DECUS

PROGRAM LIBRARY

DECUS NO.	8-73
TITLE	Matrix Inversion - Complex Numbers
AUTHOR	Prof. A.E. Sapega
COMPANY	Trinity College Hartford, Connecticut
DATE	April 18, 1967
FORMAT	

MATRIX INVERSION - COMPLEX NUMBERS

Program Library Writeup

DECUS No. 8-73

The program inverts a matrix, up to size 6x6, of complex numbers. The algorithm used is the Gauss-Jordan method, programmed to carry out complex number calculations. A unit vector of appropiate size is generated internally. Following the Gauss sweep-out, the matrix is shifted, another unit vector is generated, and the calculation proceeds. The print out of the matrices uses the symbol J to designate the imaginary part, e.g., A = a+jb.

MINIMUM HARDWARE

PDP-8, ASR-33

OTHER PROGRAMS NEEDED

FORTRAN Compiler, FORTRAN Operating System

STORAGE REQUIREMENT

This program uses essentially all core not used by the FORTRAN Operating system.

EXECUTION TIME

Actual computation takes less than 10 seconds. Data read in and read out may take up to five minutes.

MISCELLANEOUS

A data tape should be made up beforehand. The first entry is the size of the matrix. Individual data entries are by row. Each value should be written as the real part followed by the imaginary part.

```
TYPE 1
1;FORMAT(/,"MATRIX INVERSION, COMPLEX NUMBERS: (A+BI)
WILL HANDLE UP TO 6X6,   LOAD DATA TAPE, HIT CONTINUE")
PAUSE
TYPE 2
2;FORMAT(/,/,"SIZE OF MATRIX  ")
ACCEPT 3,N
3;FORMAT(I)
LAST = N*N
DIMENSION AR(42), AI(42)
TYPE 4
4;FORMAT(/,/," INPUT MATRIX VALUES ",/)
C;READ IN DATA
I1 = 1
I2 = LAST - N + 1
DO 100 J = 1,N
DO 101 I = I1,I2,N
ACCEPT 5,AR(I), AI(I)
5;FORMAT(E)
101;CONTINUE
I1 = I1 + 1
I2 = I2 + 1
100;CONTINUE
TYPE 9
9;FORMAT(/,/,"ORIGINAL MATRIX VALUES",/)
JOG = 1
GO TO 200
C;BEGIN INVERSION
C;J IS INDEX OF STAGE OF INVERSION
149;DO 150 J = 1,N
C;SET UP UNIT VECTOR
DO 105 I = 1,N
AR(LAST+I) = 0
AI(LAST+I) = 0
105;CONTINUE
AR(LAST +J) = 1.0
C;SET UP TO CLEAR PIVOT ROW
PVTR = AR(J)
PVTI = AI(J)
RADN = (PVTR*PVTR)+(PVTI*PVTI)
J3 = LAST + J
DO 106 KP = J,J3,N
C;SAVE ORIGINAL COMPLEX NUMBER
APSR = AR(KP)
APSI = AI(KP)
AR(KP) = ((APSR*PVTR)+(APSI*PVTI))/RADN
AI(KP) = ((APSI*PVTR)-(APSR*PVTI))/RADN
106;CONTINUE
C;CALCULATE REMAINING ROWS
DO 110 KRT = 1,N
IF(KRT-J) 107,110,107
107;KR1 = KRT
KR2 = KR1 + LAST
KPR = J
RWCR = AR(KR1)
RWCI = AI(KR1)
DO 109 KR = KR1,KR2,N
```

```
APR = RWCR*AR(KPR)-RWCI*AI(KPR)
API = RWCI*AR(KPR)+RWCR*AI(KPR)
C;SAVE ORIGINAL COMPLEX NUMBER
ARSR = AR(KR)
ARSI = AI(KR)
AR(KR) = ARSR-APR
AI(KR) = ARSI-API
KPR = KPR + N
109;CONTINUE
110;CONTINUE
C;SHIFT ARRAY TO ELIMINATE LEADING UNIT VECTOR
DO 111 I = 1,LAST
AR(I) = AR(I+N)
AI(I) = AI(I+N)
111;CONTINUE
150;CONTINUE
TYPE 160
160;FORMAT(/,/,"THE INVERSE MATRIX IS"/)
JOG = 2
GO TO 200
C;PRINT OUT SUBROUTINE
200;ILNE = 4
NR=NRC=1
TYPE 201,NR
201;FORMAT(/,"ROW ",I,/)
I1 = 1
I2 = LAST - N + 1
KONT = 2
DO 210 JP = 1,N
DO 213 I = I1,I2,N
TYPE 5,AR(I)
TYPE 212
212;FORMAT("+J")
TYPE 5,AI(I)
IF(N-NRC) 216,216,215
215;IF(KONT-ILNE) 214,220,214
216;IF(N-NR) 213,213,217
217;NR = NR+1
NRC = 1
KONT = 2
ILNE = 4
TYPE 201,NR
GO TO 213
220;ILNE = ILNE + 4
TYPE 221
221;FORMAT(/,/)
214;KONT = KONT + 2
NRC = NRC +1
213;CONTINUE
I1 = I1 + 1
I2 = I2 + 1
210;CONTINUE
GO TO (149,330),JOG
330;STOP
END
```

```
MATRIX INVERSION, COMPLEX NUMBERS: (A+BI)
WILL HANDLE UP TO 6X6,   LOAD DATA TAPE, HIT CONTINUE

SIZE OF MATRIX  3


 INPUT MATRIX VALUES

6   4   -3   -4   -1   2
-3   -4   6   1   -3   -3
-1   2   -3   -3   4   0


ORIGINAL MATRIX VALUES

ROW +1
+0.600000E+1     +J+0.400000E+1     -0.300000E+1     +J-0.400000E+1

-0.100000E+1     +J+0.200000E+1
ROW +2
-0.300000E+1     +J-0.400000E+1     +0.600000E+1     +J+0.100000E+1

-0.300000E+1     +J-0.300000E+1
ROW +3
-0.100000E+1     +J+0.200000E+1     -0.300000E+1     +J-0.300000E+1

+0.400000E+1     +J+0.000000E+0

THE INVERSE MATRIX IS

ROW +1
+0.139990E+0     +J+0.938361E-2     +0.510238E-1     +J+0.113802E+0

-0.739477E-2     +J+0.559705E-1
ROW +2
+0.510238E-1     +J+0.113802E+0     +0.563021E-1     +J+0.160058E+0

-0.815975E-2     +J+0.165208E+0
ROW +3
-0.739473E-2     +J+0.559705E-1     -0.815973E-2     +J+0.165208E+0

+0.146110E+0     +J+0.135477E+0     !
```