This routine is provided for completeness only and is not recommended for general work (use LU decomposition — see **MatrixLUD**). MatrixGaussJ does calculate the inverse matrix but that is not generally needed either.

See Also

Matrix Math Operations on page III-138 for more about Igor's matrix routines. The **MatrixLUD** operation.

MatrixGLM

MatrixGLM [/Z] matrixA, matrixB, waveD

The MatrixGLM operation solves the general Gauss-Markov Linear Model problem (GLM) which minimizes the 2-norm of a vector y

$$\min \|y\|_2 \quad subject\,to \quad d = Ax + By.$$

A is *matrixA* (an NxM wave), B is *matrixB* (an NxP wave), and d is provided by *waveD* which is a 1D wave of N rows. The vectors x and y are the results of the calculation; they are stored in output waves Mat_X and Mat_Y in the current data folder.

Flags

/Z	In the event of an error, MatrixGLM will not return the error to Igor, which would cause procedure execute to abort. Your code should use the V_flag output variable to detect and handle errors.

Details

All input waves must have the same numeric type. Supported types are single-precision and double-precision floating point, both real and complex. The output waves Mat_X and Mat_Y have the same numeric type as the input.

The LAPACK algorithm assumes that M <= N <= M+P and

$$rank(A) = M,$$

$$rank(AB) = N.$$

Under these assumptions there is a unique solution x and a minimal 2-norm solution y, which are obtained using a generalized QR factorization of A and B. If the operation completes successfully the variable V_Flag is set to zero. Otherwise it contains a LAPACK error code.

Output Variables

V_flag	Set to 0 if MatrixGLM succeeds or to a LAPACK error code.

See Also

Matrix Math Operations on page III-138 for more about Igor's matrix routines and for background references with details about the LAPACK libraries.

MatrixInverse

MatrixInverse [***flags***] ***srcWave***

The MatrixInverse operation calculates the inverse or the pseudo-inverse of a matrix. *srcWave* may be real or complex.

MatrixInverse saves the result in the wave M_Inverse in the current data folder.