



REVISION NOTICE

This publication replaces previous descriptions of "Matrix Inversion 3," program D1-130.0. Program designations have been changed to their current nomenclature.

FUNCTION

"Matrix Inversion 3" enables the user to replace the elements of a square matrix with the elements of its inverse. Matrices of any rank greater than one will be accommodated. Program D1-130.0 is entered and left in machine language, but it uses the program Floating Point Interpretive System 2, H1-24.1, for all calculations.

INPUT

The elements of a square matrix are stored in consecutive locations on the drum, beginning with the fraction of element 1 in location M_0 . The elements must be in extended range floating point format 2 as defined by program H1-24.1.

OUTPUT

The elements of the inverse matrix are stored in consecutive locations, beginning in M_0 , in extended range floating point format 2.

MATRIX INVERSION 3

CALLING SEQUENCE

<u>Location</u>	<u>Order</u>	<u>Address</u>
a - 1	E	0000
a	R	$L_0 + 0014$
a + 1	U	L_0
a + 2	(n at 15)	M_0
a + 3	etc.	

The E0000 order in (a - 1) is required only if the previous instructions are interpreted by program H1-24.1. In (a + 2), n is the rank of the matrix.

TIME

Approximately $1.2 n^3$ seconds are required.

STORAGE

160 locations (2 1/2 tracks) are required in memory for storage of instructions and constants. No temporary storage is used except that required by program H1-24.1. Although only $2n^2$ locations are required for the matrix elements, the routine requires $2n^2 + 2n$ locations beginning in M_0 .

NOTES

When reading in "Matrix Inversion 3," it is necessary to supply as the last word read in the first location of program H1-24.1. The floating point instructions used by this routine are the following: 800Bxxxx, 800Dxxxx, 800Mxxxx, 800Hxxxx, 800Cxxxx, 800Axxxx, 800Sxxxx, U0000, and E0000.

30.0

D1-130.0

b0231'y0113'y0114'y0161'y0033'y0034'
y0121'y0122'y0162'y0208'y0209'b0014'
a0109'y0103'b0000'y0036'm0224'h0156'
s0207'h0222'a0106'h0107'n0222'm0221'
h0216'a0036'a0155'y0218'y0141'a0222'
y0037'y0210'y0111'r0000'u0000'800b0219'
800d0000'800h0000'xu0000'xe0000'b0036'
e0206'a0207'y0115'a0216'y0116's0111'
t0113'u0050'b0115'y0123'e0110's0218'
t0121'b0141'u0057'y0129'a0109'y0131'
u0227'y0130's0141't0129'b0222's0223'
h0222't0000'u0033'xz0004'xz0001'
xz0001'xs6363'm0000'xz0001'r0000'u0000'
800m0000'800e0000'xe0000'b0115'u0041''
r0000'u0000'800s0000'xu0000'xe0000'
b0141'y0163'u0137'b0000'e0000'b0000'
c0000'u0158'y0163'u0145'b0123'a0217'
y0200'u0142'b0000's0107'y0201'u0161'
e0110's0111't0149'u0151'b0200'u0138'
b0123's0217'y0211'u0208'xz0002''
b0109'a0131'u0057'r0000'u0000'800m0000'
800a0000'800h0000'xe0000'b0163'a0226'
u0134'xs6363'xz0002'r0000'u0000'800m0000'
800e0000'xe0000'b0156'a0123'u0051',0000011'
'8's0000'40000000'4'10000000'8'40000'
'8's0156'y0132's0108'u0061'00xz0000'.0000000'