1969

Iterative methods of matrix inversion

Harvey Craig Ogden
The University of Montana

ITERATIVE METHODS OF MATRIX INVERSION

By

Harvey C. Ogden

B.S., Whitworth College, 1967

Presented in partial fulfillment of the requirements

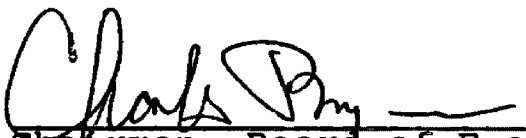
for the degree of

Master of Arts

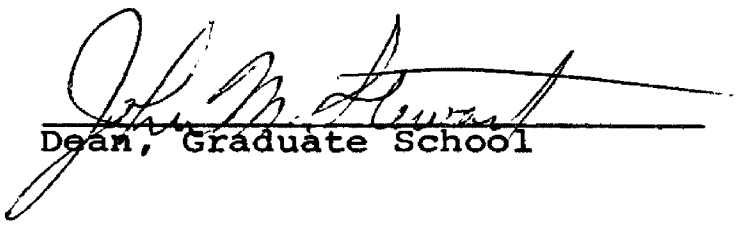
UNIVERSITY OF MONTANA

1969

Approved by:



Chairman, Board of Examiners



Dean, Graduate School

May 23, 1969

Date

ACKNOWLEDGEMENTS

I wish to express my gratitude to Professor Charles A. Bryan for his encouragement and guidance during the preparation and writing of this thesis. I also wish to thank Professors Robert P. Banaugh and Keith Yale for their helpful suggestions and critical reading of the manuscript. Finally, I wish to thank my wife, Karen, for her patience in typing the manuscript.

TABLE OF CONTENTS

INTRODUCTION

This thesis deals with the problem of improving an approximation to the inverse of a matrix by the method of successive iterations. Two basically different techniques will be introduced and compared.

The first method is well known and often is referred to as the Newton method. The second method is a natural extension of the successive overrelaxation iteration technique for solving systems of linear equations. However, to the author's knowledge, this technique has not been applied to matrix inversion before.

Since the theory of successive overrelaxation and similar iterative methods for solving systems of linear equations is well developed, much background material will be omitted. The reader may find an excellent reference in [1].

CHAPTER 1

THE NEWTON ITERATION TECHNIQUE

1.1. Preliminaries

In order to introduce the techniques, some definitions are listed. Let $A=(a_{ij})$ be an n x n matrix, then A is said to be <u>non-negative</u>, write $A \geq 0$, if all of its components are non-negative. A is said to be <u>hermitian</u> if A is equal to its conjugate transpose and A is <u>positive definite</u> if all of its eigenvalues are positive. The <u>spectrum</u> of A, $\sigma(A)$, is the set of all eigenvalues of A.

Definition 1.1. Let the <u>spectral radius</u> of A, denoted by P(A), be defined as $P(A)=\max[|\lambda_i| \ \lambda_i \in \sigma(A)]$.

Definition 1.2. Let X be an n x 1 column vector with components $x_1,\ldots,x_n$, then the <u>euclidean norm</u> of X is defined by

$$\|X\| = (X^*X)^{\frac{1}{2}} = (\sum_{i=1}^{n} |x_i|^2)^{\frac{1}{2}}.$$

Definition 1.3. Let $A=(a_{ij})$ be an n x n matrix, then the <u>spectral norm</u> of A is defined by

$$\|A\| = \sup_{\|X\|=1} \|AX\| .$$

1.2. Newton's Method

Let $A=(a_{ij})$ be an n x n matrix and suppose $G_k \doteq A^{-1}$, (i.e. G_k is approximately equal to A^{-1}). Let $\delta G_k = A^{-1} - G_k$.

Then $A=(A^{-1})^{-1}=(G_k+\delta G_k)^{-1}=[G_k(I+G_k^{-1}\delta G_k)]^{-1}$

$=(I+G_k^{-1}\delta G_k)^{-1}G_k^{-1}$.

So $AG_k=(I+G_k^{-1}\delta G_k)^{-1}$.

Make the approximation

$$(I+G_k^{-1}\delta G_k)^{-1}\doteq(I-G_k^{-1}\delta G_k).$$

Then $AG_k\doteq I-G_k^{-1}\delta G_k$.

Solving for δG_k yields

$$\delta G_k\doteq G_k(I-AG_k). \qquad (1\text{-}2)$$

Substituting this back into eq(1-1) yields

$$A^{-1}\doteq G_k+G_k(I-AG_k). \qquad (1\text{-}3)$$

This equation suggests the following iteration scheme.

$$G_{k+1}=G_k+G_k(I-AG_k) \quad k=0,1,\ldots ,$$

which can be rewritten as

$$G_{k+1}=G_k(I+E_k) \quad k=0,1,\ldots , \qquad (1\text{-}4)$$

where $E_k=I-AG_k$.

This is the Newton iteration technique for improving an approximation to the inverse of a matrix. The following theorem states the necessary and sufficient conditions for convergence of the method.

Theorem 1.1. Let A be an n x n, non-singular matrix with $G_0\doteq A^{-1}$. If G_k is defined as in eq(1-4) for k=1,2,... , then limit $G_k=A^{-1}$ if and only if $P(E_0)<1$.

Proof: $E_k=I-AG_k=I-AG_{k-1}(E_{k-1}+I)$

$=I-AG_{k-1}(I-AG_{k-1}+I)$

$=I-2AG_{k-1}+(AG_{k-1})^2$

$=(I-AG_{k-1})^2$

$=E_{k-1}^2$.

So $E_k=E_{k-1}^2=E_{k-2}^4=\ldots=E_0^{2^k}$. Now $\lim_{k\to\infty} G_k=A^{-1}$ if and only if $\lim_{k\to\infty} \delta G_k=O$, where O is the n x n matrix with all 0 components. Note that $A\delta G_k=A(A^{-1}-G_k)=I-AG_k=E_K$. So $\lim_{k\to\infty} \delta G_k=$

$A^{-1}(\lim_{k\to\infty} E_k)=A^{-1}(\lim_{k\to\infty} E_0^{2^k})=O$ if and only if $\lim_{k\to\infty} E_0^{2^k}=O$. But it is well known that if B is an n x n matrix, $\lim_{m\to\infty} B^m=O$ if and only if $P(B)<1$, [1,13]. Thus it follows that $\lim_{k\to\infty} E_0^{2^k}=O$ if and only if $P(E_0)<1$.

Since Newton's method converges if and only if $P(E_0)<1$, it is important to know what conditions on G_0 are sufficient to insure that $P(E_0)<1$, for this will be a measure of the overall usefulness of the method. To arrive at such a set of conditions it will be helpful to define another matrix norm.

Definition 1.4. If $A=(a_{ij})$ is an n x n matrix then the max norm, N(A), of A is defined by

$$N(A)=n(\max_{i,j}|a_{ij}|).$$

It is easily verified that this is indeed a norm and that $\|A\| \leq N(A)$ for an arbitrary matrix A.

Theorem 1.2. Let $A=(a_{ij})$ be an n x n matrix, then $P(A) \leq \|A\|$.

Proof: Let $\lambda \in \sigma(A)$, then there exists a non-zero column vector X such that $AX=\lambda X$. So $\|AX\| = \|\lambda X\| = |\lambda| \cdot \|X\|$, and $\frac{\|AX\|}{\|X\|} = |\lambda|$. But $\frac{\|AX\|}{\|X\|} \leq \frac{\|A\| \cdot \|X\|}{\|X\|} = \|A\|$. So $|\lambda| \leq \|A\|$ for all $\lambda \in \sigma(A)$. Thus $P(A) \leq \|A\|$. It now follows that $P(A) \leq \|A\| \leq N(A)$ or $P(A) \leq n(\max_{i,j}|a_{ij}|)$.

Thus if $N(E_0)<1$, then $P(E_0)<1$ and Newton's method will converge. But because E_0 is, in general, a full matrix with no special properties, it is very hard to find a more practical sufficient condition. So, in order to be assured that Newton's method will converge, G_0 must be a good enough approximation of A^{-1} to make the elements of E_0 uniformly small in magnitude, i.e. if $E_0=(e_{ij})$ is n x n then $|e_{ij}|<1/n$ for all $1 \leq i,j \leq n$. This condition is usually hard to satisfy, particularly if n is large, Because of this it is often impossible to use Newton's method.

Also, close examination reveals that two matrix multiplications are required per iteration, which is very time comsuming is n is large. Thus it is often impractical to

use Newton's method, even if it does converge.

CHAPTER 2

THE SUCCESSIVE OVERRELAXATION TECHNIQUE

It was shown that the Newton iteration method converges to A^{-1}, for any non-singular matrix A, as long as the initial approximation G_0 is close enough to A^{-1}. But it was pointed out that it is often very difficult to find such a G_0. Also, if n is very large, the time required for an iteration will render the method, for all practical purposes, useless. In light of this, it would be worthwhile to develop an iterative technique which is faster and does not depend on the initial approximation for its convergence.

The iterative method that will be developed in this chapter partially fulfills both of these objectives. By taking advantage of the special properties of a certain class of matrices, i.e. those matrices that arise from finite difference approximations to boundary value problems for ordinary differential equations and elliptic partial differential equations, it will be shown that this method converges or diverges independent of G_0.

It should be pointed out that matrices of this type are sparse, i.e. they have a large number of zero entries. In fact, they are usually tridiagonal or five diagonal matrices.

Also they are usually diagonally dominant. That is, the absolute value of the diagonal elements is greater than or equal to the sum of the absolute values of the other nonzero entries in that row or column.

Let $A=(a_{ij})$ be an n x n, non-singular matrix so that A^{-1} exists, then $AA^{-1}=I$. Let $A=D-E-F$ where D is a diagonal matrix, E is lower triangular and F is upper triangular. Then

$$(D-E-F)A^{-1}=I$$

$$DA^{-1}=(E+F)A^{-1}+I$$

$$A^{-1}=D^{-1}(E+F)A^{-1}+D^{-1}. \qquad (2\text{-}1)$$

This equation suggests the following iteration scheme:

$$G_{m+1}=D^{-1}(E+F)G_m+D^{-1} \quad m=0,1,\ldots, \qquad (2\text{-}2)$$

where G_0 is an arbitrary approximation to A^{-1}. This is a generalization of the point-Jacobi iteration method for solving systems of linear equations.

Theorem 2.1. If $A=(a_{ij})$ is an n x n matrix and G_m is defined by eq(2-2) for $m=0,1,\ldots$, then $\lim_{m\to\infty} G_m=A^{-1}$ if and only if $P(B)<1$ where $B=D^{-1}(E+F)$.

Proof: Let $E_m=A^{-1}-G_m$ be the error matrix for the m'th iterate. Then,

$$E_m=A^{-1}-G_m=A^{-1}-BG_{m-1}-D^{-1}$$

$$=BA^{-1}+D^{-1}-BG_{m-1}-D^{-1}$$

$$=B(A^{-1}-G_{m-1})$$

$$=BE_{m-1}.$$

So $E_m=BE_{m-1}=B^2E_{m-2}=\ldots=B^mE_0$. But $\lim_{m\to\infty} G_m=A^{-1}$ if and only if $\lim_{m\to\infty} E_m=0$. Now, $\lim_{m\to\infty} E_m=(\lim_{m\to\infty} B^m)E_0=0$ if and only if $\lim_{m\to\infty} B^m=0$, since in general $E_0\neq 0$. But $\lim_{m\to\infty} B^m=0$ if and only if $P(B)<1$.

Eq(2-2) can be rewritten as

$$g_{ij}^{(m+1)} = -\sum_{\substack{k=1\\k\neq i}}^{n}\left(\frac{a_{ik}}{a_{ii}}\right) g_{kj}^{(m)} + \frac{\delta_{ij}}{a_{ii}} \quad \text{for } i=1,\ldots, n \text{ and}$$

$j=1,\ldots, n,$ (2-3)

where δ_{ij} is the Kronecker delta. In general, all the components of G_m must be saved in order to compute the components of G_{m+1}. So it would seem intuitively attractive to replace each old component by the corresponding new one as soom as a new element is computed. This not only will reduce storage requirements for the computer, but also should produce better results. Using this procedure on eq(2-2) produces the equation

$$g_{ij}^{(m+1)} = -\sum_{k=1}^{i-1}\left(\frac{a_{ik}}{a_{ii}}\right) g_{kj}^{(m+1)} - \sum_{k=i+1}^{n}\left(\frac{a_{ik}}{a_{ii}}\right) g_{kj}^{(m)} + \frac{\delta_{ij}}{a_{ii}} \quad (2-4)$$

for i=1,..., n and j=1,..., n, or,

$$a_{ii}g_{ij}^{(m+1)} + \sum_{k=1}^{i-1} a_{ik}g_{kj}^{(m+1)} = - \sum_{k=i+1}^{n} a_{ik}g_{kj}^{(m)} + \delta_{ij}$$

for i=1,..., n and j=1,..., n. (2-5)

In the last two equations the $g_{ij}^{(m+1)}$ are to be computed by columns, left to right, in filling up the matrix G_{m+1}. In matrix notation eq(2-5) becomes

$$(D-E)G_{m+1}=FG_m+I \quad \text{or}$$

$$G_{m+1}=(D-E)^{-1}FG_m+(D-E)^{-1} \qquad m=0,1,\ldots \quad . \tag{2-6}$$

Let $L_1=(D-E)^{-1}F$, for reasons which will be apparent later, and call L_1 the Gauss-Seidel matrix. Eq(2-6) is a generalization of the Gauss-Seidel iteration method for solving systems of linear equations.

Now, in conjunction with immediate replacement, an acceleration parameter w could be introduced into the process by letting eq(2-4) define an intermediate component $\bar{g}_{ij}^{(m+1)}$ and then generating $g_{ij}^{(m+1)}$ by taking a weighted average of $\bar{g}_{ij}^{(m+1)}$ and $g_{ij}^{(m)}$. This can be written as,

$$\bar{g}_{ij}^{(m+1)} = - \sum_{k=1}^{i-1} \left(\frac{a_{ik}}{a_{ii}}\right) g_{kj}^{(m+1)} - \sum_{k=i+1}^{n} \left(\frac{a_{ik}}{a_{ii}}\right) g_{kj}^{(m)} + \frac{\delta_{ij}}{a_{ii}}$$

(2-7)

$$g_{ij}^{(m+1)} = (1-w)g_{ij}^{(m)} + w\bar{g}_{ij}^{(m+1)} . \qquad (2-8)$$

Substitution of eq(2-7) into eq (2-8) yields

$$g_{ij}^{(m+1)} = (1-w)g_{ij}^{(m)} - w \sum_{k=1}^{i-1} \left(\frac{a_{ik}}{a_{ii}} \right) g_{kj}^{(m+1)} -$$

$$w \sum_{k=i+1}^{n} \left(\frac{a_{ik}}{a_{ii}} \right) g_{kj}^{(m)} + \frac{w\delta_{ij}}{a_{ii}} \qquad (2-9)$$

or

$$a_{ii}g_{ij}^{(m+1)} + w \cdot \sum_{k=1}^{i-1} a_{ik}g_{kj}^{(m+1)} = (1-w)a_{ii}g_{ij}^{(m)} -$$

$$w \cdot \sum_{k=i+1}^{n} a_{ik}g_{kj}^{(m)} + w\delta_{ij} . \qquad (2-10)$$

In matrix notation this becomes

$$(D-wE)G_{m+1} = [(1-w)D+wF]G_m + wI \qquad m=0,1,\ldots , \qquad (2-11)$$

or, $G_{m+1} = (D-wE)^{-1}[(1-w)D+wF]G_m + w(D-wE)^{-1} \qquad m=0,1,\ldots .$ (2-12)

Let $L_w = (D-wE)^{-1}[(1-w)D+wF]$ and $K = w(D-wE)^{-1}$.

Then $G_{m+1} = L_w G_m + K \qquad m=0,1,\ldots .$ (2-13)

Note that if w=1, L_w becomes the Gauss-Seidel matrix L_1, and eq(2-12) reduces to eq(2-6). Eq(2-13) is a generalization of the successive overrelaxation iteration method for solving systems of linear equations.

Theorem 2.2. If $A=(a_{ij})$ is an n x n, non-singular matrix

and G_m is defined by eq(2-13) for m=0,1,... then $\lim_{m\to\infty} G_m = A^{-1}$ if and only if $P(L_w)<1$.

The proof of this theorem is completely analogous to the proof of theorem 2.1 and will, therefore, be omitted.

Now, the obvious question at this point is, does eq(2-13) always converge when eq(2-2) does and vice versa? To answer this question the following theorem is presented without proof [1,70].

Theorem 2.3. (Stein-Rosenberg) Let $A=(a_{ij})$ be an n x n non-singular matrix and let A=D-E-F, where D is a diagonal matrix, E is lower triangular, and F is upper triangular. Suppose $B=D^{-1}(E+F)$ is non-negative with zero diagonal entries, and let $L_1=(D+E)^{-1}F$. Then one and only one of the following mutually exclusive relations is valid:

1. $P(B)=P(L_1)=0$
2. $0<P(L_1)<P(B)<1$
3. $1=P(B)=P(L_1)$
4. $1<P(B)<P(L_1)$.

So the iteration techniques of eq(2-2) and eq(2-6) are either both convergent or both divergent, and when they both converge $P(L_1)<P(B)$. Thus it seems clear that the method of eq(2-6) should converge iteratively faster, i.e. more convergence per iteration than the method of eq(2-2), when they converge.

The reason that the acceleration parameter w was introduced into eq(2-6) was so that it could be chosen in such a manner that $P(L_w)$ would be minimized. This choice of w would give an optimal convergence rate for the method of eq(2-13). It would seem that the task of finding the w_b such that $P(L_{w_b}) \leq P(L_w)$ for all real w might be difficult. However, it can be shown, [1.110], that w_b is a known function of P(B) and that in many cases good estimates of P(B) can be determined. In any case, the following theorem due to Ostrowski will help to locate w_b [1,77].

Theorem 2.4. Let $A=D-E-E^*$ be an n x n hermitian matrix where D is hermitian and positive definite and D-wE is nonsingular for $0 \leq w \leq 2$. Then $P(L_w)<1$ if and only if A is positive definite and $0<w<2$.

It is the experience of the author that it is usually not hard to find a value for w such that eq(2-13) converges faster than eq(2-13) converges faster than eq(2-6).

A few concluding remarks are in order. As promised, the iteration technique of this chapter does not depend on the initial approximation for its convergence, but depends only on the matrix to be inverted. Also only one matrix multiplication is required per iteration for eq(2-13) as compared to two for Newton's method, thus making the time for an iteration significantly less.

SUMMARY

The appendix contains an example derived from a finite difference approximation to an ordinary differential equation. The author realizes that this example is not as good a test as should be used, but was forced into this small scale problem by limitations imposed by a small computer. None the less, this example points out the potential of the techniques of Chapter 2.

In order to compare the method of eq(2-13) with Newton's method the same initial approximations were used for both methods, with G_0 being chosen, in each case, so that $P(E_0)<1$. It was found that Newton's method would do very little converging for a few iterations and then converge very rapidly. On the other hand, the method of eq(2-13) converged at a more uniform rate. It is the author's conjecture that Newton's method requires the components of the error matrix to be reasonably uniform in magnitude before it begins to converge rapidly.

In all test cases the method of eq(2-13) performed iterations nearly twice as fast as Newton's method and took less time to reach a specified degree of accuracy. But perhaps best results could be obtained by using eq(2-13) for a few iterations and then switching to Newton's methods, to

take advantage of the higher order convergence.

BIBLIOGRAPHY

1. Varga, R.S. [1962], Matrix Iterative Analysis, Prentice-Hall Inc., Englewood Cliffs, New Jersey.

2. Faddeev, D.K. and Faddeeva, V.N. [1963], Computational Methods of Linear Algebra, W.H. Freeman Co., San Francisco.

3. Isaacson, E. and Keller, H.B. [1966], Analysis of Numerical Methods, John Wiley & Sons, Inc., New York.

4. Todd, J. [1962], Survey of Numerical Analysis, Magraw-Hill Book Co. Inc., New York.

5. Froberg, C.E. [1965], Introduction to Numerical Analysis, Addison-Wesley Co. Inc., Reading, Mass.

6. John, F. [1967], Lectures on Advanced Numerical Analysis, Gordan and Breach Science Publishers, New York.

APPENDIX

Example: Take the following boundary value problem,

$$-Y'' + x^2Y = f(x) \qquad 0 \leq x \leq 1 \qquad Y(0) = Y(1) = 0.$$

Divide [0,1] by n-1 points x_i, into n equal intervals, with $x_i - x_{i-1} = h$. Denote $Y(x_i)$ by Y_i, then approximate $-Y_i''$ by $\frac{2Y_i - Y_{i+1} - Y_{i-1}}{h^2}$. This will lead to the following matrix equation $A\bar{Y} = h^2\bar{F}$ where

$$A = \begin{bmatrix} S_1 & -1 & 0 & \cdots & 0 \\ -1 & S_2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & S_3 & -1 & \cdots & 0 \\ \vdots & & & & & \vdots \\ & & & & & -1 \\ 0 & \cdots & & & 0 & -1 & S_{n-1} \end{bmatrix} \quad \text{with } S_i = 2 + x_i^2h^2$$

$$\bar{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_{n-1} \end{bmatrix} \quad \text{and} \quad \bar{F} = \begin{bmatrix} F_1 \\ \vdots \\ F_{n-1} \end{bmatrix}$$

It is the matrix A of which the inverse is to be found. It will not be shown here, but A has special properties which

insure that $P(B)<1$, in eq(2-2) [1,164]. Thus the iteration scheme of eq(2-13) will converge. However, in order that the process of eq(2-13) might be compared with Newton's method, a G_0 must be found so that $P(E_0)<1$.

Theorem: If A is a non-singular, n x n matrix and if $G_0 = aA^*$ with $a = 1/(Tr(AA^*))$, then $P(E_0)<1$, where

$$Tr(A) = \sum_{i-1}^{n} a_{ii}.$$

Proof is trivial and is omitted [3,84].

Three programs and their selected output are listed. The following is a summary of each program.

Program 1: The output is not listed, but this program produces the matrix G_0 as defined in the above theorem. This was done for n=3,4,9,19.

Program 2: The output of program one is used as input for this program. It performs the iterations prescribed by Newton's method. At the end of each iteration it prints the iteration number m, along with $M(E_m)$ where $E_m = I-AG_m$ and $M(E_m) = \frac{1}{n}\left(\max_{1\leq j\leq n} \sum_{i=1}^{n} |e_{ij}^{(m)}|\right)$ is taken as the norm.

Program 3: It uses the same input as program two and performs the iterations specified in eq(2-13). The output is in the same format.

Also listed in the output of programs two and three,

although not direct output of either, is the time, in seconds, required to carry out an iteration and the total time, i.e. (number of iterations) x (time required per iteration).

```
      PROGRAM NO.1 (CALCULATES INITIAL APPROXIMATIONS)
      DIMENSION A(40,40)
    1 READ 2,N
    2 FORMAT(I3)
      H=N
      H=1./H
      NM1=N-1
      NM2=N-2
      DO 5 I=1,NM1
      DO 5 J=1,NM1
    5 A(I,J)=0.0
      SUM=0.0
      DO 10 I=I,NM1
      XI=I
      XI=XI*H
      S=2.+(XI*H)**2
      SUM=SUM+S**2
   10 A(I,I)=S
      SO 15 I=2,NM1
      A(I,I-1)=-1.
   15 A(I-1,I)=-1.
      FL=2*NM1-2
      SUM=SUM+FL
      SUM=1./SUM
      DO 20 I=1,NM1
      DO 20 J=1,NM1
   20 A(I,J)=SUM*A(I,J)
      DO25 I=1,NM1
   25 PUNCH 30,(A(I,J),J=1,NM1)
   30 FORMAT(5F16.8)
      GO TO 1
      END
```

```
   PROGRAM NO.2 (NEWTON ITERATION SCHEME)
   DIMENSION G(20,20),AG(20,20),STOR(20)
 1 READ 2,N
 2 FORMAT(I3)
   READ 3,CLOS
 3 FORMAT(F10.8)
   H=N
   H=1./H
   NM1=N-1
   NM2=N-2
   PUNCH 4
 4 FORMAT(9HITERATION ,5X,20HNORM OF ERROR MATRIX,/)
   ISW=0
 5 DO 6 I=1,NM1
 6 READ 7,(G(I,J),J=1,NM1)
 7 FORMAT(5F16.8)
 8 Z=NM1
   DO 10 J=1,NM1
   AG(1,J)=G(2,J)-(2.+H**4)*G(1,J)
10 AG(NM1,J)=G(NM2,J)-(2.+Z**2*H**4)*G(NM1,J)
   DO 15 I=2,NM2
   DO 15 J=1,NM1
   XI=I
   XI=XI*H
   S=2.+(XI*H)**2
15 AG(I,J)=G(I-1,J)+G(I+1,J)-S*G(I,J)
   DO 20 I=1,NM1
20 AG(I,I)=AG(I,I)+1.
   IF(SENSE SWITCH 1)21,22
21 PAUSE
22 DO 25 J=1,NM1
   STOR(J)=0.0
   DO 25 I=1,NM1
25 STOR(J)=STOR(J)+ABSF(AG(I,J))
   BIG=STOR(1)
   DO 30 J=2,NM1
   IF(BIG-STOR(J))28,30,30
28 BIG=STOR(J)
30 CONTINUE
   BIG=BIG/Z
   PUNCH 35,ISW,BIG
35 FORMAT(3X,I3,10X,E14.8)
   IF(SENSE SWITCH 1)33,34
33 PAUSE
34 IF(BIG-CLOS)1,1,36
36 DO 37 I=1,NM1
37 AG(I,I)=AG(I,I)+1.
```

```
      DO 40 I=1,NM1
      DO 40 J=1,NM1
      T=0.0
      DO 39 KT=1,NM1
   39 T=T+G(I,KT)*AG(KT,J)
   40 G(I,J)=T
      ISW=ISW+1
      GO TO 8
      END
```

```
OUTPUT (NEWTON ITERATION SCHEME)
N=3
TIME=4 SEC
TOTAL TIME=32 SEC

ITERATION      NORM OF ERROR MATRIX

      0             .37458976E+00
      1             .37569636E+00
      2             .35958460E+00
      3             .31733761E+00
      4             .23635084E+00
      5             .12055866E+00
      6             .28100900E-01
      7             .15570500E-02
      8             .35233333E-05


N=4
TIME=8 SEC
TOTAL TIME=80 SEC

ITERATION      NORM OF ERROR MATRIX

      0             .28383015E+00
      1             .28443630E+00
      2             .27676670E+00
      3             .27223217E+00
      4             .25039517E+00
      5             .21139401E+00
      6             .13673536E+00
      7             .49057062E-01
      8             .49237400E-02
      9             .36140000E-04
     10             .12750000E-06


N=9
TIME=65 SEC
TOTAL TIME=910 SEC

ITERATION      NORM OF ERROR MATRIX

      0             .11963981E+00
      1             .12399858E+00
      2             .12611528E+00
      3             .12445198E+00
```

4	.12869220E+00
5	.13076338E+00
6	.13148124E+00
7	.13163276E+00
8	.12722893E+00
9	.10997794E+00
10	.74386558E-01
11	.28744808E-01
12	.33295055E-02
13	.30195555E-04
14	.27777777E-06

N=19
TIME=472 SEC
TOTAL TIME=8024 SEC

ITERATION	NORM OF ERROR MATRIX
0	.54510247E-01
1	.55948873E-01
2	.57806963E-01
3	.59050800E-01
4	.58978163E-01
5	.60845178E-01
6	.62431621E-01
7	.63990100E-01
8	.64941710E-01
9	.66709115E-01
10	.67830710E-01
11	.65780700E-01
12	.64038321E-01
13	.56260200E-01
14	.37891149E-01
15	.14200916E-01
16	.14423831E-02
17	.93547368E-05

```
      PROGRAM NO. 3 (ITERATION SCHEME OF EQ(2-13))
      DIMENSION G(20,20),AG(20,20),STOR(20)
    1 READ 2,N
    2 FORMAT(I3)
      READ 3,CLOS
    3 FORMAT(F10.8)
      READ 4,W
    4 FORMAT(F9.6)
      H=N
      H=1./H
      ISW=0
      NM1=N-1
      NM2=N-2
      PUNCH 8
    8 FORMAT(9HITERATION,5X,20HNORM OF ERROR MATRIX/)
      DO 5 I=1,NM1
    5 READ 6,(G(I,J),J=1,NM1)
    6 FORMAT(5F16.8)
    9 Z=NM1
      DO 10 J=1,NM1
      AG(1,J)=G(2,J)-(2.+H**4)*G(1,J)
   10 AG(NM1,J0=G(NM2,J0-(2.+Z**2*H**4)*G(NM1,J)
      DO 15 I=2,NM2
      SO 15 J=1,NM1
      XI=I
      XI=XI*H
      S=2.+(XI*H)**2
   15 AG(I,J)=G(I+1,J)+G(I-1,J)-S*G(I,J)
      DO 20 I=1,NM1
   20 AG(I,I)=AG(I,I)+1.
      DO 25 J=1,NM1
      STOR(J)=0.0
      DO 25 I=1,NM1
   25 STOR(J)=STOR(J)+ABSF(AG(I,J))
      BIG=STOR(1)
      DO 30 I=2,NM1
      IF(BIG-STOR(I))28,30,30
   28 BIG=STOR(I)
   30 CONTINUE
      BIG=BIG/Z
      PUNCH 35,ISW,BIG
   35 FORMAT(3X,I3,10X,E14.8)
      IF(SENSE SWITCH 1)36,37
   36 PAUSE
   37 IF(BIG-CLOS)1,1,38
   38 DO 50 J=1,NM1
      T=G(2,J)/(2.+H**4)
      IF(J-1)40,39,40
```

```
39 T=T+1./(2.+H**4)
40 G(1,J)=(1.-W)*G(1,J)+W*T
   DO 45 I=2,NM2
   XI=I
   XI=XI*H
   S=2.+(XI*H)**2
   T=(G(I-1,J)+G(I+1,J))/S
42 T=T+1./S
45 G(I,J)=(1.-W)*G(I,J)+W*T
   S=2.+Z**2*H**4
   T=G(NM2,J)/S
   IF(J-NM1)47,46,47
46 T=T+1./S
47 G(NM1,J)-(1.-W)*G(NM1,J)+W*T
50 CONTINUE
   ISW=ISW+1
   IF(SENSE SWITCH 1)51,9
51 PAUSE
   GO TO 9
   END
```

OUTPUT (ITERATION SCHEME OF EQ(2-13))
N=3
W=1.17
TIME=3 SEC
TOTAL TIME=24 SEC

ITERATION	NORM OF ERROR MATRIX
0	.37458976E+00
1	.26455359E+00
2	.10052566E+00
3	.23602453E-01
4	.55798833E-02
5	.10484000E-02
6	.21687666E-03
7	.35586666E-04
8	.69266666E-05

N=4
W=1.25
TIME=5.2 SEC
TOTAL TIME=57.2 SEC

ITERATION	NORM OF ERROR MATRIX
0	.28383015E+00
1	.24335178E+00
2	.14898773E+00
3	.59841937E-01
4	.19431190E-01
5	.72220500E-02
6	.22082350E-02
7	.71759750E-03
8	.23440750E-03
9	.70127500E-04
10	.22407500E-04
11	.68975000E-05

N=9
W=1.525
TIME=25 SEC
TOTAL TIME=525 SEC

ITERATION	NORM OF ERROR MATRIX
0	.11963981E+00
1	.14078226E+00
2	.11232760E+00
3	.97341346E-01
4	.81115827E-01
5	.62447542E-01
6	.45164391E-01
7	.30423686E-01
8	.17174392E-01
9	.71125233E-02
10	.49689344E-02
11	.29644733E-02
12	.16702388E-02
13	.94705000E-03
14	.51271333E-03
15	.25750333E-03
16	.15433333E-03
17	.87398888E-04
18	.49837777E-04
19	.23220000E-04
20	.13006666E-04
21	.70166666E-05

N=19
W=1.724
TIME=106 SEC
TOTAL TIME=4452 SEC

ITERATION	NORM OF ERROR MATRIX
0	.54510247E-01
1	.82655489E-01
2	.71076631E-01
3	.61784473E-01
4	.53859926E-01
5	.51234519E-01
6	.48399633E-01
7	.45072432E-01
8	.41399411E-01
9	.37453810E-01
10	.33503374E-01
11	.29602665E-01
12	.25798135E-01
13	.22127237E-01
14	.18617649E-01
15	.15287269E-01
16	.12144041E-01
17	.92355736E-02
18	.63247510E-02
19	.33781968E-02
20	.28641078E-02
21	.22881000E-02
22	.17985652E-02
23	.13984789E-02
24	.10729542E-02
25	.82034000E-03
26	.63016157E-03
27	.48294263E-03
28	.36869526E-03
29	.27533842E-03
30	.20081684E-03
31	.14606684E-03
32	.11829368E-03
33	.93737368E-04
34	.72193157E-04
35	.53774736E-04
36	.41087894E-04
37	.31515789E-04
38	.23650526E-04
39	.16802105E-04

40	.13177368E-04
41	.10048421E-04
42	.76084210E-05