

## MATRIX SOLUTION

### PURPOSE

Solve a system of linear equations in the following form:

$$A * X = B$$

where A is the matrix of coefficients, B is a vector of constants, and X are the variables to be solved for.

### DESCRIPTION

In practice, solving a system of linear equations, finding the inverse of a matrix, and computing the determinant of a matrix are all closely related. Each of these can be computed from the LU decomposition of a matrix.

### SYNTAX

LET <var2> = MATRIX SOLUTION <mat> <var1> <SUBSET/EXCEPT/FOR qualification>

where <mat> is a matrix containing the coefficients of the equation;

<var1> is a vector of constants (i.e., the values for the right hand side of the equation);

<var2> is a vector where the resulting matrix solution is saved;

and where the <SUBSET/EXCEPT/FOR qualification> is optional and rarely used in this context.

### EXAMPLES

LET X = MATRIX SOLUTION A B

### NOTE 1

Matrices used to solve systems of linear equations must have the same number of rows and columns. An error message is printed if they do not.

### NOTE 2

DATAPLOT uses LINPACK routines to calculate the LU decomposition (versions prior to 7/93 use the LUDCMP and LUBKSB routines from the Numerical Recipes book).

The reciprocal of the condition number is printed. This number gives an indication of the numerical accuracy that was obtained when calculating the inverse. If this number is approximately  $10^{**(-d)}$ , then the elements of the LU decomposed matrix generally have d fewer significant digits than the original matrix.

The MATRIX ITERATIVE SOLUTION command can be used to generate a solution vector with higher precision.

### NOTE 3

The solution of a system of equations can often be significantly speeded up and the storage requirements reduced for special types of matrices. The general algorithm typically works on these cases, it just takes longer. DATAPLOT currently supports two specialized types of matrices. The TRIDIAGONAL SOLVE and TRIANGULAR SOLVE can be used to solve tridiagonal and triangular systems of equations. These commands are faster than MATRIX SOLUTION when they apply, and in the case of TRIDIAGONAL SOLVE can be used to handle larger systems of equations.

### NOTE 4

DATAPLOT does not currently support any methods involving complex matrices or any sparse matrix techniques.

### DEFAULT

None

### SYNONYMS

None

### RELATED COMMANDS

MATRIX DETERMINANT	=	Compute a matrix determinant.
MATRIX INVERSE	=	Compute a matrix inverse.
MATRIX SIMPLEX SOLUTION	=	Compute a matrix simplex solution.
TRIANGULAR INVERSE	=	Compute the inverse of a triangular matrix.
TRIANGULAR SOLVE	=	Solve a triangular system of linear equations.
TRIDIAGONAL SOLVE	=	Solve a tridiagonal system of equations.

## REFERENCE

“LINPACK User’s Guide,” Dongarra, Bunch, Moler, Stewart. Siam, 1979.

“Numerical Recipes: The Art of Scientific Programming (FORTRAN Version),” Press, Flannery, Teukolsky, and Vetterling. Cambridge University Press, 1989 (chapter 2).

## APPLICATIONS

Linear Algebra

## IMPLEMENTATION DATE

87/10

## PROGRAM

```
. PURPOSE--DETERMINE PROPER FEED RATE OF 4 INPUT COMPOSITIONS
. TO YIELD A PRE-SPECIFIED OUTPUT COMPOSITION AT A PRE-SPECIFIED OUTPUT RATE.
. ANALYSIS TECHNIQUE--SOLVING A SYSTEM OF 4 LINEAR EQUATIONS
. APPLICATION--CHEMICAL MIXING
. SOURCE--FOGIEL, THE LINEAR ALGEBRA PROBLEM SOLVER RESEARCH
. RESEARCH AND EDUCATION ASSOCIATION, 1980 (PAGE 792)
. GIVEN--INPUT PROPORTIONS-- IN1 IN2 IN3 IN4
. SULFURIC ACID 80% 0% 30% 10%
. NITRIC ACID 0% 80% 10% 10%
. WATER 16% 20% 60% 72%
. INERT 4% 0% 0% 8%
. GIVEN--DESIRED OUTPUT (2000 LB/HR) AT THE FOLLOWING PROPORTIONS--
. SULFURIC ACID 40% (800 LB PER HOUR)
. NITRIC ACID 27% (540 LB PER HOUR)
. WATER 31% (620 LB PER HOUR)
. INERT 2% (40 LB PER HOUR)
. TO FIND X1 = FLOW RATE OF INPUT STREAM 1
. X2 = FLOW RATE OF INPUT STREAM 2
. X3 = FLOW RATE OF INPUT STREAM 3
. X4 = FLOW RATE OF INPUT STREAM 4
. NOTE--TO SET UP THE 4 EQUATIONS (ONE FOR EACH COMPOUND)
. APPLY CONSERVATION OF MASS (MASS BALANCES)
. NOTE--FOR TESTING PURPOSES, THE SOLUTION IS
. 800 600 500 100 LB PER HOUR
. STEP 1-- DEFINE THE LEFT-HAND SIDE.
READ MATRIX A
.80 .00 .30 .10
.00 .80 .10 .10
.16 .20 .60 .72
.04 .00 .00 .08
END OF DATA
. STEP 2-- DEFINE THE RIGHT-HAND SIDE.
LET B = DATA 800 540 620 40
PRINT A B
. STEP 3-- SOLVE THE LINEAR SYSTEM
LET X = MATRIX SOLUTION A B
PRINT X
```

The solution vector 800, 600, 500, 100 is generated.