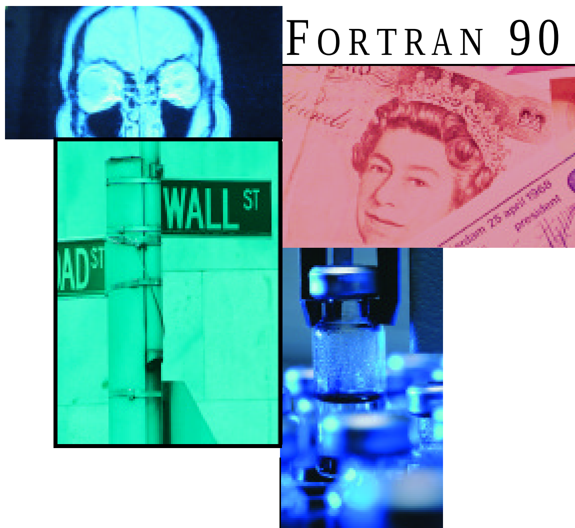
IMSL®

FORTRAN 90 MP LIBRARY



FUNCTION CATALOG

Visual Numerics

Developer of IMSL® and WAVE

[www.vni.com]

TABLE OF CONTENTS

IMSL FORTRAN 90 MP LIBRARY

Written for Fortran programmers and based on the world's most widely called numerical subroutines.

The IMSL Fortran 90 MP Library (F90MP) is a comprehensive set of over 1,000 pre-built mathematical and statistical analysis functions that Fortran programmers can embed directly into their numerical analysis applications. F90MP's functions are based upon the technology in the IMSL repository of algorithms. Visual Numerics has been providing algorithms for mathematical and statistical computations under the IMSL name since 1970.

- With F90MP, we provide "building blocks" which eliminate the need to write code from scratch. These pre-written functions allow you to focus on your expertise and reduce your development time.

- F90MP takes full advantage of the intrinsic characteristics and desirable features of the Fortran 90 language. Over 70 routines have been designed, not translated, to take full advantage of the array operators of Fortran 90.

- F90MP also supports FORTRAN 77 syntax. All of the functions in the IMSL FORTRAN Numerical Library are provided to ease your migration from FORTRAN 77 to Fortran 90.

- F90MP has also been designed to take advantage of symmetric multiprocessor (SMP) systems. Computationally intensive algorithms in areas such as linear algebra and fast Fourier transforms will leverage SMP capabilities on a variety of systems. By allowing you to replace the generic Basic Linear Algebra Subprograms ("BLAS") contained in F90MP with optimized BLAS from your hardware vendor, you can improve the performance of your numerical calculations.

- You can build applications that are portable across multiple platforms. F90MP is available for computer systems running UNIX and Windows operating systems.

- Extensive online documentation provides powerful search capabilities with hundreds of code examples.

Rely on the industry leader for software that is expertly developed, thoroughly tested, meticulously maintained and well documented. GET RELIABLE RESULTS EVERY TIME!

COST-EFFECTIVENESS AND VALUE

F90MP significantly shortens program development time and promotes standardization. You'll find that using F90MP saves time in your source code development and saves thousands of dollars in the design, development, documentation, testing and maintenance of your applications.

WIDE COMPATIBILITY AND UNIFORM OPERATION

The IMSL Fortran 90 MP Library is available for UNIX computing environments and Windows NT/95/98. Visual Numerics' commitment to regular feature and enhancement updates:

- Ensures that your software will perform to the highest standards.
- Provides for portable applications.
- Assures that Visual Numerics will keep pace with the latest hardware and software innovations.

ERROR HANDLING

Diagnostic error messages are clear and informative – designed not only to convey the error condition but also to suggest corrective action if appropriate. These error-handling features:

- Make it faster and easier for you to debug your programs.
- Provide for more productive programming and confidence that the algorithms are functioning properly in your application.

COMPREHENSIVE DOCUMENTATION

Documentation for F90MP is comprehensive, clearly written and standardized. Detailed information about each function is found in a single source within a chapter and consists of section name, purpose, synopsis, errors, return values and usage examples. Each manual's alphabetical index enables convenient cross-referencing. IMSL documentation

- Provides organized, easy-to-find information.
- Extensively documents, explains and provides references for algorithms.
- Gives at least one example of function usage, with sample input and results.

FLEXIBLE LICENSING OPTIONS

The IMSL Fortran 90 MP Library can be licensed in a number of flexible ways: licenses may be node-locked to a specific CPU, or a specified number of licenses can be purchased to "float" throughout a heterogeneous network as they are needed. This allows you to cost-effectively acquire as many seats as you need today, adding more seats when it becomes necessary. Site licenses and campus licenses are also available.

UNMATCHED PRODUCT SUPPORT

Behind every Visual Numerics license is a team of professionals ready to provide expert answers to questions about your IMSL software. Product support options include product maintenance and consultation, ensuring value and performance of your IMSL software.

Product support:

- Gives you direct access to Visual Numerics resident staff of expert product support specialists.

- Provides prompt, two-way communication with solutions to your programming needs.
- Includes product maintenance updates.

MATHEMATICAL FUNCTIONS

The IMSL Fortran 90 MP Library is a collection of the most commonly needed numerical functions customized for your programming needs. The mathematical functionality is organized into 10 sections. These capabilities range from solving systems of linear equations to optimization.

- Linear Systems, including real and complex full and sparse matrices, linear least squares, matrix decompositions, generalized inverses and vector-matrix operations.
- Eigensystem Analysis, including eigenvalues and eigenvectors of complex, real symmetric and complex Hermitian matrices.
- Interpolation and Approximation, including constrained curve-fitting splines, cubic splines, least-squares approximation and smoothing, and scattered data interpolation.
- Integration and Differentiation, including univariate, multivariate and Gauss quadrature.
- Differential Equations, using Adams-Gear and Runge-Kutta methods for stiff and nonstiff ordinary differential equations and support for partial differential equations.
- Transforms, including real and complex one- and two-dimensional fast Fourier transforms, as well as convolutions and correlations and Laplace transforms.
- Nonlinear Equations, including zeros and root finding of polynomials, zeros of a function and root of a system of equations.
- Optimization, including unconstrained, and linearly and nonlinearly constrained minimizations.
- Basic Matrix/Vector Operations, including Basic Linear Algebra Subprograms (BLAS) and matrix manipulation operations.
- Utilities, including CPU time used, error handling and machine, mathematical, physical constants, retrieval of machine constants and changing error-handling.

MATHEMATICAL SPECIAL FUNCTIONS

The IMSL Fortran 90 MP Library includes routines that evaluate the special mathematical functions that arise in applied mathematics, physics, engineering and other technical fields. The mathematical special functions are organized into 12 sections.

- Elementary Functions, including complex numbers, exponential functions and logarithmic functions.
- Trigonometric and Hyperbolic Functions, including trigonometric functions and hyperbolic functions.
- Exponential Integrals and Related Functions, including exponential integrals, logarithmic integrals and integrals of trigonometric and hyperbolic functions.
- Gamma Functions and Related Functions, including gamma functions, psi functions, Pochhammer's function and Beta functions.
- Error Functions and Related Functions, including error functions and Fresnel integrals.
- Bessel Functions, including real order complex valued Bessel functions.
- Kelvin Functions, including Kelvin functions and their derivatives.
- Airy Functions, including Airy functions and their derivatives.
- Elliptic Integrals, including complete and incomplete elliptic integrals.
- Elliptic and Related Functions, including Weierstrass P-functions and the Jacobi elliptic function.

- Probability Distribution Functions and Inverses, including statistical functions, such as chi-squared and inverse beta and many others.
- Mathieu Functions, including eigenvalues and sequence of Mathieu functions.

STATISTICAL FUNCTIONALITY

The statistical functionality is organized into 17 sections. These capabilities range from analysis of variance to random number generation.

- Basic Statistics, including univariate summary statistics, nonparametric tests, such as sign and Wilcoxon rank sum, and goodness-of-fit tests, such as chi-squared and Shapiro-Wilk.
- Regression, including stepwise regression, all best regression, multiple linear regression models, polynomial models and nonlinear models.
- Correlation, including sample variance-covariance, partial correlation and covariances, pooled variance-covariance and robust estimates of a covariance matrix and mean factor.
- Analysis of Variance, including one-way classification models, a balanced factorial design with fixed effects and the Student-Newman-Keuls multiple comparisons test.
- Categorical and Discrete Data Analysis, including chi-squared analysis of a two-way contingency table, exact probabilities in a two-way contingency table and analysis of categorical data using general linear models.
- Nonparametric Statistics, including sign tests, Wilcoxon sum tests and Cochran Q test for related observations.
- Tests of Goodness-of-Fit and Randomness, including chi-squared goodness-of-fit tests, Kolmogorov/Smirnov tests and tests for normality.
- Time Series Analysis and Forecasting, including analysis and forecasting of time series using a nonseasonal ARMA model and difference of a seasonal or nonseasonal time series.
- Covariance Structures and Factor Analysis, including principal components and factor analysis.
- Discriminant Analysis, including analysis of data using a generalized linear model and using various parametric models.
- Cluster Analysis, including hierarchical cluster analysis and k-means cluster analysis.
- Sampling, including analysis of data using a simple or stratified random sample.
- Survival Analysis, Life Testing and Reliability, including Kaplan-Meier estimates of survival probabilities.
- Multidimensional Scaling, including alternating least squares methods.
- Density and Hazard Estimation, including estimates for density and modified likelihood for hazards.
- Probability Distribution Functions and Inverses, including binomial, hypergeometric, bivariate normal, gamma and many more.
- Random Number Generation, including a generator for multivariate normal distributions and pseudorandom numbers from several distributions, including gamma, Poisson and beta.

IMSL FORTRAN 90 MP LIBRARY SUBROUTINES

The following subroutines are available in single, double, complex single and complex double precision, unless noted otherwise.

Linear Solvers

lin_sol_gen
Solves a general system of linear equations $Ax = b$.

lin_sol_self
Solves a system of linear equations $Ax = b$, where A is a self-adjoint matrix.

lin_sol_lsq
Solves a rectangular system of linear equations $Ax \cong b$, in a least-squares sense.

lin_sol_svd
Solves a rectangular least-squares system of linear equations $Ax \cong b$ using singular value decomposition $A=USV^T$.

lin_sol_tri
Solves multiple systems of linear equations $A_j x_j = y_j$, j=1,..., k.

Singular Value and Eigenvalue Decomposition

lin_svd
Computes the singular value decomposition (SVD) of a rectangular matrix, A.

lin_eig_self
Computes the eigenvalues of a self-adjoint matrix, A.

lin_eig_gen
Computes the eigenvalues of an $n \times n$ matrix, A.

lin_geig_gen
Computes the generalized eigenvalues of an $n \times n$ matrix pencil, $Av = \lambda Bv$.

Fourier Transforms

fast_dft
Computes the Discrete Fourier Transform (DFT) of a rank-1 complex array, x.

fast_2dft
Computes the Discrete Fourier Transform (2DFT) of a rank-2 complex array, x.

fast_3dft
Computes the Discrete Fourier Transform (2DFT) of a rank-3 complex array, x.

Curve and Surface Fitting with Splines

spline_constraints
Returns the derived type array result, `s_d_spline_constraints`, given optional input.

spline_values
Returns an array result, given an array of input.

spline_fitting
Performs weighted least-squares fitting by B-splines to discrete one-dimensional data.

surface_constraints
Returns the derived type array result, `s_d_surface_constraints`, given optional input.

surface_values
This rank-2 array function returns a tensor product array result, given two arrays of independent variable values.

surface_fitting
Weighted least-squares fitting by tensor product B-splines to discrete two-dimensional data is performed.

Utilities

error_post
Prints error messages that are generated by IMSL MP Library routines.

rand_gen
Generates a rank-1 array of random numbers. The output array entries are positive and less than 1 in value.

sort_real
Sorts a rank-1 array of real numbers x so the y results are algebraically nondecreasing, $y_1 \le y_2 \le \ldots y_n$.

show
Prints rank-1 or rank-2 arrays of numbers in a readable format.

Matrix Algebra Operations

Defined Array Operation	Matrix Operation	Alternative is Fortran 90
`A .x. B`	AB	`matmul(A, B)`
`.i. A`	A^{-1}	`lin_sol_gen` `lin_sol_lsq`
`.t. A, .h. A`	A^T, A^H	`transpose(A)` `conjg(transpose(A))`
`A .ix. B`	$A^{-1} B$	`lin_sol_gen` `lin_sol_lsq`
`B .xi. A`	BA^{-1}	`lin_sol_gen` `lin_sol_lsq`
`A .tx. B, or (.t. A) .x. B` `A .hx. B, or (.h. A) .x. B`	A^TB, A^HB	`matmul(transpose (A), B)` `matmul(conjg(transpose(A)), B)`
`B .xt. A, or B .x. (.t. A)` `B .xh. A, or B .x. (.h. A)`	BA^T, BA^H	`matmul(B, transpose(A))` `matmul(B, conjg(transpose(A)))`

Matrix and Utility Functions

Defined Array Functions	Matrix Operation
`S=SVD(A [,U=U, V=V])`	$A=USV^T$
`E=EIG(A [[,B=B, D=D],`	$(AV=VE), AVD=BVE$
`V=V, W=W])`	$(AW=WE), AWD=BWE$
`R=CHOL(A)`	$A=R^TR$
`Q=ORTH(A [,R=R])`	$(A=QR), Q^TQ=1$
`U=UNIT(A)`	$[u_1, \ldots] = [a_1/\|a_1\|, \ldots]$
`F=DET(A)`	$det(A) = determinant$
`K=RANK(A)`	$rank(A) = rank$
`P=NORM(A[,[type=]i])`	$p=\|A\|_1 = \max_j (\sum_{i=1}^{m} \lvert a_{ij} \rvert)$ $p=\|A\|_2 = s_1$ = largest singular value $p=\|A\|_{\infty \leftrightarrow \text{huge}(1)} = \max_i (\sum_{j=1}^{n} \lvert a_{ij} \rvert)$
`C=COND(A)`	$C = s_1 / s_{rank(A)}$
`Z=EYE(N)`	$z = I_N$
`A=DIAG(X)`	$A = diag(x_1, \ldots)$
`X=DIAGONALS(A)`	$x = (a_{11}, \ldots)$
`Y=FFT (X,[WORK=W]);` `X=IFFT(Y,[WORK=W])`	Discrete Fourier Transform, Inverse
`A=RAND(A)`	random numbers, 0 < A < 1
`L=isNaN(A)`	test for NaN, if (1) then...

Operators and Generic Functions

Operators: .x., .tx., .xt., .hx., .xh.
Computes matrix-vector and matrix-matrix products.

Operators: .t., .h.
Computes transpose and conjugate transpose of a matrix.

Operator: .i.
Computes the inverse matrix, for square non-singular matrices, or the Moore-Penrose generalized inverse matrix for singular square matrices or rectangular matrices.

Operators: .ix., .xi.
Computes the inverse matrix times a vector or matrix for square non-singular matrices or the corresponding Moore-Penrose generalized inverse matrix for singular square matrices or rectangular matrices.

CHOL
Computes the Cholesky factorization of a positive-definite, symmetric or self-adjoint matrix, A. The factor is upper triangular, $R^T R = A$.

COND
Computes the condition number of a rectangular matrix, A. The condition number is the ratio of the largest and the smallest positive singular values, $s_1 / s_{rank(A)}$ or `huge` (A), whichever is smaller.

DET
Computes the determinant of a rectangular matrix, A. The evaluation is based on the QR decomposition, $QAP = \begin{bmatrix} R_{k \times k} & 0 \\ 0 & 0 \end{bmatrix}$ and $k = rank(A)$.
Thus $det(A) = s \times det(R)$ where $s = det(Q) \times det(P) = \pm 1$.

DIAG
Constructs a square diagonal matrix from a rank-1 array or several diagonal matrices from a rank-2 array.

DIAGONALS
Extracts a rank-1 array whose values are the diagonal terms of a rank-2 array argument.

EIG
Computes the eigenvalue-eigenvector decomposition of an ordinary or generalized eigenvalue problem.

EYE
Creates a rank-2 square array whose diagonals are all the value one. The off-diagonals all have value zero.

FFT
The Discrete Fourier Transform of a complex sequence and its inverse transform.

IFFT
The inverse of the Discrete Fourier Transform of a complex sequence.

isNaN
This is a generic logical function used to test scalars or arrays for occurrence of an IEEE 754 Standard format of floating point (ANSI/IEEE 1985) NaN, or not-a-number.

NaN
Returns, as a scalar function, a value corresponding to the IEEE 754 Standard format of floating point (ANSI/IEEE 1985) for NaN. For other floating point formats a special pattern is returned that tests `.true.` using the function `isNaN( )`.

NORM
Computes the norm of a rank-1 or rank-2 array. For rank-3 arrays, the norms of each rank-2 array, in dimension 3, are computed.

ORTH
Orthogonalizes the columns of a rank-2 or rank-3 array. The decomposition $A = QR$ is computed using a forward and backward sweep of the Modified Gram-Schmidt algorithm.

RAND
Computes a scalar, rank-1, rank-2 or rank-3 array of random numbers. Each component number is positive and strictly less than one in value.

RANK
Computes the mathematical rank of a rank-2 or rank-3 array.

SVD
Computes the singular value decomposition of a rank-2 or rank-3 array, $A = USV^T$.

UNIT
Normalizes the columns of a rank-2 or rank-3 array so each has Euclidean length of value one.

Partial Differential Equations

PDE_1D_MG
Solves a system of partial differential equations

$$u_t \equiv \frac{\partial u}{\partial t} = f(u, \ldots) \quad \text{[illegible]}$$

IMSL MATH/LIBRARY

LINEAR SYSTEMS

Solution of Linear Systems, Matrix Inversion and Determinant Evaluation

REAL GENERAL MATRICES

LSARG/DLSARG (Single/Double precision)
Solves a real general system of linear equations with iterative refinement.

LSLRG/DLSLRG (Single/Double precision)
Solves a real general system of linear equations without iterative refinement.

LFCRG/DLFCRG (Single/Double precision)
Computes the *LU* factorization of a real general matrix and estimates its L_1 condition number.

LFTRG/DLFTRG (Single/Double precision)
Computes the *LU* factorization of a real general matrix.

LFSRG/DLFSRG (Single/Double precision)
Solves a real general system of linear equations given the *LU* factorization of the coefficient matrix.

LFIRG/DLFIRG (Single/Double precision)
Uses iterative refinement to improve the solution of a real general system of linear equations.

LFDRG/DLFDRG (Single/Double precision)
Computes the determinant of a real general matrix given the *LU* factorization of the matrix.

LINRG/DLINRG (Single/Double precision)
Computes the inverse of a real general matrix.

COMPLEX GENERAL MATRICES

LSACG/DLSACG (Single/Double precision)
Solves a complex general system of linear equations with iterative refinement.

LSLCG/DLSLCG (Single/Double precision)
Solves a complex general system of linear equations without iterative refinement.

LFCCG/DLFCCG (Single/Double precision)
Computes the *LU* factorization of a complex general matrix and estimates its L_1 condition number.

LFTCG/DLFTCG (Single/Double precision)
Computes the *LU* factorization of a complex general matrix.

LFSCG/DLFSCG (Single/Double precision)
Solves a complex general system of linear equations given the *LU* factorization of the coefficient matrix.

LFICG/DLFICG (Single/Double precision)
Uses iterative refinement to improve the solution of a complex general system of linear equations.

LFDCG/DLFDCG (Single/Double precision)
Computes the determinant of a complex general matrix given the *LU* factorization of the matrix.

LINCG/DLINCG (Single/Double precision)
Computes the inverse of a complex general matrix.

REAL TRIANGULAR MATRICES

LSLRT/DLSLRT (Single/Double precision)
Solves a real triangular system of linear equations.

LFCRT/DLFCRT (Single/Double precision)
Estimates the condition number of a real triangular matrix.

LFDRT/DLFDRT (Single/Double precision)
Computes the determinant of a real triangular matrix.

LINRT/DLINRT (Single/Double precision)
Computes the inverse of a real triangular matrix.

COMPLEX TRIANGULAR MATRICES

LSLCT/DLSLCT (Single/Double precision)
Solves a complex triangular system of linear equations.

LFCCT/DLFCCT (Single/Double precision)
Estimates the condition number of a complex triangular matrix.

LFDCT/DLFDCT (Single/Double precision)
Computes the determinant of a complex triangular matrix.

LINCT/DLINCT (Single/Double precision)
Computes the inverse of a complex triangular matrix.

REAL POSITIVE DEFINITE MATRICES

LSADS/DLSADS (Single/Double precision)
Solves a real symmetric positive definite system of linear equations with iterative refinement.

LSLDS/DLSLDS (Single/Double precision)
Solves a real symmetric positive definite system of linear equations without iterative refinement.

LFCDS/DLFCDS (Single/Double precision)
Computes the $R^T R$ Cholesky factorization of a real symmetric positive definite matrix and estimates its L_1 condition number.

LFTDS/DLFTDS (Single/Double precision)
Computes the $R^T R$ Cholesky factorization of a real symmetric positive definite matrix.

LFSDS/DLFSDS (Single/Double precision)
Solves a real symmetric positive definite system of linear equations given the $R^T R$ Cholesky factorization of the coefficient matrix.

LFIDS/DLFIDS (Single/Double precision)
Uses iterative refinement to improve the solution of a real symmetric positive definite system of linear equations.

LFDDS/DLFDDS (Single/Double precision)
Computes the determinant of a real symmetric positive definite matrix given the $R^T R$ Cholesky factorization of the matrix.

LINDS/DLINDS (Single/Double precision)
Computes the inverse of a real symmetric positive definite matrix.

Real Symmetric Matrices

LSASF/DLSASF (Single/Double precision)
Solves a real symmetric system of linear equations with iterative refinement.

LSLSF/DLSLSF (Single/Double precision)
Solves a real symmetric system of linear equations without iterative refinement.

LFCSF/DLFCSF (Single/Double precision)
Computes the $U D U^T$ factorization of a real symmetric matrix and estimates its L_1 condition number.

LFTSF/DLFTSF (Single/Double precision)
Computes the $U D U^T$ factorization of a real symmetric matrix.

LFSSF/DLFSSF (Single/Double precision)
Solves a real symmetric system of linear equations given the $U D U^T$ factorization of the coefficient matrix.

LFISF/DLFISF (Single/Double precision)
Uses iterative refinement to improve the solution of a real symmetric system of linear equations.

LFDSF/DLFDSF (Single/Double precision)
Computes the determinant of a real symmetric matrix given the $U D U^T$ factorization of the matrix.

Complex Hermitian Positive Definite Matrices

LSADH/DLSADH (Single/Double precision)
Solves a Hermitian positive definite system of linear equations with iterative refinement.

LSLDH/DLSLDH (Single/Double precision)
Solves a complex Hermitian positive definite system of linear equations without iterative refinement.

LFCDH/DLFCDH (Single/Double precision)
Computes the $R^H R$ factorization of a complex Hermitian positive definite matrix and estimates its L_1 condition number.

LFTDH/DLFTDH (Single/Double precision)
Computes the $R^H R$ factorization of a complex Hermitian positive definite matrix.

LFSDH/DLFSDH (Single/Double precision)
Solves a complex Hermitian positive definite system of linear equations given the $R^H R$ factorization of the coefficient matrix.

LFIDH/DLFIDH (Single/Double precision)
Uses iterative refinement to improve the solution of a complex Hermitian positive definite system of linear equations.

LFDDH/DLFDDH (Single/Double precision)
Computes the determinant of a complex Hermitian positive definite matrix given the $R^T R$ Cholesky factorization of the matrix.

Complex Hermitian Matrices

LSAHF/DLSAHF (Single/Double precision)
Solves a complex Hermitian system of linear equations with iterative refinement.

LSLHF/DLSLHF (Single/Double precision)
Solves a complex Hermitian system of linear equations without iterative refinement.

LFCHF/DLFCHF (Single/Double precision)
Computes the $U D U^H$ factorization of a complex Hermitian matrix and estimates its L_1 condition number.

LFTHF/DLFTHF (Single/Double precision)
Computes the $U D U^H$ factorization of a complex Hermitian matrix.

LFSHF/DLFSHF (Single/Double precision)
Solves a complex Hermitian system of linear equations given the $U D U^H$ factorization of the coefficient matrix.

LFIHF/DLFIHF (Single/Double precision)
Uses iterative refinement to improve the solution of

a complex Hermitian system of linear equations.

LFDHF/DLFDHF (Single/Double precision)
Computes the determinant of a complex Hermitian matrix given the $U DU^H$ factorization of the matrix.

Real Band Matrices in Band Storage Mode

LSLTR/DLSLTR (Single/Double precision)
Solves a real tridiagonal system of linear equations.

LSLCR/DLSLCR (Single/Double precision)
Computes the *L DU* factorization of a real tridiagonal matrix *A* using a cyclic reduction algorithm.

LSARB/DLSARB (Single/Double precision)
Solves a real system of linear equations in band storage mode with iterative refinement.

LSLRB/DLSLRB (Single/Double precision)
Solves a real system of linear equations in band storage mode without iterative refinement.

LFCRB/DLFCRB (Single/Double precision)
Computes the *LU* factorization of a real matrix in band storage mode and estimates its L_1 condition number.

LFTRB/DLFTRB (Single/Double precision)
Computes the *LU* factorization of a real matrix in band storage mode.

LFSRB/DLFSRB (Single/Double precision)
Solves a real system of linear equations given the *LU* factorization of the coefficient matrix in band storage mode.

LFIRB/DLFIRB (Single/Double precision)
Uses iterative refinement to improve the solution of a real system of linear equations in band storage mode.

LFDRB/DLFDRB (Single/Double precision)
Computes the determinant of a real matrix in band storage mode given the *LU* factorization of the matrix.

Real Band Symmetric Positive Definite Matrices in Band Storage Mode

LSAQS/DLSAQS (Single/Double precision)
Solves a real symmetric positive definite system of linear equations in band symmetric storage mode with iterative refinement.

LSLQS/DLSLQS (Single/Double precision)
Solves a real symmetric positive definite system of linear equations in band symmetric storage mode without iterative refinement.

LSLPB/DLSLPB (Single/Double precision)
Computes the R^TDR Cholesky factorization of a real symmetric positive definite matrix *A* in codiagonal band symmetric storage mode. Solves a system $Ax = b$.

LFCQS/DLFCQS (Single/Double precision)
Computes the R^TR Cholesky factorization of a real symmetric positive definite matrix in band symmetric storage mode and estimates its L_1 condition number.

LFTQS/DLFTQS (Single/Double precision)
Computes the R^TR Cholesky factorization of a real symmetric positive definite matrix in band symmetric storage mode.

LFSQS/DLFSQS (Single/Double precision)
Solves a real symmetric positive definite system of linear equations given the factorization of the coefficient matrix in band symmetric storage mode.

LFIQS/DLFIQS (Single/Double precision)
Uses iterative refinement to improve the solution of a real symmetric positive definite system of linear equations in band symmetric storage mode.

LFDQS/DLFDQS (Single/Double precision)
Computes the determinant of a real symmetric positive definite matrix given the R^TR Cholesky factorization of the band symmetric storage mode.

Complex Band Matrices in Band Storage Mode

LSLTQ/DLSLTQ (Single/Double precision)
Solves a complex tridiagonal system of linear equations.

LSLCQ/DLSLCQ (Single/Double precision)
Computes the *LDU* factorization of a complex tridiagonal matrix *A* using a cyclic reduction algorithm.

LSACB/DLSACB (Single/Double precision)
Solves a complex system of linear equations in band storage mode with iterative refinement.

LSLCB/DLSLCB (Single/Double precision)
Solves a complex system of linear equations in band storage mode without iterative refinement.

LFCCB/DLFCCB (Single/Double precision)
Computes the *LU* factorization of a complex matrix in band storage mode and estimates its L_1 condition number.

LFTCB/DLFTCB (Single/Double precision)
Computes the *LU* factorization of a complex matrix in band storage mode.

LFSCB/DLFSCB (Single/Double precision)
Solves a complex system of linear equations given the *LU* factorization of the coefficient matrix in band storage mode.

LFICB/DLFICB (Single/Double precision)
Uses iterative refinement to improve the solution of a complex system of linear equations in band storage mode.

LFDCB/DLFDCB (Single/Double precision)
Computes the determinant of a complex matrix given the *LU* factorization of the matrix in band storage mode.

COMPLEX BAND POSITIVE DEFINITE MATRICES IN BAND STORAGE MODE

LSAQH/DLSAQH (Single/Double precision)
Solves a complex Hermitian positive definite system of linear equations in band Hermitian storage mode with iterative refinement.

LSLQH/DLSLQH (Single/Double precision)
Solves a complex Hermitian positive definite system of linear equations in band Hermitian storage mode without iterative refinement.

LSLQB/DLSLQB (Single/Double precision)
Computes the $R^H DR$ Cholesky factorization of a complex Hermitian positive-definite matrix *A* in codiagonal band Hermitian storage mode. Solves a system $Ax = b$.

LFCQH/DLFCQH (Single/Double precision)
Computes the $R^H R$ factorization of a complex Hermitian positive definite matrix in band Hermitian storage mode and estimates its L_1 condition number.

LFTQH/DLFTQH (Single/Double precision)
Computes the $R^H R$ factorization of a complex Hermitian positive definite matrix in band Hermitian storage mode.

LFSQH/DLFSQH (Single/Double precision)
Solves a complex Hermitian positive definite system of linear equations given the factorization of the coefficient matrix in band Hermitian storage mode.

LFIQH/DLFIQH (Single/Double precision)
Uses iterative refinement to improve the solution of a complex Hermitian positive definite system of linear equations in band Hermitian storage mode.

LFDQH/DLFDQH (Single/Double precision)
Computes the determinant of a complex Hermitian positive definite matrix given the $R^T R$ Cholesky factorization in band Hermitian storage mode.

REAL SPARSE LINEAR EQUATION SOLVERS

LSLXG/DLSLXG (Single/Double precision)
Solves a sparse system of linear algebraic equations by Gaussian elimination.

LFTXG/DLFTXG (Single/Double precision)
Computes the *LU* factorization of a real general sparse matrix.

LFSXG/DLFSXG (Single/Double precision)
Solves a sparse system of linear equations given the *LU* factorization of the coefficient matrix.

COMPLEX SPARSE LINEAR EQUATION SOLVERS

LSLZG/DLSLZG (Single/Double precision)
Solves a complex sparse system of linear equations by Gaussian elimination.

LFTZG/DLFTZG (Single/Double precision)
Computes the *LU* factorization of a complex general sparse matrix.

LFSZG/DLFSZG (Single/Double precision)
Solves a complex sparse system of linear equations given the *LU* factorization of the coefficient matrix.

REAL SPARSE SYMMETRIC POSITIVE DEFINITE LINEAR EQUATIONS SOLVERS

LSLXD/DLSLXD (Single/Double precision)
Solves a sparse system of symmetric positive definite linear algebraic equations by Gaussian elimination.

LSCXD/DLSCXD (Single/Double precision)
Performs the symbolic Cholesky factorization for a sparse symmetric matrix using a minimum degree ordering or a user-specified ordering, and set up the data structure for the numerical Cholesky factorization.

LNFXD/DLNFXD (Single/Double precision)
Computes the numerical Cholesky factorization of a sparse symmetrical matrix *A*.

LFSXD/DLFSXD (Single/Double precision)
Solves a real sparse symmetric positive definite system of linear equations, given the Cholesky factorization of the coefficient matrix.

Complex Sparse Hermitian Positive Definite Linear Equations Solvers

LSLZD/DLSLZD (Single/Double precision)
Solves a complex sparse Hermitian positive definite system of linear equations by Gaussian elimination.

LNFZD/DLNFZD (Single/Double precision)
Computes the numerical Cholesky factorization of a sparse Hermitian matrix *A*.

LFSZD/DLFSZD (Single/Double precision)
Solves a complex sparse Hermitian positive definite system of linear equations, given the Cholesky factorization of the coefficient matrix.

Real Toeplitz Matrices in Toeplitz Storage Mode

LSLTO/DLSLTO (Single/Double precision)
Solves a real Toeplitz linear system.

Complex Toeplitz Matrices in Toeplitz Storage Mode

LSLTC/DLSLTC (Single/Double precision)
Solves a complex Toeplitz linear system.

Complex Cicrcular Matrices in Circulant Storage Mode

LSLCC/DLSLCC (Single/Double precision)
Solves a complex circulant linear system.

Iterative Methods

PCGRC/DPCGRC (Single/Double precision)
Solves a real symmetric definite linear system using a preconditioned conjugate gradient method with reverse communication.

JCGRC/DJCGRC (Single/Double precision)
Solves a real symmetric definite linear system using the Jacobi-preconditioned conjugate gradient method with reverse communication.

GMRES/DGMRES (Single/Double precision)
Uses GMRES with reverse communication to generate an approximate solution of $Ax = b$.

Linear Least Squares and Matrix Factorization

Least Squares, QR Decomposition and Generalized Inverse Least Squares

LSQRR/DLSQRR (Single/Double precision)
Solves a linear least-squares problem without iterative refinement.

LQRRV/DLQRRV (Single/Double precision)
Computes the least-squares solution using Householder transformations applied in blocked form.

LSBRR/DLSBRR (Single/Double precision)
Solves a linear least-squares problem with iterative refinement.

LCLSQ/DLCLSQ (Single/Double precision)
Solves a linear least-squares problem with linear constraints.

LQRRR/DLQRRR (Single/Double precision)
Computes the *QR* decomposition, $AP = QR$, using Householder transformations.

LQERR/DLQERR (Single/Double precision)
Accumulate the orthogonal matrix *Q* from its factored form given the *QR* factorization of a rectangular matrix *A*.

LQRSL/DLQRSL (Single/Double precision)
Computes the coordinate transformation, projection, and complete the solution of the least-squares problem $Ax = b$.

LUPQR/DLUPQR (Single/Double precision)
Computes an updated *QR* factorization after the rank-one matrix αxy^T is added.

Cholesky Factorization

LCHRG/DLCHRG (Single/Double precision)
Computes the Cholesky decomposition of a symmetric positive semidefinite matrix with optional column pivoting.

LUPCH/DLUPCH (Single/Double precision)
Updates the $R^T R$ Cholesky factorization of a real symmetric positive definite matrix after a rank-one matrix is added.

LDNCH/DLDNCH (Single/Double precision)
Downdates the $R^T R$ Cholesky factorization of a real symmetric positive definite matrix after a rank-one matrix is removed.

SINGULAR VALUE DECOMPOSITIONS

LSVRR/DLSVRR (Single/Double precision)
Computes the singular value decomposition of a real matrix.

LSVCR/DLSVCR (Single/Double precision)
Computes the singular value decomposition of a complex matrix.

LSGRR/DLSGRR (Single/Double precision)
Computes the generalized inverse of a real matrix.

EIGENSYSTEM ANALYSIS

Eigenvalues and (Optionally) Eigenvectors of Ax=lx

REAL GENERAL MATRICES

EVLRG/DEVLRG (Single/Double precision)
Computes all of the eigenvalues of a real matrix.

EVCRG/DEVCRG (Single/Double precision)
Computes all of the eigenvalues and eigenvectors of a real matrix.

EPIRG/DEPIRG (Single/Double precision)
Computes the performance index for a real eigensystem.

COMPLEX GENERAL MATRICES

EVLCG/DEVLCG (Single/Double precision)
Computes all of the eigenvalues of a complex matrix.

EVCCG/DEVCCG (Single/Double precision)
Computes all of the eigenvalues and eigenvectors of a complex matrix.

EPICG/DEPICG (Single/Double precision)
Computes the performance index for a complex eigensystem.

REAL SYMMETRIC MATRICES

EVLSF/DEVLSF (Single/Double precision)
Computes all of the eigenvalues of a real symmetric matrix.

EVCSF/DEVCSF (Single/Double precision)
Computes all of the eigenvalues and eigenvectors of a real symmetric matrix.

EVASF/DEVASF (Single/Double precision)
Computes the largest or smallest eigenvalues of a real symmetric matrix.

EVESF/DEVESF (Single/Double precision)
Computes the largest or smallest eigenvalues and the corresponding eigenvectors of a real symmetric matrix.

EVBSF/DEVBSF (Single/Double precision)
Computes selected eigenvalues of a real symmetric matrix.

EVFSF/DEVFSF (Single/Double precision)
Computes selected eigenvalues and eigenvectors of a real symmetric matrix.

EPISF/DEPISF (Single/Double precision)
Computes the performance index for a real symmetric eigensystem.

REAL BAND SYMMETRIC MATRICES IN BAND STORAGE MODE

EVLSB/DEVLSB (Single/Double precision)
Computes all of the eigenvalues of a real symmetric matrix in band symmetric storage mode.

EVCSB/DEVCSB (Single/Double precision)
Computes all of the eigenvalues and eigenvectors of a real symmetric matrix in band symmetric storage mode.

EVASB/DEVASB (Single/Double precision)
Computes the largest or smallest eigenvalues of a real symmetric matrix in band symmetric storage mode.

EVESB/DEVESB (Single/Double precision)
Computes the largest or smallest eigenvalues and the corresponding eigenvectors of a real symmetric matrix in band symmetric storage mode.

EVBSB/DEVBSB (Single/Double precision)
Computes the eigenvalues in a given interval of a real symmetric matrix stored in band symmetric storage mode.

EVFSB/DEVFSB (Single/Double precision)
Computes the eigenvalues in a given interval and the corresponding eigenvectors of a real symmetric matrix stored in band symmetric storage mode.

EPISB/DEPISB (Single/Double precision)
Computes the performance index for a real symmetric eigensystem in band symmetric storage mode.

Complex Hermitian Matrices

EVLHF/DEVLHF (Single/Double precision)
Computes all of the eigenvalues of a complex Hermitian matrix.

EVCHF/DEVCHF (Single/Double precision)
Computes all of the eigenvalues and eigenvectors of a complex Hermitian matrix.

EVAHF/DEVAHF (Single/Double precision)
Computes the largest or smallest eigenvalues of a complex Hermitian matrix.

EVEHF/DEHF (Single/Double precision)
Computes the largest or smallest eigenvalues and the corresponding eigenvectors of a complex Hermitian matrix.

EVBHF/DEVBHF (Single/Double precision)
Computes the eigenvalues in a given range of a complex Hermitian matrix.

EVFHF/DEVFHF (Single/Double precision)
Computes the eigenvalues in a given range and the corresponding eigenvectors of a complex Hermitian matrix.

EPIHF/DEPIHF (Single/Double precision)
Computes the performance index for a complex Hermitian eigensystem.

Real Upper Hessenberg Matrices

EVLRH/DEVLRH (Single/Double precision)
Computes all of the eigenvalues of a real upper Hessenberg matrix.

EVCRH/DEVCRH (Single/Double precision)
Computes all of the eigenvalues and eigenvectors of a real upper Hessenberg matrix.

Complex Upper Hessenberg Matrices

EVLCH/DEVLCH (Single/Double precision)
Computes all of the eigenvalues of a complex upper Hessenberg matrix.

EVCCH/DEVCCH (Single/Double precision)
Computes all of the eigenvalues and eigenvectors of a complex upper Hessenberg matrix.

Eigenvalues and (Optionally) Eigenvectors of Ax=lBx

Real General Matrices

GVLRG/DGVLRG (Single/Double precision)
Computes all of the eigenvalues of a generalized real eigensystem $Az = \lambda Bz$.

GVCRG/DGVCRG (Single/Double precision)
Computes all of the eigenvalues and eigenvectors of a generalized real eigensystem $Az = \lambda Bz$.

GPIRG/DGPIRG (Single/Double precision)
Computes the performance index for a generalized real eigensystem $Az = \lambda Bz$.

Complex General Matrices

GVLCG/DGVLCG (Single/Double precision)
Computes all of the eigenvalues of a generalized complex eigensystem $Az = \lambda Bz$.

GVCCG/DGVCCG (Single/Double precision)
Computes all of the eigenvalues and eigenvectors of a generalized complex eigensystem $Az = \lambda Bz$.

GPICG/DGPICG (Single/Double precision)
Computes the performance index for a generalized complex eigensystem $Az = \lambda Bz$.

Real Symmetric Matrices and B Positive Definite

GVLSP/DGVLSP (Single/Double precision)
Computes all of the eigenvalues of the generalized real symmetric eigenvalue problem $Az = \lambda Bz$, with B symmetric positive definite.

GVCSP/DGVCSP (Single/Double precision)
Computes all of the eigenvalues and eigenvectors of the generalized real symmetric eigenvalue problem $Az = \lambda Bz$, with B symmetric positive definite.

GPISP/DGPISP (Single/Double precision)
Computes the performance index for a generalized real symmetric eigensystem problem.

Cubic Spline Interpolation

CSIEZ/DCSIEZ (Single/Double precision)
Computes the cubic spline interpolant with the ‘not-a-knot’ condition and returns values of the interpolant at specified points.

CSINT/DCSINT (Single/Double precision)
Computes the cubic spline interpolant with the ‘not-a-knot’ condition.

CSDEC/DCSDEC (Single/Double precision)
Computes the cubic spline interpolant with specified derivative endpoint conditions.

CSHER/DCSHER (Single/Double precision)
Computes the Hermite cubic spline interpolant.

CSAKM/DCSAKM (Single/Double precision)
Computes the Akima cubic spline interpolant.

CSCON/DCSCON (Single/Double precision)
Computes a cubic spline interpolant that is consistent with the concavity of the data.

CSPER/DCSPER (Single/Double precision)
Computes the cubic spline interpolant with periodic boundary conditions.

Cubic Spline Evaluation and Integration

CSVAL/DCSVAL (Single/Double precision)
Evaluates a cubic spline.

CSDER/DCSDER (Single/Double precision)
Evaluates the derivative of a cubic spline.

CS1GD/DCS1GD (Single/Double precision)
Evaluates the derivative of a cubic spline on a grid.

CSITG/DCSITG (Single/Double precision)
Evaluates the integral of a cubic spline.

B-Spline Interpolation

SPLEZ/DSPLEZ (Single/Double precision)
Computes the values of a spline that either interpolates or fits user-supplied data.

BSINT/DBSINT (Single/Double precision)
Computes the spline interpolant, returning the B-spline coefficients.

BSNAK/DBSNAK (Single/Double precision)
Computes the “not-a-knot” spline knot sequence.

BSOPK/DBSOPK (Single/Double precision)
Computes the “optimal” spline knot sequence.

BS2IN/DBS2IN (Single/Double precision)
Computes a two-dimensional tensor-product spline interpolant, returning the tensor-product B-spline coefficients.

BS3IN/DBS3IN (Single/Double precision)
Computes a three-dimensional tensor-product spline interpolant, returning the tensor-product B-spline coefficients.

Spline Evaluation, Integration and Conversion to Piecewise Polynomial Given the B-spline Representation

BSVAL/DBSVAL (Single/Double precision)
Evaluates a spline, given its B-spline representation.

BSDER/DBSDER (Single/Double precision)
Evaluates the derivative of a spline, given its B-spline representation.

BS1GD/DBS1GD (Single/Double precision)
Evaluates the derivative of a spline on a grid, given its B-spline representation.

BSITG/DBSITG (Single/Double precision)
Evaluates the integral of a spline, given its B-spline representation.

BS2VL/DBS2VL (Single/Double precision)
Evaluates a two-dimensional tensor-product spline, given its tensor-product B-spline representation.

BS2DR/DBS2DR (Single/Double precision)
Evaluates the derivative of a two-dimensional tensor-product spline, given its tensor-product B-spline representation.

BS2GD/DBS2GD (Single/Double precision)
Evaluates the derivative of a two-dimensional tensor-product spline, given its tensor-product B-spline representation on a grid.

BS2IG/DBS2IG (Single/Double precision)
Evaluates the integral of a tensor-product spline on a rectangular domain, given its tensor-product B-spline representation.

BS3VL/DBS3VL (Single/Double precision)
Evaluates a three-dimensional tensor-product spline, given its tensor-product B-spline representation.

BS3DR/DBS3DR (Single/Double precision)
Evaluates the derivative of a three-dimensional tensor-product spline, given its tensor-product B-spline

representation.

BS3GD/DBS3GD (Single/Double precision)
Evaluates the derivative of a three-dimensional tensor-product spline, given its tensor-product B-spline representation on a grid.

BS3IG/DBS3IG (Single/Double precision)
Evaluates the integral of a tensor-product spline in three dimensions over a three-dimensional rectangle, given its tensor-product B-spline representation.

BSCPP/DBSCPP (Single/Double precision)
Converts a spline in B-spline representation to piecewise polynomial representation.

Piecewise Polynomial

PPVAL/DPPVAL (Single/Double precision)
Evaluates a piecewise polynomial.

PPDER/DPPDER (Single/Double precision)
Evaluates the derivative of a piecewise polynominal.

PP1GD/DPP1GD (Single/Double precision)
Evaluates the derivative of a piecewise polynomial on a grid.

PPITG/DPPITG (Single/Double precision)
Evaluates the integral of a piecewise polynomial.

Quadratic Polynomial Interpolation Routines for Gridded Data

QDVAL/DQDVAL (Single/Double precision)
Evaluates a function defined on a set of points using quadratic interpolation.

QDDER/DQDDER (Single/Double precision)
Evaluates the derivative of a function defined on a set of points using quadratic interpolation.

QD2VL/DQD2VL (Single/Double precision)
Evaluates a function defined on a rectangular grid using quadratic interpolation.

QD2DR/DQD2DR (Single/Double precision)
Evaluates the derivative of a function defined on a rectangular grid using quadratic interpolation.

QD3VL/DQD3VL (Single/Double precision)
Evaluates a function defined on a rectangular three-dimensional grid using quadratic interpolation.

QD3DR/DQD3DR (Single/Double precision)
Evaluates the derivative of a function defined on a rectangular three-dimensional grid using quadratic interpolation.

Scattered Data Interpolation

SURF/DSURF (Single/Double precision)
Computes a smooth bivariate interpolant to scattered data that is locally a quintic polynomial in two variables.

Least-Squares Approximation

RLINE/DRLINE (Single/Double precision)
Fits a line to a set of data points using least squares.

RCURV/DRCURV (Single/Double precision)
Fits a polynomial curve using least squares.

FNLSQ/DFNLSQ (Single/Double precision)
Computes a least-squares approximation with user-supplied basis functions.

BSLSQ/DBSLSQ (Single/Double precision)
Computes the least-squares spline approximation, and returns the B-spline coefficients.

BSVLS/DBSVLS (Single/Double precision)
Computes the variable knot B-spline least squares approximation to given data.

CONFT/DCONFT (Single/Double precision)
Computes the least-squares constrained spline approximation, returning the B-spline coefficients.

BSLS2/DBSLS2 (Single/Double precision)
Computes a two-dimensional tensor-product spline approximant using least squares, returning the tensor-product B-spline coefficients.

BSLS3/DBSLS3 (Single/Double precision)
Computes a three-dimensional tensor-product spline approximant using least squares, returning the tensor-product B-spline coefficients.

Cubic Spline Smoothing

CSSED/DCSSED (Single/Double precision)
Smooth one-dimensional data by error detection.

CSSMH/DCSSMH (Single/Double precision)
Computes a smooth cubic spline approximation to noisy data.

CSSCV/DCSSCV (Single/Double precision)
Computes a smooth cubic spline approximation to noisy data using cross-validation to estimate the smoothing parameter.

RATIONAL L_∞ APPROXIMATION

RATCH/DRATCH (Single/Double precision)
Computes a rational weighted Chebyshev approximation to a continuous function on an interval.

INTEGRATION AND DIFFERENTIATION

Univariate Quadrature

QDAGS/DQDAGS (Single/Double precision)
Integrates a function (which may have endpoint singularities).

QDAG/DQDAG (Single/Double precision)
Integrates a function using a globally adaptive scheme based on Gauss-Kronrod rules.

QDAGP/DQDAGP (Single/Double precision)
Integrates a function with singularity points given.

QDAGI/DQDAGI (Single/Double precision)
Integrates a function over an infinite or semi-infinite interval.

QDAWO/DQDAWO (Single/Double precision)
Integrates a function containing a sine or a cosine.

QDAWF/DQDAWF (Single/Double precision)
Computes a Fourier integral.

QDAWS/DQDAWS (Single/Double precision)
Integrates a function with algebraic-logarithmic singularities.

QDAWC/DQDAWC (Single/Double precision)
Integrates a function `F(X)/(X-C)` in the Cauchy principal value sense.

QDNG/DQDNG (Single/Double precision)
Integrates a smooth function using a nonadaptive rule.

Multidimensional Quadrature

TWODQ/DTWODQ (Single/Double precision)
Computes a two-dimensional iterated integral.

QAND/DQAND (Single/Double precision)
Integrates a function on a hyper-rectangle.

Gauss Rules and Three-Term Recurrences

GQRUL/DGQRUL (Single/Double precision)
Computes a Gauss, Gauss-Radau, or Gauss-Lobatto quadrature rule with various classical weight functions.

GQRCF/DGQRCF (Single/Double precision)
Computes a Gauss, Gauss-Radau or Gauss-Lobatto quadrature rule given the recurrence coefficients for the monic polynomials orthogonal with respect to the weight function.

RECCF/DRECCF (Single/Double precision)
Computes recurrence coefficients for various monic polynomials.

RECQR/DRECQR (Single/Double precision)
Computes recurrence coefficients for monic polynomials given a quadrature rule.

FQRUL/DFQRUL (Single/Double precision)
Computes a Fejér quadrature rule with various classical weight functions.

Differentiation

DERIV/DDERIV (Single/Double precision)
Computes the first, second or third derivative of a user-supplied function.

DIFFERENTIAL EQUATIONS

First-Order Ordinary Differential Equations

SOLUTION OF THE INITIAL VALUE PROBLEM FOR ODEs

IVPRK/DIVPRK (Single/Double precision)
Solves an initial-value problem for ordinary differential equations using the Runge-Kutta-Verner fifth-order and sixth-order method.

IVMRK/DIVMRK (Single/Double precision)
Solves an initial-value problem $y' = f(t, y)$ for ordinary differential equations using Runge-Kutta pairs of various orders.

IVPAG/DIVPAG (Single/Double precision)
Solves an initial-value problem for ordinary differential equations using either Adams-Moulton's or Gear's BDF method.

SOLUTION OF THE BOUNDARY VALUE PROBLEM FOR ODES

BVPFD/DBVPFD (Single/Double precision)
Solves a (parameterized) system of differential equations with boundary conditions at two points, using a variable order, variable step size finite difference method with deferred corrections.

BVPMS/DBVPMS (Single/Double precision)
Solves a (parameterized) system of differential equations with boundary conditions at two points, using a multiple-shooting method.

SOLUTION OF DIFFERENTIAL-ALGEBRAIC SYSTEMS

DASPG/DDASPG (Single/Double precision)
Solves a first order differential-algebraic system of equations, $g(t, y, y') = 0$, using the Petzold-Gear BDF method.

Partial Differential Equations

SOLUTION OF SYSTEMS OF PDES IN ONE DIMENSION

MOLCH/DMOLCH (Single/Double precision)
Solves a system of partial differential equations of the form $u_t = f(x, t, u, u_x, u_{xx})$ using the method of lines. The solution is represented with cubic Hermite polynomials.

SOLUTION OF SYSTEMS OF PDES IN TWO AND THREE DIMENSIONS

FPS2H/DFPS2H (Single/Double precision)
Solves Poisson's or Helmholtz's equation on a two-dimensional rectangle using a fast Poisson solver based on the `HODIE` finite-difference scheme on a uniform mesh.

FPS3H/DFPS3H (Single/Double precision)
Solves Poisson's or Helmholtz's equation on a three-dimensional box using a fast Poisson solver based on the `HODIE` finite-difference scheme on a uniform mesh.

SLEIG/DSLEIG (Single/Double precision)
Determines eigenvalues, eigenfunctions and/or spectral density functions for Sturm-Liouville problems in the form.

SLCNT/DSLCNT (Single/Double precision)
Calculates the indices of eigenvalues of a Sturm-Liouville problem of the form for
$-\frac{d}{dx}\left(p(x)\frac{du}{dx}\right) + q(x)u = \lambda r(x)u$ for x in $[a,b]$ with boundary conditions (at regular points)
$a_1u - a_2(pu') = \lambda(a_1'u - a_2'(pu'))$ at a
$b_1u + b_2(pu') = 0$ at b in a specified subinterval of the real line, $[\alpha, \beta]$.

TRANSFORMS

Real Trigonometric FFT

FFTRF/DFFTRF (Single/Double precision)
Computes the Fourier coefficients of a real periodic sequence.

FFTRB/DFFTRB (Single/Double precision)
Computes the real periodic sequence from its Fourier coefficients.

FFTRI/DFFTRI (Single/Double precision)
Computes parameters needed by `FFTRF` and `FFTRB`.

Complex Exponential FFT

FFTCF/DFFTCF (Single/Double precision)
Computes the Fourier coefficients of a complex periodic sequence.

FFTCB/DFFTCB (Single/Double precision)
Computes the complex periodic sequence from its Fourier coefficients.

FFTCI/DFFTCI (Single/Double precision)
Computes parameters needed by `FFTCF` and `FFTCB`.

Real Sine and Cosine FFTs

FSINT/DFSINT (Single/Double precision)
Computes the discrete Fourier sine transformation of an odd sequence.

FSINI/DFSINI (Single/Double precision)
Computes parameters needed by `FSINT`.

FCOST/DFCOST (Single/Double precision)
Computes the discrete Fourier cosine transformation of an even sequence.

FCOSI/DFCOSI (Single/Double precision)
Computes parameters needed by `FCOST`.

Real Quarter Sine and Quarter Cosine FFTs

QSINF/DQSINF (Single/Double precision)
Computes the coefficients of the sine Fourier transform with only odd wave numbers.

QSINB/DQSINB (Single/Double precision)
Computes a sequence from its sine Fourier coefficients with only odd wave numbers.

QSINI/DQSINI (Single/Double precision)
Computes parameters needed by `QSINF` and `QSINB`.

QCOSF/DQCOSF (Single/Double precision)
Computes the coefficients of the cosine Fourier transform with only odd wave numbers.

QCOSB/DQCOSB (Single/Double precision)
Computes a sequence from its cosine Fourier coefficients with only odd wave numbers.

QCOSI/DQCOSI (Single/Double precision)
Computes parameters needed by `QCOSF` and `QCOSB`.

Two- and Three- Dimensional Complex FFTs

FFT2D/DFFT2D (Single/Double precision)
Computes Fourier coefficients of a complex periodic two-dimensional array.

FFT2B/DFFT2B (Single/Double precision)
Computes the inverse Fourier transform of a complex periodic two-dimensional array.

FFT3F/DFFT3F (Single/Double precision)
Computes Fourier coefficients of a complex periodic three-dimensional array.

FFT3B/DFFT3B (Single/Double precision)
Computes the inverse Fourier transform of a complex periodic three-dimensional array.

Convolutions and Correlations

RCONV/DRCONV (Single/Double precision)
Computes the convolution of two real vectors.

CCONV/DCCONV (Single/Double precision)
Computes the convolution of two complex vectors.

RCORL/DRCORL (Single/Double precision)
Computes the correlation of two real vectors.

CCORL/DCCORL (Single/Double precision)
Computes the correlation of two complex vectors.

Laplace Transform

INLAP/DINLAP (Single/Double precision)
Computes the inverse Laplace transform of a complex function.

SINLP/DSINLP (Single/Double precision)
Computes the inverse Laplace transform of a complex function.

Nonlinear Equations

ZEROS OF A POLYNOMIAL

ZPLRC/DZPLRC (Single/Double precision)
Finds the zeros of a polynomial with real coefficients using Laguerre's method.

ZPORC/DZPORC (Single/Double precision)
Finds the zeros of a polynomial with real coefficients using the Jenkins-Traub three-stage algorithm.

ZPOCC/DZPOCC (Single/Double precision)
Finds the zeros of a polynomial with complex coefficients using the Jenkins-Traub three-stage algorithm.

ZERO(S) OF A FUNCTION

ZANLY/DZANLY (Single/Double precision)
Finds the zeros of a univariate complex function using Müller's method.

ZBREN/DZBREN (Single/Double precision)
Finds a zero of a real function that changes sign in a given interval.

ZREAL/DZREAL (Single/Double precision)
Finds the real zeros of a real function using Müller's method.

ROOT OF A SYSTEM OF EQUATIONS

NEQNF/DNEQNF (Single/Double precision)
Solves a system of nonlinear equations using a modified Powell hybrid algorithm and a finite-difference approximation to the Jacobian.

NEQNJ/DNEQNJ (Single/Double precision)
Solves a system of nonlinear equations using a modified Powell hybrid algorithm with a user-supplied Jacobian.

NEQBF/DNEQBF (Single/Double precision)
Solves a system of nonlinear equations using fac-

tored secant update with a finite-difference approximation to the Jacobian.

NEQBJ/DNEQBJ (Single/Double precision)
Solves a system of nonlinear equations using factored secant update with a user-supplied Jacobian.

OPTIMIZATION

Unconstrained Minimization

UNIVARIATE FUNCTION

UVMIF/DUVMIF (Single/Double precision)
Finds the minimum point of a smooth function of a single variable using only function evaluations.

UVMID/DUVMID (Single/Double precision)
Finds the minimum point of a smooth function of a single variable using both function evaluations and first derivative evaluations.

UVMGS/DUVMGS (Single/Double precision)
Finds the minimum point of a nonsmooth function of a single variable.

MULTIVARIATE FUNCTION

UMINF/DUMINF (Single/Double precision)
Minimizes a function of N variables using a quasi-Newton method and a finite-difference gradient.

UMING/DUMING (Single/Double precision)
Minimizes a function of N variables using a quasi-Newton method and a user-supplied gradient.

UMIDH/DUMIDH (Single/Double precision)
Minimizes a function of N variables using a modified Newton method and a finite-difference Hessian.

UMIAH/DUMIAH (Single/Double precision)
Minimizes a function of N variables using a modified Newton method and a user-supplied Hessian.

UMCGF/DUMCGF (Single/Double precision)
Minimizes a function of N variables using a conjugate gradient algorithm and a finite-difference gradient.

UMCGG/DUMCGG (Single/Double precision)
Minimizes a function of N variables using a conjugate gradient algorithm and a user-supplied gradient.

UMPOL/DUMPOL (Single/Double precision)
Minimizes a function of N variables using a direct search polytope algorithm.

NONLINEAR LEAST SQUARES

UNLSF/DUNLSF (Single/Double precision)
Solves a nonlinear least-squares problem using a modified Levenberg-Marquardt algorithm and a finite-difference Jacobian.

UNLSJ/DUNLSJ (Single/Double precision)
Solves a nonlinear least squares problem using a modified Levenberg-Marquardt algorithm and a user-supplied Jacobian.

Minimization with Simple Bounds

BCONF/DBCONF (Single/Double precision)
Minimizes a function of N variables subject to bounds on the variables using a quasi-Newton method and a finite-difference gradient.

BCONG/DBCONG (Single/Double precision)
Minimizes a function of N variables subject to bounds on the variables using a quasi-Newton method and a user-supplied gradient.

BCODH/DBCODH (Single/Double precision)
Minimizes a function of N variables subject to bounds on the variables using a modified Newton method and a finite-difference Hessian.

BCOAH/DBCOAH (Single/Double precision)
Minimizes a function of N variables subject to bounds on the variables using a modified Newton method and a user-supplied Hessian.

BCPOL/DBCPOL (Single/Double precision)
Minimizes a function of N variables subject to bounds on the variables using a direct search complex algorithm.

BCLSF/DBCLSF (Single/Double precision)
Solves a nonlinear least squares problem subject to bounds on the variables using a modified Levenberg-Marquardt algorithm and a finite-difference Jacobian.

BCLSJ/DBCLSJ (Single/Double precision)
Solves a nonlinear least squares problem subject to bounds on the variables using a modified Levenberg-Marquardt algorithm and a user-supplied Jacobian.

BCNLS/DBCNLS (Single/Double precision)
Solves a nonlinear least-squares problem subject to bounds on the variables and general linear constraints.

Linearly Constrained Minimization

DLPRS/DDLPRS (Single/Double precision)
Solves a linear programming problem via the revised simplex algorithm.

SLPRS/DSLPRS (Single/Double precision)
Solves a sparse linear programming problem via the revised simplex algorithm.

QPROG/DQPROG (Single/Double precision)
Solves a quadratic programming problem subject to linear equality/inequality constraints.

LCONF/DLCONF (Single/Double precision)
Minimizes a general objective function subject to linear equality/inequality constraints.

LCONG/DLCONG (Single/Double precision)
Minimizes a general objective function subject to linear equality/inequality constraints.

Nonlinearly Constrained Minimization

NCONF/DNCONF (Single/Double precision)
Solves a general nonlinear programming problem using the successive quadratic programming algorithm and a finite difference gradient.

NCONG/DNCONG (Single/Double precision)
Solves a general nonlinear programming problem using the successive quadratic programming algorithm and a user-supplied gradient.

Service Routines

CDGRD/DCDGRD (Single/Double precision)
Approximates the gradient using central differences.

FDGRD/DFDGRD (Single/Double precision)
Approximates the gradient using forward differences.

FDHES/DFDHES (Single/Double precision)
Approximates the Hessian using forward differences and function values.

GDHES/DGDHES (Single/Double precision)
Approximates the Hessian using forward differences and a user-supplied gradient.

FDJAC/DFDJAC (Single/Double precision)
Approximate the Jacobian of `M` functions in `N` unknowns using forward differences.

CHGRD/DCHGRD (Single/Double precision)
Checks a user-supplied gradient of a function.

CHHES/DCHHES (Single/Double precision)
Checks a user-supplied Hessian of an analytic function.

CHJAC/DCHJAC (Single/Double precision)
Checks a user-supplied Jacobian of a system of equations with `M` functions in `N` unknowns.

GGUES/DGGUES (Single/Double precision)
Generates points in an `N`-dimensional space.

BASIC MATRIX/VECTOR OPERATIONS

Basic Linear Algebra Subprograms (BLAS)

LEVEL I BLAS

ISET
Sets the components of a vector to a scalar, all integer.

SSET
Sets the components of a vector to a scalar.

CSET
Sets the components of a vector to a scalar, all complex.

ICOPY
Copies a vector x to a vector y, both integer.

SCOPY
Copies a vector x to a vector y, both single precision.

CCOPY
Copies a vector x to a vector y, both complex.

SSCAL
Multiplies a vector by a scalar, $y \leftarrow ay$, both single precision.

CSCAL
Multiplies a vector by a scalar, $y \leftarrow ay$, both complex.

CSSCAL
Multiplies a complex vector by a single-precision scalar, $y \leftarrow ay$.

SVCAL
Multiplies a vector by a scalar and stores the result in another vector, $y \leftarrow ax$, all single precision.

CVCAL
Multiplies a vector by a scalar and stores the result in another vector, $y \leftarrow ax$, all complex.

CSVCAL
Multiplies a complex vector by a single-precision scalar and stores the result in another complex vector, $y \leftarrow ax$.

IADD
Adds a scalar to each component of a vector, $x \leftarrow x + a$, all integer.

SADD
Adds a scalar to each component of a vector, $x \leftarrow x + a$, all single precision.

CADD
Adds a scalar to each component of a vector, $x \leftarrow x + a$, all complex.

ISUB
Subtract each component of a vector from a scalar, $x \leftarrow a - x$, all integer.

SSUB
Subtract each component of a vector from a scalar, $x \leftarrow a - x$, all single precision.

CSUB
Subtract each component of a vector from a scalar, $x \leftarrow a - x$, all complex.

SAXPY
Computes the scalar times a vector plus a vector, $y \leftarrow ax + y$, all single precision.

CAXP
Computes the scalar times a vector plus a vector, $y \leftarrow ax + y$, all complex.

ISWAP
Interchange vectors x and y, both integer.

SSWAP
Interchange vectors x and y, both single precision.

CSWAP
Interchange vectors x and y, both complex.

SDOT
Computes the single-precision dot product $x^T y$.

CDOTU
Computes the complex dot product $x^T y$.

CDOTC
Computes the complex conjugate dot product, $\bar{x}^T y$.

DSDOT
Computes the single-precision dot product $x^T y$ using a double precision accumulator.

CZDOTU
Computes the complex dot product $x^T y$ using a double-precision accumulator.

CZDOTC
Computes the complex conjugate dot product, $\bar{x}^T y$ using a double-precision accumulator.

SDSDOT
Computes the sum of a single-precision scalar and a single precision dot product, $a + x^T y$, using a double-precision accumulator.

CZUDOT
Computes the sum of a complex scalar plus a complex dot product, $a + x^T y$, using a double-precision accumulator.

SDDOTI
Computes the sum of a single-precision scalar plus a single precision dot product using a double-precision accumulator, which is set to the result ACC $\leftarrow a + x^T y$.

SDDOTA
Computes the sum of a single-precision scalar, a single-precision dot product and the double-precision accumulator, which is set to the result ACC $\leftarrow$ ACC $+ a + x^T y$.

CZDOTI
Computes the sum of a complex scalar plus a complex dot product using a double-complex accumulator, which is set to the result ACC $\leftarrow a + x^T y$.

CZDOTA
Computes the sum of a complex scalar, a complex dot product and the double-complex accumulator, which is set to the result ACC $\leftarrow$ ACC $+ a + x^T y$.

SHPROD
Computes the Hadamard product of two single-precision vectors.

SXYZ
Computes a single-precision *xyz* product.

ISUM
Sums the values of an integer vector.

SSUM
Sums the values of a single-precision vector.

SASUM
Sums the absolute values of the components of a single-precision vector.

SCASUM
Sums the absolute values of the real part together with the absolute values of the imaginary part of the components of a complex vector.

SNRM2
Computes the Euclidean length or L_2 norm of a single-precision vector.

SCNRM2
Computes the Euclidean norm of a complex vector.

SPRDCT
Multiplies the components of a single-precision vector.

IIMIN
Finds the smallest index of the minimum of an integer vector.

ISMIN
Finds the smallest index of the component of a single-precision vector having minimum value.

IIMAX
Finds the smallest index of the maximum component of an integer vector.

ISMAX
Finds the smallest index of the component of a single-precision vector having maximum value.

ISAMIN
Finds the smallest index of the component of a single-precision vector having minimum absolute value.

ICAMIN
Finds the smallest index of the component of a complex vector having minimum magnitude.

ISAMAX
Finds the smallest index of the component of a single-precision vector having maximum absolute value.

ICAMAX
Finds the smallest index of the component of a complex vector having maximum magnitude.

SROTG
Constructs a Givens plane rotation in single precision.

SROT
Applies a Givens plane rotation in single precision.

CSROT
Applies a complex Givens plane rotation.

SROTMG
Constructs a modified Givens plane rotation in single precision.

SROTM
Applies a modified Givens plane rotation in single precision.

CSROTM
Applies a complex modified Givens plane rotation.

Level II BLAS

SGEMV
Computes one of the matrix-vector operations:
$y \leftarrow aAx + by$, $y \leftarrow aA^T x + by$.

CGEMV
Computes one of the matrix-vector operations:
$y \leftarrow aAx + by$, $y \leftarrow a\bar{A}^T + by$.

SGBMV
Computes one of the matrix-vector operations:
$y \leftarrow aAx + by$, $y \leftarrow aA^T x + by$, where A is a matrix stored in band storage mode.

CGBMV
Computes one of the matrix-vector operations:
$y \leftarrow aAx + by$, $y \leftarrow a\bar{A}^T x + by$, where A is a matrix stored in band storage mode.

CHEMV
Compute the matrix-vector operation $y \leftarrow aAx + by$, where A is an Hermitian matrix.

CHBMV
Computes the matrix-vector operation $y \leftarrow aAx + by$ where A is an Hermitian band matrix in band Hermitian storage.

SSYMV
Computes the matrix-vector operation $y \leftarrow aAx + by$ where A is a symmetric matrix.

SSBMV
Computes the matrix-vector operation $y \leftarrow aAx + by$ where A is a symmetric matrix in band symmetric storage mode.

STRMV
Computes one of the matrix-vector operations:
$x \leftarrow Ax$ or $x \leftarrow A^T x$ where A is a triangular matrix.

CTRMV
Computes one of the matrix-vector operations:
$x \leftarrow Ax$, $x \leftarrow \bar{A}^T x$, where A is a triangular matrix.

STBMV
Computes one of the matrix-vector operations:
$x \leftarrow Ax$ or $x \leftarrow A^T x$, where A is a triangular matrix in band storage mode.

CTBMV
Computes one of the matrix-vector operations:
$x \leftarrow Ax$, $x \leftarrow \bar{A}^T x$, where A is a triangular matrix in band storage mode.

STRSV
Solves one of the triangular linear systems:
$x \leftarrow A^{-1}x$, $x \leftarrow (A^{-1})^T x$, where A is a triangular matrix.

CTRSV
Solves one of the complex triangular systems:
$x \leftarrow A^{-1}x$, $x \leftarrow (\bar{A}^T)^{-1} x$, where A is a triangular

matrix.

STBSV
Solves one of the triangular systems:
$x \leftarrow A^{-1}x$, $x \leftarrow (A^T)^{-1}x$, where A is a triangular matrix in band storage mode.

CTBSV
Solve one of the complex triangular systems:
$x \leftarrow A^{-1}x$, $x \leftarrow (\bar{A}^T)^{-1}x$, where A is a triangular matrix in band storage mode.

SGER
Computes the rank-one update of a real general matrix: $A \leftarrow A + \alpha xy^T$.

CGERU
Computes the rank-one update of a complex general matrix: $A \leftarrow A + \alpha xy^T$.

CGERC
Computes the rank-one update of a complex general matrix: $A \leftarrow A + \alpha x\bar{y}^T$.

CHER
Computes the rank-one update of an Hermitian matrix: $A \leftarrow A + \alpha x\bar{x}^T$ with x complex and α real.

CHER2
Computes a rank-two update of an Hermitian matrix: $A \leftarrow A + \alpha x\bar{y}^T + \bar{\alpha}y\bar{x}^T$.

SSYR
Computes the rank-one update of a real symmetric matrix: $A \leftarrow A + \alpha xx^T$.

SSYR2
Computes the rank-two update of a real symmetric matrix: $A \leftarrow A + \alpha xy^T + \alpha yx^T$.

Level III BLAS

SGEMM
Computes one of the matrix-matrix operations:
$C \leftarrow \alpha AB + bC$, $C \leftarrow \alpha A^T B + bC$, $C \leftarrow \alpha AB^T + bC$, $C \leftarrow \alpha A^T B^T + bC$.

CGEMM
Computes one of the matrix-matrix operations:
$C \leftarrow \alpha A\bar{B}^T + bC$, $C \leftarrow \alpha \bar{A}^T B + bC$,
$C \leftarrow \alpha A^T B^T + bC$, $C \leftarrow \alpha \bar{A}^T B^T + bC$, $C \leftarrow \alpha \bar{A}^T \bar{B}^T + bC$.

SSYMM
Computes one of the matrix-matrix operations:
$C \leftarrow \alpha AB + bC$ or $C \leftarrow \alpha BA + bC$, where A is a symmetric matrix and B and C are m by n matrices.

CSYMM
Computes one of the matrix-matrix operations:
$C \leftarrow \alpha AB + bC$ or $C \leftarrow \alpha BA + bC$, where A is a symmetric matrix and B and C are m by n matrices.

CHEMM
Computes one of the matrix-matrix operations:
$C \leftarrow \alpha AB + bC$ or $C \leftarrow \alpha BA + bC$, where A is an Hermitian matrix and B and C are m by n matrices.

STRMM
Computes one of the matrix-matrix operations:
$B \leftarrow \alpha AB$, $B \leftarrow \alpha A^T B$, $B \leftarrow \alpha BA$, $B \leftarrow \alpha BA^T$, where B is an m by n matrix and A is a triangular matrix.

CTRMM
Computes one of the matrix-matrix operations:
$B \leftarrow \alpha AB$, $B \leftarrow \alpha A^T B$, $B \leftarrow \alpha \bar{A}^T B$, or $B \leftarrow \alpha B\bar{A}^T$, where B is an m by n matrix and A is a triangular matrix.

STRSM
Solves one of the matrix equations:
$B \leftarrow \alpha A^{-1}B$, $B \leftarrow \alpha BA^{-1}$, $B \leftarrow \alpha (A^{-1})^T B$, $B \leftarrow \alpha B(A^{-1})^T$ where B is an m by n matrix and A is a triangular matrix.

CTRSM
Solves one of the complex matrix equations:
$B \leftarrow \alpha A^{-1}B$, $B \leftarrow \alpha BA^{-1}$, $B \leftarrow \alpha (\bar{A}^T)^{-1}B$,
$B \leftarrow \alpha B(\bar{A}^T)^{-1}$, where A is a triangular matrix.

SSYRK
Computes one of the symmetric rank k operations:
$C \leftarrow \alpha AA^T + bC$ or $C \leftarrow \alpha A^T A + bC$, where C is an n by n symmetric matrix and A is an n by k matrix in the first case and a k by n matrix in the second case.

CSYRK
Computes one of the symmetric rank k operations:
$C \leftarrow \alpha AA^T + bC$ or $C \leftarrow \alpha A^T A + bC$, where C is an n by n symmetric matrix and A is an n by k matrix in the first case and a k by n matrix in the second case.

CHERK
Computes one of the Hermitian rank k operations:
$C \leftarrow \alpha A\bar{A}^T + bC$ or $C \leftarrow \alpha \bar{A}^T A + bC$, where C is an n by n Hermitian matrix and A is an n by k matrix in the first case and a k by n matrix in the second case.

SSYR2K
Computes one of the symmetric rank $2k$ operations:
$C \leftarrow \alpha AB^T + \alpha BA^T + bC$ or $C \leftarrow \alpha A^T B + \alpha B^T A + bC$, where C is an n by n symmetric matrix and A

and B are n by k matrices in the first case and k by n matrices in the second case.

CSYR2K
Computes one of the symmetric rank $2k$ operations: $C \leftarrow aAB^T + aBA^T + bC$ or $C \leftarrow aA^TB + aB^TA + bC$, where C is an n by n symmetric matrix and A and B are n by k matrices in the first case and k by n matrices in the second case.

CHER2K
Computes one of the Hermitian rank $2k$ operations: $C \leftarrow \alpha A\overline{B}^T + \overline{\alpha} B\overline{A}^T + \beta C$ or $C \leftarrow \alpha \overline{A}^T B + \overline{\alpha}\overline{B}^T A + \beta C$ where C is an n by n Hermitian matrix and A and B are n by k matrices in the first case and k by n matrices in the second case.

Other Matrix/Vector Operations

Matrix Copy

CRGRG/DCRGRG (Single/Double precision)
Copies a real general matrix.

CCGCG/DCCGCG (Single/Double precision)
Copies a complex general matrix.

CRBRB/DCRBRB (Single/Double precision)
Copies a real band matrix stored in band storage mode.

CCBCB/DCCBCB (Single/Double precision)
Copies a complex band matrix stored in complex band storage mode.

Matrix Conversion

CRGRB/DCRGRB (Single/Double precision)
Converts a real general matrix to a matrix in band storage mode.

CRBRG/DCRBRG (Single/Double precision)
Converts a real matrix in band storage mode to a real general matrix.

CCGCB/DCCGCB (Single/Double precision)
Converts a complex general matrix to a matrix in complex band storage mode.

CCBCG/DCCBCG (Single/Double precision)
Converts a complex matrix in band storage mode to a complex matrix in full storage mode.

CRGCG/DCRGCG (Single/Double precision)
Copies a real general matrix to a complex general matrix.

CRRCR/DCRRCR (Single/Double precision)
Copies a real rectangular matrix to a complex rectangular matrix.

CRBCB/DCRBCB (Single/Double precision)
Converts a real matrix in band storage mode to a complex matrix in band storage mode.

CSFRG/DCSFRG (Single/Double precision)
Extends a real symmetric matrix defined in its upper triangle to its lower triangle.

CHFCG/DCHFCG (Single/Double precision)
Extends a complex Hermitian matrix defined in its upper triangle to its lower triangle.

CSBRB/DCSBRB (Single/Double precision)
Copies a real symmetric band matrix stored in band symmetric storage mode to a real band matrix stored in band storage mode.

CHBCB/DCHBCB (Single/Double precision)
Copies a complex Hermitian band matrix stored in band Hermitian storage mode to a complex band matrix stored in band storage mode.

TRNRR/DTRNRR (Single/Double precision)
Transposes a rectangular matrix.

Matrix Multiplication

MXTXF/DMXTXF (Single/Double precision)
Computes the transpose product of a matrix, A^TA.

MXTYF/DMXTYF (Single/Double precision)
Multiplies the transpose of matrix A by matrix B, A^TB.

MXYTF/DMXYTF (Single/Double precision)
Multiplies a matrix A by the transpose of a matrix B, AB^T.

MRRRR/DMRRRR (Single/Double precision)
Multiplies two real rectangular matrices, AB.

MCRCR/DMCRCR (Single/Double precision)
Multiplies two complex rectangular matrices, AB.

HRRRR/DHRRRR (Single/Double precision)
Computes the Hadamard product of two real rectangular matrices.

BLINF/DBLINF (Single/Double precision)
Computes the bilinear form x^TAy.

POLRG/DPOLRG (Single/Double precision)
Evaluates a real general matrix polynomial.

Matrix-Vector Multiplication

MURRV/DMURRV (Single/Double precision)
Multiplies a real rectangular matrix by a vector.

MURBV/DMURBV (Single/Double precision)

Multiplies a real band matrix in band storage mode by a real vector.

MUCRV/DMUCRV (Single/Double precision)
Multiplies a complex rectangular matrix by a complex vector.

MUCBV/DMUCBV (Single/Double precision)
Multiplies a complex band matrix in band storage mode by a complex vector.

MATRIX ADDITION

ARBRB/DARBRB (Single/Double precision)
Adds two band matrices, both in band storage mode.

ACBCB/DACBCB (Single/Double precision)
Adds two complex band matrices, both in band storage mode.

MATRIX NORM

NRIRR/DNRIRR (Single/Double precision)
Computes the infinity norm of a real matrix.

NR1RR/DNR1RR (Single/Double precision)
Computes the 1-norm of a real matrix.

NR2RR/DNR2RR (Single/Double precision)
Computes the Frobenius norm of a real rectangular matrix.

NR1RB/DNR1RB (Single/Double precision)
Computes the 1-norm of a real band matrix in band storage mode.

NR1CB/DNR1CB (Single/Double precision)
Computes the 1-norm of a complex band matrix in band storage mode.

DISTANCE BETWEEN TWO POINTS

DISL2/DDISL2 (Single/Double precision)
Computes the Euclidean (2-norm) distance between two points.

DISL1/DDISL1 (Single/Double precision)
Computes the 1-norm distance between two points.

DISLI/DDISLI (Single/Double precision)
Computes the infinity norm distance between two points.

VECTOR CONVOLUTIONS

VCONR/DVCONR (Single/Double precision)
Computes the convolution of two real vectors.

VCONC/DVCONC (Single/Double precision)
Computes the convolution of two complex vectors.

Extended Precision Arithmetic
(no single precision equivalent)

DQINI
Initializes an extended-precision accumulator with a double-precision scalar.

DQSTO
Stores a double-precision approximation to an extended-precision scalar.

DQADD
Adds a double-precision scalar to the accumulator in extended precision.

DQMUL
Multiplies double-precision scalars in extended precision.

ZQINI
Initializes an extended-precision complex accumulator to a double complex scalar.

ZQSTO
Stores a double complex approximation to an extended-precision complex scalar.

ZQADD
Adds a double complex scalar to the accumulator in extended precision.

ZQMUL
Multiplies double complex scalars using extended precision.

UTILITIES

Print

WRRRN
Prints a real rectangular matrix with integer row and column labels.

WRRRL
Prints a real rectangular matrix with a given format and labels.

WRIRN
Prints an integer rectangular matrix with integer row and column labels

WRIRL
Prints an integer rectangular matrix with a given format and labels.

WRCRN
Prints a complex rectangular matrix with integer row and column labels.

WRCRL
Prints a complex rectangular matrix with a given format and labels.

WROPT
Sets or Retrieves an option for printing a matrix.

PGOPT
Sets or Retrieves page width and length for printing.

Permute

PERMU
Rearranges the elements of an array as specified by a permutation.

PERMA
Permutes the rows or columns of a matrix.

Sort

SVRGN
Sorts a real array by algebraically increasing value.

SVRGP
Sorts a real array by algebraically increasing value and returns the permutation that rearranges the array.

SVIGN
Sorts an integer array by algebraically increasing value.

SVIGP
Sorts an integer array by algebraically increasing value and returns the permutation that rearranges the array.

SVRBN
Sorts a real array by nondecreasing absolute value.

SVRBP
Sorts a real array by nondecreasing absolute value and returns the permutation that rearranges the array.

SVIBN
Sorts an integer array by nondecreasing absolute value.

SVIBP
Sorts an integer array by nondecreasing absolute value and returns the permutation that rearranges the array.

Search

SRCH
Searches a sorted vector for a given scalar and returns its index.

ISRCH
Searches a sorted integer vector for a given integer and returns its index.

SSRCH
Searches a character vector, sorted in ascending ASCII order, for a given string and returns its index.

Character String Manipulation

ACHAR
Returns a character given its ASCII value.

IACHAR
Returns the integer ASCII value of a character argument.

ICASE
Returns the ASCII value of a character converted to uppercase.

IICSR
Compares two character strings using the ASCII collating sequence but without regard to case.

IIDEX
Determines the position in a string at which a given character sequence begins without regard to case.

CVTSI
Converts a character string containing an integer number into the corresponding integer form.

Time, Date and Version

CPSEC
Returns CPU time used in seconds.

TIMDY
Gets time of day.

TDATE
Gets today's date.

NDAYS
Computes the number of days from January 1, 1900, to the given date.

NDYIN
Gives the date corresponding to the number of days since January 1, 1900

IDYWK
Computes the day of the week for a given date.

VERML
Obtains IMSL MATH/LIBRARY-related version, system and serial numbers.

Random Number Generation

RNGET
Retrieves the current value of the seed used in the IMSL random number generators.

RNSET
Initializes a random seed for use in the IMSL random number generators.

RNOPT
Selects the uniform (0, 1) multiplicative congruential pseudorandom number generator.

RNUNF
Generates a pseudorandom number from a uniform (0, 1) distribution.

RNUN
Generates pseudorandom numbers from a uniform (0, 1) distribution.

Options Manager

IUMAG
This routine handles MATH/LIBRARY and STAT/LIBRARY type `INTEGER` options.

SUMAG
This routine handles MATH/LIBRARY and STAT/LIBRARY type `SINGLE PRECISION` options.

DUMAG
This routine handles MATH/LIBRARY and STAT/LIBRARY type `DOUBLE PRECISION` options.

Line Printer Graphics

PLOTP
Prints a plot of up to 10 sets of points.

Miscellaneous

PRIME
Decomposes an integer into its prime factors.

CONST
Returns the value of various mathematical and physical constants.

CUNIT
Converts `X` in units `XUNITS` to `Y` in units `YUNITS`.

HYPOT
Computes `SQRT(A**2 + B**2)` without underflow or overflow.

Special Functions

Elementary Functions

CARG
Evaluates the argument of a complex number.

CBRT
Evaluates the cube root.

CCBRT
Evaluates the complex cube root.

EXPRL
Evaluates the exponential function factored from first order, (EXP(X) – 1.0)/X.

CEXPRL
Evaluates the complex exponential function factored from first order.

CLOG10
Evaluates the principal value of the complex common logarithm.

ALNREL
Evaluates the natural logarithm of one plus the argument.

CLNREL
Evaluates the principal value of the complex natural logarithm of one plus the argument.

Trigonometric and Hyperbolic Functions

Trigonometric Functions

CTAN
Evaluates the complex tangent.

COT
Evaluates the cotangent.

CCOT
Evaluates the complex cotangent.

SINDG
Evaluates the sine for the argument in degrees.

COSDG
Evaluates the cosine for the argument in degrees.

CASIN
Evaluates the complex arc sine.

CACOS
Evaluates the complex arc cosine.

CATAN
Evaluates the complex arc tangent.

CATAN2
Evaluates the complex arc tangent of a ratio.

Hyperbolic Functions

CSINH
Evaluates the complex hyperbolic sine.

CCOSH
Evaluates the complex hyperbolic cosine.

CTANH
Evaluates the complex hyperbolic tangent.

Inverse Hyperbolic Functions

ASINH
Evaluates the arc hyperbolic sine.

CASINH
Evaluates the complex arc hyperbolic sine.

ACOSH
Evaluates the arc hyperbolic cosine.

CACOSH
Evaluates the complex arc hyperbolic cosine.

ATANH
Evaluates the arc hyperbolic tangent.

CATANH
Evaluates the complex arc hyperbolic tangent.

Exponential Integrals and Related Functions

EI
Evaluates the exponential integral for arguments greater than zero and the Cauchy principal value for arguments less than zero.

E1
Evaluates the exponential integral for arguments greater than zero and the Cauchy principal value of the integral for arguments less than zero.

ENE
Evaluates the exponential integral of integer order for arguments greater than zero scaled by EXP(X).

ALI
Evaluates the logarithmic integral.

SI
Evaluates the sine integral.

CI
Evaluates the cosine integral.

CIN
Evaluates a function closely related to the cosine integral.

SHI
Evaluates the hyperbolic sine integral.

CHI
Evaluates the hyperbolic cosine integral.

CINH
Evaluates a function closely related to the hyperbolic cosine integral.

Gamma Function and Related Functions

Factorial Function

FAC
Evaluates the factorial of the argument.

BINOM
Evaluates the binomial coefficient.

Gamma Function

GAMMA
Evaluates the complete gamma function.

CGAMMA
Evaluates the complete gamma function.

GAMR
Evaluates the reciprocal gamma function.

CGAMR
Evaluates the reciprocal complex gamma function.

ALNGAM
Evaluates the logarithm of the absolute value of the gamma function.

CLNGAM
Evaluates the complex natural logarithm of the gamma function.

ALGAMS
Returns the logarithm of the absolute value of the gamma function and the sign of gamma.

Incomplete Gamma Function

GAMI
Evaluates the incomplete gamma function.

GAMIC
Evaluates the complementary incomplete gamma function.

GAMIT
Evaluates the Tricomi form of the incomplete gamma function.

Psi Function

PSI
Evaluates the logarithmic derivative of the gamma function.

CPSI
Evaluates the logarithmic derivative of the gamma function for a complex argument.

Pochhammer's Function

POCH
Evaluates a generalization of Pochhammer's symbol.

POCH1
Evaluates a generalization of Pochhammer's symbol starting from the first order.

Beta Function

BETA
Evaluates the complete beta function.

CBETA
Evaluates the complex complete beta function.

ALBETA
Evaluates the natural logarithm of the complete beta function for positive arguments.

CLBETA
Evaluates the complex logarithm of the complete beta function.

BETAI
Evaluates the incomplete beta function ratio.

Error Function and Related Functions

Error Function

ERF
Evaluates the error function.

ERFC
Evaluates the complementary error function.

ERFCE
Evaluates the exponentially scaled complementary error function.

CERFE
Evaluates the complex scaled complemented error function.

ERFI
Evaluates the inverse error function.

ERFCI
Evaluates the inverse complementary error function.

DAWS
Evaluates Dawson's function.

Fresnel Integrals

FREC
Evaluates the cosine Fresnel integral.

FRESS
Evaluates the sine Fresnel integral.

Bessel Functions

Bessel Functions of Orders 0 and 1

BSJ0
Evaluates the Bessel function of the first kind of order zero.

BSJ1
Evaluates the Bessel function of the first kind of order one.

BSY0
Evaluates the Bessel function of the second kind of order zero.

BSY1
Evaluates the Bessel function of the second kind of order one.

BS10
Evaluates the modified Bessel function of the first kind of order zero.

BSI1
Evaluates the modified Bessel function of the first kind of order one.

BSK0
Evaluates the modified Bessel function of the third kind of order zero.

BSK1
Evaluates the modified Bessel function of the third kind of order one.

BSI0E
Evaluates the exponentially scaled modified Bessel function of the first kind of order zero.

BSI1E
Evaluates the exponentially scaled modified Bessel function of the first kind of order one.

BSK0E
Evaluates the exponentially scaled modified Bessel function of the third kind of order zero.

BSK1E
Evaluates the exponentially scaled modified Bessel function of the third kind of order one.

Series of Bessel Functions, Integer Order

BSJNS
Evaluates a sequence of Bessel functions of the first kind with integer order and real arguments.

CBJNS
Evaluates a sequence of Bessel functions of the first kind with integer order and complex arguments.

BSINS
Evaluates a sequence of modified Bessel functions of the first kind with integer order and real arguments.

CBINS
Evaluates a sequence of modified Bessel functions of the first kind with integer order and complex arguments.

Series of Bessel Functions, Real Order and Argument

BSJS
Evaluates a sequence of Bessel functions of the first kind with real order and real positive arguments

BSYS
Evaluates a sequence of Bessel functions of the second kind with real nonnegative order and real positive arguments.

BSIS
Evaluates a sequence of modified Bessel functions of the first kind with real order and real positive arguments.

BSIES
Evaluates a sequence of exponentially scaled modified Bessel functions of the first kind with nonnegative real order and real positive arguments.

BSKS
Evaluates a sequence of modified Bessel functions of the third kind of fractional order.

BSKES
Evaluates a sequence of exponentially scaled modified Bessel functions of the third kind of fractional order.

Series of Bessel Functions, Real Order and Complex Argument

CBJS
Evaluates a sequence of Bessel functions of the first kind with real order and complex arguments.

CBYS
Evaluates a sequence of Bessel functions of the second kind with real order and complex arguments.

CBIS
Evaluates a sequence of modified Bessel functions of the first kind with real order and complex arguments.

CBKS
Evaluates a sequence of modified Bessel functions of the second kind with real order and complex arguments.

Kelvin Functions

BER0
Evaluates the Kelvin function of the first kind, ber, of order zero.

BE10
Evaluates the Kelvin function of the first kind, bei, of order zero.

AKER0
Evaluates the Kelvin function of the second kind, ker, of order zero.

AKEI0
Evaluates the Kelvin function of the second kind, kei, of order zero.

BERP0
Evaluates the derivative of the Kelvin function of the first kind, ber, of order zero.

BEIP0
Evaluates the derivative of the Kelvin function of the first kind, bei, of order zero.

AKERP0
Evaluates the derivative of the Kelvin function of the second kind, ker, of order zero.

AKEIP0
Evaluates the Kelvin function of the second kind, kei, of order zero.

BER1
Evaluates the Kelvin function of the first kind, ber, of order one.

BEI1
Evaluates the Kelvin function of the first kind, bei, of order one.

AKER1
Evaluates the Kelvin function of the second kind, ker, of order one.

AKEI1
Evaluates the Kelvin function of the second kind, kei, of order one.

Airy Functions

AI
Evaluates the Airy function.

BI
Evaluates the Airy function of the second kind.

AID
Evaluates the derivative of the Airy function.

BID
Evaluates the derivative of the Airy function of the second kind.

AIE
Evaluates the exponentially scaled Airy function.

BIE
Evaluates the exponentially scaled Airy function of

the second kind.

AIDE
Evaluates the exponentially scaled derivative of the Airy function.

BIDE
Evaluates the exponentially scaled derivative of the Airy function of the second kind.

Elliptic Integrals

ELK
Evaluates the complete elliptic integral of the kind K(x).

ELE
Evaluates the complete elliptic integral of the second kind $E(x)$.

ELRF
Evaluates Carlson's incomplete elliptic integral of the first kind R_F(X, Y, Z).

ELRD
Evaluates Carlson's incomplete elliptic integral of the second kind R_D(X, Y, Z).

ELRJ
Evaluates Carlson's incomplete elliptic integral of the third kind R_J(X, Y, Z, RHO).

ELRC
Evaluates an elementary integral from which inverse circular functions, logarithms and inverse hyperbolic functions can be computed.

Elliptic and Related Functions

Weierstrass Elliptic and Related Functions

CWPL
Evaluates the Weierstrass' ℘ function in the lemniscatic case for complex argument with unit period parallelogram.

CWPLD
Evaluates the first derivative of the Weierstrass' ℘ function in the lemniscatic case for complex argument with unit period parallelogram.

CWPQ
Evaluates the Weierstrass' ℘ function in the equianharmonic case for complex argument with unit period parallelogram.

CWPQD
Evaluates the first derivative of the Weierstrass' ℘ function in the equianharmonic case for complex argument with unit period parallelogram.

Jacobi Elliptic Functions

EJSN
Evaluates the Jacobi elliptic function sn(x, m).

CEJSN
Evaluates the complex Jacobi elliptic function sn(z, m).

EJCN
Evaluates the Jacobi elliptic function cn(x, m).

CEJCN
Evaluates the complex Jacobi elliptic integral cn(z, m).

EJDN
Evaluates the Jacobi elliptic function dn(x, m).

CEJDN
Evaluates the complex Jacobi elliptic integral dn(z, m).

Probability Distribution Functions and Inverses

Discrete Random Variables: Distribution Functions and Probability Functions

BINDF
Evaluates the binomial distribution function.

BINPR
Evaluates the binomial probability function.

HYPDF
Evaluates the hypergeometric distribution function.

HYPPR
Evaluates the hypergeometric probability function.

POIDF
Evaluates the Poisson distribution function.

POIPR
Evaluates the Poisson probability function.

Continuous Random Variables: Distribution Functions and Their Inverses

AKS1DF
Evaluates the distribution function of the one-sided

Kolmogorov-Smirnov goodness of fit D^+ or D^- test statistic based on continuous data for one sample.

AKS2DF
Evaluates the distribution function of the Kolmogorov-Smirnov goodness of fit *D* test statistic based on continuous data for two samples.

ANORDF
Evaluates the standard normal (Gaussian) distribution function.

ANORIN
Evaluates the inverse of the standard normal (Gaussian) distribution function.

BETDF
Evaluates the beta probability distribution function.

BETIN
Evaluates the inverse of the beta distribution function.

BNRDF
Evaluates the bivariate normal distribution function.

CHIDF
Evaluates the chi-squared distribution function.

CHIIN
Evaluates the inverse of the chi-squared distribution function.

CSNDF
Evaluates the noncentral chi-squared distribution function.

FDF
Evaluates the *F* distribution function.

FIN
Evaluates the inverse of the *F* distribution function

GAMDF
Evaluates the gamma distribution function

TDF
Evaluates the Student's *t* distribution function.

TIN
Evaluates the inverse of the Student's *t* distribution function.

TNDF
Evaluates the noncentral Student's *t* distribution function.

General Continuous Random Variables

GCDF
Evaluates a general continuous cumulative distribution function given ordinates of the density.

GCIN
Evaluates the inverse of a general continuous cumulative distribution function given ordinates of the density.

Mathieu Functions

MATEE
Evaluates the eigenvalues for the periodic Mathieu functions

MATCE
Evaluates a sequence of even, periodic, integer order, real Mathieu functions.

MATSE
Evaluates a sequence of odd, periodic, integer order, real Mathieu functions.

Miscellaneous Functions

SPENC
Evaluates a form of Spence's integral.

INITS
Initializes the orthogonal series so the function value is the number of terms needed to insure the error is no larger than the requested accuracy.

CSEVL
Evaluates the N-term Chebyshev series.

IMSL STAT/LIBRARY

Basic Statistics

Frequency Tabulations

OWFRQ
Tallies observations into a one-way frequency table.

TWFRQ
Tallies observations into a two-way frequency table.

FREQ
Tallies multivariate observations into a multiway frequency table.

Univariate Summary Statistics

UVSTA
Computes basic univariate statistics.

Ranks and Order Statistics

RANKS
Computes the ranks, normal scores, or exponential scores for a vector of observations.

LETTR
Produces a letter value summary.

ORDST
Determines order statistics.

EQTIL
Computes empirical quantiles.

Parametric Estimates and Tests

(See also Univariate Summary Statistics)

TWOMV
Computes statistics for mean and variance inferences using samples from two normal populations.

BINES
Estimates the parameter *p* of the binomial distribution.

POIES
Estimates the parameter of the Poisson distribution.

NRCES
Computes maximum likelihood estimates of the mean and variance from grouped and/or censored normal data.

Grouped Data

GRPES
Computes basic statistics from grouped data.

Continous Data in a Table

CSTAT
Computes cell frequencies, cell means, and cell sums of squares for multivariate data.

MEDPL
Computes a median polish of a two-way table.

Regression

Simple Linear Regression

RLINE
Fits a line to a set of data points using least squares.

RONE
Analyzes a simple linear regression model.

RINCF
Performs response control given a fitted simple linear regression model.

RINPF
Performs inverse prediction given a fitted simple linear regression model.

Multivariate General Linear Model Analysis

Model Fitting

RLSE
Fits a multiple linear regression model using least squares.

RCOV
Fits a multivariate linear regression model given the variance-covariance matrix.

RGIVN
Fits a multivariate linear regression model via fast Givens transformations.

RGLM
Fits a multivariate general linear model.

RLEQU
Fits a multivariate linear regression model with linear equality restrictions $H B = G$ imposed on the regression parameters given results from routine

`RGIVN` after `IDO` = 1 and `IDO` = 2 and prior to `IDO` = 3.

Statistical Inference

RSTAT
Computes statistics related to a regression fit given the coefficient estimates.

RCOVB
Computes the estimated variance-covariance matrix of the estimated regression coefficients given the *R* matrix.

CESTI
Constructs an equivalent completely testable multivariate general linear hypothesis $H\,BU = G$ from a partially testable hypothesis $H_pBU = G_p$.

RHPSS
Computes the matrix of sums of squares and crossproducts for the multivariate general linear hypothesis $H\,BU = G$ given the coefficient estimates.

RHPTE
Performs tests for a multivariate general linear hypothesis $H\,BU = G$ given the hypothesis sums of squares and crossproducts matrix S_H and the error sums of squares and crossproducts matrix S_E.

RLOFE
Computes a lack of fit test based on exact replicates for a fitted regression model.

RLOFN
Computes a lack of fit test based on near replicates for a fitted regression model.

RCASE
Computes case statistics and diagnostics given data points, coefficient estimates.

ROTIN
Computes diagnostics for detection of outliers and influential data points given residuals and the *R* matrix for a fitted general linear model.

Utilities for Classification Variables

GCLAS
Gets the unique values of each classification variable.

GRGLM
Generates regressors for a general linear model.

Variables Selection

RBEST
Selects the best multiple linear regression models.

RSTEP
Builds multiple linear regression models using forward selection, backward selection, or stepwise selection.

GSWEP
Performs a generalized sweep of a row of a nonnegative definite matrix.

RSUBM
Retrieves a symmetric submatrix from a symmetric matrix.

Polynominal Regression and Second-Order Models

Polynominal Regression Analysis

RCURV
Fits a polynomial curve using least squares.

RPOLY
Analyzes a polynomial regression model.

Second-Order Model Design

RCOMP
Generates an orthogonal central composite design.

Utility Routines for Polynomial Models and Second-Order Models

Service Routines

RFORP
Fits an orthogonal polynomial regression model.

RSTAP
Computes summary statistics for a polynomial regression model given the fit based on orthogonal polynomials.

RCASP
Computes case statistics for a polynomial regression model given the fit based on orthogonal polynomials.

OPOLY
Generates orthogonal polynomials with respect to *x*-values and specified weights.

GCSCP
Generates centered variables, squares, and crossproducts.

TCSCP
Transforms coefficients from a second order response surface model generated from squares and crossproducts of centered variables to a model using uncentered variables.

Nonlinear Regression Analysis

RNLIN
Fits a nonlinear regression model.

Fitting Linear Models Based on Criteria Other Than Least Squares

RLAV
Fits a multiple linear regression model using the least absolute values criterion.

RLLP
Fits a multiple linear regression model using the L_p norm criterion.

RLMV
Fits a multiple linear regression model using the minimax criterion.

Correlation

The Correlation Matrix

CORVC
Computes the variance-covariance or correlation matrix.

COVPL
Computes a pooled variance-covariance matrix from the observations.

PCORR
Computes partial correlations or covariances from the covariance or correlation matrix.

RBCOV
Computes a robust estimate of a covariance matrix and mean vector.

Correlation Measures for a Contingency Table

CTRHO
Estimates the bivariate normal correlation coefficient using a contingency table.

TETCC
Categorizes bivariate data and computes the tetrachoric correlation coefficient.

A Dichotomous Variable with a Classification Variable

BSPBS
Computes the biserial and point-biserial correlation coefficients for a dichotomous variable and a numerically measurable classification variable.

BSCAT
Computes the biserial correlation coefficient for a dichotomous variable and a classification variable.

Measures Based Upon Ranks

CNCRD
Calculates and test the significance of the Kendall coefficient of concordance.

KENDL
Computes and test Kendall’s rank correlation coefficient.

KENDP
Computes the frequency distribution of the total score in Kendall’s rank correlation coefficient.

Analysis of Variance

General Analysis

AONEW
Analyzes a one-way classification model.

AONEC
Analyzes a one-way classification model with covariates.

ATWOB
Analyzes a randomized block design or a two-way balanced design.

ABIBD
Analyzes a balanced incomplete block design or a

balanced lattice design.

ALATN
Analyzes a Latin square design.

ANWAY
Analyzes a balanced *n*-way classification model with fixed effects.

ABALD
Analyzes a balanced complete experimental design for a fixed, random, or mixed model.

ANEST
Analyzes a completely nested random model with possibly unequal numbers in the subgroups.

Inference on Means and Variance Components

CTRST
Computes contrast estimates and sums of squares.

SCIPM
Computes simultaneous confidence intervals on all pairwise differences of means.

SNKMC
Performs Student-Newman-Keuls multiple comparison test.

CIDMS
Computes a confidence interval on a variance component estimated as proportional to the difference in two mean squares in a balanced complete experimental design.

Service Routine

ROREX
Reorders the responses from a balanced complete experimental design.

Categorical and Discrete Data Analysis

Statistics in the Two-Way Table

CTTWO
Performs a chi-squared analysis of a 2 by 2 contingency table.

CTCHI
Performs a chi-squared analysis of a two-way contingency table.

CTPRB
Computes exact probabilities in a two-way contingency table.

CTEPR
Computes Fisher's exact test probability and a hybrid approximation to the Fisher exact test probability for a contingency table using the network algorithm.

Log-Linear Models

PRPFT
Performs iterative proportional fitting of a contingency table using a loglinear model.

CTLLN
Computes model estimates and associated statistics for a hierarchical log-linear model.

CTPAR
Computes model estimates and covariances in a fitted log-linear model.

CTASC
Computes partial association statistics for log-linear models in a multidimensional contingency table.

CTSTP
Builds hierarchical log-linear models using forward selection, backward selection, or stepwise selection.

Randomization Tests

CTRAN
Performs generalized Mantel-Haenszel tests in a stratified contingency table.

Generalized Categorical Models

CTGLM
Analyzes categorical data using logistic, Probit, Poisson, and other generalized linear models.

Weighted Least-Squares Analysis

CTWLS
Performs a generalized linear least-squares analysis of transformed probabilities in a two-dimensional contingency table.

Nonparametric Statistics

One Sample or Matched Samples

Test of Location

SIGNT
Performs a sign test of the hypothesis that a given value is a specified quantile of a distribution.

SNRNK
Performs a Wilcoxon signed rank test.

Tests for Trends

NCTRD
Performs the Noether test for cyclical trend.

SDPLC
Performs the Cox and Stuart sign test for trends in dispersion and location.

Ties

NTIES
Computes tie statistics for a sample of observations.

Two Independent Samples

RNKSM
Performs the Wilcoxon rank sum test.

INCLD
Performs an includance test.

More than Two Samples

One Way Tests of Location

KRSKL
Performs a Kruskal-Wallis test for identical population medians.

BHAKV
Performs a Bhapkar *V* test.

Two-way Tests of Location

FRDMN
Performs Friedman's test for a randomized complete block design.

QTEST
Performs a Cochran *Q* test for related observations.

Tests for Trends

KTRND
Performs *k*-sample trends test against ordered alternatives.

Tests of Goodness-of-Fit and Randomness

General Goodness-of-Fit Tests for a Specified Distribution

KSONE
Performs a Kolmogorov-Smirnov one-sample test for continuous distributions.

CHIGF
Performs a chi-squared goodness-of-fit test.

SPWLK
Performs a Shapiro-Wilk *W*-test for normality.

LILLF
Performs Lilliefors test for an exponential or normal distribution.

MVMMT
Computes Mardia's multivariate measures of skewness and kurtosis and test for multivariate normality.

Two Sample Tests

KSTWO
Performs a Kolmogorov-Smirnov two-sample test.

Tests for Randomness

RUNS
Performs a runs up test.

PAIRS
Performs a pairs test.

DSQAR
Performs a d^2 test.

DCUBE
Performs a triplets test.

Time Series Analysis and Forecasting

GENERAL METHODOLOGY

TRANSFORMATION OF DATA

BCTR
Performs a forward or an inverse Box-Cox (power) transformation.

DIFF
Differences a time series.

SAMPLE CORRELATION FUNCTION

ACF
Computes the sample autocorrelation function of a stationary time series.

PACF
Computes the sample partial autocorrelation function of a stationary time series.

CCF
Computes the sample cross-correlation function of two stationary time series.

MCCF
Computes the multichannel cross-correlation function of two mutually stationary multichannel time series.

TIME DOMAIN METHODOLOGY

NONSEASONAL AUTOREGRESSIVE MOVING AVERAGE MODEL

ARMME
Computes method of moments estimates of the autoregressive parameters of an ARMA model.

MAMME
Computes method of moments estimates of the moving average parameters of an ARMA model.

NSPE
Computes preliminary estimates of the autoregressive and moving average parameters of an ARMA model.

NSLSE
Computes least-squares estimates of parameters for a nonseasonal ARMA model.

SPWF
Computes the Wiener forecast operator for a stationary stochastic process.

NSBJF
Computes Box-Jenkins forecasts and their associated probability limits for a nonseasonal ARMA model.

TRANSFER FUNCTION MODEL

IRNSE
Computes estimates of the impulse response weights and noise series of a univariate transfer function model.

TFPE
Computes preliminary estimates of parameters for a univariate transfer function model.

MULTICHANNEL TIME SERIES

MLSE
Computes least-squares estimates of a linear regression model for a multichannel time series with a specified base channel.

MWFE
Computes least-squares estimates of the multichannel Wiener filter coefficients for two mutually stationary multichannel time series.

KALMN
Performs Kalman filtering and evaluates the likelihood function for the state-space model.

DIAGNOSTICS

LOFCF
Performs lack-of-fit test for a univariate time series or transfer function given the appropriate correlation function.

FREQUENCY DOMAIN METHODOLOGY

SMOOTHING FUNCTIONS

DIRIC
Computes the Dirichlet kernel.

FEJER
Computes the Fejér kernel.

Spectral Density Estimation

PFFT
Computes the periodogram of a stationary time series using a fast Fourier transform.

SSWD
Estimates the nonnormalized spectral density of a stationary time series using a spectral window given the time series data.

SSWP
Estimates the nonnormalized spectral density of a stationary time series using a spectral window given the periodogram.

SWED
Estimates the nonnormalized spectral density of a stationary time series based on specified periodogram weights given the time series data.

SWEP
Estimates the nonnormalized spectral density of a stationary time series based on specified periodogram weights given the periodogram.

Cross-Spectral Density Estimation

CPFFT
Computes the cross periodogram of two stationary time series using a fast Fourier transform.

CSSWD
Estimates the nonnormalized cross-spectral density of two stationary time series using a spectral window given the time series data.

CSSWP
Estimates the nonnormalized cross-spectral density of two stationary time series using a spectral window given the spectral densities and cross periodogram.

CSWED
Estimates the nonnormalized cross-spectral density of two stationary time series using a weighted cross periodogram given the time series data.

CSWEP
Estimates the nonnormalized cross-spectral density of two stationary time series using a weighted cross periodogram given the spectral densities and cross periodogram.

Covariance Structures and Factor Analysis

Principal Components

PRINC
Computes principal components from a variance-covariance matrix or a correlation matrix.

KPRIN
Maximum likelihood or least-squares estimates for principal components from one or more matrices.

Factor Analysis

Factor Extraction

FACTR
Extracts initial factor loading estimates in factor analysis.

Factor Rotation and Summarization

FROTA
Computes an orthogonal rotation of a factor loading matrix using a generalized orthomax criterion, including quartimax, varimax, and equamax rotations.

FOPCS
Computes an orthogonal Procrustes rotation of a factor-loading matrix using a target matrix.

FDOBL
Computes a direct oblimin rotation of a factor loading matrix.

FPRMX
Computes an oblique Promax or Procrustes rotation of a factor loading matrix using a target matrix, including pivot and power vector options.

FHARR
Computes an oblique rotation of an unrotated factor loading matrix using the Harris-Kaiser method.

FGCRF
Computes direct oblique rotation according to a generalized fourth-degree polynomial criterion.

FIMAG
Computes the image transformation matrix.

FRVAR
Computes the factor structure and the variance explained by each factor.

Factor Scores

FCOEF
Computes a matrix of factor score coefficients for input to the routine FSCOR.

FSCOR
Computes a set of factor scores given the factor score coefficient matrix.

Residual Correlation

FRESI
Computes communalities and the standardized factor residual correlation matrix.

Independence of Sets of Variables and Canonical Correlation Analysis

MVIND
Computes a test for the independence of k sets of multivariate normal variables.

CANCR
Performs canonical correlation analysis from a data matrix.

CANVC
Performs canonical correlation analysis from a variance-covariance matrix or a correlation matrix.

Discriminant Analysis

Parametric Discrimination

DSCRM
Performs a linear or a quadratic discriminant function analysis among several known groups.

DMSCR
Uses Fisher's linear discriminant analysis method to reduce the number of variables.

Nonparametric Discrimination

NNBRD
Performs k nearest neighbor discrimination.

Cluster Analysis

Hierarchical Cluster Analysis

CDIST
Computes a matrix of dissimilarities (or similarities) between the columns (or rows) of a matrix.

CLINK
Performs a hierarchical cluster analysis given a distance matrix.

CNUMB
Computes cluster membership for a hierarchical cluster tree.

K-means Cluster Analysis

KMEAN
Performs a *K*-means (centroid) cluster analysis.

Sampling

SMPPR
Computes statistics for inferences regarding the population proportion and total given proportion data from a simple random sample.

SMPPS
Computes statistics for inferences regarding the population proportion and total given proportion data from a stratified random sample.

SMPRR
Computes statistics for inferences regarding the population mean and total using ratio or regression estimation, or inferences regarding the population ratio given a simple random sample.

SMPRS
Computes statistics for inferences regarding the population mean and total using ratio or regression estimation given continuous data from a stratified random sample.

SMPSC
Computes statistics for inferences regarding the population mean and total using single stage cluster sampling with continuous data.

SMPSR
Computes statistics for inferences regarding the population mean and total, given data from a simple random sample.

SMPSS
Computes statistics for inferences regarding the population mean and total, given data from a stratified random sample.

SMPST
Computes statistics for inferences regarding the population mean and total given continuous data from a two-stage sample with equisized primary units.

Survival Analysis, Life Testing and Reliability

Survival Analysis

KAPMR
Computes Kaplan-Meier estimates of survival probabilities in stratified samples.

KTBLE
Prints Kaplan-Meier estimates of survival probabilities in stratified samples.

TRNBL
Computes Turnbull's generalized Kaplan-Meier estimates of survival probabilities in samples with interval censoring.

PHGLM
Analyzes time event data via the proportional hazards model.

SVGLM
Analyzes censored survival data using a generalized linear model.

STBLE
Estimates survival probabilities and hazard rates for various parametric models.

Actuarial Tables

ACTBL
Produces population and cohort life tables.

Multidimensional Scaling

Multidimensional Scaling

MSIDV
Performs individual-differences multidimensional scaling for metric data using alternating least squares.

Utility Routines

MSDST
Computes distances in a multidimensional scaling model.

MSSTN
Transforms dissimilarity/similarity matrices and replace missing values by estimates to obtain standardized dissimilarity matrices.

MSDBL
Obtains normalized product-moment (double centered) matrices from dissimilarity matrices.

MSINI
Computes initial estimates in multidimensional scaling models.

MSTRS
Computes various stress criteria in multidimensional scaling.

Density and Hazard Estimation

Estimates for a Density

DESPL
Performs nonparametric probability density function estimation by the penalized likelihood method.

DESKN
Performs nonparametric probability density function estimation by the kernel method.

DNFFT
Computes Gaussian kernel estimates of a univariate density via the fast Fourier transform over a fixed interval.

DESPT
Estimates a probability density function at specified points using linear or cubic interpolation.

Modified Likelihood Estimates for Hazards

HAZRD
Performs nonparametric hazard rate estimation using kernel functions and quasi-likelihoods.

HAZEZ
Performs nonparametric hazard rate estimation using kernel functions. Easy-to-use version of HAZRD.

HAZST
Performs hazard rate estimation over a grid of points using a kernel function.

Histograms

VHSTP
Prints a vertical histogram.

VHS2P
Prints a vertical histogram with every bar subdivided into two parts.

HHSTP
Prints a horizontal histogram.

Scatterplots

SCTP
Prints a scatter plot of several groups of data

Exploratory Data Analysis

BOXP
Prints boxplots for one or more samples.

STMLP
Prints a stem-and-leaf plot.

Empirical Probability Distribution

CDFP
Prints a sample cumulative distribution function (CDF), a theoretical CDF, and confidence band information.

CDF2P
Prints a plot of two sample cumulative distribution functions.

PROBP
Prints a probability plot.

Other Graphics Routines

PLOTP
Prints a plot of up to 10 sets of points.

TREEP
Prints a binary tree.

Discrete Random Variables: Distribution Functions and Probability Functions

BINDF
Evaluates the binomial distribution function.

BINPR
Evaluates the binomial probability function.

HYPDF
Evaluates the hypergeometric distribution function.

HYPPR
Evaluates the hypergeometric probability function.

POIDF
Evaluates the Poisson distribution function.

POIPR
Evaluates the Poisson probability function.

Continuous Random Variables: Distribution Functions and Their Inverses

AKS1DF
Evaluates the distribution function of the one-sided Kolmogorov-Smirnov goodness of fit D^+ or D^- test statistic based on continuous data for one sample.

AKS2DF
Evaluates the distribution function of the Kolmogorov-Smirnov goodness of fit D test statistic based on continuous data for two samples.

ANORDF
Evaluates the standard normal (Gaussian) distribution function.

ANORIN
Evaluates the standard normal (Gaussian) distribution function.

BETDF
Evaluates the beta probability distribution function.

BETIN
Evaluates the inverse of the beta distribution function.

BNRDF
Evaluates the bivariate normal distribution function.

CHIDF
Evaluates the chi-squared distribution function.

CHIIN
Evaluates the inverse of the chi-squared distribution function.

CSNDF
Evaluates the noncentral chi-squared distribution function.

CSNIN
Evaluates the inverse of the noncentral chi-squared function.

FDF
Evaluates the *F* distribution function.

FIN
Evaluates the inverse of the *F* distribution function.

GAMDF
Evaluates the gamma distribution function.

GAMIN
Evaluates the inverse of the gamma distribution function.

TDF
Evaluates the Student's *t* distribution function.

TIN
Evaluates the inverse of the Student's *t* distribution function.

TNDF
Evaluates the noncentral Student's *t* distribution function.

TNIN
Evaluates the inverse of the noncentral Student's t distribution function.

General Continuous Random Variables

GCDF
Evaluates a general continuous cumulative distribution function given ordinates of the density.

GCIN
Evaluates the inverse of a general continuous cumulative distribution function given ordinates of the density.

GFNIN
Evaluates the inverse of a general continuous cumulative distribution function given in a subprogram.

Random Number Generation

Utility Routines for Random Number Generators

RNOPT
Selects the uniform (0,1) multiplicative congruential pseudorandom number generator.

RNOPG
Retrieves the indicator of the type of uniform random number generator.

RNSET
Initializes a random seed for use in the IMSL random number generators.

RNGET
Retrieves the current value of the seed used in the IMSL random number generators.

RNSES
Initializes the table in the IMSL random number generators that use shuffling.

RNGES
Retrieves the current value of the table in the IMSL random number generators that use shuffling.

RNSEF
Retrieves the array used in the IMSL `GFSR` random number generator.

RNGEF
Retrieves the current value of the array used in the IMSL `GFSR` random number generator.

RNISD
Determines a seed that yields a stream beginning 100,000 numbers beyond the beginning of the stream yielded by a given seed used in IMSL multiplicative congruential generators (with no shufflings).

Basic Uniform Distribution

RNUN
Generates pseudorandom numbers from a uniform (0, 1) distribution.

RNUNF
Generates a pseudorandom number from a uniform (0, 1) distribution.

Univariate Discrete Distributions

RNBIN
Generates pseudorandom numbers from a binomial distribution.

RNGDA
Generates pseudorandom numbers from a general discrete distribution using an alias method.

RNGDS
Sets up table to generate pseudorandom numbers from a general discrete distribution.

RNGDT
Generates pseudorandom numbers from a general discrete distribution using a table lookup method.

RNGEO
Generates pseudorandom numbers from a geometric distribution.

RNHYP
Generates pseudorandom numbers from a hypergeometric distribution.

RNLGR
Generates pseudorandom numbers from a logarithmic distribution.

RNNBN
Generates pseudorandom numbers from a negative binomial distribution.

RNPOI
Generates pseudorandom numbers from a Poisson distribution.

RNUND
Generates pseudorandom numbers from a discrete uniform distribution.

Univariate Continuous Distributions

RNBET
Generates pseudorandom numbers from a beta distribution.

RNCHI
Generates pseudorandom numbers from a chi-squared distribution.

RNCHY
Generates pseudorandom numbers from a Cauchy distribution.

RNEXP
Generates pseudorandom numbers from a standard exponential distribution.

RNEXT
Generates pseudorandom numbers from a mixture of two exponential distributions.

RNGAM
Generates pseudorandom numbers from a standard gamma distribution.

RNGCS
Sets up table to generate pseudorandom numbers from a general continuous distribution.

RNGCT
Generates pseudorandom numbers from a general continuous distribution.

RNLNL
Generates pseudorandom numbers from a lognormal distribution.

RNNOA
Generates pseudorandom numbers from a standard normal distribution using an acceptance/rejection method.

RNNOF
Generates a pseudorandom number from a standard normal distribution.

RNNOR
Generates pseudorandom numbers from a standard normal distribution using an inverse CDF method.

RNSTA
Generates pseudorandom numbers from a stable distribution.

RNSTT
Generates pseudorandom numbers from a Student's *t* distribution.

RNTRI
Generates pseudorandom numbers from a triangular distribution on the interval (0, 1).

RNVMS
Generates pseudorandom numbers from a von Mises distribution.

RNWIB
Generates pseudorandom numbers from a Weibull distribution.

Multivariate Distributions

RNCOR
Generates a pseudorandom orthogonal matrix or a correlation matrix.

RNDAT
Generates pseudorandom numbers from a multivariate distribution determined from a given sample.

RNMTN
Generates pseudorandom numbers from a multinomial distribution.

RNMVN
Generates pseudorandom numbers from a multivari-

ate normal distribution.

RNSPH
Generates pseudorandom points on a unit circle or K-dimensional sphere.

RNTAB
Generates a pseudorandom two-way table.

Order Statistics

RNNOS
Generates pseudorandom order statistics from a standard normal distribution.

RNUNO
Generates pseudorandom order statistics from a uniform (0, 1) distribution.

Stochastic Processes

RNARM
Generates a time series from a specified ARMA model.

RNNPP
Generates pseudorandom numbers from a nonhomogenous Poisson process.

Samples and Permutations

RNPER
Generates a pseudorandom permutation.

RNSRI
Generates a simple pseudorandom sample of indices.

RNSRS
Generates a simple pseudorandom sample from a finite population.

Utilities

Print

WRRRN
Prints a real rectangular matrix with integer row and column labels.

WRRRL
Prints a real rectangular matrix with a given format and labels.

WRIRN
Prints an integer rectangular matrix with integer row and column labels.

WRIRL
Prints an integer rectangular matrix with a given format and labels.

WROPT
Sets or retrieves an option for printing a matrix.

PGOPT
Sets or retrieves page width and length for printing.

Permute

PERMU
Rearranges the elements of an array as specified by a permutation.

PERMA
Permutes the rows or columns of a matrix.

RORDM
Reorders rows and columns of a symmetric matrix.

MVNAN
Moves any rows of a matrix with the IMSL missing value code NaN (not a number) in the specified columns to the last rows of the matrix.

Sort

SVRGN
Sorts a real array by algebraically increasing value.

SVRGP
Sorts a real array by algebraically increasing value and returns the permutation that rearranges the array.

SVIGN
Sorts an integer array by algebraically increasing value.

SVIGP
Sorts an integer array by algebraically increasing value and returns the permutation that rearranges the array.

SCOLR
Sorts columns of a real rectangular matrix using keys in rows.

SROWR
Sorts rows of a real rectangular matrix using keys in columns.

Search

SRCH
Searches a sorted vector for a given scalar and returns its index.

ISRCH
Searches a sorted integer vector for a given integer and returns its index.

SSRCH
Searches a character vector, sorted in ascending ASCII order, for a given string and returns its index.

Character String Manipulation

ACHAR
Returns a character given its ASCII value.

IACHAR
Returns the integer ASCII value of a character argument.

ICASE
Returns the ASCII value of a character converted to uppercase.

IICSR
Compares two character strings using the ASCII collating sequence but without regard to case.

IIDEX
Determines the position in a string at which a given character sequence begins without regard to case.

CVTSI
Converts a character string containing an integer number into the corresponding integer form.

Time, Date and Version

CPSEC
Returns CPU time used in seconds.

TIMDY
Gets time of day.

TDATE
Gets today's date.

NDAYS
Computes the number of days from January 1, 1900, to the given date.

NDYIN
Gives the date corresponding to the number of days since January 1, 1900.

IDYWK
Computes the day of the week for a given date.

VERSL
Obtains STAT/LIBRARY-related version, system and serial numbers.

Retrieval of Data Sets

GDATA
Retrieves a commonly analyzed data set.

Library Environments Utilities

The following routines are documented in the Reference Material sections of the IMSL® MATH/LIBRARY® and IMSL® STAT/LIBRARY® User's Manuals.

ERSET
Sets error handler default print and stop actions.

IERCD
Retrieves the code for an informational error.

N1RTY
Retrieves an error type for the most recently called IMSL routine.

IMACH
Retrieves interger machine constants.

AMACH
Retrieves machine constants.

IFNAN
Checks if a floating-point number is NaN (not a number).

UMACH
Sets or Retrieves input or output device unit numbers.

IMSL C NUMERICAL LIBRARY

The IMSL C Numerical Library (CNL) is a comprehensive set of over 300 pre-built mathematical and statistical analysis functions that C or C++ programmers can embed directly into their numerical analysis applications. CNL's functions are based upon the same algorithms contained in the company's highly regarded IMSL Fortran 90 MP Library. Visual Numerics, Inc. has been providing algorithms for mathematical and statistical computations under the IMSL name since 1970.

CNL significantly shortens program development time by taking full advantage of the intrinsic characteristics and desirable features of the C language. Variable argument lists simplify calling sequences. The concise set of required arguments contains only the information necessary for usage. Optional arguments provide added functionality and power to each function. You'll find that using CNL saves significant effort in your source code development and thousands of dollars in the design, development, testing and maintenance of your application.

JNL – A NUMERICAL LIBRARY FOR JAVA

JNL is a 100% Pure Java numerical library for the Java environment. The library extends core Java numerics and allows developers to seamlessly integrate advanced mathematical functions into their Java applications.

JNL is an object-oriented implementation of several important classes of mathematical functions drawn from the IMSL algorithm repository. Visual Numerics has taken individual algorithms and reimplemented them as object-oriented, Java methods. JNL is designed with extensibility in mind; new classes may be derived from existing ones to add functionality to satisfy particular requirements.

Because JNL is a 100% Pure Java class library, it can be deployed on any platform that supports Java. A JNL-based application will work seamlessly on a PC, a Macintosh, a UNIX workstation or any other Java-enabled platform.

JNL can be used to write client-side applets, server-side applications or even non-networked desktop applications. JNL applets perform all processing on the Java client, whether it is a thin client, such as a network computer, a PC or workstation equipped with a Java Virtual Machine. Client-side processing reduces the number of "round trips" to a networked server, which in turn minimizes network traffic and system latency. JNL is Visual Numerics' contribution to the worldwide Java development community, and is available free of charge via our Website.

WORLDWIDE OFFICES

USA
Visual Numerics, Inc.
1300 West Sam Houston Pkwy South
Suite 150
Houston, TX 77042
UNITED STATES
Tel: 800-222-4675
713-784-3131
E-mail: info@vni.com

United Kingdom
Visual Numerics International, Ltd.,
United Kingdom
Centennial Court
Easthampstead Road
Bracknell, Berkshire RG12 1YQ
England
UNITED KINGDOM
Tel: +44-1-344-45-8700
E-mail: info@vniuk.co.uk

France
Visual Numerics SARL
Tour Europe
33 Place des Corolles Cedex
F-92049 Paris La Defense
FRANCE
Tel: +33-1-46-93-94-20
E-mail: info@vni-paris.fr

Germany
Visual Numerics International GmbH,
Germany
Zettachring 10, D-70567
Stuttgart
GERMANY
Tel: +49-711-1328-70
E-mail: info@visual-numerics.de

Mexico
Visual Numerics S.A. de C.V.
Cerrada de Berna 3, Tercer Piso
Col. Juarez
Mexico, D.F. C.P. 06600
MEXICO
Tel: +52-5514-9730 or 9628
avadillo@mail.internet.com.mx

Japan
Visual Numerics Japan, Inc.
Gobancho Hikari Building, 4th Floor
14 Goban-cho Chiyoda-Ku
Tokyo
JAPAN
Tel: +81-3-5211-7760
E-mail: vnijapan@vnij.co.jp

Korea
Visual Numerics, Inc., Korea
Rm. 801, Hanshin Bldg.,
136-1, Mapo-dong, Mapo-gu,
Seoul 121-050
KOREA
Tel: +82-2-3273-2632
E-mail: info@vni.co.kr

Taiwan
Visual Numerics, Inc., Taiwan Branch
7/F, #510 Chung Hsiao E. Road, Sect. 5
Taipei, Taiwan 110
REPUBLIC OF CHINA

Visual Numerics®
Developer of IMSL® and WAVE

[www.vni.com]