

Computer Programs by Chapter and Section

1.0	flmoon	calculate phases of the moon by date
1.1	julday	Julian Day number from calendar date
1.1	badluk	Friday the 13th when the moon is full
1.1	caldat	calendar date from Julian day number
2.1	gaussj	Gauss-Jordan matrix inversion and linear equation solution
2.3	ludcmp	linear equation solution, <i>LU</i> decomposition
2.3	lubksb	linear equation solution, backsubstitution
2.4	tridag	solution of tridiagonal systems
2.4	banmul	multiply vector by band diagonal matrix
2.4	bandec	band diagonal systems, decomposition
2.4	banbks	band diagonal systems, backsubstitution
2.5	mprove	linear equation solution, iterative improvement
2.6	svbksb	singular value backsubstitution
2.6	svdcmp	singular value decomposition of a matrix
2.6	pythag	calculate $(a^2 + b^2)^{1/2}$ without overflow
2.7	cyclic	solution of cyclic tridiagonal systems
2.7	sprsin	convert matrix to sparse format
2.7	spr sax	product of sparse matrix and vector
2.7	sprstx	product of transpose sparse matrix and vector
2.7	sprstp	transpose of sparse matrix
2.7	sprspm	pattern multiply two sparse matrices
2.7	sprstm	threshold multiply two sparse matrices
2.7	linbcg	biconjugate gradient solution of sparse systems
2.7	snrm	used by linbcg for vector norm
2.7	atimes	used by linbcg for sparse multiplication
2.7	asolve	used by linbcg for preconditioner
2.8	vander	solve Vandermonde systems
2.8	toeplz	solve Toeplitz systems
2.9	choldc	Cholesky decomposition
2.9	cholsl	Cholesky backsubstitution
2.10	qrdcmp	QR decomposition
2.10	qrsolv	QR backsubstitution
2.10	rsolv	right triangular backsubstitution
2.10	qrupdt	update a QR decomposition
2.10	rotate	Jacobi rotation used by qrupdt
3.1	polint	polynomial interpolation
3.2	ratint	rational function interpolation
3.3	spline	construct a cubic spline
3.3	splint	cubic spline interpolation
3.4	locate	search an ordered table by bisection

World Wide Web sample page from NUMERICAL RECIPES IN C: THE ART OF SCIENTIFIC COMPUTING (ISBN 0-521-43108-5)
 Copyright (C) 1988-1992 by Cambridge University Press. Programs Copyright (C) 1988-1992 by Numerical Recipes Software.
 Permission is granted for internet users to make one paper copy for their own personal use. Further reproduction, or any copying of machine-readable files (including this one), to any server computer, is strictly prohibited. To order Numerical Recipes books, diskettes, or CDROMs visit website <http://www.nr.com> or call 1-800-872-7423 (North America only), or send email to trade@cup.cam.ac.uk (outside North America).

3.4	hunt	search a table when calls are correlated
3.5	polcoe	polynomial coefficients from table of values
3.5	polcof	polynomial coefficients from table of values
3.6	polin2	two-dimensional polynomial interpolation
3.6	bcucof	construct two-dimensional bicubic
3.6	bcuint	two-dimensional bicubic interpolation
3.6	splie2	construct two-dimensional spline
3.6	splin2	two-dimensional spline interpolation
4.2	trapzd	trapezoidal rule
4.2	qtrap	integrate using trapezoidal rule
4.2	qsimp	integrate using Simpson's rule
4.3	qromb	integrate using Romberg adaptive method
4.4	midpnt	extended midpoint rule
4.4	qromo	integrate using open Romberg adaptive method
4.4	midinf	integrate a function on a semi-infinite interval
4.4	midsq1	integrate a function with lower square-root singularity
4.4	midsqu	integrate a function with upper square-root singularity
4.4	midexp	integrate a function that decreases exponentially
4.5	qgaus	integrate a function by Gaussian quadratures
4.5	gauleg	Gauss-Legendre weights and abscissas
4.5	gaulag	Gauss-Laguerre weights and abscissas
4.5	gauher	Gauss-Hermite weights and abscissas
4.5	gaujac	Gauss-Jacobi weights and abscissas
4.5	gaucocf	quadrature weights from orthogonal polynomials
4.5	orthog	construct nonclassical orthogonal polynomials
4.6	quad3d	integrate a function over a three-dimensional space
5.1	eulsum	sum a series by Euler-van Wijngaarden algorithm
5.3	ddpoly	evaluate a polynomial and its derivatives
5.3	poldiv	divide one polynomial by another
5.3	ratval	evaluate a rational function
5.7	dfridr	numerical derivative by Ridders' method
5.8	chebft	fit a Chebyshev polynomial to a function
5.8	chebev	Chebyshev polynomial evaluation
5.9	chder	derivative of a function already Chebyshev fitted
5.9	chint	integrate a function already Chebyshev fitted
5.10	chebpc	polynomial coefficients from a Chebyshev fit
5.10	pcshft	polynomial coefficients of a shifted polynomial
5.11	pccheb	inverse of chebpc; use to economize power series
5.12	pade	Padé approximant from power series coefficients
5.13	ratlsq	rational fit by least-squares method
6.1	gammln	logarithm of gamma function
6.1	factrl	factorial function
6.1	bico	binomial coefficients function
6.1	factln	logarithm of factorial function

World Wide Web sample page from NUMERICAL RECIPES IN C: THE ART OF SCIENTIFIC COMPUTING (ISBN 0-521-43108-5)
 Copyright (C) 1988-1992 by Cambridge University Press. Programs Copyright (C) 1988-1992 by Numerical Recipes Software.
 Permission is granted for internet users to make one paper copy for their own personal use. Further reproduction, or any copying of machine-readable files (including this one), to any server computer, is strictly prohibited. To order Numerical Recipes books, diskettes, or CDROMs visit website <http://www.nr.com> or call 1-800-872-7423 (North America only), or send email to trade@cup.cam.ac.uk (outside North America).

6.1	beta	beta function
6.2	gammp	incomplete gamma function
6.2	gammq	complement of incomplete gamma function
6.2	gser	series used by gammp and gammq
6.2	gcf	continued fraction used by gammp and gammq
6.2	erff	error function
6.2	erffc	complementary error function
6.2	erfcc	complementary error function, concise routine
6.3	expint	exponential integral E_n
6.3	ei	exponential integral Ei
6.4	betai	incomplete beta function
6.4	betacf	continued fraction used by betai
6.5	bessj0	Bessel function J_0
6.5	bessy0	Bessel function Y_0
6.5	bessj1	Bessel function J_1
6.5	bessy1	Bessel function Y_1
6.5	bessy	Bessel function Y of general integer order
6.5	bessj	Bessel function J of general integer order
6.6	bessi0	modified Bessel function I_0
6.6	bessk0	modified Bessel function K_0
6.6	bessi1	modified Bessel function I_1
6.6	bessk1	modified Bessel function K_1
6.6	bessk	modified Bessel function K of integer order
6.6	bessi	modified Bessel function I of integer order
6.7	bessjy	Bessel functions of fractional order
6.7	beschb	Chebyshev expansion used by bessjy
6.7	bessik	modified Bessel functions of fractional order
6.7	airy	Airy functions
6.7	sphbes	spherical Bessel functions j_n and y_n
6.8	plgnldr	Legendre polynomials, associated (spherical harmonics)
6.9	frenel	Fresnel integrals $S(x)$ and $C(x)$
6.9	cisi	cosine and sine integrals Si and Si
6.10	dawson	Dawson's integral
6.11	rf	Carlson's elliptic integral of the first kind
6.11	rd	Carlson's elliptic integral of the second kind
6.11	rj	Carlson's elliptic integral of the third kind
6.11	rc	Carlson's degenerate elliptic integral
6.11	ellf	Legendre elliptic integral of the first kind
6.11	elle	Legendre elliptic integral of the second kind
6.11	ellpi	Legendre elliptic integral of the third kind
6.11	sncndn	Jacobian elliptic functions
6.12	hypgeo	complex hypergeometric function
6.12	hypser	complex hypergeometric function, series evaluation
6.12	hypdrv	complex hypergeometric function, derivative of
7.1	ran0	random deviate by Park and Miller minimal standard
7.1	ran1	random deviate, minimal standard plus shuffle

7.1	ran2	random deviate by L'Ecuyer long period plus shuffle
7.1	ran3	random deviate by Knuth subtractive method
7.2	expdev	exponential random deviates
7.2	gasdev	normally distributed random deviates
7.3	gamdev	gamma-law distribution random deviates
7.3	poidev	Poisson distributed random deviates
7.3	bnldev	binomial distributed random deviates
7.4	irbit1	random bit sequence
7.4	irbit2	random bit sequence
7.5	psdes	"pseudo-DES" hashing of 64 bits
7.5	ran4	random deviates from DES-like hashing
7.7	sobseq	Sobol's quasi-random sequence
7.8	vegas	adaptive multidimensional Monte Carlo integration
7.8	rebin	sample rebinning used by vegas
7.8	miser	recursive multidimensional Monte Carlo integration
7.8	ranpt	get random point, used by miser
8.1	piksr1	sort an array by straight insertion
8.1	piksr2	sort two arrays by straight insertion
8.1	shell	sort an array by Shell's method
8.2	sort	sort an array by quicksort method
8.2	sort2	sort two arrays by quicksort method
8.3	hpsort	sort an array by heapsort method
8.4	indexx	construct an index for an array
8.4	sort3	sort, use an index to sort 3 or more arrays
8.4	rank	construct a rank table for an array
8.5	select	find the N th largest in an array
8.5	selip	find the N th largest, without altering an array
8.5	hpse1	find M largest values, without altering an array
8.6	eclass	determine equivalence classes from list
8.6	eclazz	determine equivalence classes from procedure
9.0	scrsho	graph a function to search for roots
9.1	zbrac	outward search for brackets on roots
9.1	zbrak	inward search for brackets on roots
9.1	rtbis	find root of a function by bisection
9.2	rtflsp	find root of a function by false-position
9.2	rtsec	find root of a function by secant method
9.2	zriddr	find root of a function by Ridders' method
9.3	zbrent	find root of a function by Brent's method
9.4	rtnewt	find root of a function by Newton-Raphson
9.4	rtsafe	find root of a function by Newton-Raphson and bisection
9.5	laguer	find a root of a polynomial by Laguerre's method
9.5	zroots	roots of a polynomial by Laguerre's method with deflation
9.5	zrhqr	roots of a polynomial by eigenvalue methods
9.5	qroot	complex or double root of a polynomial, Bairstow

9.6	<code>mnewt</code>	Newton's method for systems of equations
9.7	<code>lnsrch</code>	search along a line, used by <code>newt</code>
9.7	<code>newt</code>	globally convergent multi-dimensional Newton's method
9.7	<code>fdjac</code>	finite-difference Jacobian, used by <code>newt</code>
9.7	<code>fmin</code>	norm of a vector function, used by <code>newt</code>
9.7	<code>broydn</code>	secant method for systems of equations
10.1	<code>mnbrak</code>	bracket the minimum of a function
10.1	<code>golden</code>	find minimum of a function by golden section search
10.2	<code>brent</code>	find minimum of a function by Brent's method
10.3	<code>dbrent</code>	find minimum of a function using derivative information
10.4	<code>amoeba</code>	minimize in N -dimensions by downhill simplex method
10.4	<code>amotry</code>	evaluate a trial point, used by <code>amoeba</code>
10.5	<code>powell</code>	minimize in N -dimensions by Powell's method
10.5	<code>linmin</code>	minimum of a function along a ray in N -dimensions
10.5	<code>f1dim</code>	function used by <code>linmin</code>
10.6	<code>frprmn</code>	minimize in N -dimensions by conjugate gradient
10.6	<code>dlinmin</code>	minimum of a function along a ray using derivatives
10.6	<code>df1dim</code>	function used by <code>dlinmin</code>
10.7	<code>dfpmin</code>	minimize in N -dimensions by variable metric method
10.8	<code>simplx</code>	linear programming maximization of a linear function
10.8	<code>simp1</code>	linear programming, used by <code>simplx</code>
10.8	<code>simp2</code>	linear programming, used by <code>simplx</code>
10.8	<code>simp3</code>	linear programming, used by <code>simplx</code>
10.9	<code>anneal</code>	traveling salesman problem by simulated annealing
10.9	<code>revcst</code>	cost of a reversal, used by <code>anneal</code>
10.9	<code>reverse</code>	do a reversal, used by <code>anneal</code>
10.9	<code>trncst</code>	cost of a transposition, used by <code>anneal</code>
10.9	<code>trnspt</code>	do a transposition, used by <code>anneal</code>
10.9	<code>metrop</code>	Metropolis algorithm, used by <code>anneal</code>
10.9	<code>amebsa</code>	simulated annealing in continuous spaces
10.9	<code>amotsa</code>	evaluate a trial point, used by <code>amebsa</code>
11.1	<code>jacobi</code>	eigenvalues and eigenvectors of a symmetric matrix
11.1	<code>eigsrt</code>	eigenvectors, sorts into order by eigenvalue
11.2	<code>tred2</code>	Householder reduction of a real, symmetric matrix
11.3	<code>tqli</code>	eigensolution of a symmetric tridiagonal matrix
11.5	<code>balanc</code>	balance a nonsymmetric matrix
11.5	<code>elmhes</code>	reduce a general matrix to Hessenberg form
11.6	<code>hqr</code>	eigenvalues of a Hessenberg matrix
12.2	<code>four1</code>	fast Fourier transform (FFT) in one dimension
12.3	<code>twofft</code>	fast Fourier transform of two real functions
12.3	<code>realft</code>	fast Fourier transform of a single real function
12.3	<code>sinft</code>	fast sine transform
12.3	<code>cosft1</code>	fast cosine transform with endpoints
12.3	<code>cosft2</code>	"staggered" fast cosine transform

12.4	fourn	fast Fourier transform in multidimensions
12.5	rlft3	FFT of real data in two or three dimensions
12.6	fourfs	FFT for huge data sets on external media
12.6	fourew	rewind and permute files, used by fourfs
13.1	convlv	convolution or deconvolution of data using FFT
13.2	correl	correlation or autocorrelation of data using FFT
13.4	spctrm	power spectrum estimation using FFT
13.6	memcof	evaluate maximum entropy (MEM) coefficients
13.6	fixrts	reflect roots of a polynomial into unit circle
13.6	predic	linear prediction using MEM coefficients
13.7	evlmem	power spectral estimation from MEM coefficients
13.8	period	power spectrum of unevenly sampled data
13.8	fasper	power spectrum of unevenly sampled larger data sets
13.8	spread	extrapolate value into array, used by fasper
13.9	dftcor	compute endpoint corrections for Fourier integrals
13.9	dftint	high-accuracy Fourier integrals
13.10	wt1	one-dimensional discrete wavelet transform
13.10	daub4	Daubechies 4-coefficient wavelet filter
13.10	pwtset	initialize coefficients for pwt
13.10	pwt	partial wavelet transform
13.10	wtn	multidimensional discrete wavelet transform
14.1	moment	calculate moments of a data set
14.2	ttest	Student's <i>t</i> -test for difference of means
14.2	avevar	calculate mean and variance of a data set
14.2	tutest	Student's <i>t</i> -test for means, case of unequal variances
14.2	tptest	Student's <i>t</i> -test for means, case of paired data
14.2	fctest	<i>F</i> -test for difference of variances
14.3	chsone	chi-square test for difference between data and model
14.3	chstwo	chi-square test for difference between two data sets
14.3	ksone	Kolmogorov-Smirnov test of data against model
14.3	kstwo	Kolmogorov-Smirnov test between two data sets
14.3	probks	Kolmogorov-Smirnov probability function
14.4	cntab1	contingency table analysis using chi-square
14.4	cntab2	contingency table analysis using entropy measure
14.5	pearsn	Pearson's correlation between two data sets
14.6	spear	Spearman's rank correlation between two data sets
14.6	crank	replaces array elements by their rank
14.6	kendl1	correlation between two data sets, Kendall's tau
14.6	kendl2	contingency table analysis using Kendall's tau
14.7	ks2d1s	K-S test in two dimensions, data vs. model
14.7	quadct	count points by quadrants, used by ks2d1s
14.7	quadv1	quadrant probabilities, used by ks2d1s
14.7	ks2d2s	K-S test in two dimensions, data vs. data
14.8	savgol	Savitzky-Golay smoothing coefficients

15.2	<code>fit</code>	least-squares fit data to a straight line
15.3	<code>fitexy</code>	fit data to a straight line, errors in both x and y
15.3	<code>chixy</code>	used by <code>fitexy</code> to calculate a χ^2
15.4	<code>lfit</code>	general linear least-squares fit by normal equations
15.4	<code>covsrt</code>	rearrange covariance matrix, used by <code>lfit</code>
15.4	<code>svdfit</code>	linear least-squares fit by singular value decomposition
15.4	<code>svdvar</code>	variances from singular value decomposition
15.4	<code>fpoly</code>	fit a polynomial using <code>lfit</code> or <code>svdfit</code>
15.4	<code>fleg</code>	fit a Legendre polynomial using <code>lfit</code> or <code>svdfit</code>
15.5	<code>mrqmin</code>	nonlinear least-squares fit, Marquardt's method
15.5	<code>mrqcof</code>	used by <code>mrqmin</code> to evaluate coefficients
15.5	<code>fgauss</code>	fit a sum of Gaussians using <code>mrqmin</code>
15.7	<code>medfit</code>	fit data to a straight line robustly, least absolute deviation
15.7	<code>rofunc</code>	fit data robustly, used by <code>medfit</code>
16.1	<code>rk4</code>	integrate one step of ODEs, fourth-order Runge-Kutta
16.1	<code>rkdumb</code>	integrate ODEs by fourth-order Runge-Kutta
16.2	<code>rkqs</code>	integrate one step of ODEs with accuracy monitoring
16.2	<code>rkck</code>	Cash-Karp-Runge-Kutta step used by <code>rkqs</code>
16.2	<code>odeint</code>	integrate ODEs with accuracy monitoring
16.3	<code>mmid</code>	integrate ODEs by modified midpoint method
16.4	<code>bsstep</code>	integrate ODEs, Bulirsch-Stoer step
16.4	<code>pzextr</code>	polynomial extrapolation, used by <code>bsstep</code>
16.4	<code>rzextr</code>	rational function extrapolation, used by <code>bsstep</code>
16.5	<code>stoerm</code>	integrate conservative second-order ODEs
16.6	<code>stiff</code>	integrate stiff ODEs by fourth-order Rosenbrock
16.6	<code>jacobn</code>	sample Jacobian routine for <code>stiff</code>
16.6	<code>derivs</code>	sample derivatives routine for <code>stiff</code>
16.6	<code>simpr</code>	integrate stiff ODEs by semi-implicit midpoint rule
16.6	<code>stifbs</code>	integrate stiff ODEs, Bulirsch-Stoer step
17.1	<code>shoot</code>	solve two point boundary value problem by shooting
17.2	<code>shootf</code>	ditto, by shooting to a fitting point
17.3	<code>solvde</code>	two point boundary value problem, solve by relaxation
17.3	<code>bksub</code>	backsubstitution, used by <code>solvde</code>
17.3	<code>pinvs</code>	diagonalize a sub-block, used by <code>solvde</code>
17.3	<code>red</code>	reduce columns of a matrix, used by <code>solvde</code>
17.4	<code>sfroid</code>	spheroidal functions by method of <code>solvde</code>
17.4	<code>difeq</code>	spheroidal matrix coefficients, used by <code>sfroid</code>
17.4	<code>sphoot</code>	spheroidal functions by method of <code>shoot</code>
17.4	<code>sphfpt</code>	spheroidal functions by method of <code>shootf</code>
18.1	<code>fred2</code>	solve linear Fredholm equations of the second kind
18.1	<code>fredin</code>	interpolate solutions obtained with <code>fred2</code>
18.2	<code>voltra</code>	linear Volterra equations of the second kind
18.3	<code>wghts</code>	quadrature weights for an arbitrarily singular kernel
18.3	<code>kermom</code>	sample routine for moments of a singular kernel

18.3	quadmx	sample routine for a quadrature matrix
18.3	fredex	example of solving a singular Fredholm equation
19.5	sor	elliptic PDE solved by successive overrelaxation method
19.6	mglin	linear elliptic PDE solved by multigrid method
19.6	rstrct	half-weighting restriction, used by mglin, mgfas
19.6	interp	bilinear prolongation, used by mglin, mgfas
19.6	addint	interpolate and add, used by mglin
19.6	slvsml	solve on coarsest grid, used by mglin
19.6	relax	Gauss-Seidel relaxation, used by mglin
19.6	resid	calculate residual, used by mglin
19.6	copy	utility used by mglin, mgfas
19.6	fill0	utility used by mglin
19.6	mgfas	nonlinear elliptic PDE solved by multigrid method
19.6	relax2	Gauss-Seidel relaxation, used by mgfas
19.6	slvsm2	solve on coarsest grid, used by mgfas
19.6	lop	applies nonlinear operator, used by mgfas
19.6	matadd	utility used by mgfas
19.6	matsub	utility used by mgfas
19.6	anorm2	utility used by mgfas
20.1	machar	diagnose computer's floating arithmetic
20.2	igray	Gray code and its inverse
20.3	icrc1	cyclic redundancy checksum, used by icrc
20.3	icrc	cyclic redundancy checksum
20.3	decchk	decimal check digit calculation or verification
20.4	hufmak	construct a Huffman code
20.4	hufapp	append bits to a Huffman code, used by hufmak
20.4	hufenc	use Huffman code to encode and compress a character
20.4	hufdec	use Huffman code to decode and decompress a character
20.5	arcmak	construct an arithmetic code
20.5	arcode	encode or decode a character using arithmetic coding
20.5	arcsum	add integer to byte string, used by arcode
20.6	mpops	multiple precision arithmetic, simpler operations
20.6	mpmul	multiple precision multiply, using FFT methods
20.6	mpinv	multiple precision reciprocal
20.6	mpdiv	multiple precision divide and remainder
20.6	mpsqrtr	multiple precision square root
20.6	mp2dfrr	multiple precision conversion to decimal base
20.6	mppi	multiple precision example, compute many digits of π