COMPUTER PROJECT NO: ONE

SOLVING LINEAR SYSTEMS USING HOUSEHOLDER TRANSFORMATIONS

CPT S 531 / MATH 544

INSTRUCTOR: DR. ALAN GENZ

10/10 Good work

BEN M CHEN

60883883

10-01-88

CPT S 531/MATH 544 Computer Project 1

Due 10/10/88

Write a subroutine called HSOLVE to "solve" a linear system AX = B using Householder transformations. The subroutine should take as input integers M and N (M ≥ N), an MxN matrix A, an M-vector B and an integer JOB. If JOB = 0 then HSOLVE should reduce A to upper triangular form using a sequence of N Householder transformations, overwriting A and saving the betas and diagonal elements in vectors BETA and R (see pages 148-149 in the text). If JOB = 1 then HSOLVE should reduce A, transform B using the same sequence of Householder transformations that were used for A and solve for X using the first N components of the transformed B. If JOB = 2 then HSOLVE should assume that A has already been transformed using a previous call of HSOLVE and use the entries in A, R and BETA to transform B and solve for X. When JOB = 1 or 2 HSOLVE should return the solution in an output N-vector output parameter X. HSOLVE should begin like:

```
SUBROUTINE HSOLVE(LDA, A, M, N, B, BETA, R, X, JOB)
INTEGER LDA, M, N, JOB
REAL A(LDA,N), B(M), BETA(N), R(N), X(N)
```

HSOLVE should be clearly written with good structure and appropriate comments.

Use your subroutine in a main program to:

a) solve the Hilbert system $H^{(n)}x = b^{(n)}$, where $h_{ij}^{(n)} = 1/(i+j-1)$ and $b_i^{(n)} = \sum_{j=1}^{n} 1/(i+j-1)$, for n = 2, 3, 4, 5, 6,

b) solve the system:

$$\begin{bmatrix} 1 & -3 & 9 & -27 \\ 1 & -2 & 4 & -8 \\ 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \end{bmatrix} x = b,$$

for b^T = (13, 7, 3, 1, 1, 3, 7) and (10.57, 6.68, 2.99, 1, 1.01, 3.32, 9.43).

c) for n = 10, 20, 40: generate 100 random nxn matrices and use HSOLVE to compute their solutions when the right-hand side b vectors are produced using true solutions x^T = (1, 1, ..., 1). Determine the average ∞ norm error for each n, and compare this with the average error obtained using the LINPACK subroutines SGEFA and SGESL. The entries in the matrices should be uniform random numbers from [-1, 1].

The program listing and relelvant output should be handed in, along with a brief discussion of the results that you obtained.

```
C     ------------------------------------------------------------------PRO00010
C     Computer Project 1:     CPT S 531/MATH 544          Ben M. Chen PRO00020
C     ------------------------------------------------------------------PRO00030
      DIMENSION A(40,40),B(40),BETA(40),X(40),R(40),C(40,40),D(40)      PRO00040
      DIMENSION IPVT(40)                                                PRO00050
      INTEGER  LDA, M, N, JOB, INFO, IPVT                               PRO00060
      REAL AVERRH, AVERRL, MAX                                          PRO00070
C     ------------------------------------------------------------------ PRO00080
C     Parameters and variables in main program:                         PRO00090
C         Refer the subroutine HSOLVE for more information about the    PRO00100
C         parameters and variables, except                              PRO00110
C                   C  -- copy of A & used in Part C for comparision    PRO00120
C                   D  -- copy of B & used in Part C for comparision    PRO00130
C                INFO  -- creat by LINPACK subroutine.                  PRO00140
C                IPVT  -- pivoting information created by LINPACK.      PRO00150
C                 MAX  -- maximun value of a vector.                    PRO00160
C              AVERRH  -- average infinite norm error from HSOLVE.      PRO00170
C              AVERRL  -- average infinite norm error from LINPACK.     PRO00180
C              VNI(0)  -- random number uniformly in (-1, 1)            PRO00190
C     ------------------------------------------------------------------PRO00200
C     Part A: Using HSOLVE to solve the Hilbert system  Hx=b for        PRO00210
C     n=2,3,4,5,6                                                       PRO00220
C     ------------------------------------------------------------------PRO00230
      DO 440 I=2,6                                                      PRO00240
           DO 420 J=1,I                                                 PRO00250
                B(J)=0.                                                 PRO00260
                DO 410 K=1,I                                            PRO00270
                     A(J,K)=1./(J+K-1)                                  PRO00280
                     B(J)=B(J)+A(J,K)                                   PRO00290
410             CONTINUE                                                PRO00300
420        CONTINUE                                                     PRO00310
           LDA=40                                                       PRO00320
           M=I                                                          PRO00330
           N=I                                                          PRO00340
           JOB=1                                                        PRO00350
           CALL HSOLVE(LDA,A,M,N,B,BETA,R,X,JOB)                        PRO00360
C     ------------------------------------------------------------------PRO00370
C          * * *  P R I N T I N G  * * *                                PRO00380
C     ------------------------------------------------------------------PRO00390
           WRITE(6,*) ' '                                               PRO00400
           WRITE(6,*)'Solution For Hilbert System; N=',N                PRO00410
           WRITE(6,*)' '                                                PRO00420
           DO 430 J=1,N                                                 PRO00430
                WRITE(6,*) X(J)                                         PRO00440
430        CONTINUE                                                     PRO00450
440   CONTINUE                                                          PRO00460
C     ------------------------------------------------------------------ PRO00470
C     Part B: Using HSOLVE to solve the given systems.                  PRO00480
C     ------------------------------------------------------------------ PRO00490
      M=7                                                               PRO00500
      N=4                                                               PRO00510
      WRITE(6,*) ' '                                                    PRO00520
      WRITE(6,*)'Reading the matrix A by row-wise.'                     PRO00530
      WRITE(6,*)' '                                                     PRO00540
           READ *, ((A(I,J),J=1,N),I=1,M)                               PRO00550
```

```
      WRITE(6,*) ' '                                                    PR000560
      WRITE(6,*)'Reading the first vector  B:'                          PR000570
      WRITE(6,*)' '                                                     PR000580
           READ *, (B(I),I=1,M)                                         PR000590
      JOB=1                                                             PR000600
      CALL HSOLVE(LDA,A,M,N,B,BETA,R,X,JOB)                             PR000610
C     ------------------------------------------------------------------PR000620
C          * * *   P R I N T I N G   * * *                              PR000630
C     ------------------------------------------------------------------PR000640
      WRITE(6,*)' '                                                     PR000650
      WRITE(6,*)'Solution For The First System of Part B:  X='          PR000660
      WRITE(6,*)' '                                                     PR000670
      DO 510 I=1,N                                                      PR000680
           WRITE(6,*) X(I)                                              PR000690
510   CONTINUE                                                          PR000700
C     ------------------------------------------------------------------PR000710
      WRITE(6,*) ' '                                                    PR000720
      WRITE(6,*)'Reading the second vector  B:'                         PR000730
           READ *, (B(I),I=1,M)                                         PR000740
      JOB=2                                                             PR000750
      CALL HSOLVE(LDA,A,M,N,B,BETA,R,X,JOB)                             PR000760
C     ------------------------------------------------------------------PR000770
C          * * *  P R I N T I N G  * * *                               PR000780
C     ------------------------------------------------------------------PR000790
      WRITE(6,*) ' '                                                    PR000800
      WRITE(6,*)'Solution For the Second System of Part B:  X='         PR000810
      WRITE(6,*) ' '                                                    PR000820
      DO 520 I=1,N                                                      PR000830
           WRITE(6,*) X(I)                                              PR000840
520   CONTINUE                                                          PR000850
C     ------------------------------------------------------------------PR000860
C     Part C: Using HSOLVE to determine the average infinite norm       PR000870
C             errors and comparing to the average errors obtained       PR000880
C             from LINPACK subroutine SGEFA and SGESL.                  PR000890
C                                                                       PR000900
C     *  Because both HSOLVE and SGEFA overwrite the matix A and B,     PR000910
C        we introduce  C  &  D  to momerize  A  &  B, respectively.     PR000920
C     ------------------------------------------------------------------ PR000930
C                                                                       PR000940
      DO 660 I=0,2                                                      PR000950
           N=(2**I)*10                                                  PR000960
           M=N                                                          PR000970
           AVERRH=0.                                                    PR000980
           AVERRL=0.                                                    PR000990
           DO 650 I1=1,100                                              PR001000
                DO 620 J=1,N                                            PR001010
                     B(J)=0.                                            PR001020
                     DO 610 K=1,N                                       PR001030
C     ------------------------------------------------------------------PR001040
C      * * * C R E A T I N G   R A N D O M   M A T R I C E S  * * *     PR001050
C     ------------------------------------------------------------------PR001060
                          A(J,K)=VNI(0)                                 PR001070
                          C(J,K)=A(J,K)                                 PR001080
                          B(J)=B(J)+A(J,K)                              PR001090
610                  CONTINUE                                           PR001100
```

```
                    D(J)=B(J)                                              PRO01110
620             CONTINUE                                                   PRO01120
                JOB=1                                                      PRO01130
                CALL HSOLVE(LDA,A,M,N,B,BETA,R,X,JOB)                      PRO01140
C       ---------------------------------------------------------------    PRO01150
C       Finding the infinite norm error in HSOLVE.                         PRO01160
C       ---------------------------------------------------------------    PRO01170
                MAX=0.                                                     PRO01180
                DO 630 K=1,N                                               PRO01190
                    IF (ABS(X(K)-1.).GT.MAX) MAX=ABS(X(K)-1.)              PRO01200
630             CONTINUE                                                   PRO01210
                AVERRH=AVERRH+MAX/100.                                     PRO01220
C       ---------------------------------------------------------------    PRO01230
                CALL SGEFA(C,LDA,N,IPVT,INFO)                              PRO01240
                CALL SGESL(C,LDA,N,IPVT,D,0)                               PRO01250
C       ---------------------------------------------------------------    PRO01260
C       Finding the infinte norm error in LINPACK.                         PRO01270
C       ---------------------------------------------------------------    PRO01280
                MAX=0.                                                     PRO01290
                DO 640 K=1,N                                               PRO01300
                    IF (ABS(D(K)-1.).GT.MAX) MAX=ABS(D(K)-1.)              PRO01310
640             CONTINUE                                                   PRO01320
                AVERRL=AVERRL+MAX/100.                                     PRO01330
C       ---------------------------------------------------------------    PRO01340
650         CONTINUE                                                       PRO01350
C       ---------------------------------------------------------------    PRO01360
C           * * *  P R I N T I N G  * * *                                  PRO01370
C       ---------------------------------------------------------------    PRO01380
            WRITE(6,*) ' '                                                 PRO01390
            WRITE(6,*)'For N=',N,',the HSOLVE average norm error = '       PRO01400
            WRITE(6,*) ' '                                                 PRO01410
            WRITE(6,*) AVERRH                                              PRO01420
            WRITE(6,*) ' '                                                 PRO01430
            WRITE(6,*)'For N=',N,',the LINPACK average norm error = '      PRO01440
            WRITE(6,*) ' '                                                 PRO01450
            WRITE(6,*) AVERRL                                              PRO01460
660     CONTINUE                                                           PRO01470
        STOP                                                               PRO01480
        END                                                                PRO01490
C                                                                          PRO01500
C       ---------------------------------------------------------------    PRO01510
C       This subroutine, named HSOLVE, is written for solving a linear     PRO01520
C       system  AX=B  using Householder transformations,  where  A  is     PRO01530
C       an  M-by-N  matrix  and  B  is an  M-vector.                       PRO01540
C       ---------------------------------------------------------------    PRO01550
C       Parameters & varibles in this subroutine:                          PRO01560
C            LDA   --   leading dimesion of the matrix  A .                PRO01570
C            A     --   an  M-by-N  matrix.                                PRO01580
C            M     --   number of rows in matrix  A  and  vector  B.       PRO01590
C            N     --   number of columns in matrix  A .                   PRO01600
C            B     --   column vector with  M  elements.                   PRO01610
C            BETA  --   storing the betas in Householder transformation    PRO01620
C            R     --   storing the diagonal elements of trasformed  A     PRO01630
C            X     --   storing the solution of the linear system.         PRO01640
C            JOB   --   JOB=0, HSOLVE only reduces matrix  A .             PRO01650
```

```
C                    JOB=1, HSOLVE reduce A and solve for X          PR001660
C                    JOB=2, HSOLVE assumes that A has been reduced,  PR001670
C                           and return the solution in X.            PR001680
C     ------------------------------------------------------------------PR001690
C                                                                      PR001700
      SUBROUTINE HSOLVE(LDA,A,M,N,B,BETA,R,X,JOB)                      PR001710
      INTEGER LDA,M,N,JOB                                              PR001720
      REAL A(LDA,N),B(M),BETA(N),R(N),X(N),ALFA,MAX,SIG                PR001730
      JOB=JOB-1                                                        PR001740
      IF (JOB) 100,100,200                                             PR001750
C     ------------------------------------------------------------------PR001760
C     Following is the program to reduce matrix  A  to a upper tran-   PR001770
C     gular form using a sequence of  N  householder transformations.  PR001780
C                                                                      PR001790
C     This for JOB=0 and JOB=1                                         PR001800
C     ------------------------------------------------------------------PR001810
100   DO 35 I=1,N                                                      PR001820
          MAX=0.                                                       PR001830
          DO 10 J=I,M                                                  PR001840
               IF (ABS(A(J,I)).GT.MAX) MAX=ABS(A(J,I))                 PR001850
10        CONTINUE                                                     PR001860
          ALFA=0.                                                      PR001870
          DO 15 J=I,M                                                  PR001880
               A(J,I)=A(J,I)/MAX                                       PR001890
               ALFA=ALFA+A(J,I)**2                                     PR001900
15        CONTINUE                                                     PR001910
          ALFA=SQRT(ALFA)                                              PR001920
          BETA(I)=1./(ALFA*(ALFA+ABS(A(I,I))))                         PR001930
          IF (A(I,I).LT.0.) THEN                                       PR001940
               SIG=-1.                                                 PR001950
               ELSE                                                    PR001960
                    SIG=1.                                             PR001970
          ENDIF                                                        PR001980
          A(I,I)=A(I,I)+ SIG*ALFA                                      PR001990
          R(I)=- SIG*ALFA*MAX                                          PR002000
          DO 30 K=I+1,N                                                PR002010
               ALFA=0.                                                 PR002020
               DO 20 J=I,M                                             PR002030
                    ALFA=ALFA+A(J,I)*A(J,K)                            PR002040
20             CONTINUE                                                PR002050
               ALFA=ALFA*BETA(I)                                       PR002060
               DO 25 J=I,M                                             PR002070
                    A(J,K)=A(J,K)-ALFA*A(J,I)                          PR002080
25             CONTINUE                                                PR002090
30        CONTINUE                                                     PR002100
35    CONTINUE                                                         PR002110
      IF (JOB.EQ.-1) GOTO 300                                          PR002120
C     ------------------------------------------------------------------ PR002130
C     Following program is to transform  B  using the same sequences   PR002140
C     of Householder transformations that were used for  A  .          PR002150
C                                                                      PR002160
C     This is for JOB=1 and JOB=2.                                     PR002170
C     ------------------------------------------------------------------ PR002180
200   DO 50 I=1,N                                                      PR002190
          ALFA=0.                                                      PR002200
```

```
            DO 40 J=I,M                                                    PRO02210
                 ALFA=ALFA+A(J,I)*B(J)                                     PRO02220
40          CONTINUE                                                       PRO02230
            DO 45 J=I,M                                                    PRO02240
                 B(J)=B(J)-ALFA*BETA(I)*A(J,I)                             PRO02250
45          CONTINUE                                                       PRO02260
50        CONTINUE                                                         PRO02270
C         ---------------------------------------------------------------- PRO02280
C         Solving for X. (Note: A has been reduced in upper trangular.)    PRO02290
C         ---------------------------------------------------------------- PRO02300
          DO 60 I=N,1,-1                                                   PRO02310
               X(I)=B(I)                                                   PRO02320
               DO 55 J=I+1,N                                               PRO02330
                    X(I)=X(I)-A(I,J)*X(J)                                  PRO02340
55             CONTINUE                                                    PRO02350
               X(I)=X(I)/R(I)                                              PRO02360
60        CONTINUE                                                         PRO02370
300       RETURN                                                           PRO02380
          END                                                              PRO02390
                                                                           PRO02400
```

1. Results For Part A: Hilbert Systems

Solution For Hilbert System; N = 2

0.999995708
1.00000477

Solution For Hilbert System; N = 3

1.00000763
0.999951303
1.00004578

Solution For Hilbert System; N = 4

0.999977589
1.00019169
0.999580860
1.00025749

Solution For Hilbert System; N = 5

0.999479711
1.00935936
0.960700333
1.05814934
0.972006679

Solution For Hilbert System; N = 6

0.999503314
1.01546097
0.894065619
1.27455330
0.699263990
1.11751842

2. Results For Part B: Solving The Given Systems

Solution For The First System of Part B: X =

1.00000000
-1.00000286
0.999998808
0.446724812E-06

Solution For the Second System of Part B: X =

0.999998152
-1.24952698
0.999998212
0.116667390

3. Results For Part C: Determine The Infinite Norm Errors

For N = 10 ,the HSOLVE average norm error =

0.572258868E-04

For N = 10 ,the LINPACK average norm error =

0.288394076E-04

For N = 20 ,the HSOLVE average norm error =

0.967520464E-04

For N = 20 ,the LINPACK average norm error =

0.517345616E-04

For N = 40 ,the HSOLVE average norm error =

0.129582477E-02

For N = 40 ,the LINPACK average norm error =

0.365520362E-03

DISCUSSION:

This program is written and run on CMS in FORTRAN.

Although we have 10 digits in each number we obtained, it is easy to check that only the first seven digits are useful, that is t=7.

From the results we obtained, we see that:

1. In Part A, the true solution for any Hilbert system must be the form of {1,1,...,1}'. In our solution, however, only the lower order systems (n=2,3,4) have the close results. For the higher order system, say n=6, the results are useless.
2. The computer gave quite nice solutions for the system given in Part B.
3. For Part C, we see that the subroutines in LINPACK gave us a little better results than that we obtained from HSOLVE (the average infinite norm errors from HSOLVE are about twice much the average infinite norm errors form LINPACK. Also, we note that the average infinite norm errors are increasing while the order of system increased.

Ben M Chen

Oct. 1, 88

Finding Eigenvalus and Eigenvectors of an Symmetric Matrix

by Using Serial Jacobi Algorithm

Ben M Chen

10/10
Good work

COMPUTER PROJECT NO: 2

Computer Science 531 / Mathematics 544

November 25, 1988

MATH 544/CPT S 531

Homework 6: Due 11/11/88: 7.1-3, 7.1-8, 7.2-5, 7.3-3, 7.3-4, 7.4-3, 7.4-8

Computer Project 2: Due 11/30/87

Write a subroutine called JACOBI to find all of the eigenvalues of an NxN symmetric matrix A, using the algorithm given in the text on page 299. The subroutine should have for input parameters, LDA (the actual leading dimension of A and an output matrix U), N, A and DELTA (required accuracy). The output should be a vector EIGENS of eigenvalues, and a matrix U that has the eigenvectors of A for its columns. An additional output parameter COUNT, should give the number of iterations that were required for convergence. The matrix A should be overwritten during the calculations.

Test your subroutine with DELTA = 0.00001 using 20 randomly generated matrices for N = 10, 20, 40. For each N you should compute the average number of flops (using COUNT) for convergence, and the average Frobenius norm of AU - UD, where D is a diagonal matrix containing the computed eigenvalues of A. The entries in the random symmetric matrices should be taken from the interval [-1,1].

```
C       ------------------------------------------------------------------JAC00010
C        COMPUTER PROJECT NO:2     CPT S 531/MATH 544       BEN M CHEN  JAC00020
C       ------------------------------------------------------------------JAC00030
        DIMENSION A(40,40),U(40,40),D(40,40),EIGENS(40)                   JAC00040
        INTEGER LDA,N,COUNT,P,AFLOPS                                      JAC00050
        REAL FNORM,FNORM1,DELTA                                           JAC00060
        LDA=40                                                            JAC00070
        DELTA=0.00001                                                     JAC00080
        DO 160 K=0,2                                                      JAC00090
             N=(2**K)*10                                                  JAC00100
             FNORM=0.                                                     JAC00110
             AFLOPS=0                                                     JAC00120
             DO 150 L=1,20                                                JAC00130
                  DO 110 I=1,N                                            JAC00140
                       DO 100 J=I,N                                       JAC00150
                            A(I,J)=VNI(0)                                 JAC00160
                            A(J,I)=A(I,J)                                 JAC00170
C                                                                         JAC00180
                            D(I,J)=A(I,J)                                 JAC00190
                            D(J,I)=A(J,I)                                 JAC00200
100                    CONTINUE                                           JAC00210
110                 CONTINUE                                              JAC00220
                    CALL JACOBI(LDA,D,N,DELTA,EIGENS,U,COUNT)             JAC00230
                    AFLOPS=AFLOPS+COUNT*4*N*N*N                           JAC00240
                    FNORM1=0.                                             JAC00250
                    DO 140 I=1,N                                          JAC00260
                         DO 130 J=1,N                                     JAC00270
                              D(I,J)=0.                                   JAC00280
                              DO 120 P=1,N                                JAC00290
                                   D(I,J)=D(I,J)+A(I,P)*U(P,J)            JAC00300
120                           CONTINUE                                    JAC00310
                              D(I,J)=D(I,J)-EIGENS(J)*U(I,J)              JAC00320
                              FNORM1=FNORM1+D(I,J)*D(I,J)                 JAC00330
130                    CONTINUE                                           JAC00340
140                 CONTINUE                                              JAC00350
                    FNORM=FNORM+SQRT(FNORM1)                              JAC00360
150             CONTINUE                                                  JAC00370
                AFLOPS=AFLOPS/20                                          JAC00380
                FNORM=FNORM/20.                                           JAC00390
C       - - - - - - - - - - - -  P R I N T I N G  - - - - - - - - - - - - JAC00400
                WRITE(6,*)'F O R   N = ',N                                JAC00410
                WRITE(6,*)'AVERAGE NUMBER OF FLOPS FOR CONV. = ',AFLOPS   JAC00420
                WRITE(6,*)'AVERAGE FROBENIUS NORM OF AU - UD = ',FNORM    JAC00430
160         CONTINUE                                                      JAC00440
            STOP                                                          JAC00450
            END                                                           JAC00460
C                                                                         JAC00470
C                                                                         JAC00480
C       ------------------------------------------------------------------JAC00490
            SUBROUTINE JACOBI(LDA,A,N,DELTA,EIGENS,U,COUNT)               JAC00500
            INTEGER LDA,N,COUNT                                           JAC00510
            REAL A(LDA,N),U(LDA,N),DELTA,EIGENS(N),FNORM,OFFA             JAC00520
C       ------------------------------------------------------------------JAC00530
C           THIS SUBROUTINE IS FOR FINDING THE EIGENVALUES & EIGENVECTORS JAC00540
C           OF AN N-BY-N SYMMETRIC MATRIX USING SERIAL JACOBI ALGORITHM.  JAC00550
```

```
C                * * *   PARAMETERS & VARIABLES :   * * *                JAC00560
C        A       ---  AN N-BY-N SYMMETRIC MATRIX.                         JAC00570
C        U       ---  AN N-BY-N MATRIX CONSISTS OF EIGENVECTORS OF A.      JAC00580
C        LDA     ---  LEADING DIMENSION OF A AND U.                       JAC00590
C        DELTA   ---  REQUIRED ACCURACY FOR THE COMPUTATION.              JAC00600
C        EIGENS ---  AN N-VECTOR CONSISTS OF EIGENVALUES OF A.            JAC00610
C        COUNT   ---  NUMBER OF ITERATIONS REQUIRED FOR CONVERGENCE.      JAC00620
C       ------------------------------------------------------------------JAC00630
        DO 15 I=1,N                                                       JAC00640
             DO 10 J=1,N                                                  JAC00650
                  IF (I.EQ.J) THEN                                        JAC00660
                            U(I,J)=1.                                     JAC00670
                       ELSE                                               JAC00680
                            U(I,J)=0.                                     JAC00690
                  ENDIF                                                   JAC00700
10           CONTINUE                                                     JAC00710
15      CONTINUE                                                          JAC00720
        COUNT=0                                                           JAC00730
        FNORM=0.                                                          JAC00740
        DO 20 I=1,N                                                       JAC00750
             FNORM=FNORM+A(I,I)*A(I,I)                                    JAC00760
20      CONTINUE                                                          JAC00770
C                                                                         JAC00780
25      OFFA=0.                                                           JAC00790
        DO 35 I=1,N-1                                                     JAC00800
             DO 30 J=I+1,N                                                JAC00810
                  OFFA=OFFA+A(I,J)*A(I,J)                                 JAC00820
30           CONTINUE                                                     JAC00830
35      CONTINUE                                                          JAC00840
        OFFA=OFFA*2.                                                      JAC00850
        IF (COUNT.NE.0) GOTO 40                                           JAC00860
        FNORM=FNORM+OFFA                                                  JAC00870
        DELTA2=DELTA*DELTA*FNORM                                          JAC00880
40      IF (OFFA.LE.DELTA2) GOTO 65                                       JAC00890
        COUNT=COUNT+1                                                     JAC00900
        DO 60 I=1,N-1                                                     JAC00910
             DO 55 J=I+1,N                                                JAC00920
                  IF(ABS(A(I,J)).LE.DELTA) GOTO 55                        JAC00930
                  TAU=(A(I,I)-A(J,J))*.5/A(I,J)                           JAC00940
                  IF (TAU.LT.0) THEN                                      JAC00950
                            SIGN=-1.                                      JAC00960
                       ELSE                                               JAC00970
                            SIGN=1.                                       JAC00980
                  ENDIF                                                   JAC00990
                  TAU2=1+TAU*TAU                                          JAC01000
                  T=SIGN/(ABS(TAU)+SQRT(TAU2))                            JAC01010
                  C=1+T*T                                                 JAC01020
                  C=1/SQRT(C)                                             JAC01030
                  S=T*C                                                   JAC01040
                  DO 45 K=1,N                                             JAC01050
                       AKI=C*A(K,I)+S*A(K,J)                              JAC01060
                       A(K,J)=-S*A(K,I)+C*A(K,J)                          JAC01070
                       A(K,I)=AKI                                         JAC01080
C                                                                         JAC01090
                       UKI=C*U(K,I)+S*U(K,J)                              JAC01100
```

```
                    U(K,J)=-S*U(K,I)+C*U(K,J)                          JAC01110
                    U(K,I)=UKI                                         JAC01120
45                CONTINUE                                             JAC01130
                  A(I,I)=C*A(I,I)+S*A(J,I)                             JAC01140
                  A(J,J)=-S*A(I,J)+C*A(J,J)                            JAC01150
                  A(I,J)=0.                                            JAC01160
                  DO 50 K=1,N                                          JAC01170
                    A(I,K)=A(K,I)                                      JAC01180
                    A(J,K)=A(K,J)                                      JAC01190
50                CONTINUE                                             JAC01200
55              CONTINUE                                               JAC01210
60            CONTINUE                                                 JAC01220
              GOTO 25                                                  JAC01230
65            DO 70 I=1,N                                              JAC01240
                EIGENS(I)=A(I,I)                                       JAC01250
70            CONTINUE                                                 JAC01260
              RETURN                                                   JAC01270
              END                                                      JAC01280
```

COMPUTER PROJECT NO:2 -- SERIAL JACOBI ALGORITHM BEN M CHEN

R E S U L T S

1) F O R N = 10

AVERAGE NUMBER OF FLOPS FOR CONV. = 18600

AVERAGE FROBENIUS NORM OF AU - UD = 0.891845557E-04

2) F O R N = 20

AVERAGE NUMBER OF FLOPS FOR CONV. = 176000

AVERAGE FROBENIUS NORM OF AU - UD = 0.312740682E-03

3) F O R N = 40

AVERAGE NUMBER OF FLOPS FOR CONV. = 1548800

AVERAGE FROBENIUS NORM OF AU - UD = 0.972041860E-03