



# DECUS

## PROGRAM LIBRARY

DECUS NO.	8-72
TITLE	Matrix Inversion - Real Numbers
AUTHOR	Prof. A. E. Sapega
COMPANY	Trinity College Hartford, Connecticut
DATE	April 18, 1967
FORMAT	

## MATRIX INVERSION - REAL NUMBERS

### Program Library Writeup

DECUS No. 8-72

This program inverts a matrix, up to size 12x12, of real numbers. The algorithm used is the Gauss-Jordan method. A unit vector of appropriate size is generated internally at each stage. Following the Gauss sweep-out, the matrix is shifted in storage, another unit vector is generated, and the calculation proceeds.

### 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 5 minutes.

### MISCELLANEOUS

A data tape should be prepared beforehand. The first entry on the tape is the size of the matrix. The matrix entries are then read in by rows.

```

TYPE 1
1;FORMAT(/,"MATRIX INVERSION.... WILL HANDLE UP TO 12X12
LOAD DATA TAPE --- HIT CONTINUE")
PAUSE
TYPE 2
2;FORMAT(/,/, "SIZE OF MATRIX = ")
ACCEPT 3,N
3;FORMAT(I)
LAST = N*N
DIMENSION A(156)
TYPE 4
4;FORMAT(/,/, "MATRIX ENTRIES IN ORDER A11,A12,A13,ETC.")
C;READ IN DATA
I1 = 1
I2 = LAST - N + 1
DO 100 J = 1,N
DO 101 I = I1,I2,N
ACCEPT 5,A(I)
5;FORMAT(E)
101;CONTINUE
I1 = I1 + 1
I2 = I2 + 1
100;CONTINUE
TYPE 8
8;FORMAT(/,/,/, "MATRIX INVERSION RESULTS",/)
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
A(LAST+I) = 0.0
105;CONTINUE
A(LAST+J) = 1.0
C;SET UP TO CLEAR PIVOT ROW
PVT = A(J)
J3 = LAST + J
DO 106 KP = J,J3,N
A(KP) = A(KP)/PVT
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
RWC = A(KR1)
DO 109 KR = KR1,KR2,N
A(KR) = A(KR) - RWC * A(KPR)
KPR = KPR + N
109;CONTINUE
110;CONTINUE
C;SHIFT ARRAY TO ELIMINATE LEADING UNIT VECTOR
DO 111 I = 1, LAST
A(I) = A(I+N)
111;CONTINUE
150;CONTINUE
TYPE 160
160;FORMAT(/,/, "THE INVERSE MATRIX IS"/)
JOG = 2
GO TO 200
161;STOP
C;PRINT OUT SUBROUTINE
200;ILNE = 4
NR=NRC=1
TYPE 201,NR
201;FORMAT(/, "ROW ", I, /)
I1 = 1
I2 = LAST - N + 1
KONT = 1
DO 210 JP = 1,N
DO 213 I = I1,I2,N
TYPE 212,A(I)
212;FORMAT(E)
IF(N-NRC) 216,216,215
216;IF(N-NR) 213,213,217
217;NR = NR+1
NRC = KONT = 1
ILNE = 4
TYPE 201,NR
GO TO 213
215;IF(KONT-ILNE) 214,220,214
220;ILNE = ILNE + 4
TYPE 221
221;FORMAT(/,/)
214;KONT = KONT + 1
NRC=NRC+1
213;CONTINUE
I1 = I1 + 1
I2 = I2 + 1
210;CONTINUE
GO TO (149,161),JOG
END

```

MATRIX INVERSION.... WILL HANDLE UP TO 12X12  
LOAD DATA TAPE --- HIT CONTINUE

SIZE OF MATRIX = 8

MATRIX ENTRIES IN ORDER A11,A12,A13,ETC.

+0.100000E+1  
+0.500000E+0  
+0.333333E+0  
+0.250000E+0  
+0.200000E+0  
+0.166666E+0  
+0.142857E+0  
+0.125000E+0  
+0.500000E+0  
+0.333333E+0  
+0.250000E+0  
+0.200000E+0  
+0.166666E+0  
+0.142857E+0  
+0.125000E+0  
+0.111111E+0  
+0.333333E+0  
+0.250000E+0  
+0.200000E+0  
+0.166666E+0  
+0.142857E+0  
+0.125000E+0  
+0.111111E+0  
+0.100000E+0  
+0.250000E+0  
+0.200000E+0  
+0.166666E+0  
+0.142857E+0

+0.125000E+0  
+0.111111E+0  
+0.100000E+0  
+0.909090E-1  
+0.200000E+0  
+0.166666E+0  
+0.142857E+0  
+0.125000E+0  
+0.111111E+0  
+0.100000E+0  
+0.909090E-1  
+0.833333E-1  
+0.166666E+0  
+0.142857E+0  
+0.125000E+0  
+0.111111E+0  
+0.100000E+0  
+0.909090E-1  
+0.833333E-1  
+0.769231E-1  
+0.142857E+0  
+0.125000E+0  
+0.111111E+0  
+0.100000E+0  
+0.909090E-1  
+0.833333E-1  
+0.769231E-1  
+0.714285E-1  
+0.125000E+0  
+0.111111E+0  
+0.100000E+0  
+0.909090E-1  
+0.833333E-1  
+0.769231E-1  
+0.714285E-1  
+0.666666E-1

# MATRIX INVERSION RESULTS

## ORIGINAL MATRIX VALUES

ROW +1				
+0.100000E+1	+0.500000E+0	+0.333332E+0	+0.250000E+0	
+0.200000E+0	+0.166666E+0	+0.142857E+0	+0.125000E+0	
ROW +2				
+0.500000E+0	+0.333332E+0	+0.250000E+0	+0.200000E+0	
+0.166666E+0	+0.142857E+0	+0.125000E+0	+0.111110E+0	
ROW +3				
+0.333332E+0	+0.250000E+0	+0.200000E+0	+0.166666E+0	
+0.142857E+0	+0.125000E+0	+0.111110E+0	+0.100000E+0	
ROW +4				
+0.250000E+0	+0.200000E+0	+0.166666E+0	+0.142857E+0	
+0.125000E+0	+0.111110E+0	+0.100000E+0	+0.909090E-1	
ROW +5				
+0.200000E+0	+0.166666E+0	+0.142857E+0	+0.125000E+0	
+0.111110E+0	+0.100000E+0	+0.909090E-1	+0.833332E-1	
ROW +6				
+0.166666E+0	+0.142857E+0	+0.125000E+0	+0.111110E+0	
+0.100000E+0	+0.909090E-1	+0.833332E-1	+0.769231E-1	
ROW +7				
+0.142857E+0	+0.125000E+0	+0.111110E+0	+0.100000E+0	
+0.909090E-1	+0.833332E-1	+0.769231E-1	+0.714285E-1	
ROW +8				
+0.125000E+0	+0.111110E+0	+0.100000E+0	+0.909090E-1	
+0.833332E-1	+0.769231E-1	+0.714285E-1	+0.666665E-1	

MATRIX INVERSION..... WILL HANDLE UP TO 12X12  
LOAD DATA TAPE --- HIT CONTINUE

SIZE OF MATRIX = 4

MATRIX ENTRIES IN ORDER A11,A12,A13,ETC.+0.100000E+1

+0.500000E+0  
+0.333333E+0  
+0.250000E+0  
+0.500000E+0  
+0.333333E+0  
+0.250000E+0  
+0.200000E+0  
+0.333333E+0  
+0.250000E+0  
+0.200000E+0  
+0.166666E+0  
+0.250000E+0  
+0.200000E+0  
+0.166666E+0  
+0.142857E+0

MATRIX INVERSION RESULTS

ORIGINAL MATRIX VALUES

ROW +1			
+0.100000E+1	+0.500000E+0	+0.333332E+0	+0.250000E+0
ROW +2			
+0.500000E+0	+0.333332E+0	+0.250000E+0	+0.200000E+0
ROW +3			
+0.333332E+0	+0.250000E+0	+0.200000E+0	+0.166666E+0
ROW +4			
+0.250000E+0	+0.200000E+0	+0.166666E+0	+0.142857E+0

THE INVERSE MATRIX IS

ROW +1			
+0.159725E+2	-0.119655E+3	+0.239128E+3	-0.139416E+3
ROW +2			
-0.119654E+3	+0.119573E+4	-0.268928E+4	+0.167286E+4
ROW +3			
+0.239126E+3	-0.268927E+4	+0.645314E+4	-0.418214E+4
ROW +4			
-0.139415E+3	+0.167284E+4	-0.418212E+4	+0.278812E+4

!



# MATRIX INVERSION RESULTS

## ORIGINAL MATRIX VALUES

ROW +1				
+0.100000E+1	+0.500000E+0	+0.333332E+0	+0.250000E+0	
+0.200000E+0				
ROW +2				
+0.500000E+0	+0.333332E+0	+0.250000E+0	+0.200000E+0	
+0.166666E+0				
ROW +3				
+0.333332E+0	+0.250000E+0	+0.200000E+0	+0.166666E+0	
+0.142857E+0				
ROW +4				
+0.250000E+0	+0.200000E+0	+0.166666E+0	+0.142857E+0	
+0.125000E+0				
ROW +5				
+0.200000E+0	+0.166666E+0	+0.142857E+0	+0.125000E+0	
+0.111110E+0				

## THE INVERSE MATRIX IS

ROW +1				
+0.126073E+2	-0.767119E+2	+0.111613E+3	-0.920288E+1	
-0.407754E+2				
ROW +2				
-0.515380E+2	+0.326738E+3	-0.108908E+3	-0.962137E+3	
+0.825129E+3				
ROW +3				
-0.694729E+2	+0.124880E+4	-0.524054E+4	+0.775907E+4	
-0.373930E+4				
ROW +4				
+0.342884E+3	-0.448185E+4	+0.140935E+5	-0.158744E+5	
+0.584403E+4				
ROW +5				
-0.241959E+3	+0.308767E+4	-0.916851E+4	+0.936258E+4	
-0.293182E+4	!			

THE INVERSE MATRIX IS

ROW +1			
+0.272634E+2	-0.357763E+3	+0.130014E+4	-0.141413E+4
-0.329614E+3	+0.312726E+3	+0.165345E+4	-0.119708E+4
ROW +2			
-0.356676E+3	+0.630141E+4	-0.254948E+5	+0.256585E+5
+0.241019E+5	-0.352589E+5	-0.169460E+5	+0.221324E+5
ROW +3			
+0.128535E+4	-0.252997E+5	+0.102323E+6	-0.569573E+5
-0.285921E+6	+0.410354E+6	-0.777362E+5	-0.688629E+5
ROW +4			
-0.135440E+4	+0.246516E+5	-0.540229E+5	-0.296322E+6
+0.124483E+7	-0.156502E+7	+0.665992E+6	-0.172452E+5
ROW +5			
-0.415422E+3	+0.255213E+5	-0.289462E+6	+0.124128E+7
-0.245451E+7	+0.228681E+7	-0.848236E+6	+0.380734E+5
ROW +6			
+0.321370E+3	-0.348104E+5	+0.401734E+6	-0.152542E+7
+0.222715E+7	-0.653729E+6	-0.113237E+7	+0.718552E+6
ROW +7			
+0.172594E+4	-0.191468E+5	-0.597868E+5	+0.604021E+6
-0.748710E+6	-0.119720E+7	+0.274985E+7	-0.133427E+7
ROW +8			
-0.123858E+4	+0.232816E+5	-0.774314E+5	+0.107290E+5
-0.766784E+4	+0.755308E+6	-0.134583E+7	+0.645083E+6

!

MATRIX INVERSION.... WILL HANDLE UP TO 12X12  
LOAD DATA TAPE --- HIT CONTINUE

SIZE OF MATRIX = 5

MATRIX ENTRIES IN ORDER A11,A12,A13,ETC.

+0.100000E+1  
+0.500000E+0  
+0.333333E+0  
+0.250000E+0  
+0.200000E+0  
+0.500000E+0  
+0.333333E+0  
+0.250000E+0  
+0.200000E+0  
+0.166666E+0  
+0.333333E+0  
+0.250000E+0  
+0.200000E+0  
+0.166666E+0  
+0.142857E+0  
+0.250000E+0  
+0.200000E+0  
+0.166666E+0  
+0.142857E+0  
+0.125000E+0  
+0.200000E+0  
+0.166666E+0  
+0.14257E+0  
+0.125000E+0  
+0.111111E+0