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, LU 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	`sprsax`	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

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	midsql	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	gaucof	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

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	`plgndr`	Legendre polynomials, associated (spherical harmonics)
6.9	`frenel`	Fresnel integrals $S(x)$ and $C(x)$
6.9	`cisi`	cosine and sine integrals Ci 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	`piksrt`	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 Nth largest in an array
8.5	`selip`	find the Nth largest, without altering an array
8.5	`hpsel`	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	`mnewt`	Newton's method for systems of equations
9.7	`lnsrch`	search along a line, used by `newt`
9.7	`newt`	globally convergent multi-dimensional Newton's method
9.7	`fdjac`	finite-difference Jacobian, used by `newt`
9.7	`fmin`	norm of a vector function, used by `newt`
9.7	`broydn`	secant method for systems of equations
10.1	`mnbrak`	bracket the minimum of a function
10.1	`golden`	find minimum of a function by golden section search
10.2	`brent`	find minimum of a function by Brent's method
10.3	`dbrent`	find minimum of a function using derivative information
10.4	`amoeba`	minimize in N-dimensions by downhill simplex method
10.4	`amotry`	evaluate a trial point, used by `amoeba`
10.5	`powell`	minimize in N-dimensions by Powell's method
10.5	`linmin`	minimum of a function along a ray in N-dimensions
10.5	`f1dim`	function used by `linmin`
10.6	`frprmn`	minimize in N-dimensions by conjugate gradient
10.6	`dlinmin`	minimum of a function along a ray using derivatives
10.6	`df1dim`	function used by `dlinmin`
10.7	`dfpmin`	minimize in N-dimensions by variable metric method
10.8	`simplx`	linear programming maximization of a linear function
10.8	`simp1`	linear programming, used by `simplx`
10.8	`simp2`	linear programming, used by `simplx`
10.8	`simp3`	linear programming, used by `simplx`
10.9	`anneal`	traveling salesman problem by simulated annealing
10.9	`revcst`	cost of a reversal, used by `anneal`
10.9	`reverse`	do a reversal, used by `anneal`
10.9	`trncst`	cost of a transposition, used by `anneal`
10.9	`trnspt`	do a transposition, used by `anneal`
10.9	`metrop`	Metropolis algorithm, used by `anneal`
10.9	`amebsa`	simulated annealing in continuous spaces
10.9	`amotsa`	evaluate a trial point, used by `amebsa`
11.1	`jacobi`	eigenvalues and eigenvectors of a symmetric matrix
11.1	`eigsrt`	eigenvectors, sorts into order by eigenvalue
11.2	`tred2`	Householder reduction of a real, symmetric matrix
11.3	`tqli`	eigensolution of a symmetric tridiagonal matrix
11.5	`balanc`	balance a nonsymmetric matrix
11.5	`elmhes`	reduce a general matrix to Hessenberg form
11.6	`hqr`	eigenvalues of a Hessenberg matrix
12.2	`four1`	fast Fourier transform (FFT) in one dimension
12.3	`twofft`	fast Fourier transform of two real functions
12.3	`realft`	fast Fourier transform of a single real function
12.3	`sinft`	fast sine transform
12.3	`cosft1`	fast cosine transform with endpoints
12.3	`cosft2`	"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`	extirpolate 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 t-test for difference of means
14.2	`avevar`	calculate mean and variance of a data set
14.2	`tutest`	Student's t-test for means, case of unequal variances
14.2	`tptest`	Student's t-test for means, case of paired data
14.2	`ftest`	F-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	`quadvl`	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	`fit`	least-squares fit data to a straight line
15.3	`fitexy`	fit data to a straight line, errors in both x and y
15.3	`chixy`	used by `fitexy` to calculate a χ^2
15.4	`lfit`	general linear least-squares fit by normal equations
15.4	`covsrt`	rearrange covariance matrix, used by `lfit`
15.4	`svdfit`	linear least-squares fit by singular value decomposition
15.4	`svdvar`	variances from singular value decomposition
15.4	`fpoly`	fit a polynomial using `lfit` or `svdfit`
15.4	`fleg`	fit a Legendre polynomial using `lfit` or `svdfit`
15.5	`mrqmin`	nonlinear least-squares fit, Marquardt's method
15.5	`mrqcof`	used by `mrqmin` to evaluate coefficients
15.5	`fgauss`	fit a sum of Gaussians using `mrqmin`
15.7	`medfit`	fit data to a straight line robustly, least absolute deviation
15.7	`rofunc`	fit data robustly, used by `medfit`
16.1	`rk4`	integrate one step of ODEs, fourth-order Runge-Kutta
16.1	`rkdumb`	integrate ODEs by fourth-order Runge-Kutta
16.2	`rkqs`	integrate one step of ODEs with accuracy monitoring
16.2	`rkck`	Cash-Karp-Runge-Kutta step used by `rkqs`
16.2	`odeint`	integrate ODEs with accuracy monitoring
16.3	`mmid`	integrate ODEs by modified midpoint method
16.4	`bsstep`	integrate ODEs, Bulirsch-Stoer step
16.4	`pzextr`	polynomial extrapolation, used by `bsstep`
16.4	`rzextr`	rational function extrapolation, used by `bsstep`
16.5	`stoerm`	integrate conservative second-order ODEs
16.6	`stiff`	integrate stiff ODEs by fourth-order Rosenbrock
16.6	`jacobn`	sample Jacobian routine for `stiff`
16.6	`derivs`	sample derivatives routine for `stiff`
16.6	`simpr`	integrate stiff ODEs by semi-implicit midpoint rule
16.6	`stifbs`	integrate stiff ODEs, Bulirsch-Stoer step
17.1	`shoot`	solve two point boundary value problem by shooting
17.2	`shootf`	ditto, by shooting to a fitting point
17.3	`solvde`	two point boundary value problem, solve by relaxation
17.3	`bksub`	backsubstitution, used by `solvde`
17.3	`pinvs`	diagonalize a sub-block, used by `solvde`
17.3	`red`	reduce columns of a matrix, used by `solvde`
17.4	`sfroid`	spheroidal functions by method of `solvde`
17.4	`difeq`	spheroidal matrix coefficients, used by `sfroid`
17.4	`sphoot`	spheroidal functions by method of `shoot`
17.4	`sphfpt`	spheroidal functions by method of `shootf`
18.1	`fred2`	solve linear Fredholm equations of the second kind
18.1	`fredin`	interpolate solutions obtained with `fred2`
18.2	`voltra`	linear Volterra equations of the second kind
18.3	`wwghts`	quadrature weights for an arbitrarily singular kernel
18.3	`kermom`	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	`mpsqrt`	multiple precision square root
20.6	`mp2dfr`	multiple precision conversion to decimal base
20.6	`mppi`	multiple precision example, compute many digits of π