The Numerical Efficiency of the Redheffer Star-Product in Linear Least-Squares Estimation

Molander, Per

1976

Document Version:
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):
Molander, P. (1976). *The Numerical Efficiency of the Redheffer Star-Product in Linear Least-Squares Estimation*. (Research Reports TFRT-3135). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
1

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

TFRT-3135

THE NUMERICAL EFFICIENCY OF THE REDHEFFER STAR-PRODUCT IN LINEAR LEAST-SQUARES ESTIMATION

P. MOLANDER

Report 7623 (C) May 1976
Department of Automatic Control
Lund Institute of Technology

[redacted]

THE NUMERICAL EFFICIENCY OF THE REDHEFFER STAR-PRODUCT IN LINEAR LEAST-SQUARES ESTIMATION

Per Molander

Abstract

This paper discusses the numerical properties of a new algorithm for finding the steady-state solution of $P(t+1) = FP(t)F^t + Q - FP(t)H^t[I + HP(t)H^t]^{-1}HP(t)F^t$.

Introduction

In a recent paper by Friedlander, Kailath, and Ljung [1], a new algorithm is derived for finding the steady state solution of the Riccati-type equation

$$P(t+1) = FP(t)F^t + Q - FP(t)H^t[I + HP(t)H^t]^{-1}HP(t)F^t \tag{1}$$

which arises e.g. in Kalman filter theory. (1) may converge quite slowly, and there is need for a fast algorithm to overcome this difficulty.

The new machinery developed by Friedlander et al is based on the so-called Redheffer star-product, defined thus:

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} * \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} = \begin{pmatrix} B_{11}(I-A_{12}B_{21})^{-1}A_{11} & B_{12}+B_{11}A_{12}(I-B_{21}A_{12})^{-1}B_{22} \\ A_{21}+A_{22}B_{21}(I-A_{12}B_{21})^{-1}A_{11} & A_{22}(I-B_{21}A_{12})^{-1}B_{22} \end{pmatrix}$$

Here, A_{11}, B_{11} etc are nxn-matrices; A_{12} and B_{21} must fulfil certain requirements for the inverse to exist (e.g. $A_{12}B_{21}$ may have all its eigenvalues in the LHP, as in the case below.) It is readily verified that the star-product is associative.

Defining $S(t) = \begin{pmatrix} \Phi(t) & P(t) \\ W(t) & \Phi^t(t) \end{pmatrix}$ and $M(t) = \begin{pmatrix} F(t) & Q(t) \\ -H^t(t)H(t) & F^t(t) \end{pmatrix}$ we note that (1) is exactly the equation for the (1,2)-element of $S(t)*M(t)$ using the ABCD-lemma. If $M(t)$ is independent of t, $S(t+k) = S(t)*M^{*k}$.

Since (1) converges to a positively definite solution regardless of $P(0)$ (provided $P(0)$ is positively definite), we may choose $P(0) = Q$ and get the following doubling formula:

$$S(2^p-1) = M^{*2p} \qquad (2)$$

The four recursive equations from (2) are

(3) $\Phi_{k+1} = \Phi_k(I-P_kW_k)^{-1}\,\Phi_k \qquad \Phi_0 = F$

(4) $P_{k+1} = P_k + \Phi_k P_k (I-W_kP_k)^{-1}\,\Phi_k^t \qquad P_0 = Q$

(5) $W_{k+1} = W_k + \Phi_k^t\,W_k(I-P_kW_k)^{-1}\,\Phi_k \qquad W_0 = -H^tH$

(6) equals the transpose of (3)

Convergence properties

The iteration (1) converges if the system $[H,F]$ is detectable, i.e. at least all unstable modes are observable.

If Q is positively definite, so is $P(\infty)$; and convergence difficulties do not arise in general.

If Q is positively semidefinite, $P(\infty)$ is positively definite or semidefinite, depending on whether the system $[B,F]$ is controllable or not; here, $Q = B^tB$. In this case, difficulties may arise due to the numerical inaccuracy of the computer. Due to this inaccuracy, $P(t)$ may become positively definite or indefinite. In the former case, $P(t)$ will converge to a positively definite solution, which is not the correct solution with respect to the given initial data. - In the latter case, $P(t)$ is not a proper initial value for further computation, but the iteration may still converge to the correct solution, or instability may occur. (For a discussion of these problems in the continuous case, see [2].)

From a purely analytical point of view, all these qualities are inherited by the algorithm. However, unlike (1), (4) does not necessarily converge to a solution of the algebraic equation

$$P = FPF^t + Q - FPH^t[I + HPH^t]^{-1} HPF^t$$

since it is a doubling formula, and may thus produce false results if perturbed during the computation. However, provided that F is not pathological in some sense, such inaccuracies will not grow fast enough to affect the result.

Numerical tests

The algorithm has been tested versus direct iteration of (1) for various dimensions and numbers of output signals. Its main disadvantage seems to be its large memory requirement. On the other hand, the computing time is in general so reduced by using the algorithm, that it has proved to be economically advantageous for almost all cases tested.

The inverse to be computed in (4) is of the same dimension as the state vector, whereas in (1) it is of the same order as the number of output signals. This implies that for systems of sufficiently large dimension with few outputs the direct iteration will supersede the algorithm; for the single output case, the limiting dimension seems to be 10, judging by the test.

The stopping condition used was $\| P(t+1) - P(t)\| < \varepsilon$, where $\| \cdot \|$ is a modified matrix norm. In general, nothing can be said about the ε necessary to acquire a certain degree of accuracy, and two different ε's should be used.

One disadvantage of (4) is that its operations do not automatically produce symmetric results; in fact, the output matrix $P(\infty)$ showed a discrepancy of 0.1 per cent in the numerically most unfavourable case.

Finally, it should be stressed that the execution times in the appendix should not be taken too seriously, since the convergence rate depends heavily on the matrix entries and not only on the dimensions involved.

References

[1] Friedlander, Kailath, Ljung: Scattering Theory and Linear Least-Squares estimation, Part II - Discrete Time Problems. J Franklin Inst 30 (1976) 71-82.

[2] Mårtensson: On the Matrix Riccati Equation. Information Sciences 3 (1971).

Appendix I

The matrices of the testbatch:

1.
$$F = \begin{pmatrix} -1 & 0 & -3 \\ -3 & -3 & 4 \\ 0 & 0 & -2 \end{pmatrix} \qquad H_a = \begin{bmatrix} 1 & 1.5 & 1 \end{bmatrix}$$

$$H_b = \begin{bmatrix} 1 & 1.5 & 1 \\ 1.5 & 2 & -1 \\ -0.5 & 3 & 2 \end{bmatrix}$$

2.
$$F = \begin{pmatrix} -20 & 10 & 10 \\ -18 & 17 & 22 \\ 13 & -13 & -17 \end{pmatrix}$$
H_a, H_b same as above

3.
$$F = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & -1 \\ -1 & 1 & 1 & 0 & 0 & 1 \\ 1 & -1 & 1 & 1 & 0 & -1 \\ -1 & 1 & -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 \end{pmatrix} \qquad H_a = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$H_b = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -1.2 & 2 & -1 & -1 & 1.5 & 1.5 \\ 0.4 & 1.5 & 2 & 0.4 & 0.4 & -1.2 \end{bmatrix}$$

$$H_c = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -1.2 & 2 & -1 & -1 & 1.5 & 1.5 \\ 0.4 & 1.5 & 2 & 0.4 & 0.4 & -1.2 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1.2 & 1.2 & 0 \\ 1 & 0.5 & 0.5 & 0 & 0 & 0 \end{bmatrix}$$

4.
$$F = \begin{pmatrix} -0.1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 10 & 10 & 5 \\ 0 & 3 & 0 & -3 & 1 & 0 \\ 7 & 2 & 0 & 0 & -10 & 0 \\ 32 & 15 & 0 & 0 & 100 & -50 \end{pmatrix} \qquad H_a = \begin{bmatrix} -2 & 4 & -5 & 7 & -8 & 10 \end{bmatrix}$$

H_b, H_c same as above

5.

$$F = \begin{pmatrix} -0.1 & -0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.1 & -0.2 & -0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.1 & -0.2 & -0.3 & -0.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.1 & -0.2 & -0.3 & -0.4 & -0.4 & 0 & 0 & 0 & 0 & 0 \\ -0.1 & -0.2 & -0.3 & -0.4 & -0.5 & -0.5 & 0 & 0 & 0 & 0 \\ -0.1 & -0.2 & -0.3 & -0.4 & -0.5 & -0.6 & -0.6 & 0 & 0 & 0 \\ -0.1 & -0.2 & -0.3 & -0.4 & -0.5 & -0.6 & -0.7 & -0.7 & 0 & 0 \\ -0.1 & -0.2 & -0.3 & -0.4 & -0.5 & -0.6 & -0.7 & -0.8 & -0.8 & 0 \\ -0.1 & -0.2 & -0.3 & -0.4 & -0.5 & -0.6 & -0.7 & -0.8 & -0.9 & -0.9 \\ -0.1 & -0.2 & -0.3 & -0.4 & -0.5 & -0.6 & -0.7 & -0.8 & -0.9 & -1.0 \end{pmatrix}$$

$$H_a = \begin{bmatrix} 10 & 50 & -25 & -50 & 10 & -25 & -50 & 10 & -50 & 100 \end{bmatrix}$$

$$\begin{bmatrix} 10 & 50 & -25 & -50 & 10 & -25 & -50 & 10 & -50 & 100 \\ 1 & 0 & 0 & 1.5 & 1 & 0 & 0 & 0 & 10 & 10 \\ 10 & 0 & 0 & 0 & 2 & -1 & 0 & 10 & 0 & 0 \end{bmatrix}$$

$$H_c = \begin{bmatrix} 10 & 50 & -25 & -50 & 10 & -25 & -50 & 10 & -50 & 100 \\ 1 & 0 & 0 & 1.5 & 1 & 0 & 0 & 0 & 10 & 10 \\ 10 & 0 & 0 & 0 & 2 & -1 & 0 & 10 & 0 & 0 \\ 1 & 1 & 0 & 1 & -1.2 & 0 & 0 & 0 & 0 & 2 \\ -1 & 0 & 0 & 0 & 10 & 0 & 0 & 10 & 0 & 2 \\ -1 & 1.5 & 0 & 0 & 0 & 1.5 & 0.4 & -5 & 0 & -5 \\ 10 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & -5 \\ -1 & 0 & -1 & -1 & -1 & -5 & -5 & 0 & 0 & -5 \\ 2 & 2 & 0 & 2 & 2 & -1 & 0 & -1 & 0 & -1 \\ 0 & -5 & 2 & 2 & 2 & -1 & 1.5 & 2 & 0.4 & 0.4 \end{bmatrix}$$

Q was generated by $Q_{ii} = 2$

$Q_{ij} = -1$ if $i-j = 1$, else 0.

Appendix II

Approximate execution time of the loop in milliseconds for the algorithm and direct iteration respectively

Run No.	Algorithm	Iteration
1 a	30	*
1 b	25	40
2 a	35	*
2 b	30	*
3 a	190	*
3 b	160	285
3 c	120	200
4 a	110	*
4 b	120	*
4 c	120	*
5 a	810	800
5 b	610	*
5 c	500	*

(*) means that execution was interrupted due to maximum number of iterations. In any of these cases the execution time was already longer than that of the algorithm.

```
OP3
-01/28/76-09:01:27 (,1)

PRO RAM

GE USED: CODE(1) 000331; DATA(0) 001062; BLANK COMMON(2) 000000

NAL REFERENCES (BLOCK, NAME)

  IMPD
  INTP$
  RDU$
  T02$
  WDU$
  I01$
  STOP$

GE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   000772 101F      0000   000773 102F      0000   000776 103F      0000    0010
   001007 108F      0000   001013 109F      0000   001017 110F      0000    0010
   000026 120G      0001   000027 123G      0001   000044 133G      0001    0000
   000071 152G      0001   000103 160G      0001   000104 163G      0001    0001
   000145 207G      0001   000146 212G      0001   000170 224G      0001    0001
   000210 241G      0001   000232 253G      0001   000237 257G      0001    0003
 R 000766 EPS       0000 R 000000 F         0000 R 000310 H         0000 I  0007
 I 000770 J         0000 I 000764 N         0000 I 000765 NUT       0000 R  0006
 R 000454 R
```

```
      DIMENSION F(10,10),Q(10,10),H(10,10),R(10,10)
     *,PINF(10,10)
  101 FORMAT( )
  102 FORMAT(10X,10E11.5)
  103 FORMAT(//,10X,20HNUMBER OF ITERATIONS,I5)
  104 FORMAT(10X,8HMATRIX F,/)
  108 FORMAT(/,10X,8HMATRIX Q,/)
  109 FORMAT(/,10X,8HMATRIX H,/)
  110 FORMAT(//,10X,27HEXECUTION TERMINATED AT K =,I4)
  113 FORMAT(////,31H           STEADY-STATE SOLUTION,/)
  114 FORMAT(/,10X,8HMATRIX R,/)
      N=10
      NUT=1
      EPS=1.0E-9
      DO 2 I=1,N
      DO 2 J=1,N
    2 READ(5,101) F(I,J)
      DO 50 I=1,N
      DO 50 J=1,N
   50 F(I,J)=0.1*F(I,J)
      WRITE(6,104)
      DO 19 I=1,N
   19 WRITE(6,102) (F(I,J),J=1,N)
      DO 1 I=1,N
      DO 1 J=1,N
    1 READ(5,101) Q(I,J)
      WRITE(6,108)
      DO 25 I=1,N
   25 WRITE(6,102) (Q(I,J),J=1,N)
      DO 3 I=1,NUT
```

```
00211              DO 3 J=1,N
00214            3 READ(5,101) H(I,J)
00221              WRITE(6,109)
00223              DO 26 I=1,NUT
00226           26 WRITE(6,102) (H(I,J),J=1,N)
00235              DO 30 I=1,NUT
00240              DO 30 J=1,NUT
00243           30 READ(5,101) R(I,J)
00250              WRITE(6,114)
00252              DO 31 I=1,NUT
00255           31 WRITE(6,102) (R(I,J),J=1,NUT)
00264              CALL LIMPD(F,Q,H,R,N,NUT,10,10,EPS,PINF,IERR)
00265           40 WRITE(6,113)
00267              DO 12 I=1,N
00272           12 WRITE(6,102) (PINF(I,J),J=1,N)
00301              WRITE(6,105) IERR
00304          105 FORMAT(//,10X,6HIERR =,I3)
00305              STOP
00306              END

     END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
/28/76-09:01:30 (,0)

 E LIMPD      ENTRY POINT 000453

 SED: CODE(1) 000556; DATA(0) 001071; BLANK COMMON(2) 000000

 REFERENCES (BLOCK, NAME)

  SCPU
 SECOM
 OLVR
 EDHEF
 NDU$
 IO2$
 IO1$
 ERR3$

 SSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 001001 103F      0000    001010 105F     0001    000205 11L      0001    000107
 000125 126G      0000    000773 130F     0000    000776 131F     0001    000127
 000155 145G      0001    000156 150G     0001    000171 156G     0001    000172
 000270 201G      0001    000402 21L      0001    000330 213G     0001    000331
 000371 233G      0001    000404 8L       0000 R  000144 FT       0000 R  000000
 001027 INJP$     0000 I  000765 ISING    0000 I  000764 IO       0000 I  000772
 000771 K         0003 I  000000 MSCPU    0000 R  000770 S        0000 R  000310
 000620 X4


            SUBROUTINE LIMPD(F,Q,H,R,N,NUT,IA,IUT,EPS,X2,IERR)
      C
      C     AUTHOR PER MOLANDER AUG 12TH, 1975
      C     REF. FRIEDLANDER,KAILATH,LJUNG,
      C          THE STAR-PRODUCT IN DISCRETE TIME LINEAR LEAST SQUARES
      C          APPROXIMATION
      C
      C     CALCULATES THE STEADY-STATE VALUE OF P IN
      C     P(T+1)=F*P(T)*F(TRANSP)+Q-F*P(T)*H(TRANSP)*
      C                                   -1
      C     *(R+H*P(T)*H(TRANSP))   *H*P(T)*F(TRANSP)
      C     PERTAINING TO THE SYSTEM
      C
      C     X(T+1)=F*X(T)+U(T)
      C     Y(T)=H*X(T)+V(T)
      C
      C     F IS OF ORDER NXN
      C     H IS OF ORDER NUTXN
      C     Q, OF ORDER NXN, AND R,OF ORDER NUTXNUT, ARE THE COVARIANCE
      C     MATRICES OF U AND V RESPECTIVELY
      C     IA DIMENSION PARAMETER
      C     THE MAXIMUM VALUE OF N AND NUT IS 10
      C     EPS,TEST QUANTITY,IS A BOUND FOR THE DISTANCE
      C     BETWEEN TWO CONSECUTIVE APPROXIMATIONS
      C     X2 IS THE SOLUTION MATRIX
      C     IF STEADY STATE IS NOT REACHED IN LESS THAN 25 STEPS, IERR
      C     IS 1, ELSE 0
      C     WARNING F,Q,H, AND R  ARE DESTROYED
```

```
00101      C        ********
00101      C
00101      C        SUBROUTINES REQUIRED
00101      C             REDHEF
00101      C             DECOM
00101      C             SOLVB
00101      C
00103               DIMENSION F(IA,IA),Q(IA,IA),H(IUT,IA),R(IUT,IUT),
00103              *HTH(10,10),FT(10,10),X1(10,10),X2(10,10),X3(10,10),
00104               IERR=0
00105               I0=MSCPU(0)
00106               CALL DECOM(R,NUT,IUT,1.0E-7,ISING)
00107               WRITE(6,130) ISING
00112           130 FORMAT(6H ISING,I2)
00113               CALL SOLVB(H,X1,NUT,N,IA)
00114               DO 4 I=1,N
00117               DO 4 J=1,NUT
00122             4 X3(I,J)=H(J,I)
00122      C        HTH=X3*X1
00125               DO 41 I=1,N
00130               DO 41 J=1,N
00133               S=0.0
00134               DO 42 K=1,NUT
00137            42 S=S+X3(I,K)*X1(K,J)
00141            41 HTH(I,J)=S
00144               DO 5 I=1,N
00147               DO 5 J=1,N
00152             5 HTH(I,J)=-HTH(I,J)
00155               DO 6 I=1,N
00160               DO 6 J=1,N
00163             6 FT(I,J)=F(J,I)
00166               K=0
00166      C
00166      C        LOOP
00167            11 K=K+1
00170               IF(K-10) 20,20,21
00173            20 CALL REDHEF(F,Q,HTH,FT,F,Q,HTH,FT,X1,X2,X3,X4,N,IA
00174               S=0.0
00175               DO 7 I=1,N
00200               DO 7 J=1,N
00203             7 S=S+ABS(Q(I,J)-X2(I,J))
00206               S=S/N
00207               IF(S-EPS) 8,8,9
00212             9 DO 10 I=1,N
00215               DO 10 J=1,N
00220               F(I,J)=X1(I,J)
00221               Q(I,J)=X2(I,J)
00222               HTH(I,J)=X3(I,J)
00223            10 FT(I,J)=X4(I,J)
00226               DO 51 I=1,N
00231            51 WRITE(6,131)(Q(I,J)          ,J=1,N)
00240           131 FORMAT(10X,10E11.5)
00241               GO TO 11
00241      C        END OF LOOP
00241      C
00242            21 IERR=1
00243             8 CONTINUE
00244               WRITE(6,103) K
00247           103 FORMAT(1H0,9X,20HNUMBER OF ITERATIONS,I5,//)
00250               I1=MSCPU(0)
00251               I0=(I1-I0)/K
00252               WRITE(6,105) I0
00255           105 FORMAT(1H0,9X,22HAVERAGE TIME/ITERATION,I4,3H MS)
00256               END
```

```
FOR,IS REDHEF
FOR S0E3-01/28/76-09:01:36 (,)

SUBROUTINE REDHEF     ENTRY POINT 000665

STORAGE USED: CODE(1) 000742; DATA(0) 000376; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003    DECOM
 0004    SOLV3
 0005    NDU$
 0006    T02$
 0007    ERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000057 105G     0001    000062 110G     0001    000060 114G
 0000    000315 130F     0001    000122 135G     0001    000203 150G
 0001    000237 167G     0001    000241 172G     0001    000243 176G
 0001    000277 215G     0001    000331 225G     0001    000335 230G
 0001    000363 245G     0001    000407 255G     0001    000410 260G
 0001    000500 303G     0001    000502 307G     0001    000533 317G
 0001    000570 336G     0001    000571 341G     0001    000613 347G
 0000 R  000000 D1       0000 R  000144 D2       0000 I  000310 I
 0000 I  000311 J        0000 I  000313 K        0000 R  000312 S


00101         SUBROUTINE REDHEF(A1,A2,A3,A4,B1,B2,B3,B4,C1,C2,C
00101      C  AUTHO. PER HOLANDER AUG 12TH, 1975
00101      C  REF. FRIEDLANDER,KAILATH,LJUNG
00101      C
00101      C  CALCULATES THE STAR-PRODUCT BETWEEN
00101      C
00101      C  A1 A2          B1 B2
00101      C           AND
00101      C  A3 A4          B3 B4
00101      C
00101      C  WHERE A1,B1 ETC ARE MATRICES OF ORDER NXN
00101      C  THE MAXIMUM VALUE OF N IS 10
00101      C  C1 ETC. ARE THE OUTPUT MATRICES
00101      C  IA DIMENSION PARAMETER
00101      C  SUBROUTINES REQUIRED
00101      C      DECOM
00101      C      SOLV3
00101      C
00101         DIMENSION A1(IA,IA),A2(IA,IA),A3(IA,IA),A4(IA,I
00103        *B1(IA,IA),B2(IA,IA),B3(IA,IA),B4(IA,IA),
00103        *C1(10,10),C2(10,10),C3(10,10),C4(10,10),D1(10,
00103         D1=A2*B3
00103      C
00104         DO 11 I=1,N
00107         DO 11 J=1,N
00112         S=0.0
00113         DO 12 K=1,N
00116      12 S=S+A2(I,K)*B3(K,J)
00120      11 D1(I,J)=S
00123         DO 1 I=1,N
```

```
00126         DO 1 J=1,N
00131       1 D1(I,J)=-J1(I,J)
00134         DO 2 I=1,N
00137       2 D1(I,I)=D1(I,I)+1.0
00141         CALL DECOM(D1,N,10,1.0E-7,ISING)
00142         CALL SOLVB(A1,D2,N,N,10)
00143         WRITE(6,130) ISING
00146     130 FORMAT(6H ISING,I2)
00146   C     C1=B1*D2
00147         DO 13 I=1,N
00152         DO 13 J=1,N
00155         S=0.0
00156         DO 14 K=1,N
00161      14 S=S+B1(I,K)*D2(K,J)
00163      13 C1(I,J)=S
00163   C     D1=B3*D2
00166         DO 15 I=1,N
00171         DO 15 J=1,N
00174         S=0.0
00175         DO 16 K=1,N
00200      16 S=S+B3(I,K)*D2(K,J)
00202      15 D1(I,J)=S
00202   C     D2=A4*D1
00205         DO 17 I=1,N
00210         DO 17 J=1,N
00213         S=0.0
00214         DO 18 K=1,N
00217      18 S=S+A4(I,K)*D1(K,J)
00221      17 D2(I,J)=S
00224         DO 3 I=1,N
00227         DO 3 J=1,N
00232       3 C3(I,J)=A3(I,J)+D2(I,J)
00232   C     D1=B3*A2
00235         DO 19 I=1,N
00240         DO 19 J=1,N
00243         S=0.0
00244         DO 20 K=1,N
00247      20 S=S+B3(I,K)*A2(K,J)
00251      19 D1(I,J)=S
00254         DO 4 I=1,N
00257         DO 4 J=1,N
00262       4 D1(I,J)=-J1(I,J)
00265         DO 5 I=1,N
00270       5 D1(I,I)=D1(I,I)+1.0
00272         CALL DECOM(D1,N,10,1.0E-7,ISING)
00273         CALL SOLVB(B4,D2,N,N,10)
00274         WRITE(6,130) ISING
00274   C     D1=A2*D2
00277         DO 21 I=1,N
00302         DO 21 J=1,N
00305         S=0.0
00306         DO 22 K=1,N
00311      22 S=S+A2(I,K)*D2(K,J)
00313      21 D1(I,J)=S
00313   C     C2=B1*D1
00316         DO 23 I=1,N
00321         DO 23 J=1,N
00324         S=0.0
00325         DO 24 K=1,N
00330      24 S=S+B1(I,K)*D1(K,J)
00332      23 C2(I,J)=S
00335         DO 6 I=1,N
00340         DO 6 J=1,N
00343       6 C2(I,J)=C2(I,J)+D2(I,J)
```

```
00343        C      C4=A4*D2
00346               DO 25 I=1,N
00351               DO 25 J=1,N
00354               S=0.0
00355               DO 26 K=1,N
00360            26 S=S+A4(I,K)*D2(K,J)
00362            25 C4(I,J)=S
00365               RETURN
00366               END
```

END OF COMPILATION: NO DIAGNOSTICS.