NAG C Library

Mark 6 News

1 Introduction

At Mark 6, the NAG C Library is offering extended coverage in a number of areas, primarily optimization and statistics.

All functions in the NAG C Library conform fully to ANSI C, and functions introduced at this mark take advantage of the `const` keyword to allow for safer code which can be more easily optimised. Although existing functions have not yet been modified, it is our intention to extend the use of `const` and other features of the C99 standard in future releases.

The nonlinear optimizer, nag_opt_nlp (e04ucc), is perhaps the most popular routine in the NAG C Library, with particular use in portfolio management. At Mark 6, we are offering nag_opt_nlp_sparse (e04ugc), which provides the same functionality as nag_opt_nlp (e04ucc) but is for use with large sparse problems.

A set of Time Series routines have been introduced to generate, forecast and estimate Generalized Autoregressive Conditional Heteroskedastic (GARCH) processes. These routines can model univariate regression-GARCH processes of any order and also take into account various types of asymmetry. One of the main applications of GARCH at present is the modelling of financial data with time varying volatility.

In addition we have added a full suite of functions on non-parametric statistics in Chapter g08. Non-parametric statistical analysis is particularly useful when assumptions cannot be made about the nature of the population on which inference is made.

Functions have also been added to Chapters g01 (Simple Calculations on Statistical Data), g02 (Correlation and Regression Analysis), g04 (Analysis of Variance), g11 (Contingency Table Analysis) and g12 (Survival Analysis).

Finally, in cooperation with CERN, we have added a number of special functions to Chapter s, and polynomial solvers for cubic and quartic equations to Chapter c02.

Several function documents have been revised at the mark. In some cases (Chapters d01, e01 and e02) this is to further clarify the use of NAG's memory handling scheme. In other cases we have removed constructions for pre-ANSI compilers or made changes to the examples to provide clearer illustrations of use.

We are also in the process of revising the style of the NAG C Library documentation. The major change is that each of the parameters has been numbered and its type stated explicitly. This change has been applied to new and significantly revised documentation at this mark. The changes, we believe, aid clarity but do not impact in any way on the usage; therefore existing documents will be modified only when other revisions are applied.

All new example programs at Mark 6 are also in a new style, showing typical C language usage; for example, array storage is now allocated dynamically.

2 New Functions

`c02akc`	`nag_cubic_roots` Zeros of a cubic polynomial with real coefficients
`c02alc`	`nag_quartic_roots` Zeros of a real quartic polynomial with real coefficients
`e04ugc`	`nag_opt_nlp_sparse` NLP problem (sparse)
`g01aec`	`nag_frequency_table` Frequency table from raw data
`g01gbc`	`nag_prob_non_central_students_t` Computes probabilities for the non-central Student's t-distribution

g01gcc nag_prob_non_central_chi_sq
Computes probabilities for the non-central χ^2 distribution

g01gdc nag_prob_non_central_f_dist
Computes probabilities for the non-central F-distribution

g01gec nag_prob_non_central_beta_dist
Computes probabilities for the non-central beta distribution

g01hbc nag_multi_normal
Computes probabilities for the multivariate Normal distribution

g02byc nag_partial_corr
Computes partial correlation/variance-covariance matrix from correlation/variance-covariance matrix computed by nag_corr_cov (g02bxc)

g04bcc nag_anova_row_col
Analysis of variance, general row and column design, treatment means and standard errors

g04dbc nag_anova_confid_interval
Computes confidence intervals for differences between means computed by nag_anova_random (g04bbc) or nag_anova_row_col (g04bcc)

g04eac nag_dummy_vars
Computes orthogonal polynomials or dummy variables for factor/classification variable

g05hkc nag_generate_agarchI
Univariate time series, generate n terms of either a symmetric GARCH process or a GARCH process with asymmetry of the form $(\epsilon_{t-1} + \gamma)^2$

g05hlc nag_generate_agarchII
Univariate time series, generate n terms of a GARCH process with asymmetry of the form $(|\epsilon_{t-1}| + \gamma\epsilon_{t-1})^2$

g05hmc nag_generate_garchGJR
Univariate time series, generate n terms of an asymmetric Glosten, Jagannathan and Runkle (GJR) GARCH process

g08aac nag_sign_test
Sign test on two paired samples

g08acc nag_median_test
Median test on two samples of unequal size

g08aec nag_friedman_test
Friedman two-way analysis of variance on k matched samples

g08afc nag_kruskal_wallis_test
Kruskal–Wallis one-way analysis of variance on k samples of unequal size

g08agc nag_wilcoxon_test
Performs the Wilcoxon one-sample (matched pairs) signed rank test

g08amc nag_mann_whitney
Performs the Mann–Whitney U test on two independent samples

g08cbc nag_1_sample_ks_test
Performs the one-sample Kolmogorov–Smirnov test for standard distributions

g08cdc nag_2_sample_ks_test
Performs the two-sample Kolmogorov–Smirnov test

g08cgc nag_chi_sq_goodness_of_fit_test
Performs the χ^2 goodness of fit test, for standard continuous distributions

g08eac nag_runs_test
Performs the runs up or runs down test for randomness

g08ebc nag_pairs_test
Performs the pairs (serial) test for randomness

g08ecc nag_triplets_test
Performs the triplets test for randomness

g08edc nag_gaps_test
Performs the gaps test for randomness

g10abc nag_smooth_spline_fit
Fit cubic smoothing spline, smoothing parameter given

g10acc nag_smooth_spline_estim
Fit cubic smoothing spline, smoothing parameter estimated

`g10bac` `nag_kernel_density_estim`
Kernel density estimate using Gaussian kernel

`g10zac` `nag_order_data`
Reorder data to give ordered distinct observations

`g11bac` `nag_tabulate_stats`
Computes multiway table from set of classification factors using selected statistic

`g11bbc` `nag_tabulate_percentile`
Computes multiway table from set of classification factors using given percentile/quantile

`g12bac` `nag_surviv_cox_model`
Fits Cox's proportional hazard model

`g13asc` `nag_tsa_resid_corr`
Univariate time series, diagnostic checking of residuals, following
nag_tsa_multi_inp_model_estim (g13bec)

`g13fac` `nag_estimate_agarchI`
Univariate time series, parameter estimation for either a symmetric GARCH process or a GARCH process with asymmetry of the form $(\epsilon_{t-1}+\gamma)^2$

`g13fbc` `nag_forecast_agarchI`
Univariate time series, forecast function for either a symmetric GARCH process or a GARCH process with asymmetry of the form $(\epsilon_{t-1}+\gamma)^2$

`g13fcc` `nag_estimate_agarchII`
Univariate time series, parameter estimation for a GARCH process with asymmetry of the form $(|\epsilon_{t-1}|+\gamma\epsilon_{t-1})^2$

`g13fdc` `nag_forecast_agarchII`
Univariate time series, forecast function for a GARCH process with asymmetry of the form $(|\epsilon_{t-1}|+\gamma\epsilon_{t-1})^2$

`g13fec` `nag_estimate_garchGJR`
Univariate time series, parameter estimation for an asymmetric Glosten, Jagannathan and Runkle (GJR) GARCH process

`g13ffc` `nag_forecast_garchGJR`
Univariate time series, forecast function for an asymmetric Glosten, Jagannathan and Runkle (GJR) GARCH process

`s14aec` `nag_real_polygamma`
Derivative of the psi function $\psi(x)$

`s14afc` `nag_complex_polygamma`
Derivative of the psi function $\psi(z)$

`s17alc` `nag_bessel_zeros`
Zeros of Bessel functions $J_\alpha(x)$, $J'_\alpha(x)$, $Y_\alpha(x)$ or $Y'_\alpha(x)$

`s18ecc` `nag_bessel_i_nu_scaled`
Scaled modified Bessel function $e^{-x}I_{\nu/4}(x)$

`s18edc` `nag_bessel_k_nu_scaled`
Scaled modified Bessel function $e^{x}K_{\nu/4}(x)$

`s18eec` `nag_bessel_i_nu`
Modified Bessel function $I_{\nu/4}(x)$

`s18efc` `nag_bessel_k_nu`
Modified Bessel function $K_{\nu/4}(x)$

`s18egc` `nag_bessel_k_alpha`
Modified Bessel functions $K_{\alpha+n}(x)$ for real $x>0$, selected values of $\alpha\geq 0$ and $n=0,1,\ldots,N$

`s18ehc` `nag_bessel_k_alpha_scaled`
Scaled modified Bessel functions $e^{x}K_{\alpha+n}(x)$ for real $x>0$, selected values of $\alpha\geq 0$ and $n=0,1,\ldots,N$

`s18ejc` `nag_bessel_i_alpha`
Modified Bessel functions $I_{\alpha+n-1}(x)$ or $I_{\alpha-n+1}(x)$ for real $x\neq 0$, non-negative $\alpha<1$ and $n=1,2,\ldots,|N|+1$

`s18ekc` `nag_bessel_j_alpha`
Bessel functions $J_{\alpha+n-1}(x)$ or $J_{\alpha-n+1}(x)$ for real $x\neq 0$, non-negative $\alpha<1$ and $n=1,2,\ldots,|N|+1$

`s21cbc` `nag_jacobian_elliptic`
Jacobian elliptic functions sn, cn and dn with complex arguments

`s21ccc` `nag_jacobian_theta`
Jacobian theta functions with real arguments

`s21dac` `nag_elliptic_integral_f`
Elliptic integrals of the second kind with complex arguments

`s22aac` `nag_legendre_p`
Legendre and associated Legendre functions of the first kind with real arguments

3 Functions Withdrawn or Scheduled for Withdrawal

The functions listed below are scheduled for withdrawal from the NAG C Library, because improved functions have now been included in the Library. Users are advised to stop using functions which are scheduled for withdrawal and to use recommended replacement functions instead.

The following functions will be withdrawn at Mark 8.

Function Scheduled for Withdrawal	**Recommended Replacement**
`e04jbc`	`nag_opt_nlp (e04ucc)`
`e04kbc`	`nag_opt_nlp (e04ucc)`