPORIG = 2.381E+00
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.993E+01 DR= 2.124E+01

==

M= 17 N= 18 AVG EXACT ITER=15.333 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.116E+01 DR= 2.011E+01

==

M= 17 N= 19 AVG EXACT ITER=13.077 EXACT No.= 91
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.875E+01 DR= 1.961E+01

==

M= 17 N= 20 AVG EXACT ITER=13.791 EXACT No.= 91
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.836E+01 DR= 1.934E+01

==

M= 18 N= 2 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02

==

M= 18 N= 3 AVG EXACT ITER= 7.200 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.761E+01 DR= 6.933E+01

==

M= 18 N= 4 AVG EXACT ITER= 7.590 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 6.250E+01 DR= 6.250E+01

==

M= 18 N= 5 AVG EXACT ITER= 9.636 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.779E+01 DR= 4.750E+01

==

M= 18 N= 6 AVG EXACT ITER= 9.064 EXACT No.= 94
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.948E+01 DR= 3.812E+01

==

M= 18 N= 7 AVG EXACT ITER= 9.663 EXACT No.= 92
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.696E+01 DR= 3.649E+01

==

M= 18 N= 8 AVG EXACT ITER= 9.842 EXACT No.= 95
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.456E+01 DR= 3.382E+01

==

M= 18 N= 9 AVG EXACT ITER= 9.307 EXACT No.= 75
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.119E+01 DR= 3.007E+01

==

M= 18 N= 10 AVG EXACT ITER=11.167 EXACT No.= 84
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.632E+01 DR= 2.571E+01

==

M= 18 N= 11 AVG EXACT ITER= 9.295 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.487E+01 DR= 2.541E+01

==

M= 18 N= 12 AVG EXACT ITER=12.228 EXACT No.= 79
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.489E+01 DR= 2.468E+01

==

M= 18 N= 13 AVG EXACT ITER=13.410 EXACT No.= 83
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.390E+01 DR= 2.298E+01

==

M= 18 N= 14 AVG EXACT ITER= 9.988 EXACT No.= 82

PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.317E+01 DR= 2.152E+01
===
M= 18 N= 15 AVG EXACT ITER=13.966 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.146E+01 DR= 2.084E+01
===
M= 18 N= 16 AVG EXACT ITER=16.023 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.884E+01 DR= 1.997E+01
===
M= 18 N= 17 AVG EXACT ITER=14.157 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.829E+01 DR= 1.970E+01
===
M= 18 N= 18 AVG EXACT ITER=13.132 EXACT No.= 91
TOTAL OF: KINFORIG= 5.451E+17 KINFCHAN= 3.406E+09 KINFHAM= 1.312E+19
KPPORIG = 1.173E+13 KPPCHAN = 2.330E+07 KPPHAM = 2.040E+15
LOG TOTAL: KINFCHAN/KINFORIG=-8.204E+00 KINFHAM/KINFORIG= 1.381E+00
KPPCHAN/KPPORIG =-5.702E+00 KPPHAM/KPPORIG = 2.240E+00
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.880E+01 DR= 1.929E+01
===
M= 18 N= 19 AVG EXACT ITER=15.511 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.869E+01 DR= 1.842E+01
===
M= 18 N= 20 AVG EXACT ITER=14.568 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.970E+01 DR= 2.017E+01
===
M= 19 N= 2 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02
===
M= 19 N= 3 AVG EXACT ITER= 5.780 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 7.063E+01 DR= 7.200E+01
===
M= 19 N= 4 AVG EXACT ITER= 9.051 EXACT No.= 98
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.269E+01 DR= 5.179E+01
===
M= 19 N= 5 AVG EXACT ITER= 8.883 EXACT No.= 94
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.670E+01 DR= 4.787E+01
===
M= 19 N= 6 AVG EXACT ITER= 8.289 EXACT No.= 90
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.170E+01 DR= 4.204E+01
===
M= 19 N= 7 AVG EXACT ITER= 8.663 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.378E+01 DR= 3.372E+01
===
M= 19 N= 8 AVG EXACT ITER=11.830 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.931E+01 DR= 2.813E+01
===
M= 19 N= 9 AVG EXACT ITER= 9.920 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.164E+01 DR= 3.093E+01
===
M= 19 N= 10 AVG EXACT ITER=10.916 EXACT No.= 83
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.720E+01 DR= 2.590E+01
===
M= 19 N= 11 AVG EXACT ITER=11.580 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.452E+01 DR= 2.448E+01
===

M= 19 N= 12 AVG EXACT ITER=12.326 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.442E+01 DR= 2.461E+01

==

M= 19 N= 13 AVG EXACT ITER=12.198 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.613E+01 DR= 2.576E+01

==

M= 19 N= 14 AVG EXACT ITER=11.200 EXACT No.= 85
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.037E+01 DR= 2.042E+01

==

M= 19 N= 15 AVG EXACT ITER=12.022 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.111E+01 DR= 2.157E+01

==

M= 19 N= 16 AVG EXACT ITER=15.548 EXACT No.= 84
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.187E+01 DR= 2.180E+01

==

M= 19 N= 17 AVG EXACT ITER=12.115 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.924E+01 DR= 1.941E+01

==

M= 19 N= 18 AVG EXACT ITER=12.278 EXACT No.= 90
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.865E+01 DR= 1.809E+01

==

M= 19 N= 19 AVG EXACT ITER=16.348 EXACT No.= 89
TOTAL OF: KINFORIG= 1.148E+16 KINFCHAN= 4.358E+08 KINFHAM= 1.959E+17
KPPORIG = 1.029E+14 KPPCHAN = 2.159E+06 KPPHAM = 1.370E+14
LOG TOTAL: KINFCHAN/KINFORIG=-7.421E+00 KINFHAM/KINFORIG= 1.232E+00
KPPCHAN/KPPORIG =-7.678E+00 KPPHAM/KPPORIG = 1.243E-01
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.070E+01 DR= 2.099E+01

==

M= 19 N= 20 AVG EXACT ITER=14.864 EXACT No.= 88
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.806E+01 DR= 1.778E+01

==

M= 20 N= 2 AVG EXACT ITER= 2.000 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.000E+02 DR= 1.000E+02

==

M= 20 N= 3 AVG EXACT ITER= 7.610 EXACT No.= 100
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 7.000E+01 DR= 7.100E+01

==

M= 20 N= 4 AVG EXACT ITER= 9.898 EXACT No.= 98
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 5.781E+01 DR= 5.689E+01

==

M= 20 N= 5 AVG EXACT ITER= 9.446 EXACT No.= 92
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.609E+01 DR= 4.565E+01

==

M= 20 N= 6 AVG EXACT ITER= 9.341 EXACT No.= 85
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 4.000E+01 DR= 3.922E+01

==

M= 20 N= 7 AVG EXACT ITER= 9.721 EXACT No.= 86
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.634E+01 DR= 3.505E+01

==

M= 20 N= 8 AVG EXACT ITER=10.057 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.534E+01 DR= 3.491E+01

==

M= 20 N= 9 AVG EXACT ITER=10.146 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 3.062E+01 DR= 3.059E+01

==
M= 20 N= 10 AVG EXACT ITER=10.976 EXACT No.= 85
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.341E+01 DR= 2.388E+01
==
M= 20 N= 11 AVG EXACT ITER=10.393 EXACT No.= 84
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.476E+01 DR= 2.413E+01
==
M= 20 N= 12 AVG EXACT ITER=13.057 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.414E+01 DR= 2.538E+01
==
M= 20 N= 13 AVG EXACT ITER=11.765 EXACT No.= 85
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.288E+01 DR= 2.308E+01
==
M= 20 N= 14 AVG EXACT ITER=12.685 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.056E+01 DR= 2.159E+01
==
M= 20 N= 15 AVG EXACT ITER=14.936 EXACT No.= 94
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.840E+01 DR= 1.851E+01
==
M= 20 N= 16 AVG EXACT ITER=13.404 EXACT No.= 89
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 2.084E+01 DR= 2.086E+01
==
M= 20 N= 17 AVG EXACT ITER=13.688 EXACT No.= 93
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.898E+01 DR= 1.860E+01
==
M= 20 N= 18 AVG EXACT ITER=16.931 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.713E+01 DR= 1.769E+01
==
M= 20 N= 19 AVG EXACT ITER=11.621 EXACT No.= 87
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.747E+01 DR= 1.779E+01
==
M= 20 N= 20 AVG EXACT ITER=14.466 EXACT No.= 88
TOTAL OF: KINFORIG= 1.364E+14 KINFCHAN= 1.734E+08 KINFHAM= 2.561E+17
KPPORIG = 3.791E+11 KPPCHAN = 9.069E+06 KPPHAM = 6.906E+14
LOG TOTAL: KINFCHAN/KINFORIG=-5.896E+00 KINFHAM/KINFORIG= 3.274E+00
KPPCHAN/KPPORIG =-4.621E+00 KPPHAM/KPPORIG = 3.261E+00
PERCENTAGE OF ELEMENTS THAT CONVERGED EXACTLY: DL= 1.875E+01 DR= 1.955E+01

VITA

Chung-Chuan Wang

Candidate for the Degree of

Master of Science

Thesis: SURVEY AND COMPARISON OF MATRIX SCALING METHODS

Major Field: Computer Science

Biographical:

Personal Data: Born in Taiwan, Republic of China, November 9, 1960, the son of Wen-Chih Wang and Bee-Eun Wang-Lai.

Education: Received Bachelor of Science Degree in Hydraulic and Ocean Engineer from National Chen-Kung University, Taiwan, R.O.C. June 1983; completed requirements for the Master of Science degree at Oklahoma State University in December, 1993.

Professional Experience: System Programmer, Yamani Computer System, Inc., Taiwan, R.O.C., from March , 1987, to September, 1987; Software Engineer, Acer Incorporated, Taiwan, R.O.C., from January, 1988, to September, 1989; System Analyst, Long Bang Construction Company, Taiwan, R.O.C., from September, 1989, to August, 1990.