

Numerical Recipes Subroutines and Functions

This page gives access to the individual Numerical Recipes (that is, subroutines and functions). Scroll down this page and click the left mouse button on any particular Recipe that you want to **view**, **print**, or **save to disk**. (Note: Be sure to initialize your path with the Set Path button before you save any files to disk.)

We do not recommend copying all the Recipes to disk in bulk. ([Here is why.](#)) Instead, we recommend copying just those Recipes that you need for a specific project, directly into the working directory for that project. However, if you really want to copy all the Recipes to a single directory, here is how:

1. Set your path using the Set Path button on the button bar.
2. Click here: [Copy all Subroutines and Functions to Disk](#)

LIST OF NUMERICAL RECIPES

ADDINT.FOR	interpolate and add, used by mglin [19.6]
AIRY.FOR	Airy functions [6.7]
AMEBSA.FOR	simulated annealing in continuous spaces [10.9]
AMOEB.A.FOR	minimize in N-dimensions by downhill simplex method [10.4]
AMOTRY.FOR	evaluate a trial point, used by amoeba [10.4]
AMOTSA.FOR	evaluate a trial point, used by amebsa [10.9]
ANNEAL.FOR	traveling salesman problem by simulated annealing [10.9]
ANORM2.FOR	utility used by mgfas [19.6]
ARCM.AK.FOR	construct an arithmetic code [20.5]
ARCODE.FOR	encode or decode a character using arithmetic coding [20.5]
ARCSUM.FOR	add integer to byte string, used by arcode [20.5]
ASOLVE.FOR	used by linbcg for preconditioner [2.7]
ATIME.S.FOR	used by linbcg for sparse multiplication [2.7]
AVEVAR.FOR	calculate mean and variance of a data set [14.2]
BADLUK.FOR	Friday the 13th when the moon is full [1.1]
BALANC.FOR	balance a nonsymmetric matrix [11.5]
BANBK.S.FOR	band diagonal systems, backsubstitution [2.4]
BANDEC.FOR	band diagonal systems, decomposition [2.4]
BANMUL.FOR	multiply vector by band diagonal matrix [2.4]
BCUCOF.FOR	construct two-dimensional bicubic [3.6]
BCUINT.FOR	two-dimensional bicubic interpolation [3.6]
BESCHB.FOR	Chebyshev expansion used by bessjy [6.7]
BESSI0.FOR	modified Bessel function I ₀ [6.6]
BESSI1.FOR	modified Bessel function I ₁ [6.6]
BESSI.FOR	modified Bessel function I of integer order [6.6]
BESSIK.FOR	modified Bessel functions of fractional order [6.7]
BESSJ0.FOR	Bessel function J ₀ [6.5]
BESSJ1.FOR	Bessel function J ₁ [6.5]
BESSJ.FOR	Bessel function J of general integer order [6.5]
BESSJY.FOR	Bessel functions of fractional order [6.7]
BESSK0.FOR	modified Bessel function K ₀ [6.6]
BESSK1.FOR	modified Bessel function K ₁ [6.6]
BESSK.FOR	modified Bessel function K of integer order [6.6]
BESSY0.FOR	Bessel function Y ₀ [6.5]
BESSY1.FOR	Bessel function Y ₁ [6.5]
BESSY.FOR	Bessel function Y of general integer order [6.5]
BETA.FOR	beta function [6.1]
BETACF.FOR	continued fraction used by betai [6.4]
BETAI.FOR	incomplete beta function [6.4]
BICO.FOR	binomial coefficients function [6.1]
BKSUB.FOR	backsubstitution, used by SOLVDE [17.3]
BNLDEV.FOR	binomial distributed random deviates [7.3]
BRENT.FOR	find minimum of a function by Brent's method [10.2]

[BROYDN.FOR](#) secant method for systems of equations [9.7]
[BSSTEP.FOR](#) integrate ODEs, Bulirsch-Stoer step [16.4]
[CALDAT.FOR](#) calendar date from Julian day number [1.1]
[CHDER.FOR](#) derivative of a function already Chebyshev fitted [5.9]
[CHEBEV.FOR](#) Chebyshev polynomial evaluation [5.8]
[CHEBFT.FOR](#) fit a Chebyshev polynomial to a function [5.8]
[CHEBPC.FOR](#) polynomial coefficients from a Chebyshev fit [5.10]
[CHINT.FOR](#) integrate a function already Chebyshev fitted [5.9]
[CHIXY.FOR](#) used by [fitexy](#) to calculate a χ^2 [15.3]
[CHOLDC.FOR](#) Cholesky decomposition [2.9]
[CHOLSL.FOR](#) Cholesky backsubstitution [2.9]
[CHSONE.FOR](#) chi-square test for difference between data and model [14.3]
[CHSTWO.FOR](#) chi-square test for difference between two data sets [14.3]
[CISI.FOR](#) cosine and sine integrals C_i and S_i [6.9]
[CNTAB1.FOR](#) contingency table analysis using chi-square [14.4]
[CNTAB2.FOR](#) contingency table analysis using entropy measure [14.4]
[CONVLV.FOR](#) convolution or deconvolution of data using FFT [13.1]
[COPY.FOR](#) utility used by [mglin](#), [mgfas](#) [19.6]
[CORREL.FOR](#) correlation or autocorrelation of data using FFT [13.2]
[COSFT1.FOR](#) fast cosine transform with endpoints [12.3]
[COSFT2.FOR](#) "staggered" fast cosine transform [12.3]
[COVSRT.FOR](#) rearrange covariance matrix, used by [lfit](#) [15.4]
[CRANK.FOR](#) replaces array elements by their rank [14.6]
[CYCLIC.FOR](#) solution of cyclic tridiagonal systems [2.7]
[DAUB4.FOR](#) Daubechies 4-coefficient wavelet filter [13.10]
[DAWSON.FOR](#) Dawson's integral [6.10]
[DBRENT.FOR](#) find minimum of a function using derivative information [10.3]
[DDPOLY.FOR](#) evaluate a polynomial and its derivatives [5.3]
[DECCHK.FOR](#) decimal check digit calculation or verification [20.3]
[DERIVS.FOR](#) sample derivatives routine for [stiff](#) [16.6]
[DF1DIM.FOR](#) alternative function used by [linmin](#) [10.6]
[DFPMIN.FOR](#) minimize in N-dimensions by variable metric method [10.7]
[DFRIDR.FOR](#) numerical derivative by Ridders' method [5.7]
[DFTCOR.FOR](#) compute endpoint corrections for Fourier integrals [13.9]
[DFTINT.FOR](#) high-accuracy Fourier integrals [13.9]
[DIFEQ.FOR](#) spheroidal matrix coefficients, used by SFROID [17.4]
[ECLASS.FOR](#) determine equivalence classes from list [8.6]
[ECLAZZ.FOR](#) determine equivalence classes from procedure [8.6]
[EI.FOR](#) exponential integral E_i [6.3]
[EIGSRT.FOR](#) eigenvectors, sorts into order by eigenvalue [11.1]
[ELLE.FOR](#) Legendre elliptic integral of the second kind [6.11]
[ELLF.FOR](#) Legendre elliptic integral of the first kind [6.11]
[ELLPI.FOR](#) Legendre elliptic integral of the third kind [6.11]
[ELMHES.FOR](#) reduce a general matrix to Hessenberg form [11.5]
[ERF.FOR](#) error function [6.2]
[ERFC.FOR](#) complementary error function [6.2]
[ERFCC.FOR](#) complementary error function, concise routine [6.2]
[EULSUM.FOR](#) sum a series by Euler--van Wijngaarden algorithm [5.1]
[EVLMEM.FOR](#) power spectral estimation from MEM coefficients [13.7]
[EXPDEV.FOR](#) exponential random deviates [7.2]
[EXPINT.FOR](#) exponential integral E_n [6.3]
[F1DIM.FOR](#) function used by [linmin](#) [10.5]
[FACTLN.FOR](#) logarithm of factorial function [6.1]
[FACTRL.FOR](#) factorial function [6.1]
[FASPER.FOR](#) power spectrum of unevenly sampled larger data sets [13.8]
[FDJAC.FOR](#) finite-difference Jacobian, used by [newt](#) [9.7]
[FGAUSS.FOR](#) fit a sum of Gaussians using [mrqmin](#) [15.5]

[FILL0.FOR](#)
[FIT.FOR](#)
[FITEXY.FOR](#)
[FIXRTS.FOR](#)
[FLEG.FOR](#)
[FLMOON.FOR](#)
[FMIN.FOR](#)
[FOUR1.FOR](#)
[FOUREW.FOR](#)
[FOURFS.FOR](#)
[FOURN.FOR](#)
[FPOLY.FOR](#)
[FRED2.FOR](#)
[FREDEX.FOR](#)
[FREDIN.FOR](#)
[FRENEL.FOR](#)
[FRPRMN.FOR](#)
[FTEST.FOR](#)
[GAMDEV.FOR](#)
[GAMMLN.FOR](#)
[GAMMP.FOR](#)
[GAMMQ.FOR](#)
[GASDEV.FOR](#)
[GAUCOF.FOR](#)
[GAUHER.FOR](#)
[GAUJAC.FOR](#)
[GAULAG.FOR](#)
[GAULEG.FOR](#)
[GAUSSJ.FOR](#)
[GCF.FOR](#)
[GOLDEN.FOR](#)
[GSER.FOR](#)
[HPSEL.FOR](#)
[HPSORT.FOR](#)
[HQR.FOR](#)
[HUFAPP.FOR](#)
[HUFDEC.FOR](#)
[HUFENC.FOR](#)
[HUFMAK.FOR](#)
[HUNT.FOR](#)
[HYPDRV.FOR](#)
[HYPGEO.FOR](#)
[HYPSER.FOR](#)
[ICRC1.FOR](#)
[ICRC.FOR](#)
[IGRAY.FOR](#)
[INDEXX.FOR](#)
[INTERP.FOR](#)
[IRBIT1.FOR](#)
[IRBIT2.FOR](#)
[JACOBI.FOR](#)
[JACOBN.FOR](#)
[JULDAY.FOR](#)
[KENDL1.FOR](#)
[KENDL2.FOR](#)
[KERMOM.FOR](#)
[KS2D1S.FOR](#)

utility used by [mglin](#) [19.6]
 least-squares fit data to a straight line [15.2]
 fit data to a straight line, errors in both x and y [15.3]
 reflect roots of a polynomial into unit circle [13.6]
 fit a Legendre polynomial using [lfit](#) or [svdfit](#) [15.4]
 calculate phases of the moon by date [1.0]
 norm of a vector function, used by [newt](#) [9.7]
 fast Fourier transform (FFT) in one dimension [12.2]
 rewind and permute files, used by [fourfs](#) [12.6]
 FFT for huge data sets on external media [12.6]
 fast Fourier transform in multidimensions [12.4]
 fit a polynomial using [lfit](#) or [svdfit](#) [15.4]
 solve linear Fredholm equations of the second kind [18.1]
 example of solving a singular Fredholm equation [18.3]
 interpolate solutions obtained with [fred2](#) [18.1]
 Fresnel integrals S(x) and C(x) [6.9]
 minimize in N-dimensions by conjugate gradient [10.6]
 F-test for difference of variances [14.2]
 gamma-law distribution random deviates [7.3]
 logarithm of gamma function [6.1]
 incomplete gamma function [6.2]
 complement of incomplete gamma function [6.2]
 normally distributed random deviates [7.2]
 quadrature weights from orthogonal polynomials [4.5]
 Gauss-Hermite weights and abscissas [4.5]
 Gauss-Jacobi weights and abscissas [4.5]
 Gauss-Laguerre weights and abscissas [4.5]
 Gauss-Legendre weights and abscissas [4.5]
 Gauss-Jordan matrix inversion and linear equation solution [2.1]
 continued fraction used by [gammp](#) and [gammq](#) [6.2]
 find minimum of a function by [golden](#) section search [10.1]
 series used by [gammp](#) and [gammq](#) [6.2]
 find M largest values, without altering an array [8.5]
 sort an array by heapsort method [8.3]
 eigenvalues of a Hessenberg matrix [11.6]
 append bits to a Huffman code, used by [hufmak](#) [20.4]
 use Huffman code to decode and decompress a character [20.4]
 use Huffman code to encode and compress a character [20.4]
 construct a Huffman code [20.4]
 search a table when calls are correlated [3.4]
 complex hypergeometric function, derivative of [6.12]
 complex hypergeometric function [6.12]
 complex hypergeometric function, series evaluation [6.12]
 cyclic redundancy checksum, used by [icrc](#) [20.3]
 cyclic redundancy checksum [20.3]
 Gray code and its inverse [20.2]
 construct an index for an array [8.4]
 bilinear prolongation, used by [mglin](#), [mgfas](#) [19.6]
 random bit sequence [7.4]
 random bit sequence [7.4]
 eigenvalues and eigenvectors of a symmetric matrix [11.1]
 sample Jacobian routine for [stiff](#) [16.6]
 Julian Day number from calendar date [1.1]
 correlation between two data sets, Kendall's tau [14.6]
 contingency table analysis using Kendall's tau [14.6]
 sample routine for moments of a singular kernel [18.3]
 K--S test in two dimensions, data vs. model [14.7]

[KS2D2S.FOR](#) K--S test in two dimensions, data vs. data [14.7]
[KSONE.FOR](#) Kolmogorov-Smirnov test of data against model [14.3]
[KSTWO.FOR](#) Kolmogorov-Smirnov test between two data sets [14.3]
[LAGUER.FOR](#) find a root of a polynomial by Laguerre's method [9.5]
[LFIT.FOR](#) general linear least-squares fit by normal equations [15.4]
[LINBCG.FOR](#) biconjugate gradient solution of sparse systems [2.7]
[LINMIN.FOR](#) minimum of a function along a ray in N-dimensions [10.5]
[LNSRCH.FOR](#) search along a line, used by [newt](#) [9.7]
[LOCATE.FOR](#) search an ordered table by bisection [3.4]
[LOP.FOR](#) applies nonlinear operator, used by [mgfas](#) [19.6]
[LUBKSB.FOR](#) linear equation solution, backsubstitution [2.3]
[LUDCMP.FOR](#) linear equation solution, LU decomposition [2.3]
[MACHAR.FOR](#) diagnose computer's floating arithmetic [20.1]
[MALOC.FOR](#) memory allocation utility used by [mglin](#), [mgfas](#) [19.6]
[MATADD.FOR](#) utility used by [mgfas](#) [19.6]
[MATSUB.FOR](#) utility used by [mgfas](#) [19.6]
[MEDFIT.FOR](#) fit data to a straight line robustly, least absolute deviation [15.7]
[MEMCOF.FOR](#) evaluate maximum entropy (MEM) coefficients [13.6]
[METROP.FOR](#) Metropolis algorithm, used by [anneal](#) [10.9]
[MGFAS.FOR](#) nonlinear elliptic PDE solved by multigrid method [19.6]
[MGLIN.FOR](#) linear elliptic PDE solved by multigrid method [19.6]
[MIDEXP.FOR](#) integrate a function that decreases exponentially [4.4]
[MIDINF.FOR](#) integrate a function on a semi-infinite interval [4.4]
[MIDPNT.FOR](#) extended midpoint rule [4.4]
[MIDSQL.FOR](#) integrate a function with lower square-root singularity [4.4]
[MIDSQU.FOR](#) integrate a function with upper square-root singularity [4.4]
[MISER.FOR](#) recursive multidimensional Monte Carlo integration [7.8]
[MMID.FOR](#) integrate ODEs by modified midpoint method [16.3]
[MNBRAK.FOR](#) bracket the minimum of a function [10.1]
[MNEWT.FOR](#) Newton's method for systems of equations [9.6]
[MOMENT.FOR](#) calculate moments of a data set [14.1]
[MP2DFR.FOR](#) multiple precision conversion to decimal base [20.6]
[MPDIV.FOR](#) multiple precision divide and remainder [20.6]
[MPINV.FOR](#) multiple precision reciprocal [20.6]
[MPMUL.FOR](#) multiple precision multiply, using FFT methods [20.6]
[MPOPS.FOR](#) multiple precision arithmetic, simpler operations [20.6]
[MPPI.FOR](#) multiple precision example, compute many digits of pi [20.6]
[MPROVE.FOR](#) linear equation solution, iterative improvement [2.5]
[MPSQRT.FOR](#) multiple precision square root [20.6]
[MRQCOF.FOR](#) used by [mrqmin](#) to evaluate coefficients [15.5]
[MRQMIN.FOR](#) nonlinear least-squares fit, Marquardt's method [15.5]
[NEWT.FOR](#) globally convergent multi-dimensional Newton's method [9.7]
[ODEINT.FOR](#) integrate ODEs with accuracy monitoring [16.2]
[ORTHOG.FOR](#) construct nonclassical orthogonal polynomials [4.5]
[PADE.FOR](#) Pade approximant from power series coefficients [5.12]
[PCCHEB.FOR](#) inverse of chebpc; use to economize power series [5.11]
[PCSHFT.FOR](#) polynomial coefficients of a shifted polynomial [5.10]
[PEARSN.FOR](#) Pearson's correlation between two data sets [14.5]
[PERIOD.FOR](#) power spectrum of unevenly sampled data [13.8]
[PIKSR2.FOR](#) sort two arrays by straight insertion [8.1]
[PIKSRT.FOR](#) sort an array by straight insertion [8.1]
[PINVS.FOR](#) diagonalize a sub-block, used by SOLVDE [17.3]
[PLGNDR.FOR](#) Legendre polynomials, associated (spherical harmonics) [6.8]
[POIDEV.FOR](#) Poisson distributed random deviates [7.3]
[POLCOE.FOR](#) polynomial coefficients from table of values [3.5]
[POLCOF.FOR](#) polynomial coefficients from table of values [3.5]
[POLDIV.FOR](#) divide one polynomial by another [5.3]

[POLIN2.FOR](#) two-dimensional polynomial interpolation [3.6]
[POLINT.FOR](#) polynomial interpolation [3.1]
[POWELL.FOR](#) minimize in N-dimensions by Powell's method [10.5]
[PREDIC.FOR](#) linear prediction using MEM coefficients [13.6]
[PROBKS.FOR](#) Kolmogorov-Smirnov probability function [14.3]
[PSDES.FOR](#) "pseudo-DES" hashing of 64 bits [7.5]
[PWT.FOR](#) partial wavelet transform [13.10]
[PWTSET.FOR](#) initialize coefficients for [pwt](#) [13.10]
[PYTHAG.FOR](#) calculate $(a^2+b^2)^{1/2}$ without overflow [2.6]
[PZEXTR.FOR](#) polynomial extrapolation, used by [bsstep](#) [16.4]
[QGAUS.FOR](#) integrate a function by Gaussian quadratures [4.5]
[QRDCMP.FOR](#) QR decomposition [2.10]
[QROMB.FOR](#) integrate using Romberg adaptive method [4.3]
[QROMO.FOR](#) integrate using open Romberg adaptive method [4.4]
[QROOT.FOR](#) complex or double root of a polynomial, Bairstow [9.5]
[QRSOLV.FOR](#) QR backsubstitution [2.10]
[QRUPDT.FOR](#) update a QR decomposition [2.10]
[QSIMP.FOR](#) integrate using Simpson's rule [4.2]
[QTRAP.FOR](#) integrate using trapezoidal rule [4.2]
[QUAD3D.FOR](#) integrate a function over a three-dimensional space [4.6]
[QUADCT.FOR](#) count points by quadrants, used by [ks2d1s](#) [14.7]
[QUADMX.FOR](#) sample routine for a quadrature matrix [18.3]
[QUADVL.FOR](#) quadrant probabilities, used by [ks2d1s](#) [14.7]
[RAN0.FOR](#) random deviate by Park and Miller minimal standard [7.1]
[RAN1.FOR](#) random deviate, minimal standard plus shuffle [7.1]
[RAN2.FOR](#) random deviate by L'Ecuyer long period plus shuffle [7.1]
[RAN3.FOR](#) random deviate by Knuth subtractive method [7.1]
[RAN4.FOR](#) random deviates from DES-like hashing [7.5]
[RANK.FOR](#) construct a rank table for an array [8.4]
[RANPT.FOR](#) get random point, used by [miser](#) [7.8]
[RATINT.FOR](#) rational function interpolation [3.2]
[RATLSQ.FOR](#) rational fit by least-squares method [5.13]
[RATVAL.FOR](#) evaluate a rational function [5.3]
[RC.FOR](#) Carlson's degenerate elliptic integral [6.11]
[RD.FOR](#) Carlson's elliptic integral of the second kind [6.11]
[REALFT.FOR](#) fast Fourier transform of a single real function [12.3]
[REBIN.FOR](#) sample rebinning used by [vegas](#) [7.8]
[RED.FOR](#) reduce columns of a matrix, used by SOLVDE [17.3]
[RELAX2.FOR](#) Gauss-Seidel relaxation, used by [mgfas](#) [19.6]
[RELAX.FOR](#) Gauss-Seidel relaxation, used by [mglin](#) [19.6]
[RESID.FOR](#) calculate residual, used by [mglin](#) [19.6]
[REVCST.FOR](#) cost of a reversal, used by [anneal](#) [10.9]
[REVERS.FOR](#) do a reversal, used by [anneal](#) [10.9]
[RF.FOR](#) Carlson's elliptic integral of the first kind [6.11]
[RJ.FOR](#) Carlson's elliptic integral of the third kind [6.11]
[RK4.FOR](#) integrate one step of ODEs, fourth-order Runge-Kutta [16.1]
[RKCK.FOR](#) Cash-Karp-Runge-Kutta step used by [rkqs](#) [16.2]
[RKDUMB.FOR](#) integrate ODEs by fourth-order Runge-Kutta [16.1]
[RKQS.FOR](#) integrate one step of ODEs with accuracy monitoring [16.2]
[RLFT3.FOR](#) FFT of real data in two or three dimensions [12.5]
[ROTATE.FOR](#) Jacobi rotation used by [grupdt](#) [2.10]
[RSOLV.FOR](#) right triangular backsubstitution [2.10]
[RSTRCT.FOR](#) half-weighting restriction, used by [mglin](#), [mgfas](#) [19.6]
[RTBIS.FOR](#) find root of a function by bisection [9.1]
[RTFLSP.FOR](#) find root of a function by false-position [9.2]
[RTNEWT.FOR](#) find root of a function by Newton-Raphson [9.4]
[RTSAFE.FOR](#) find root of a function by Newton-Raphson and bisection [9.4]

[RTSEC.FOR](#) find root of a function by secant method [9.2]
[RZEXTR.FOR](#) rational function extrapolation, used by [bsstep](#) [16.4]
[SAVGOL.FOR](#) Savitzky-Golay smoothing coefficients [14.8]
[SCRSHO.FOR](#) graph a function to search for roots [9.0]
[SELECT.FOR](#) find the Nth largest in an array [8.5]
[SELIP.FOR](#) find the Nth largest, without altering an array [8.5]
[SFROID.FOR](#) spheroidal functions by method of SOLVDE [17.4]
[SHELL.FOR](#) sort an array by Shell's method [8.1]
[SHOOT.FOR](#) solve two point boundary value problem by shooting [17.1]
[SHOOTF.FOR](#) ditto, by shooting to a fitting point [17.2]
[SIMP1.FOR](#) linear programming, used by [simplx](#) [10.8]
[SIMP2.FOR](#) linear programming, used by [simplx](#) [10.8]
[SIMP3.FOR](#) linear programming, used by [simplx](#) [10.8]
[SIMPLX.FOR](#) linear programming maximization of a linear function [10.8]
[SIMPR.FOR](#) integrate stiff ODEs by semi-implicit midpoint rule [16.6]
[SINFT.FOR](#) fast sine transform [12.3]
[SLVSM2.FOR](#) solve on coarsest grid, used by [mgfas](#) [19.6]
[SLVSM1.FOR](#) solve on coarsest grid, used by [mglin](#) [19.6]
[SNCNDN.FOR](#) Jacobian elliptic functions [6.11]
[SNRM.FOR](#) used by [linbcg](#) for vector norm [2.7]
[SOBSEQ.FOR](#) Sobol's quasi-random sequence [7.7]
[SOLVDE.FOR](#) two point boundary value problem, solve by relaxation [17.3]
[SOR.FOR](#) elliptic PDE solved by successive overrelaxation method [19.5]
[SORT2.FOR](#) sort two arrays by quicksort method [8.2]
[SORT3.FOR](#) sort, use an index to sort 3 or more arrays [8.4]
[SORT.FOR](#) sort an array by quicksort method [8.2]
[SPCTRM.FOR](#) power spectrum estimation using FFT [13.4]
[SPEAR.FOR](#) Spearman's rank correlation between two data sets [14.6]
[SPHBES.FOR](#) spherical Bessel functions j_n and y_n [6.7]
[SPHEPT.FOR](#) spheroidal functions by method of [shootf](#) [17.4]
[SPHOOT.FOR](#) spheroidal functions by method of [shoot](#) [17.4]
[SPLIE2.FOR](#) construct two-dimensional spline [3.6]
[SPLIN2.FOR](#) two-dimensional spline interpolation [3.6]
[SPLINE.FOR](#) construct a cubic spline [3.3]
[SPLINT.FOR](#) cubic spline interpolation [3.3]
[SPREAD.FOR](#) extirpolate value into array, used by [fasper](#) [13.8]
[SPRSAX.FOR](#) product of sparse matrix and vector [2.7]
[SPRSIN.FOR](#) convert matrix to sparse format [2.7]
[SPRSPM.FOR](#) pattern multiply two sparse matrices [2.7]
[SPRSTM.FOR](#) threshold multiply two sparse matrices [2.7]
[SPRSTP.FOR](#) transpose of sparse matrix [2.7]
[SPRSTX.FOR](#) product of transpose sparse matrix and vector [2.7]
[STIFBS.FOR](#) integrate stiff ODEs, Bulirsch-Stoer step [16.6]
[STIFF.FOR](#) integrate stiff ODEs by fourth-order Rosenbrock [16.6]
[STOERM.FOR](#) integrate conservative second-order ODEs [16.5]
[SVBKSB.FOR](#) singular value backsubstitution [2.6]
[SVDCMP.FOR](#) singular value decomposition of a matrix [2.6]
[SVDFIT.FOR](#) linear least-squares fit by singular value decomposition [15.4]
[SVDVAR.FOR](#) variances from singular value decomposition [15.4]
[TOEPLZ.FOR](#) solve Toeplitz systems [2.8]
[TPTEST.FOR](#) Student's t-test for means, case of paired data [14.2]
[TQLI.FOR](#) eigensolution of a symmetric tridiagonal matrix [11.3]
[TRAPZD.FOR](#) trapezoidal rule [4.2]
[TRED2.FOR](#) Householder reduction of a real, symmetric matrix [11.2]
[TRIDAG.FOR](#) solution of tridiagonal systems [2.4]
[TRNCST.FOR](#) cost of a transposition, used by [anneal](#) [10.9]
[TRNSPT.FOR](#) do a transposition, used by [anneal](#) [10.9]

<u>TTEST.FOR</u>	Student's t-test for difference of means [14.2]
<u>TUTEST.FOR</u>	Student's t-test for means, case of unequal variances [14.2]
<u>TWOFFT.FOR</u>	fast Fourier transform of two real functions [12.3]
<u>VANDER.FOR</u>	solve Vandermonde systems [2.8]
<u>VEGAS.FOR</u>	adaptive multidimensional Monte Carlo integration [7.8]
<u>VOLTRA.FOR</u>	linear Volterra equations of the second kind [18.2]
<u>WT1.FOR</u>	one-dimensional discrete wavelet transform [13.10]
<u>WTN.FOR</u>	multidimensional discrete wavelet transform [13.10]
<u>WWGHTS.FOR</u>	quadrature weights for an arbitrarily singular kernel [18.3]
<u>ZBRAC.FOR</u>	outward search for brackets on roots [9.1]
<u>ZBRAK.FOR</u>	inward search for brackets on roots [9.1]
<u>ZBRENT.FOR</u>	find root of a function by Brent's method [9.3]
<u>ZRHQR.FOR</u>	roots of a polynomial by eigenvalue methods [9.5]
<u>ZRIDDR.FOR</u>	find root of a function by Ridder's method [9.2]
<u>ZROOTS.FOR</u>	roots of a polynomial by Laguerre's method with deflation [9.5]