

GAMS Index for the NAG C Library

This index classifies NAG C Library routines according to Version 2 of the GAMS classification scheme described in [1]. Note that only those GAMS classes which contain Library routines, either directly or in a subclass, are included below.

C	Elementary and special functions (<i>search also class L5</i>)
C3	Polynomials
C3a	Orthogonal
C3a2	Chebyshev, Legendre
	e02aec nag_ld_cheb_eval
	Evaluates the coefficients of a Chebyshev series polynomial
C4	Elementary transcendental functions
C4c	Hyperbolic, inverse hyperbolic
	s10aac nag_tanh
	Hyperbolic tangent, $\tanh x$
	s10abc nag_sinh
	Hyperbolic sine, $\sinh x$
	s10acc nag_cosh
	Hyperbolic cosine, $\cosh x$
	s11aac nag_arctanh
	Inverse hyperbolic tangent, $\operatorname{arctanh} x$
	s11abc nag_arcsinh
	Inverse hyperbolic sine, $\operatorname{arcsinh} x$
	s11acc nag_arccosh
	Inverse hyperbolic cosine, $\operatorname{arccosh} x$
C5	Exponential and logarithmic integrals
	s13aac nag_exp_integral
	Exponential integral $E_1(x)$
C6	Cosine and sine integrals
	s13acc nag_cos_integral
	Cosine integral $\operatorname{Ci}(x)$
	s13adc nag_sin_integral
	Sine integral $\operatorname{Si}(x)$
C7	Gamma
C7a	Gamma, log gamma, reciprocal gamma
	s14aac nag_gamma
	Gamma function $\Gamma(x)$
	s14abc nag_log_gamma
	Log Gamma function $\ln(\Gamma(x))$
C7e	Incomplete gamma
	s14bac nag_incomplete_gamma
	Incomplete gamma functions $P(a, x)$ and $Q(a, x)$
C8	Error functions
C8a	Error functions, their inverses, integrals, including the normal distribution function
	s15abc nag_cumul_normal
	Cumulative normal distribution function, $P(x)$
	s15acc nag_cumul_normal_complem
	Complement of cumulative normal distribution function, $Q(x)$
	s15adc nag_erfc
	Complement of error function, $\operatorname{erfc} x$
	s15aec nag_erf
	Error function, $\operatorname{erf} x$
C8b	Fresnel integrals
	s20acc nag_fresnel_s
	Fresnel integral $S(x)$
	s20adc nag_fresnel_c
	Fresnel integral $C(x)$
C10	Bessel functions
C10a	J, Y, H_1, H_2
C10a1	Real argument, integer order
	s17acc nag_bessel_y0
	Bessel function $Y_0(x)$
	s17adc nag_bessel_y1
	Bessel function $Y_1(x)$
	s17aec nag_bessel_j0
	Bessel function $J_0(x)$

		s17afc	nag_bessel_j1 Bessel function $J_1(x)$
C10b	I, K		
C10b1	Real argument, integer order		
		s18acc	nag_bessel_k0 Modified Bessel function $K_0(x)$
		s18adc	nag_bessel_k1 Modified Bessel function $K_1(x)$
		s18aec	nag_bessel_i0 Modified Bessel function $I_0(x)$
		s18afc	nag_bessel_i1 Modified Bessel function $I_1(x)$
		s18ccc	nag_bessel_k0_scaled Scaled modified Bessel function $e^x K_0(x)$
		s18cdc	nag_bessel_k1_scaled Scaled modified Bessel function $e^x K_1(x)$
		s18cec	nag_bessel_i0_scaled Scaled modified Bessel function $e^{- x } I_0(x)$
		s18cfc	nag_bessel_i1_scaled Scaled modified Bessel function $e^{- x } I_1(x)$
C10c	Kelvin functions		
		s19aac	nag_kelvin_ber Kelvin function ber x
		s19abc	nag_kelvin_bei Kelvin function bei x
		s19acc	nag_kelvin_ker Kelvin function ker x
		s19adc	nag_kelvin_kei Kelvin function kei x
C10d	Airy and Scorer functions		
		s17agc	nag_airy_ai Airy function $\text{Ai}(x)$
		s17ahc	nag_airy_bi Airy function $\text{Bi}(x)$
		s17ajc	nag_airy_ai_deriv Airy function $\text{Ai}'(x)$
		s17akc	nag_airy_bi_deriv Airy function $\text{Bi}'(x)$
C14	Elliptic integrals		
		s21bac	nag_elliptic_integral_rc Degenerate symmetrised elliptic integral of 1st kind $R_C(x, y)$
		s21bbc	nag_elliptic_integral_rf Symmetrised elliptic integral of 1st kind $R_F(x, y, z)$
		s21bcc	nag_elliptic_integral_rd Symmetrised elliptic integral of 2nd kind $R_D(x, y, z)$
		s21bdc	nag_elliptic_integral_rj Symmetrised elliptic integral of 3rd kind $R_J(x, y, z, r)$
D	Linear Algebra		
D1	Elementary vector and matrix operations		
D1a	Elementary vector operations		
D1a10	Convolutions		
		c06ekc	nag_convolution_real Circular convolution or correlation of two real vectors
D1b4	Multiplication by vector		
		f06pac	dgemv Matrix-vector product, real rectangular matrix
		f06pbc	dgbmv Matrix-vector product, real rectangular band matrix
		f06pcc	dsymv Matrix-vector product, real symmetric matrix
		f06pdc	dsbmv Matrix-vector product, real symmetric band matrix
		f06pec	dspmv Matrix-vector product, real symmetric packed matrix
		f06pfc	dtrmv Matrix-vector product, real triangular matrix
		f06pgc	dtbmv Matrix-vector product, real triangular band matrix

		f06phc	dtpmv Matrix-vector product, real triangular packed matrix
		f06sac	zgemv Matrix-vector product, complex rectangular matrix
		f06sbc	zgbmv Matrix-vector product, complex rectangular band matrix
		f06scc	zhemv Matrix-vector product, complex Hermitian matrix
		f06sdc	zhbmh Matrix-vector product, complex Hermitian band matrix
		f06sec	zhpmv Matrix-vector product, complex Hermitian packed matrix
		f06sfc	ztrmv Matrix-vector product, complex triangular matrix
		f06sgc	ztbmh Matrix-vector product, complex triangular band matrix
		f06shc	ztpmv Matrix-vector product, complex triangular packed matrix
D1b5	Addition, subtraction		
		f06pmc	dger Rank-1 update, real rectangular matrix
		f06ppc	dsyr Rank-1 update, real symmetric matrix
		f06pqc	dspr Rank-1 update, real symmetric packed matrix
		f06prc	dsyr2 Rank-2 update, real symmetric matrix
		f06psc	dspr2 Rank-2 update, real symmetric packed matrix
		f06smc	zgeru Rank-1 update, complex rectangular matrix, unconjugated vector
		f06snc	zgerc Rank-1 update, complex rectangular matrix, conjugated vector
		f06spc	zher Rank-1 update, complex Hermitian matrix
		f06sqc	zhpr Rank-1 update, complex Hermitian packed matrix
		f06src	zher2 Rank-2 update, complex Hermitian matrix
		f06ssc	zhpr2 Rank-2 update, complex Hermitian packed matrix
		f06ypc	dsyrk Rank- k update of a real symmetric matrix
		f06zpc	zherk Rank- k update of a complex Hermitian matrix
		f06zrc	zher2k Rank- $2k$ update of a complex Hermitian matrix
		f06zuc	zsyrk Rank- k update of a complex symmetric matrix
		f06zwc	zsyr2k Rank- $2k$ update of a complex symmetric matrix
D1b6	Multiplication		
		f01qdc	nag_real_apply_q Compute QB or $Q^T B$ after factorization by nag_real_qr (f01qcc)
		f06yac	dgemm Matrix-matrix product, two real rectangular matrices
		f06ycc	dsymm Matrix-matrix product, one real symmetric matrix, one real rectangular matrix
		f06yfc	dtrmm Matrix-matrix product, one real triangular matrix, one real rectangular matrix
		f06yrc	dsyr2k Rank- $2k$ update of a real symmetric matrix
		f06zac	zgemm Matrix-matrix product, two complex rectangular matrices
		f06zcc	zhemm Matrix-matrix product, one complex Hermitian matrix, one complex rectangular matrix
		f06zfc	ztrmm Matrix-matrix product, one complex triangular matrix, one complex rectangular matrix

		f06ztc	zsymm Matrix-matrix product, one complex symmetric matrix, one complex rectangular matrix
D1b9	Storage mode conversion	f11zac	nag_sparse_nsym_sort Sparse sort (nonsymmetric)
		f11zbc	nag_sparse_sym_sort Sparse sort (symmetric)
D2	Solution of systems of linear equations (including inversion, <i>LU</i> and related decompositions)		
D2a	Real nonsymmetric matrices		
D2a1	General	f03afc	nag_real_lu <i>LU</i> factorization and determinant of real matrix
		f04ajc	nag_real_lu_solve_mult_rhs Approximate solution of real simultaneous linear equations (coefficient matrix already factorized by nag_real_lu (f03afc))
		f04arc	nag_real_lin_eqn Approximate solution of real simultaneous linear equations, one right-hand side
D2a3	Triangular	f06pjc	dtrsv System of equations, real triangular matrix
		f06pkc	dtbsv System of equations, real triangular band matrix
		f06plc	dtpsv System of equations, real triangular packed matrix
		f06yjc	dtrsm Solves a system of equations with multiple right-hand sides, real triangular coefficient matrix
D2a4	Sparse	f11dac	nag_sparse_nsym_fac Incomplete <i>LU</i> factorization (nonsymmetric)
		f11dcc	nag_sparse_nsym_fac_sol Solver with incomplete <i>LU</i> preconditioning (nonsymmetric)
		f11dec	nag_sparse_nsym_sol Solver with no/Jacobi/SSOR/preconditioning (nonsymmetric)
D2b	Real symmetric matrices		
D2b1	General		
D2b1b	Positive-definite	f03aec	nag_real_cholesky <i>LL^T</i> factorization and determinant of real symmetric positive-definite matrix
		f04agc	nag_real_cholesky_solve_mult_rhs Approximate solution of real symmetric positive-definite simultaneous linear equations (coefficient matrix already factorized by nag_real_cholesky (f03aec))
D2b2	Positive-definite banded	f01mcc	nag_real_cholesky_skyline <i>LDL^T</i> factorization of real symmetric positive-definite variable-bandwidth (skyline) matrix
		f04mcc	nag_real_cholesky_skyline_solve Approximate solution of real symmetric positive-definite variable-bandwidth simultaneous linear equations (coefficient matrix already factorized by nag_real_cholesky_skyline (f01mcc))
D2b4	Sparse	f11jac	nag_sparse_sym_chol_fac Incomplete Cholesky factorization (symmetric)
		f11jcc	nag_sparse_sym_chol_sol Solver with incomplete Cholesky preconditioning (symmetric)
		f11jec	nag_sparse_sym_sol Solver with Jacobi, SSOR, or no preconditioning (symmetric)
D2c	Complex non-Hermitian matrices		
D2c1	General	f03ahc	nag_complex_lu <i>LU</i> factorization and determinant of complex matrix
		f04adc	nag_complex_lin_eqn_mult_rhs Approximate solution of complex simultaneous linear equations with multiple right-hand sides
		f04akc	nag_complex_lu_solve_mult_rhs Approximate solution of complex simultaneous linear equations (coefficient matrix already factorized by nag_complex_lu (f03ahc))

D2c3	Triangular	
		f06sjc <code>ztrsv</code> System of equations, complex triangular matrix
		f06skc <code>ztbsv</code> System of equations, complex triangular band matrix
		f06slc <code>ztpsv</code> System of equations, complex triangular packed matrix
		f06zjc <code>ztrsm</code> Solves system of equations with multiple right-hand sides, complex triangular coefficient matrix
D2d	Complex Hermitian matrices	
D2d1	General	
D2d1b	Positive-definite	
		f01bnc <code>nag_complex_cholesky</code> UU^H factorization of complex Hermitian positive-definite matrix
		f04awc <code>nag_hermitian_lin_eqn_mult_rhs</code> Approximate solution of complex Hermitian positive-definite simultaneous linear equations (coefficient matrix already factorized by <code>nag_complex_cholesky</code> (f01bnc))
D2e	Associated operations (e.g., matrix reorderings)	
		f11zac <code>nag_sparse_nsym_sort</code> Sparse sort (nonsymmetric)
		f11zbc <code>nag_sparse_sym_sort</code> Sparse sort (symmetric)
D3	Determinants	
D3a	Real nonsymmetric matrices	
D3a1	General	
		f03afc <code>nag_real_lu</code> LU factorization and determinant of real matrix
D3b	Real symmetric matrices	
D3b1	General	
D3b1b	Positive-definite	
		f03aec <code>nag_real_cholesky</code> LL^T factorization and determinant of real symmetric positive-definite matrix
D4	Eigenvalues, eigenvectors	
D4a	Ordinary eigenvalue problems ($Ax = \lambda x$)	
D4a1	Real symmetric	
		f02aac <code>nag_real_symm_eigenvalues</code> All eigenvalues of real symmetric matrix
		f02abc <code>nag_real_symm_eigensystem</code> All eigenvalues and eigenvectors of real symmetric matrix
D4a2	Real nonsymmetric	
		f02afc <code>nag_real_eigenvalues</code> All eigenvalues of real matrix
		f02agc <code>nag_real_eigensystem</code> All eigenvalues and eigenvectors of real matrix
		f02ecc <code>nag_real_eigensystem_sel</code> Computes selected eigenvalues and eigenvectors of a real general matrix
D4a3	Complex Hermitian	
		f02awc <code>nag_hermitian_eigenvalues</code> All eigenvalues of complex Hermitian matrix
		f02axc <code>nag_hermitian_eigensystem</code> All eigenvalues and eigenvectors of complex Hermitian matrix
D4a4	Complex non-Hermitian	
D4b	Generalized eigenvalue problems (e.g. $Ax = \lambda Bx$)	
D4b1	Real symmetric	
		f02adc <code>nag_real_symm_general_eigenvalues</code> All eigenvalues of generalized real symmetric-definite eigenproblem
		f02aec <code>nag_real_symm_general_eigensystem</code> All eigenvalues and eigenvectors of generalized real symmetric-definite eigenproblem
D4b2	Real general	
		f02bjc <code>nag_real_general_eigensystem</code> All eigenvalues and optionally eigenvectors of real generalized eigenproblem, by QZ algorithm
D5	QR decomposition, Gram–Schmidt orthogonalization	
		f01qcc <code>nag_real_qr</code> QR factorization of real m by n matrix ($m \geq n$)
		f01qec <code>nag_real_form_q</code> Form columns of Q after factorization by <code>nag_real_qr</code> (f01qcc)

		f01rcc	<code>nag_complex_qr</code> QR factorization of complex m by n matrix ($m \geq n$)
		f01rdc	<code>nag_complex_apply_q</code> Compute QB or $Q^H B$ after factorization by <code>nag_complex_qr</code> (f01rcc)
		f01rec	<code>nag_complex_form_q</code> Form columns of Q after factorization by <code>nag_complex_qr</code> (f01rcc)
D6	Singular value decomposition		
		f02wec	<code>nag_real_svd</code> SVD of real matrix
		f02xec	<code>nag_complex_svd</code> SVD of complex matrix
D9	Singular, overdetermined or underdetermined systems of linear equations, generalized inverses		
D9b	Constrained		
D9b1	Least squares (L_2) solution		
		e04ncc	<code>nag_opt_lin_lsq</code> Solves linear least-squares and convex quadratic programming problems (non-sparse)
E	Interpolation		
E1	Univariate data (curve fitting)		
E1a	Polynomial splines (piecewise polynomials)		
		e01bac	<code>nag_1d_spline_interpolant</code> Interpolating function, cubic spline interpolant, one variable
		e01bec	<code>nag_monotonic_interpolant</code> Interpolating function, monotonicity-preserving, piecewise cubic Hermite, one variable
		e02bac	<code>nag_1d_spline_fit_knots</code> Least-squares curve cubic spline fit (including interpolation), one variable
E1b	Polynomials		
		e02afc	<code>nag_1d_cheb_interp_fit</code> Computes the coefficients of a Chebyshev series polynomial for interpolated data
E2	Multivariate data (surface fitting)		
E2a	Gridded		
		e01dac	<code>nag_2d_spline_interpolant</code> Interpolating function, bicubic spline interpolant, two variables
E2b	Scattered		
		e01sac	<code>nag_2d_scatter_interpolant</code> A function to generate a two-dimensional surface interpolating a set of data points, using either the method of Renka and Cline or using the modified Shepard's method
E3	Service routines for interpolation		
		e01szc	<code>nag_2d_scatter_free</code> Freeing function for use with <code>nag_2d_scatter_eval</code> (e01sbc)
E3a	Evaluation of fitted functions, including quadrature		
E3a1	Function evaluation		
		e01bfc	<code>nag_monotonic_evaluate</code> Evaluation of interpolant computed by <code>nag_monotonic_interpolant</code> (e01bec), function only
		e01sbc	<code>nag_2d_scatter_eval</code> A function to evaluate at a set of points, the two-dimensional interpolant function generated by <code>nag_2d_scatter_interpolant</code> (e01sac)
		e02aec	<code>nag_1d_cheb_eval</code> Evaluates the coefficients of a Chebyshev series polynomial
		e02bbc	<code>nag_1d_spline_evaluate</code> Evaluation of fitted cubic spline, function only
		e02bcc	<code>nag_1d_spline_deriv</code> Evaluation of fitted cubic spline, function and derivatives
		e02dec	<code>nag_2d_spline_eval</code> Evaluation of bicubic spline, at a set of points
		e02dfc	<code>nag_2d_spline_eval_rect</code> Evaluation of bicubic spline, at a mesh of points
E3a2	Derivative evaluation		
		e01bgc	<code>nag_monotonic_deriv</code> Evaluation of interpolant computed by <code>nag_monotonic_interpolant</code> (e01bec), function and first derivative
		e02bcc	<code>nag_1d_spline_deriv</code> Evaluation of fitted cubic spline, function and derivatives
E3a3	Quadrature		
		e01bhc	<code>nag_monotonic_intg</code> Evaluation of interpolant computed by <code>nag_monotonic_interpolant</code> (e01bec), definite integral

- e02bdc nag_1d_spline_intg
Evaluation of fitted cubic spline, definite integral
- F** Solution of nonlinear equations
- F1** Single equation
- F1a** Polynomial
- F1a1** Real coefficients
 - c02agc nag_zeros_real_poly
Zeros of a polynomial with real coefficients
- F1a2** Complex coefficients
 - c02afc nag_zeros_complex_poly
Zeros of a polynomial with complex coefficients
- F1b** Nonpolynomial
 - c05adc nag_zero_cont_func_bd
Zero of a continuous function of one variable
 - c05sdc nag_zero_cont_func_bd_1
Zero of a continuous function of one variable, thread-safe
- F2** System of equations
 - c05nbc nag_zero_nonlin_eqns
Solution of a system of nonlinear equations (function values only)
 - c05pbc nag_zero_nonlin_eqns_deriv
Solution of a system of nonlinear equations (using first derivatives)
 - c05tbc nag_zero_nonlin_eqns_1
Solution of a system of nonlinear equations (function values only), thread-safe
 - c05ubc nag_zero_nonlin_eqns_deriv_1
Solution of a system of nonlinear equations (using first derivatives), thread-safe
 - c05zbc nag_check_deriv
Derivative checker for nag_zero_nonlin_eqns_deriv (c05pbc)
- F3** Service routines (e.g., check user-supplied derivatives)
 - c05zcc nag_check_deriv_1
Derivative checker for nag_zero_nonlin_eqns_deriv (c05pbc), thread-safe
 - e04hcc nag_opt_check_deriv
Derivative checker for use with nag_opt_bounds_deriv (e04kbc)
 - e04hdc nag_opt_check_2nd_deriv
Checks 2nd derivatives of a user-defined function.
 - e04kbc nag_opt_bounds_deriv
Bound constrained nonlinear minimization (first derivatives required)
- G** Optimization (*search also classes K, L8*)
- G1** Unconstrained
- G1a** Univariate
- G1a1** Smooth function
- G1a1a** User provides no derivatives
 - e04abc nag_opt_one_var_no_deriv
Minimizes a function of one variable, using function values only
- G1a1b** User provides first derivatives
 - e04bbc nag_opt_one_var_deriv
Minimizes a function of one variable, requires first derivatives
- G1b** Multivariate
- G1b1** Smooth function
- G1b1a** User provides no derivatives
 - e04jbc nag_opt_bounds_no_deriv
Bound constrained nonlinear minimization (no derivatives required)
- G1b1b** User provides first derivatives
 - e04dgc nag_opt_conj_grad
Unconstrained minimization using conjugate gradients
- G1b2** General function (no smoothness assumed)
 - e04ccc nag_opt_simplex
Unconstrained minimization using simplex algorithm
- G2** Constrained
- G2a** Linear programming
- G2a1** Dense matrix of constraints
 - e04mfc nag_opt_lp
Linear programming
 - e04ncc nag_opt_lin_lsq
Solves linear least-squares and convex quadratic programming problems (non-sparse)
 - e04nfc nag_opt_qp
Quadratic programming

G2a2	Sparse matrix of constraints	e04nkc <code>nag_opt_sparse_convex_qp</code> Solves sparse linear programming or convex quadratic programming problems
G2b	Transportation and assignments problem	h03abc <code>nag_transport</code> Classical transportation algorithm
G2c	Integer programming	
G2c1	Zero/one	h02bbc <code>nag_ip_bb</code> Solves integer programming problems using a branch and bound method
G2c6	Pure integer programming	h02bbc <code>nag_ip_bb</code> Solves integer programming problems using a branch and bound method
G2c7	Mixed integer programming	h02bbc <code>nag_ip_bb</code> Solves integer programming problems using a branch and bound method
G2e	Quadratic programming	
G2e1	Positive-definite Hessian (i.e., convex problem)	e04ncc <code>nag_opt_lin_lsq</code> Solves linear least-squares and convex quadratic programming problems (non-sparse) e04nfc <code>nag_opt_qp</code> Quadratic programming e04nkc <code>nag_opt_sparse_convex_qp</code> Solves sparse linear programming or convex quadratic programming problems
G2e2	Indefinite Hessian	e04nfc <code>nag_opt_qp</code> Quadratic programming e04nkc <code>nag_opt_sparse_convex_qp</code> Solves sparse linear programming or convex quadratic programming problems
G2h	General nonlinear programming	
G2h1	Simple bounds	
G2h1a	Smooth function	
G2h1a1	User provides no derivatives	e04jbc <code>nag_opt_bounds_no_deriv</code> Bound constrained nonlinear minimization (no derivatives required) e04ucc <code>nag_opt_nlp</code> Minimization with nonlinear constraints using a sequential QP method e04unc <code>nag_opt_nlin_lsq</code> Solves nonlinear least-squares problems using the sequential QP method
G2h1a2	User provides first derivatives	e04ucc <code>nag_opt_nlp</code> Minimization with nonlinear constraints using a sequential QP method e04unc <code>nag_opt_nlin_lsq</code> Solves nonlinear least-squares problems using the sequential QP method
G2h1a3	User provides first and second derivatives	e041bc <code>nag_opt_bounds_2nd_deriv</code> Solves bound constrained problems. 1st and 2nd derivatives are required.
G2h2	Linear equality or inequality constraints	
G2h2a	Smooth function	
G2h2a1	User provides no derivatives	e04ucc <code>nag_opt_nlp</code> Minimization with nonlinear constraints using a sequential QP method e04unc <code>nag_opt_nlin_lsq</code> Solves nonlinear least-squares problems using the sequential QP method
G2h2a2	User provides first derivatives	e04ucc <code>nag_opt_nlp</code> Minimization with nonlinear constraints using a sequential QP method e04unc <code>nag_opt_nlin_lsq</code> Solves nonlinear least-squares problems using the sequential QP method
G2h3	Nonlinear constraints	
G2h3a	Equality constraints only	
G2h3a1	Smooth function and constraints	e04ucc <code>nag_opt_nlp</code> Minimization with nonlinear constraints using a sequential QP method e04unc <code>nag_opt_nlin_lsq</code> Solves nonlinear least-squares problems using the sequential QP method

G2h3b	Equality and inequality constraints
G2h3b1	Smooth function and constraints
G2h3b1a	User provides no derivatives
	e04ucc nag_opt_nlp
	Minimization with nonlinear constraints using a sequential QP method
	e04unc nag_opt_nlin_lsq
	Solves nonlinear least-squares problems using the sequential QP method
G2h3b1b	User provides first derivatives of function and constraints
	e04ucc nag_opt_nlp
	Minimization with nonlinear constraints using a sequential QP method
	e04unc nag_opt_nlin_lsq
	Solves nonlinear least-squares problems using the sequential QP method
G4	Service routines
G4a	Problem input (e.g., matrix generation)
	e04mzc nag_opt_sparse_mps_read
	Read MPSX data for sparse LP or QP problem from a file
	h02buc nag_ip_mps_read
	Read MPSX data for IP, LP or QP problem from a file
G4c	Check user-supplied derivatives
	e04hcc nag_opt_check_deriv
	Derivative checker for use with nag_opt_bounds_deriv (e04kbc)
	e04hdc nag_opt_check_2nd_deriv
	Checks 2nd derivatives of a user-defined function.
	e04kbc nag_opt_bounds_deriv
	Bound constrained nonlinear minimization (first derivatives required)
	e04yac nag_opt_lsq_check_deriv
	Least-squares derivative checker for use with nag_opt_lsq_deriv (e04gbc)
G4d	Find feasible point
	e04mfc nag_opt_lp
	Linear programming
	e04ncc nag_opt_lin_lsq
	Solves linear least-squares and convex quadratic programming problems (non-sparse)
	e04nfc nag_opt_qp
	Quadratic programming
	e04nkc nag_opt_sparse_convex_qp
	Solves sparse linear programming or convex quadratic programming problems
	e04ucc nag_opt_nlp
	Minimization with nonlinear constraints using a sequential QP method
	e04unc nag_opt_nlin_lsq
	Solves nonlinear least-squares problems using the sequential QP method
G4f	Other
	e04myc nag_opt_sparse_mps_free
	Free memory allocated by nag_opt_sparse_mps_read (e04mzc)
	e04xac nag_opt_estimate_deriv
	Computes an approximation to the gradient vector and/or the Hessian matrix for use with nag_opt_nlp (e04ucc) and other nonlinear optimization functions
	e04xxc nag_opt_init
	Initialisation function for option setting
	e04xyc nag_opt_read
	Read options from a textfile
	e04xzc nag_opt_free
	NAG memory freeing function for use with option setting
	h02bvc nag_ip_mps_free
	Free memory allocated by nag_ip_mps_read (h02buc)
	h02xxc nag_ip_init
	Initialize option structure to null values
	h02xyc nag_ip_read
	Read optional parameter values from a file
	h02xzc nag_ip_free
	Free NAG allocated memory from option structures
H	Differentiation, integration
H1	Numerical differentiation
	e04xac nag_opt_estimate_deriv
	Computes an approximation to the gradient vector and/or the Hessian matrix for use with nag_opt_nlp (e04ucc) and other nonlinear optimization functions
H2	Quadrature (numerical evaluation of definite integrals)
H2a	One-dimensional integrals
H2a1	Finite interval (general integrand)

H2a1a	Integrand available via user-defined procedure
H2a1a1	Automatic (user need only specify required accuracy) d01ajc nag_ld_quad_gen 1-D adaptive quadrature, allowing for badly-behaved integrands d01akc nag_ld_quad_osc 1-D adaptive quadrature, suitable for oscillating functions d01alc nag_ld_quad_brkpts 1-D adaptive quadrature, allowing for singularities at specified points d01sjc nag_ld_quad_gen_l 1-D adaptive quadrature, allowing for badly-behaved integrands, thread-safe d01skc nag_ld_quad_osc_l 1-D adaptive quadrature, suitable for oscillating functions, thread-safe d01slc nag_ld_quad_brkpts_l 1-D adaptive quadrature, allowing for singularities at specified points, thread-safe
H2a1a2	Nonautomatic d01bac nag_ld_quad_gauss 1-D Gaussian quadrature rule evaluation d01tac nag_ld_quad_gauss_l 1-D Gaussian quadrature rule evaluation, thread-safe
H2a1b	Integrand available only on grid
H2a1b2	Nonautomatic d01gac nag_ld_quad_vals 1-D integration of a function defined by data values only
H2a2	Finite interval (specific or special type integrand including weight functions, oscillating and singular integrands, principal value integrals, splines, etc.)
H2a2a	Integrand available via user-defined procedure
H2a2a1	Automatic (user need only specify required accuracy) d01anc nag_ld_quad_wt_trig 1-D adaptive quadrature, finite interval, sine or cosine weight functions d01apc nag_ld_quad_wt_alglog 1-D adaptive quadrature, weight function with end-point singularities of algebraic-logarithmic type d01aqc nag_ld_quad_wt_cauchy 1-D adaptive quadrature, weight function $1/(x - c)$, Cauchy principal value d01skc nag_ld_quad_osc_l 1-D adaptive quadrature, suitable for oscillating functions, thread-safe d01slc nag_ld_quad_brkpts_l 1-D adaptive quadrature, allowing for singularities at specified points, thread-safe d01snc nag_ld_quad_wt_trig_l 1-D adaptive quadrature, finite interval, sine or cosine weight functions, thread-safe d01spc nag_ld_quad_wt_alglog_l 1-D adaptive quadrature, weight function with end-point singularities of algebraic-logarithmic type, thread-safe d01sqc nag_ld_quad_wt_cauchy_l 1-D adaptive quadrature, weight function $1/(x - c)$, Cauchy principal value, thread-safe
H2a2b	Integrand available only on grid
H2a2b1	Automatic (user need only specify required accuracy) e02bdc nag_ld_spline_intg Evaluation of fitted cubic spline, definite integral
H2a3	Semi-infinite interval (including e^{-x} weight function)
H2a3a	Integrand available via user-defined procedure
H2a3a1	Automatic (user need only specify required accuracy) d01amc nag_ld_quad_inf 1-D adaptive quadrature over infinite or semi-infinite interval d01asc nag_ld_quad_inf_wt_trig 1-D adaptive quadrature, semi-infinite interval, sine or cosine weight function d01smc nag_ld_quad_inf_l 1-D adaptive quadrature over infinite or semi-infinite interval, thread-safe d01ssc nag_ld_quad_inf_wt_trig_l 1-D adaptive quadrature, semi-infinite interval, sine or cosine weight function, thread-safe
H2a3a2	Nonautomatic d01bac nag_ld_quad_gauss 1-D Gaussian quadrature rule evaluation d01tac nag_ld_quad_gauss_l 1-D Gaussian quadrature rule evaluation, thread-safe
H2a4	Infinite interval (including e^{-x^2} weight function)
H2a4a	Integrand available via user-defined procedure

- H2a4a1** Automatic (user need only specify required accuracy)
 - d01amc** nag_ld_quad_inf
1-D adaptive quadrature over infinite or semi-infinite interval
 - d01smc** nag_ld_quad_inf_1
1-D adaptive quadrature over infinite or semi-infinite interval, thread-safe
- H2a4a2** Nonautomatic
 - d01bac** nag_ld_quad_gauss
1-D Gaussian quadrature rule evaluation
 - d01tac** nag_ld_quad_gauss_1
1-D Gaussian quadrature rule evaluation, thread-safe
- H2b** Multidimensional integrals
 - H2b1** One or more hyper-rectangular regions (includes iterated integrals)
 - H2b1a** Integrand available via user-defined procedure
 - H2b1a1** Automatic (user need only specify required accuracy)
 - d01fcc** nag_multid_quad_adapt
Multi-dimensional adaptive quadrature
 - d01gbc** nag_multid_quad_monte_carlo
Multi-dimensional quadrature, using Monte Carlo method
 - d01wcc** nag_multid_quad_adapt_1
Multi-dimensional adaptive quadrature, thread-safe
 - d01xbc** nag_multid_quad_monte_carlo_1
Multi-dimensional quadrature, using Monte Carlo method, thread-safe
- I** Differential and integral equations
 - I1** Ordinary differential equations (ODE's)
 - I1a** Initial value problems
 - I1a1** General, nonstiff or mildly stiff
 - I1a1a** One-step methods (e.g., Runge–Kutta)
 - d02pcc** nag_ode_ivp_rk_range
Ordinary differential equations solver, initial value problems over a range using Runge–Kutta methods
 - d02pdc** nag_ode_ivp_rk_onestep
Ordinary differential equations solver, initial value problems, one time step using Runge–Kutta methods
 - I1a1b** Multistep methods (e.g. Adams predictor-corrector)
 - d02cjc** nag_ode_ivp_adams_gen
Ordinary differential equation solver using a variable-order variable-step Adams method (black box)
 - d02qfc** nag_ode_ivp_adams_roots
Ordinary differential equation solver using Adams method (sophisticated use)
 - I1a2** Stiff and mixed algebraic- differential equations
 - d02ejc** nag_ode_ivp_bdf_gen
Ordinary differential equations solver, stiff, initial value problems using the Backward Differentiation Formulae
 - I1b** Multipoint boundary value problems
 - I1b1** Linear
 - d02gbc** nag_ode_bvp_fd_lin_gen
Ordinary differential equations solver, for general linear two-point boundary value problems, using a finite difference technique with deferred correction
 - I1b2** Nonlinear
 - d02gac** nag_ode_bvp_fd_nonlin_fixedbc
Ordinary differential equations solver, for simple nonlinear 2-point boundary value problems, using a finite difference technique with deferred correction
 - d02rac** nag_ode_bvp_fd_nonlin_gen
Ordinary differential equations solver, for general non-linear two-point boundary value problems, using a finite difference technique with deferred correction
 - I1c** Service routines (e.g., interpolation of solutions, error handling, test programs)
 - d02ppc** nag_ode_ivp_rk_free
Freeing function for use with the Runge–Kutta suite (d02p functions)
 - d02pvc** nag_ode_ivp_rk_setup
Set-up function for use with nag_ode_ivp_rk_range (d02pcc) and/or nag_ode_ivp_rk_onestep (d02pdc)
 - d02pwc** nag_ode_ivp_rk_reset_tend
A function to re-set the end point following a call to nag_ode_ivp_rk_onestep (d02pdc)
 - d02pxc** nag_ode_ivp_rk_interp
Ordinary differential equations solver, computes the solution by interpolation anywhere on an integration step taken by nag_ode_ivp_rk_onestep (d02pdc)

		d02pzc	nag_ode_ivp_rk_errass	A function to provide global error assessment during an integration with either nag_ode_ivp_rk_range (d02pcc) or nag_ode_ivp_rk_onestep (d02pdc)
		d02qwc	nag_ode_ivp_adams_setup	Set-up function for nag_ode_ivp_adams_roots (d02qfc)
		d02qyc	nag_ode_ivp_adams_free	Freeing function for use with nag_ode_ivp_adams_roots (d02qfc)
		d02qzc	nag_ode_ivp_adams_interp	Interpolation function for use with nag_ode_ivp_adams_roots (d02qfc)
J	Integral transforms			
J1	Trigonometric transforms including fast Fourier transforms			
J1a	One-dimensional			
J1a1	Real	c06eac	nag_fft_real	Single 1-D real discrete Fourier transform
		c06fpc	nag_fft_multiple_real	Multiple 1-D real discrete Fourier transforms
J1a2	Complex	c06ebc	nag_fft_hermitian	Single 1-D Hermitian discrete Fourier transform
		c06ecc	nag_fft_complex	Single 1-D complex discrete Fourier transform
		c06fqc	nag_fft_multiple_hermitian	Multiple 1-D Hermitian discrete Fourier transforms
		c06frc	nag_fft_multiple_complex	Multiple 1-D complex discrete Fourier transforms
		c06gbc	nag_conjugate_hermitian	Complex conjugate of Hermitian sequence
		c06gcc	nag_conjugate_complex	Complex conjugate of complex sequence
		c06gqc	nag_multiple_conjugate_hermitian	Complex conjugate of multiple Hermitian sequences
		c06gsc	nag_multiple_hermitian_to_complex	Convert Hermitian sequences to general complex sequences
		c06gzc	nag_fft_init_trig	Initialisation function for other c6 functions
J1a3	Sine and cosine transforms	c06hac	nag_fft_multiple_sine	Discrete sine transform
		c06hbc	nag_fft_multiple_cosine	Discrete cosine transform
		c06hcc	nag_fft_multiple_qtr_sine	Discrete quarter-wave sine transform
		c06hdc	nag_fft_multiple_qtr_cosine	Discrete quarter-wave cosine transform
J1b	Multidimensional	c06fuc	nag_fft_2d_complex	2-D complex discrete Fourier transform
J2	Convolutions	c06ekc	nag_convolution_real	Circular convolution or correlation of two real vectors
J4	Hilbert transforms	d01aqc	nag_1d_quad_wt_cauchy	1-D adaptive quadrature, weight function $1/(x - c)$, Cauchy principal value
		d01s qc	nag_1d_quad_wt_cauchy_1	1-D adaptive quadrature, weight function $1/(x - c)$, Cauchy principal value, thread-safe
K	Approximation (<i>search also class L8</i>)			
K1	Least squares (L_2) approximation			
K1a	Linear least squares (<i>search also classes D5, D6, D9</i>)			
K1a1	Unconstrained			
K1a1a	Univariate data (curve fitting)			
K1a1a1	Polynomial splines (piecewise polynomials)	e02bac	nag_1d_spline_fit_knots	Least-squares curve cubic spline fit (including interpolation), one variable
		e02bec	nag_1d_spline_fit	Least-squares cubic spline curve fit, automatic knot placement, one variable

K1a1a2	Polynomials	
	e02adc	nag_1d_cheb_fit Computes the coefficients of a Chebyshev series polynomial for arbitrary data
	e02afc	nag_1d_cheb_interp_fit Computes the coefficients of a Chebyshev series polynomial for interpolated data
K1a1b	Multivariate data (surface fitting)	
	e02dcc	nag_2d_spline_fit_grid Least-squares bicubic spline fit with automatic knot placement, two variables (rectangular grid)
	e02ddc	nag_2d_spline_fit_scatter Least-squares bicubic spline fit with automatic knot placement, two variables (scattered data)
K1b	Nonlinear least squares	
K1b1	Unconstrained	
K1b1a	Smooth functions	
K1b1a1	User provides no derivatives	
	e04fcc	nag_opt_lsq_no_deriv Unconstrained nonlinear least squares (no derivatives required)
K1b1a2	User provides first derivatives	
	e04gbc	nag_opt_lsq_deriv Unconstrained nonlinear least squares (first derivatives required)
K1b2	Constrained	
K1b2b	Nonlinear constraints	
	e04unc	nag_opt_nlin_lsq Solves nonlinear least-squares problems using the sequential QP method
K6	Service routines for approximation	
K6a	Evaluation of fitted functions, including quadrature	
K6a1	Function evaluation	
	e02aec	nag_1d_cheb_eval Evaluates the coefficients of a Chebyshev series polynomial
	e02bbc	nag_1d_spline_evaluate Evaluation of fitted cubic spline, function only
	e02bcc	nag_1d_spline_deriv Evaluation of fitted cubic spline, function and derivatives
K6a2	Derivative evaluation	
	e02bcc	nag_1d_spline_deriv Evaluation of fitted cubic spline, function and derivatives
K6a3	Quadrature	
	e02bdc	nag_1d_spline_intg Evaluation of fitted cubic spline, definite integral
L	Statistics, probability	
L1	Data summarization	
L1a	One-dimensional data	
L1a1	Raw data	
	g01aac	nag_summary_stats_1var Mean, variance, skewness, kurtosis etc, one variable, from raw data
	g01alc	nag_5pt_summary_stats five-point summary (median, hinges and extremes)
	g07dac	nag_median_1var Robust estimation, median, median absolute deviation, robust standard deviation
	g07dbc	nag_robust_m_estim_1var Robust estimation, M -estimate of location and scale parameters, standard weight function
	g07ddc	nag_robust_trimmed_1var Trimmed and winsorized mean of a sample with estimates of the variances of the two means
L1c	Multi-dimensional data	
L1c1	Raw data	
L1c1b	Covariance, correlation	
	g02bxc	nag_corr_cov Product-moment correlation, unweighted/weighted correlation and covariance matrix, allows variables to be disregarded
	g02hkc	nag_robust_corr_estim Robust estimation of a correlation matrix, Huber's weight function
L1c2	Raw data containing missing values (<i>search also class L1c1</i>)	
	g02brc	nag_ken_spe_corr_coeff Kendall and/or Spearman non-parametric rank correlation coefficients, allows variables and observations to be selectively disregarded

L10	Time series analysis (<i>search also class J</i>)
L10a	Univariate (<i>search also classes L3a6 and L3a7</i>)
L10a2	Time domain analysis
L10a2a	Summary statistics
L10a2a1	Autocorrelations and autocovariances
	g13abc nag_tsa_auto_corr
	Sample autocorrelation function
L10a2a2	Partial autocorrelations
	g13acc nag_tsa_auto_corr_part
	Partial autocorrelation function
L10a2c	Autoregressive models
L10a2c1	Model identification
	g13acc nag_tsa_auto_corr_part
	Partial autocorrelation function
L10a2d	ARMA and ARIMA models (including Box–Jenkins methods)
L10a2d2	Parameter estimation
	g13bec nag_tsa_multi_inp_model_estim
	Estimation for time series models
L10a2e	State-space analysis (e.g., Kalman filtering)
	g13eac nag_kalman_sqrt_filt_cov_var
	One iteration step of the time-varying Kalman filter recursion using the square root covariance implementation
	g13ebc nag_kalman_sqrt_filt_cov_invar
	One iteration step of the time-invariant Kalman filter recursion using the square root covariance implementation with (A, C) in lower observer Hessenberg form
	g13ecc nag_kalman_sqrt_filt_info_var
	One iteration step of the time-varying Kalman filter recursion using the square root information implementation
	g13edc nag_kalman_sqrt_filt_info_invar
	One iteration step of the time-invariant Kalman filter recursion using the square root information implementation with $(A^{-1}, A^{-1}B)$ in upper controller Hessenberg form
	g13ewc nag_trans_hessenberg_observer
	Unitary state-space transformation to reduce (A, C) to lower or upper observer Hessenberg form
	g13exc nag_trans_hessenberg_controller
	Unitary state-space transformation to reduce (B, A) to lower or upper controller Hessenberg form
L10a3	Frequency domain analysis (<i>search also class J1</i>)
L10a3a	Spectral analysis
L10a3a3	Spectrum estimation using the periodogram
	g13cbc nag_tsa_spectrum_univar
	Univariate time series, smoothed sample spectrum using spectral smoothing by the trapezium frequency (Daniell) window
L10b	Two time series (<i>search also classes L3b3c, L10c, and L10d</i>)
L10b2	Time domain analysis
L10b2b	Transfer function models
	g13bec nag_tsa_multi_inp_model_estim
	Estimation for time series models
	g13bjc nag_tsa_multi_inp_model_forecast
	Forecasting function
L10b3	Frequency domain analysis (<i>search also class J1</i>)
L10b3a	Cross-spectral analysis
L10b3a3	Cross-spectrum estimation using the cross-periodogram
	g13cdc nag_tsa_spectrum_bivar
	Multivariate time series, smoothed sample cross spectrum using spectral smoothing by the trapezium frequency (Daniell) window
L10b3a6	Spectral functions
	g13cec nag_tsa_cross_spectrum_bivar
	Multivariate time series, cross amplitude spectrum, squared coherency, bounds, univariate and bivariate (cross) spectra
	g13cfc nag_tsa_gain_phase_bivar
	Multivariate time series, gain, phase, bounds, univariate and bivariate (cross) spectra
	g13cgc nag_tsa_noise_spectrum_bivar
	Multivariate time series, noise spectrum, bounds, impulse response function and its standard error
L12	Discriminant analysis
	g03acc nag_mv_canon_var
	Canonical variate analysis

- g03dac nag_mv_discrim
Test for equality of within-group covariance matrices
 - g03dbc nag_mv_discrim_mahaldist
Mahalanobis squared distances, following nag_mv_discrim (g03dac)
 - g03dcc nag_mv_discrim_group
Allocation of observations to groups, following nag_mv_discrim (g03dac)
- L13** Covariance structure models
- L13a** Factor analysis
 - g03bac nag_mv_orthomax
Orthogonal rotations for loading matrix
 - g03bcc nag_mv_procustes
Procrustes rotations
 - g03cac nag_mv_factor
Maximum likelihood estimates of parameters
 - g03ccc nag_mv_fac_score
Factor score coefficients, following nag_mv_factor (g03cac)
- L13b** Principal components analysis
 - g03aac nag_mv_prin_comp
Principal component analysis
- L13c** Canonical correlation
 - g03acc nag_mv_canon_var
Canonical variate analysis
 - g03adc nag_mv_canon_corr
Canonical correlation analysis
- L14** Cluster analysis
- L14a** One-way
- L14a1** Unconstrained
- L14a1a** Nested
- L14a1a1** Joining (e.g., single link)
 - g03ecc nag_mv_hierar_cluster_analysis
Performs hierarchical cluster analysis
 - g03ehc nag_mv_dendrogram
Construct dendrogram following nag_mv_hierar_cluster_analysis (g03ecc)
 - g03ejc nag_mv_cluster_indicator
Construct clusters following nag_mv_hierar_cluster_analysis (g03ecc)
- L14a1b** Non-nested (e.g., K means)
 - g03efc nag_mv_kmeans_cluster_analysis
K-means
- L14d** Service routines (e.g., compute distance matrix)
 - g03eac nag_mv_distance_mat
Compute distance (dissimilarity) matrix
 - g03xzc nag_mv_dend_free
Frees memory allocated to the dendrogram array in nag_mv_dendrogram (g03ehc)
 - g13bxc nag_tsa_options_init
Initialisation function for option setting
 - g13byc nag_tsa_transf_orders
Function to allocate memory to transfer function model orders
 - g13bzc nag_tsa_trans_free
Freeing function for the structure holding the transfer function model orders
- L15** Life testing, survival analysis
 - g12aac nag_prod_limit_surviv_fn
Kaplan-Meier (product-limit) estimates of survival probabilities
- L16** Multidimensional scaling
 - g03fac nag_mv_prin_coord_analysis
Principal co-ordinate analysis
 - g03fcc nag_mv_ordinal_multidimscale
Multidimensional scaling
- L2** Data manipulation
- L2a** Transform (*search also classes L10a1, N6, and N8*)
 - g03zac nag_mv_z_scores
Standardize values of a data matrix
- L4** Elementary data analysis
- L4a** One-dimensional data
- L4a1** Raw data
- L4a1a** Parametric analysis
- L4a1a2** Probability plots
- L4a1a2n** Negative binomial, normal
 - g01dhc nag_ranks_and_scores
Ranks, Normal scores, approximate Normal scores or exponential (Savage) scores

L4a1a4	Parameter estimates and tests
L4a1a4n	Normal
	g01ddc nag_shapiro_wilk_test Shapiro and Wilk's W test for Normality
	g07cac nag_2_sample_t_test t -test statistic, for a difference in means between two Normal populations, confidence interval
L4a5	Categorical data
	g11aac nag_chi_sq_2_way_table χ^2 statistic for two-way contingency table
L5	Function evaluation (<i>search also class C</i>)
L5a	Univariate
L5a1	Cumulative distribution functions, probability density functions
L5a1b	Beta, binomial
	g01bjc nag_binomial_dist Binomial distribution function
	g01eec nag_prob_beta_dist Upper and lower tail probabilities and probability density function for the beta distribution
L5a1c	Cauchy, χ^2
	g01ecc nag_prob_chi_sq Probabilities for χ^2 distribution
L5a1e	Error function, exponential, extreme value
	s15adc nag_erfc Complement of error function, erfc x
	s15aec nag_erf Error function, erf x
L5a1f	F distribution
	g01edc nag_prob_f_dist Probabilities for F -distribution
L5a1g	Gamma, general, geometric
	g01efc nag_gamma_dist Probabilities for the gamma distribution
L5a1h	Halfnormal, hypergeometric
	g01blc nag_hypergeom_dist Hypergeometric distribution function
L5a1n	Negative binomial, normal
	g01eac nag_prob_normal Probabilities for the standard Normal distribution
	s15abc nag_cumul_normal Cumulative normal distribution function, $P(x)$
	s15acc nag_cumul_normal_complem Complement of cumulative normal distribution function, $Q(x)$
L5a1p	Pareto, Poisson
	g01bkc nag_poisson_dist Poisson distribution function
L5a1t	t distribution
	g01ebc nag_prob_students_t Probabilities for Student's t -distribution
L5a2	Inverse distribution functions, sparsity functions
L5a2b	Beta, binomial
	g01fec nag_deviates_beta Deviates for the beta distribution
L5a2c	Cauchy, χ^2
	g01fcc nag_deviates_chi_sq Deviates for the χ^2 distribution
L5a2f	F distribution
	g01fdc nag_deviates_f_dist Deviates for the F -distribution
L5a2g	Gamma, general, geometric
	g01ffc nag_deviates_gamma_dist Deviates for the gamma distribution
L5a2n	Negative binomial, normal, normal order statistics
	g01cec nag_deviates_normal_dist Deviate of Normal distribution function
	g01fac nag_deviates_normal Deviates for the Normal distribution

L5a2t	<i>t</i> distribution	
		g01fbc <code>nag_deviates_students_t</code> Deviates for Student's <i>t</i> -distribution
L5b	Multivariate	
L5b1	Cumulative multivariate distribution functions, probability density functions	
L5b1n	Normal	
		g01hac <code>nag_bivariate_normal_dist</code> Probability for the bivariate Normal distribution
L6	Random number generation	
L6a	Univariate	
		g05eyc <code>nag_return_discrete</code> Pseudo-random integer from reference vector
L6a2	Beta, binomial, Boolean	
		g05edc <code>nag_ref_vec_binomial</code> Set up reference vector for generating pseudo-random integers, binomial distribution
		g05fec <code>nag_random_beta</code> Pseudo-random real numbers from the beta distribution
L6a5	Exponential, extreme value	
		g05dbc <code>nag_random_exp</code> Pseudo-random real number, (negative) exponential distribution
L6a7	Gamma, general (continuous, discrete), geometric	
		g05exc <code>nag_ref_vec_discrete_pdf_cdf</code> Set up reference vector from supplied cumulative distribution function or probability distribution function
		g05ffc <code>nag_random_gamma</code> Pseudo-random real numbers from the gamma distribution
L6a14	Negative binomial, normal, normal order statistics	
		g05ddc <code>nag_random_normal</code> Pseudo-random real number, Normal distribution
L6a16	Pareto, Pascal, permutations, Poisson	
		g05ecc <code>nag_ref_vec_poisson</code> Set up reference vector for generating pseudo-random integers, Poisson distribution
		g05ehc <code>nag_ran_permut_vec</code> Pseudo-random permutation of a vector of integers
L6a19	Samples, stable distribution	
		g05ejc <code>nag_ran_sample_vec</code> Pseudo-random sample without replacement from an integer vector
L6a20	<i>t</i> distribution, time series, triangular	
		g05hac <code>nag_arma_time_series</code> ARMA time series of <i>n</i> terms
L6a21	Uniform (continuous, discrete), uniform order statistics	
		g05cac <code>nag_random_continuous_uniform</code> Pseudo-random real number, uniform distribution over (0,1)
		g05dac <code>nag_random_continuous_uniform_ab</code> Pseudo-random real number, uniform distribution over (<i>a</i> , <i>b</i>)
		g05dyc <code>nag_random_discrete_uniform</code> Pseudo-random integer from uniform distribution
L6b	Multivariate	
L6b14	Normal	
		g05eac <code>nag_ref_vec_multi_normal</code> Set up reference vector for multivariate Normal distribution
		g05ezc <code>nag_return_multi_normal</code> Pseudo-random multivariate Normal vector from reference vector
L6c	Service routines (e.g., seed)	
		g05cbc <code>nag_random_init_repeatable</code> Initialise random number generating functions to give repeatable sequence
		g05ccc <code>nag_random_init_nonrepeatable</code> Initialise random number generating functions to give non-repeatable sequence
		g05cfc <code>nag_save_random_state</code> Save state of random number generating functions
		g05cgc <code>nag_restore_random_state</code> Restore state of random number generating functions
L7	Analysis of variance (including analysis of covariance)	
L7a	One-way	
L7a1	Parametric	
		g04bbc <code>nag_anova_random</code> General block design or completely randomized design

L7b	Two-way (<i>search also class L7d</i>)
	g04bbc nag_anova_random General block design or completely randomized design
L7d	Multi-way
L7d1	Balanced complete data (e.g., factorial designs)
	g04cac nag_anova_factorial Complete factorial design
L7f	Generate experimental designs
	g02dac nag_regsn_mult_linear Fits a general (multiple) linear regression model
	g02dnc nag_regsn_mult_linear_est_func Estimate of an estimable function for a general linear regression model
L7g	Service routines
	g04czc nag_anova_factorial_free Complete factorial design
L8	Regression (<i>search also classes D5, D6, D9, G, K</i>)
L8a	Simple linear (i.e., $y = b_0 + b_1x$) (<i>search also class L8h</i>)
L8a1	Ordinary least squares
L8a1a	Parameter estimation
L8a1a1	Unweighted data
	g02cac nag_simple_linear_regression Simple linear regression with or without a constant term, data may be weighted
	g02cbc nag_regress_confid_interval Simple linear regression confidence intervals for the regression line and individual points
L8b	Polynomial (e.g., $y = b_0 + b_1x + b_2x^2$) (<i>search also class L8c</i>)
L8b1	Ordinary least squares
L8b1b	Parameter estimation
L8b1b2	Using orthogonal polynomials
	e02adc nag_ld_cheb_fit Computes the coefficients of a Chebyshev series polynomial for arbitrary data
L8c	Multiple linear (i.e. $y = b_0 + b_1x_1 + \dots + b_px_p$)
L8c1	Ordinary least squares
L8c1a	Variable selection
L8c1a1	Using raw data
	g02ddc nag_regsn_mult_linear_upd_model Estimates of regression parameters from an updated model
	g02dec nag_regsn_mult_linear_add_var Add a new independent variable to a general linear regression model
	g02dfc nag_regsn_mult_linear_delete_var Delete an independent variable from a general linear regression model
L8c1b	Parameter estimation (<i>search also class L8c1a</i>)
L8c1b1	Using raw data
	g02dac nag_regsn_mult_linear Fits a general (multiple) linear regression model
	g02dcc nag_regsn_mult_linear_addrem_obs Add/delete an observation to/from a general linear regression model
	g02ddc nag_regsn_mult_linear_upd_model Estimates of regression parameters from an updated model
	g02dec nag_regsn_mult_linear_add_var Add a new independent variable to a general linear regression model
	g02dfc nag_regsn_mult_linear_delete_var Delete an independent variable from a general linear regression model
	g02dkc nag_regsn_mult_linear_tran_model Estimates of parameters of a general linear regression model for given constraints
	g02dnc nag_regsn_mult_linear_est_func Estimate of an estimable function for a general linear regression model
L8c1c	Analysis (<i>search also classes L8c1a and L8c1b</i>)
	g02fac nag_regsn_std_resid_influence Calculate standardized residuals and influence statistics
L8c1d	Inference (<i>search also classes L8c1a and L8c1b</i>)
	g02dnc nag_regsn_mult_linear_est_func Estimate of an estimable function for a general linear regression model
L8c2	Several regressions
	g02dgc nag_regsn_mult_linear_newyvar Fits a general linear regression model to new dependent variable
L8c4	Robust
	g02hac nag_robust_m_regsn_estim Robust regression, standard M -estimates

- L8e** Nonlinear (i.e., $y = F(X, b)$) (*search also class L8h*)
- g02gbc** `nag_glm_binomial`
Fits a generalised linear model with binomial errors
 - g02gcc** `nag_glm_poisson`
Fits a generalised linear model with Poisson errors
 - g02gdc** `nag_glm_gamma`
Fits a generalised linear model with gamma errors
 - g02gkc** `nag_glm_tran_model`
Estimates and standard errors of the parameters of a general linear model for given constraints
 - g02gnc** `nag_glm_est_func`
Estimatable function and the standard error of a generalized linear model
- L8e1** Ordinary least squares
- L8e1b** Parameter estimation (*search also class L8e1a*)
- e04ycc** `nag_opt_lsq_covariance`
Covariance matrix for nonlinear least-squares
 - g02gac** `nag_glm_normal`
Fits a Generalised linear model with Normal errors
- L8e1b1** Unweighted data, user provides no derivatives
- e04fcc** `nag_opt_lsq_no_deriv`
Unconstrained nonlinear least squares (no derivatives required)
 - e04unc** `nag_opt_nlin_lsq`
Solves nonlinear least-squares problems using the sequential QP method
- L8e1b2** Unweighted data, user provides derivatives
- e04gbc** `nag_opt_lsq_deriv`
Unconstrained nonlinear least squares (first derivatives required)
 - e04unc** `nag_opt_nlin_lsq`
Solves nonlinear least-squares problems using the sequential QP method
- L8g** Spline (i.e., piecewise polynomial)
- e02bac** `nag_1d_spline_fit_knots`
Least-squares curve cubic spline fit (including interpolation), one variable
 - e02bec** `nag_1d_spline_fit`
Least-squares cubic spline curve fit, automatic knot placement, one variable
- L8h** EDA (e.g., smoothing)
- g10cac** `nag_running_median_smoother`
Smoothed data sequence using running median smoother
- L9** Categorical data analysis
- L9b** Two-way tables (*search also class L9d*)
- g11aac** `nag_chi_sq_2_way_table`
 χ^2 statistic for two-way contingency table
- L9c** Log-linear model
- g02gcc** `nag_glm_poisson`
Fits a generalised linear model with Poisson errors
 - g02gkc** `nag_glm_tran_model`
Estimates and standard errors of the parameters of a general linear model for given constraints
 - g02gnc** `nag_glm_est_func`
Estimatable function and the standard error of a generalized linear model
- N** Data handling (*search also class L2*)
- N5** Searching
- N5a** Extreme value
- m01fsc** `nag_search_vector`
Searches a vector for either the first or last match to a given value
- N6** Sorting
- N6a** Internal
- N6a1** Passive (i.e. construct pointer array, rank)
- m01dsc** `nag_rank_sort`
Rank sort of set of values of arbitrary data type
- N6a1b** Real
- g01dhc** `nag_ranks_and_scores`
Ranks, Normal scores, approximate Normal scores or exponential (Savage) scores
- N6a2** Active
- m01csc** `nag_quicksort`
Quicksort of set of values of arbitrary data type
 - m01ctc** `nag_stable_sort`
Stable sort of set of values of arbitrary data type
 - m01cuc** `nag_chain_sort`
Chain sort of linked list

N6a2b	Real	m01cac	nag_double_sort Quicksort of set of values of data type double
N8	Permuting	m01esc	nag_reorder_vector Reorders set of values of arbitrary data type into the order specified by a set of indices
		m01zac	nag_make_indices Inverts a permutation converting a rank vector to an index vector or vice versa
R	Service routines	a00aac	nag_implementation_details Library identification, details of implementation and mark
R1	Machine-dependent constants	X01AAC	nag_pi π
		X01ABC	nag_euler_constant Euler's constant, γ
		X02AHC	nag_max_sine_argument Largest permissible argument for <code>sin</code> and <code>cos</code> functions
		X02AJC	nag_machine_precision Machine precision
		X02AKC	nag_real_smallest_number Smallest positive model number
		X02ALC	nag_real_largest_number Largest positive model number
		X02AMC	nag_real_safe_small_number Safe range of floating-point arithmetic
		X02BEC	nag_decimal_digits Maximum number of decimal digits that can be represented
		X02BHC	nag_real_base Parameter b of model of floating-point arithmetic
		X02BJC	nag_real_base_digits Parameter p of model of floating-point arithmetic
		X02BKC	nag_real_min_exponent Parameter $e_{\min}$ of model of floating-point arithmetic
		X02BLC	nag_real_max_exponent Parameter $e_{\max}$ of model of floating-point arithmetic
		X02DAC	nag_underflow_flag Switch for taking precautions to avoid underflow
		X02DJC	nag_real_arithmetic_rounds Parameter ROUNDS of model of floating-point arithmetic

References

- Boisvert R F, Howe S E and Kahaner D K (1990) The guide to available mathematical software problem classification scheme. *Report NISTIR 4475* Applied and Computational Mathematics Division, National Institute of Standards and Technology.
- Boisvert R F, Howe S E and Kahaner D K (1985) GAMS – a framework for the management of scientific software. *ACM Trans. Math. Software* **11** 313-355.
- Boisvert R F (1989) The guide to available mathematical software advisory system. *Math. Comput. Simul.* **31** 453-464.
-