SIMULATION OPTIMIZATION SYSTEMS
Research Laboratory

INVCH - A FORTRAN PACKAGE FOR CALCULATING THE INVERSE OF A PERTURBED MATRIX

J.W. Bandler and Q.J. Zhang

SOS-84-22-L

December 1984

McMASTER UNIVERSITY
Hamilton, Canada L8S 4L7
Department of Electrical and Computer Engineering

INVCH - A FORTRAN PACKAGE FOR CALCULATING THE INVERSE OF A PERTURBED MATRIX

J.W. Bandler and Q.J. Zhang

SOS-84-22-L

December 1984

INVCH – A FORTRAN PACKAGE FOR CALCULATING THE INVERSE OF A PERTURBED MATRIX

J.W. Bandler and Q.J. Zhang

Abstract

INVCH is a package of subroutines for calculating the inverse of a perturbed matrix. The package uses an interative procedure to calculate large change effects. The package and documentation have been developed for use on the CDC 170/815 system with the NOS 2.2-602/587 operating system and the Fortran Extended (FTN) version 4.8 compiler. This document contains a listing of the INVCH package.

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant A7239.

The authors are with the Simulation Optimization Systems Research Laboratory and the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4L7.

I. INTRODUCTION

INVCH is a package of Fortran subroutines for calculating the inverse of a perturbed matrix. It uses large change formula (64) of our previous report [1], yielding efficient computation when the variables affect the matrix in such a way that the deviation matrix has a small rank compared to the order of the matrix.

The whole package is written in Fortran IV for the CDC 170/815 system with the NOS 2.2-602/587 operating system. It is available at McMaster University in the form of a library of binary relocatable subroutines in the group indirect file LIBILCH under the charge RJWBAND.

This document includes a listing of the package INVCH. The user's manual presented together with illustrative examples is found in [2]. The listing contains 303 lines, of which 149 are comments. It has been modularized into 4 subroutines. The list of all subroutines is given in Table I.

TABLE I

LIST OF SUBROUTINES OF THE INVCH PACKAGE

	Subroutine	Number of Lines (source text)	Listing from Page
1	INVCH	79	4
2	INVER0	90	5
3	INVCHA	66	6
4	LUFACT	62	7

II. REFERENCES

[1] J.W. Bandler and Q.J. Zhang, "A unified approach to first-order and large change sensitivity computations in linear systems", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-84-20-R, 1984.

[2] J.W. Bandler and Q.J. Zhang, "INVCH - A Fortran package for calculating the inverse of a perturbed matrix", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-84-22-U, 1984.

III. LISTING OF THE INVCH PACKAGE

```
      SUBROUTINE INVCH(N,LW0,A,R,W0,ICH)                                  000001
C                                                                         000002
C     AN MAN-MACHINE INTERACTIVE PROGRAM TO CALCULATE THE INVERSE OF      000003
C     THE PERTURBED MATRIX OF [A] USING THE ITERATIVE SCHEME OF (64) OF   000004
C     REFERENCE [1].                                                      000005
C                                                                         000006
C     LIST OF ARGUMENTS :  (  ....  FOR INPUT ARGUMENTS.                  000007
C                             ----  FOR OUTPUT ARGUMENTS.)                000008
C                                                                         000009
C      N     ....  NO. OF ROWS OR COLUMNS OF [A].                         000010
C      LW0   ....  AN INTEGER DEFINING THE DIMENSION OF ARRAY W0.         000011
C                  THE RECOMMENDED VALUE IS N*(2+4*N).                    000012
C      A     ..--  REAL ARRAY OF DIMENSION N BY N. ON ENTRY, IT MUST      000013
C                  BE SET TO MATRIX [A].  ON EXIT, IT CONTAINS THE        000014
C                  LU FACTORS OF [A].                                     000015
C      R     ----  REAL ARRAY OF DIMENSION N BY N, PROVIDING SPACES      000016
C                  FOR THE INVERSE OF [A] UNDER VARIOUS CHANGES.          000017
C      W0    ----  REAL ARRAY OF DIMENSION LW0, USED AS WORKING SPACES.   000018
C      ICH   ....  AN INTEGER DEFINING THE CHANNEL NUMBER FOR OUTPUT      000019
C                  PRINT-OUT.                                             000020
C                                                                         000021
      DIMENSION A(N,N),R(N,N),W0(LW0)                                     000022
C                                                                         000023
      WRITE(ICH,80)                                                       000024
      DO 5 I=1,N                                                          000025
    5 WRITE(ICH,60) (A(I,J),J=1,N)                                        000026
      WRITE(ICH,70)                                                       000027
      PRINT*,"************************************************************"000028
      PRINT*," FIND THE INVERSE OF ( [A]+[V][D][W]' ) BY LARGE CHANGE"    000029
      PRINT*," FORMULA ."                                                 000030
      PRINT*," "                                                          000031
C                                                                         000032
C     CALL SUBROUTINE LUFACT TO PERFORM LU-FACTORIZATION AND FORWARD      000033
C     AND BACKWARD SUBSTITUTIONS TO OBTAIN THE INVERSE OF [A] BEFORE      000034
C     LARGE CHANGE.  UNIT N-VECTORS ARE FORMULATED BY ARRAY W0 FOR        000035
C     CORRESPONDING R.H.S.                                                000036
C                                                                         000037
      DO 6 J=1,N                                                          000038
      DO 4 I=1,N                                                          000039
    4 W0(I)=0.                                                            000040
      W0(J)=1.                                                            000041
      MODE=1                                                              000042
      IF(J.GT.1) MODE=3                                                   000043
      CALL LUFACT(N,A,W0,MODE)                                            000044
      DO 6 I=1,N                                                          000045
    6 R(I,J)=W0(I)                                                        000046
C                                                                         000047
      PRINT*,"      THE INVERSE OF [A] BEFORE CHANGE :"                   000048
      DO 8 I=1,N                                                          000049
    8 WRITE(ICH,60)  (R(I,J),J=1,N)                                       000050
      WRITE(ICH,70)                                                       000051
C                                                                         000052
   10 PRINT*,"************************************************************"000053
      PRINT*," ENTER IR1 AND IR2    ( SUCH THAT [D] IS OF IR1 BY IR2 )"   000054
      READ*,IR1,IR2                                                       000055
C                                                                         000056
C     CALCULATE RELEVANT PARAMETERS FOR MEMORY SPACE DISTRIBUTION.        000057
C                                                                         000058
      N2=1+N*IR1                                                          000059
      N3=N2+IR1*IR2                                                       000060
      N4=N3+N*IR2                                                         000061
      N5=N4+N                                                             000062
      N6=N5+N                                                             000063
C                                                                         000064
C     CALL THE MAIN SUBROUTINE TO FIND THE INVERSE OF ( [A]+[V][D][W]' )  000065
```

```
C     USING LARGE CHANGE FORMULA (64) OF REF. [1].                          000066
C                                                                           000067
      CALL INVER0(N,IR1,IR2,R,W0(1),W0(N2),W0(N3),W0(N4),W0(N5),W0(N6),     000068
     +ICH)                                                                  000069
      PRINT*," TRY ANOTHER SET OF  [V], [D] AND [W] ?     "                 000070
      PRINT*," ENTER 1 OR 2  ( Y OR N ) "                                   000071
      READ*,KKK                                                             000072
      IF(KKK.EQ.1) GO TO 10                                                 000073
C                                                                           000074
      RETURN                                                                000075
   60 FORMAT(/5X,10F12.5/)                                                  000076
   70 FORMAT(//)                                                            000077
   80 FORMAT(1H1,6X,"THE ORIGINAL MATRIX [A] : ")                           000078
      END                                                                   000079
C                                                                           000080
C                                                                           000081
      SUBROUTINE INVER0(N,IR1,IR2,R0,V,D,W,S,G,R,ICH)                       000082
C                                                                           000083
C     THIS SUBROUTINE ORGANIZES [V],[D] AND [W] AND CONTROLS THE            000084
C     WHOLE CALCULATION PROCEDURE.                                          000085
C                                                                           000086
C     LIST OF ARGUMENTS:  ( .... FOR INPUT ARGUMENTS.                       000087
C                         ---- FOR OUTPUT ARGUMENTS. )                     000088
C      N    ....  NO. OF ROWS OR COLUMNS OF [A].                           000089
C      IR1  ....  NO. OF ROWS OF [D].                                       000090
C      IR2  ....  NO. OF COLUMNS OF [D].                                    000091
C      R0   ....  REAL ARRAY OF DIMENSION N BY N, CONTAINING THE            000092
C                 INVERSE OF [A], I.E. BEFORE CHANGE.                       000093
C      V,D,W --   REAL ARRAYS OF DIMENSIONS N BY IR1, IR1 BY IR2 AND        000094
C                 N BY IR2 RESPECTIVELY, USED TO STORE [V],[D] AND [W].     000095
C      S,G ----   BOTH ARE REAL ARRAYS OF DIMENSION N, USED AS WORKING      000096
C                 SPACES.                                                   000097
C      R    ----  REAL ARRAY OF DIMENSION N BY N, CONTAINING THE INVERSE    000098
C                 OF ( [A]+[V][D][W]' ), I.E., AFTER CHANGE.                000099
C                                                                           000100
      DIMENSION R0(N,N),R(N,N),V(N,IR1),D(IR1,IR2),W(N,IR2),S(N),G(N)       000101
C                                                                           000102
      PRINT*," [V],[D] AND [W] ARE MATRICES OF DIMENSIONS N BY IR1,"        000103
      PRINT*,"                 IR1 BY IR2 AND N BY IR2 RESPECTIVELY"        000104
      PRINT*,"                 SUCH THAT  DELTA([A])=[V][D][W]' ."          000105
      PRINT*," ENTER [V]  ( COLUMN BY COLUMN ) "                            000106
      READ*,V                                                               000107
      PRINT*," ENTER [W]  ( COLUMN BY COLUMN ) "                            000108
      READ*,W                                                               000109
      WRITE(ICH,100) N,IR1,IR2                                              000110
      PRINT*,"     MATRIX [V] : "                                           000111
      DO 5 I=1,N                                                            000112
    5 WRITE(ICH,80) (V(I,J),J=1,IR1)                                        000113
      WRITE(ICH,85)                                                         000114
      PRINT*,"     MATRIX [W] : "                                           000115
      DO 10 I=1,N                                                           000116
   10 WRITE(ICH,80) (W(I,J),J=1,IR2)                                        000117
      WRITE(ICH,85)                                                         000118
C                                                                           000119
      KKK=0                                                                 000120
   20 PRINT*," [D] IS OF ",IR1," BY ",IR2,"."                               000121
      PRINT*," ENTER [D]  ( COLUMN BY COLUMN ) "                            000122
      READ*,D                                                               000123
      PRINT*," "                                                            000124
C                                                                           000125
C     IF KKK IS NON-ZERO, THEN [V] AND [W] ARE THE SAME AS THOSE            000126
C     WITH THE PREVIOUS [D].                                                000127
C                                                                           000128
      IF(KKK.NE.0) WRITE(ICH,110)                                           000129
      PRINT*,"     MATRIX [D] : "                                           000130
```

```
      DO 25 I=1,IR1                                                          000131
   25 WRITE(ICH,80) (D(I,J),J=1,IR2)                                         000132
      WRITE(ICH,85)                                                          000133
C                                                                            000134
C     R SHOULD CONTAIN THE INITIAL INVERSE OF [A] BEFORE ENTERING            000135
C     SUBROUTINE INVCHA. IT IS ALTERED BY INVCHA TO CONTAIN THE INVERSE      000136
C     OF [A] AFTER CHANGE.                                                   000137
C                                                                            000138
      DO 30 I=1,N                                                            000139
      DO 30 J=1,N                                                            000140
   30 R(I,J)=R0(I,J)                                                         000141
C                                                                            000142
C     CALCULATE THE INVERSE OF ( [A]+[V][D][W]' ) BY SUBROUTINE INVCHA.      000143
C                                                                            000144
      CALL INVCHA(N,IR1,IR2,R,V,D,W,S,G,IFLAG)                               000145
C                                                                            000146
C     IF  IFLAG IS ZERO,  THEN THE CALCULATION IS SUCCESSFUL,                000147
C     OTHERWISE FAILED DUE TO ZERO DENOMINATOR IN THE ITERATIVE              000148
C     PROCEDURE.                                                             000149
C                                                                            000150
      IF(IFLAG.NE.0) GO TO 50                                                000151
      PRINT*,"     THE INVERSE OF ( [A]+[V][D][W]' ) :"                      000152
      DO 40 I=1,N                                                            000153
   40 WRITE(ICH,80) (R(I,J),J=1,N)                                           000154
      WRITE(ICH,85)                                                          000155
C                                                                            000156
C                                                                            000157
   50 PRINT*," TRY ANOTHER [D] ?    ENTER 1 OR 2  ( Y OR N ) "                000158
      READ*,KKK                                                              000159
      IF(KKK.EQ.1) GO TO 20                                                  000160
      RETURN                                                                 000161
C                                                                            000162
   80 FORMAT(/5X,10F12.5/)                                                   000163
   85 FORMAT(//)                                                             000164
  100 FORMAT(1H1,                                                            000165
     +4X,"NUMBER OF ROWS OR COLUMNS OF [A] ( N )  .........",I3//            000166
     +5X,"NUMBER OF ROWS OF [D] ( IR1 )           .........",I3//            000167
     +5X,"NUMBER OF COLUMNS OF [D] ( IR2 )        .........",I3//)           000168
  110 FORMAT(1H1,5X," [V] AND [W] ARE THE SAME AS THOSE WITH THE",           000169
     +" PREVIOUS [D]."//)                                                    000170
      END                                                                    000171
C                                                                            000172
C                                                                            000173
      SUBROUTINE INVCHA(N,IR1,IR2,R,V,D,W,S,G,IFLAG)                         000174
C                                                                            000175
C     SUBROUTINE TO CALCULATE THE INVERSE OF ( [A]+[V][D][W]' ) BY           000176
C     LARGE CHANGE FORMULA (64) OF REF. [1].                                 000177
C                                                                            000178
C     LIST OF ARGUMENTS : ( ....  FOR INPUT ARGUMENTS.                       000179
C                         ----  FOR OUTPUT ARGUMENTS.)                      000180
C      N     ....  NO. OF ROWS OR COLUMNS OF [A].                            000181
C      IR1   ....  NO. OF ROWS OF [D].                                       000182
C      IR2   ....  NO. OF COLUMNS OF [D].                                    000183
C      R     ..--  REAL ARRAY OF DIMENSION N BY N. ON ENTRY, IT CONTAINS     000184
C                  THE INVERSE OF [A].  ON EXIT, IT CONTAINS THE INVERSE     000185
C                  OF ( [A]+[V][D][W]' ).                                    000186
C      V     ....  REAL ARRAY OF DIMENSION N BY IR1, CONTAINING [V].         000187
C      D     ....  REAL ARRAY OF DIMENSION IR1 BY IR2,CONTAINING [D].        000188
C      W     ....  REAL ARRAY OF DIMENSION N BY IR2, CONTAINING [W].         000189
C      S,G   ----  BOTH ARE REAL ARRAYS OF DIMENSION N, USED AS WORKING      000190
C                  SPACES.                                                   000191
C      IFALG ---   =0,  SUCCESSFULLY EXIT.                                   000192
C                  =1,  ZERO DENOMINATOR OCCURS IN THE ITERATIVE             000193
C                       PROCEDURE, CALCULATION TERMINATED.                   000194
C                                                                            000195
```

```
      DIMENSION R(N,N),V(N,IR1),D(IR1,IR2),W(N,IR2),S(N),G(N)                  000196
C                                                                              000197
      IFLAG=0                                                                  000198
C                                                                              000199
C     THE ITERATIVE PROCEDURE STARTS HERE.                                     000200
C                                                                              000201
      DO 70 IT=1,IR2                                                           000202
C                                                                              000203
C     CALCULATION OF THE [S] VECTOR IN (64) OF REF. [1].  THE RESULT IS        000204
C     STORED IN ARRAY S.                                                       000205
C                                                                              000206
      DO 10 I=1,N                                                              000207
      G(I)=0.                                                                  000208
      DO 10 J=1,IR1                                                            000209
   10 G(I)=G(I)+V(I,J)*D(J,IT)                                                 000210
      DO 20 I=1,N                                                              000211
      S(I)=0.                                                                  000212
      DO 20 J=1,N                                                              000213
   20 S(I)=S(I)+R(I,J)*G(J)                                                    000214
C                                                                              000215
C     CALCULATE THE DENOMINATOR IN (64).                                       000216
C                                                                              000217
      TEMP=1.                                                                  000218
      DO 30 I=1,N                                                              000219
   30 TEMP=TEMP+W(I,IT)*S(I)                                                   000220
C                                                                              000221
      IF(ABS(TEMP).GT.1.E-20) GO TO 40                                         000222
      PRINT*," ZERO DENOMINATOR OCCURS IN THE",IT,"-TH ITERATION "             000223
      IFLAG=1                                                                  000224
      RETURN                                                                   000225
C                                                                              000226
C     UPDATE THE INVERSE OF [A] .                                              000227
C                                                                              000228
   40 DO 60 J=1,N                                                              000229
      TEMP1=0.                                                                 000230
      DO 50 K=1,N                                                              000231
   50 TEMP1=TEMP1+W(K,IT)*R(K,J)                                               000232
      TEMP1=TEMP1/TEMP                                                         000233
      DO 60 I=1,N                                                              000234
   60 R(I,J)=R(I,J)-S(I)*TEMP1                                                 000235
   70 CONTINUE                                                                 000236
C                                                                              000237
      RETURN                                                                   000238
      END                                                                      000239
C                                                                              000240
C                                                                              000241
      SUBROUTINE LUFACT(N,A,B,MODE)                                            000242
C                                                                              000243
C     PERFORM LU-FACTORIZATION AND/OR FORWARD AND BACKWARD                     000244
C     SUBSTITUTION ( FBS ) FOR THE SOLUTION OF [A][X]=[B].                     000245
C                                                                              000246
C     LIST OF ARGUMENTS :  (  .... FOR INPUT ARGUMENTS.                        000247
C                             ---- FOR OUTPUT ARGUMENTS.)                     000248
C      N     ....  NO. OF ROWS OR COLUMNS OF [A].                              000249
C      A     ..--  REAL ARRAY OF DIMENSION N BY N. ON ENTRY, IT SHOULD        000250
C                  BE SET TO [A] WHEN MODE=1 OR, SET TO THE LU FACTORS        000251
C                  OF [A] IF MODE=3.  ON EXIT, IT CONTAINS THE LU FACTORS     000252
C                  OF [A].                                                    000253
C      B     ..--  REAL ARRAY OF DIMENSION N.  ON ENTRY, IT CONTAINS         000254
C                  THE R.H.S. OF THE LINEAR EQUATIONS, I.E, [B]. ON           000255
C                  EXIT, IT CONTAINS THE SOLUTION VECTOR [X].                 000256
C      MODE ....   =1,  LU-FACTORIZATION AND FBS.                             000257
C                  =3,  FBS ONLY.                                             000258
C                                                                              000259
C                                                                              000260
```

```
      DIMENSION A(N,N),B(N)                                              000261
C                                                                        000262
      N2=N-1                                                             000263
      N1=N+1                                                             000264
      IF(MODE.EQ.3.OR.N.EQ.1) GO TO 50                                   000265
C                                                                        000266
C     CROUT'S ALGORITHM OF LU-FACTORIZATION.                             000267
C                                                                        000268
      DO 20 II=1,N2                                                      000269
      ZZ=A(II,II)                                                        000270
      IF(ABS(ZZ).LT.1.E-20) GO TO 100                                    000271
      NN=II+1                                                            000272
      DO 20 J=NN,N                                                       000273
      A(II,J)=A(II,J)/ZZ                                                 000274
      DO 10 K=NN,N                                                       000275
   10 A(K,J)=A(K,J)-A(K,II)*A(II,J)                                      000276
   20 CONTINUE                                                           000277
C                                                                        000278
C     FORWARD SUBSTITUTION                                               000279
C                                                                        000280
   50 IF(ABS(A(1,1)).LT.1.E-20) GO TO 100                                000281
      B(1)=B(1)/A(1,1)                                                   000282
      IF(N.EQ.1) RETURN                                                  000283
      DO 70 II=2,N                                                       000284
      ZZ=A(II,II)                                                        000285
      IF(ABS(ZZ).LT.1.E-20) GO TO 100                                    000286
      IN=II-1                                                            000287
      DO 60 K=1,IN                                                       000288
   60 B(II)=B(II)-A(II,K)*B(K)                                           000289
   70 B(II)=B(II)/ZZ                                                     000290
C                                                                        000291
C     BACKWARD SUBSTITUTION                                              000292
C                                                                        000293
      DO 80 L=2,N                                                        000294
      II=N1-L                                                            000295
      IP=II+1                                                            000296
      DO 80 K=IP,N                                                       000297
   80 B(II)=B(II)-A(II,K)*B(K)                                           000298
      RETURN                                                             000299
C                                                                        000300
  100 PRINT*,"INDEFINITE VALUE IN LUFACT DUE TO ZERO ON THE DIAGONAL"    000301
      RETURN                                                             000302
      END                                                                000303
```