method	**Solution Method**
GJ	Gauss Jordan.
LU	LU decomposition.
SV	Singular Value decomposition.

Details

The array b can be a matrix containing a number of b vectors and the output matrix M_x will contain a corresponding set of solution vectors.

V_flag is set to zero if success, 1 if singular matrix using GJ or LU and 1 if SV fails to converge.

For normal problems you should use LU. GJ is provided only for completeness and has no practical use.

When using SV, singular values smaller than 10^{-6} times the largest singular value are set to zero before back substitution.

Generates an error if the dimensions of the input matrices are not appropriate.

See Also

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

MatrixSparse

`MatrixSparse [`*`flags`*`] keyword=`*`value`*

The MatrixSparse operation provides support for basic matrix operations on sparse matrices. The MatrixSparse operation was added in Igor Pro 9.00.

The following sections use terms and concepts explained under Sparse Matrices. You will need to understand those terms and concepts to understand the following material.

Flags

/LM	Limited memory hint. Specifies that the operation can allocate limited memory proportional to vector size. If this flag is omitted, the operation allocates memory aggressively up to the size of the sparse matrix.
/Q	Quiet - do not print diagnostic information to the history area.
/Z	Errors are not fatal and do not abort procedure execution. Your procedure can inspect the V_flag variable to see if the operation succeeded. V_flag will be zero if it succeeded or nonzero if it failed.

Keywords

alpha=*value*	Specifies the real part of the alpha value for the MM, MV, and TRSV operations. The default value is 1.
alphai=*value*	Specifies the imaginary part of the alpha value for the MM, MV, and TRSV operations on complex matrices. The default value is 0. alphai should be specified only when operating on complex matrices.
beta=*value*	Specifies the real part of the beta value for the MM and MV operations. The default value is 0.
betai=*value*	Specifies the imaginary part of the beta value for the MM and MV operations. The default value is 0. betai should be specified only when operating on complex matrices.
colsA=*nCols*	Specifies the number of columns of the sparse matrix A. colsA is a required keyword for all MatrixSparse commands even in the rare cases where neither cooA, cscA, nor csrA are used.

colsG=*nCols* Specifies the number of columns of the sparse matrix G.

colsG is a required keyword for operations involving sparse matrix G.

cooA={*valuesWave*, *rowsWave*, *colsWave*}

Specifies sparse matrix A in COO format using three 1D waves. See **Sparse Matrix Formats** on page III-151 for background information on cooA format.

The COO format is used only for conversion operations, not for math operations which always use CSR format.

valuesWave is an wave of length *nnz* (number of non-zero values) containing the non-zero values of A. The wave must be single or double precision real or complex and must not contain any NaNs or INFs.

rowsWave and *colsWave* are 64-bit integer 1D waves created with Make/L that contain nnz row and column indices respectively.

When using cooA, you must also specify the rowsA and colsA keywords.

cscA={*valuesWave*, *rowsWave*, *ptrWave*}

Specifies sparse matrix A in CSC format A using three 1D waves. See **Sparse Matrix Formats** on page III-151 for background information on cscA format.

The CSC format is used only for conversion operations, not for math operations which always use CSR format.

valuesWave is an wave of length *nnz* (number of non-zero values) containing the non-zero values of A. The wave must be single or double precision real or complex and must not contain any NaNs or INFs.

rowsWave and *ptrWave* are 64-bit integer 1D waves created with Make/L.

rowsWave consists of nnz entries that specify the zero-based row index of each entry in *valuesWave*.

ptrWave contains nCols or nCols+1 entries the first of which must be zero. *ptrWave*[j] gives the index in valuesWave of the first non-zero value in column j. The optional longer version of *ptrWave* stores nnz in the last wave point.

When using cscA, you must also specify the rowsA and colsA keywords.

csrA={*valuesWave*, *colsWave*, *ptrWave*}

Specifies sparse matrix A in CSR format using three 1D waves. See **Sparse Matrix Formats** on page III-151 for background information on csrA format.

The CSR format is used for conversion operations and for math operations.

valuesWave is an wave of length *nnz* (number of non-zero values) containing the non-zero values of A. The wave must be single or double precision real or complex and must not contain any NaNs or INFs.

colsWave and *ptrWave* are 64-bit integer 1D waves created with Make/L.

colsWave consists of nnz entries that specify the zero-based column index of each entry in *valuesWave*.

ptrWave contains nRows or nRows+1 entries the first of which must be zero. *ptrWave*[i] gives the index in *valuesWave* of the first non-zero value of row i. The optional longer version of *ptrWave* stores nnz in the last wave point.

When using csrA, you must also specify the rowsA and colsA keywords.

cooG={*valuesWave*, *rowsWave*, *colsWave*}

Similar to cooA but applies to sparse matrix G.

cscG={*valuesWave*, *rowsWave*, *ptrWave*}

Similar to cscA but applies to sparse matrix G.

csrG={*valuesWave*, *colsWave*, *ptrWave*}	
	Similar to csrA but applies to sparse matrix G.
matrixB=*wb*	Designates the 2D wave *wb* as matrix B for the MM operation.
	The wave *wb* must be of the same data type as sparse matrix A and must not contain INFs or NaNs.
matrixC=*wc*	Designates the 2D wave *wc* as matrix C for the MM operation.
	The wave *wc* must be of the same data type as sparse matrix A and must not contain INFs or NaNs.
opA=*opName*	Specifies a transformation that is applied to sparse matrix A. *opName* is a single letter:
	T: Transpose
	H: Hermitian
	N: No transformation (default)
	See **MatrixSparse Transformations** on page III-156 for details.
opG=*opName*	Specifies a transformation that is applied to sparse matrix G. *opName* is a single letter:
	T: Transpose
	H: Hermitian
	N: No transformation (default)
	See **MatrixSparse Transformations** on page III-156 for details.
operation=*opN*	Specifies the operation name. This is a required keyword.
	opN is one of the following:

ADD	Adds sparse matrices. See **MatrixSparse ADD** on page III-157 for details.
MM	Computes the product of a sparse matrix and a dense matrix. See **MatrixSparse MM** on page III-157 for details.
MV	Computes the product of a sparse matrix and a vector. See **MatrixSparse MV** on page III-158 for details.
SMSM	Computes the product of two sparse matrices. See **MatrixSparse SMSM** on page III-158 for details.
TOCOO	Creates a sparse matrix in COO format from a sparse matrix or dense matrix. See **MatrixSparse TOCOO** on page III-159 for details.
TOCSC	Creates a sparse matrix in CSC format from a sparse matrix or dense matrix. See **MatrixSparse TOCSC** on page III-159 for details.
TOCSR	Creates a sparse matrix in CSR format from a sparse matrix or dense matrix. See **MatrixSparse TOCSR** on page III-160 for details.
TODENSE	Creates a dense matrix from a sparse matrix. See **MatrixSparse TODENSE** on page III-160 for details.
TRSV	Solves a system of linear equations for a triangular sparse matrix. See **MatrixSparse TRSV** on page III-161 for details.

rowsA=*nRows*	Specifies the number of rows of the sparse matrix A.
	rowsA is a required keyword for all MatrixSparse commands even in the rare cases where neither cooA, cscA, nor csrA are used.
rowsG=*nRows*	Specifies the number of rows of the sparse matrix G.
	rowsG is a required keyword for operations involving sparse matrix G.