1 Introduction

In this document we describe the differences between the last release ‘Release 2’ and this latest release, now named ‘Mark 21’, of the NAG SMP Library. There are two main differences:

(i) Release 2 was based on Mark 19 of the NAG Fortran Library while this latest release is based on Mark 21 of the NAG Fortran Library; and

(ii) in Mark 21, a larger number of routines have been specially tuned to maximize their performance on shared memory parallel systems.

1.1 New Functionality

At Mark 21 of the NAG SMP Library new functionality has been introduced in addition to improvements in existing areas. The Library now contains 1533 user-callable routines, all of which are documented, and 375 are new at this mark.

New chapters on large scale eigenproblems and mesh generation have been introduced, and extensions have been included in the areas of optimization, dense and banded linear algebra, direct solution of large scale linear systems, simple statistical calculations, regression, random numbers, zeros of polynomials, partial differential equations, time series analysis, and special functions.

In addition the provision of thread safe versions of existing routines has been significantly extended in Chapter C05 (Roots of One or More Transcendental Equations), Chapter D03 (Partial Differential Equations), Chapter E04 (Minimizing or Maximizing a Function) and Chapter G05 (Random Number Generators) to aid users developing multithreaded applications.

The new Chapter D06 (Mesh Generation) has routines for generating 2-D meshes together with a number of associated utility routines and the new Chapter F12 (Large Scale Eigenproblems) has routines for the solution of symmetric and nonsymmetric standard and generalized large scale eigenvalue problems. Chapter F11 has been renamed as Large Scale Linear Systems, and new routines for the direct solution of sparse problems have been added.

Chapter E04 (Minimizing or Maximizing a Function) has been updated with new routines for the solution of LP, QP and nonlinear programming problems with sparse linear constraints.

Chapter F07 (Linear Equations (LAPACK)) and Chapter F08 (Least-squares and Eigenvalue Problems (LAPACK)) have been extended to include all the LAPACK driver routines, thus allowing the solution of most problems with a call to a single routine rather than multiple calls to LAPACK computational routines. A comprehensive suite of driver routines for the solution of dense and banded linear equations has also been added to Chapter F04 (Simultaneous Linear Equations).

Routines for finding the roots of real and complex cubic and quartic equations have been added to Chapter C02 (Zeros of Polynomials).

Chapter D03 (Partial Differential Equations) now includes routines for solving Black–Scholes equations.

Routines for Landau and Vavilov distributions have been added to Chapter G01 (Simple Calculations on Statistical Data), new routines for stepwise regression and mixed effects regression have been included in Chapter G02 (Correlation and Regression Analysis), and a number of new random number generators, including Copulas and improved quasi-random number generators, generation of univariate GARCH, asymmetric GARCH and EGARCH processes, and generators for further distributions have been added to Chapter G05 (Random Number Generators).

Chapter G13 (Time Series Analysis) has been extended with routines for parameter estimation and forecasting for univariate regression GARCH, asymmetric GARCH and EGARCH processes.

Chapter S (Approximations of Special Functions) has new routines for polygamma functions, zeros of Bessel functions, Jacobian functions, elliptic integrals and Legendre and associated Legendre functions. Variant routines for the log Gamma function, and Bessel function of the 1st kind have also been added.
1.2 New SMP parallelism and other optimizations

SMP Parallel versions of many of the new Chapter F11 sparse direct solvers, and Chapter F12 sparse
eigenproblem are included, and parallelism has been extended to many more LAPACK routines. In
addition, improvements in underlying NAG auxiliary routines improves the usage of the SMP optimized
dense linear algebra by routines in the areas of optimization, ODEs and PDEs, and statistics.

1.3 Documentation

The NAG Fortran Library Manual has undergone a fundamental change since Release 2 and the Essential
Introduction and the Introduction to the NAG SMP Library are essential reading for all users of the NAG
Fortran Library.

2 Tuned Routines

The following is a list of user-callable routines that have been parallelized, or otherwise optimized, since
the last release. There are 67 of these routines at this release in the area of dense and sparse linear algebra.
See the document ‘Tuned and Enhanced Routines in the NAG SMP Library’ for a full list of tuned
routines.

Note: on some implementations, the equivalent vendor library routines may be substituted for some of the
following list – consult the Users’ Note for your implementation for further information.

- F04AFF Solution of real symmetric positive-definite simultaneous linear equations using iterative
  refinement (coefficient matrix already factorized by F03AEF)
- F04AGF Solution of real symmetric positive-definite simultaneous linear equations (coefficient matrix
  already factorized by F03AEF)
- F04AHF Solution of real simultaneous linear equations using iterative refinement (coefficient matrix
  already factorized by F03AFF)
- F04AJF Solution of real simultaneous linear equations (coefficient matrix already factorized by
  F03AFF)
- F07AHF Refined solution with error bounds of real system of linear equations, multiple right-hand sides
- F07AVF Refined solution with error bounds of complex system of linear equations, multiple right-hand sides
- F07BDF LU factorization of real $m$ by $n$ band matrix
- F07BEF Solution of real band system of linear equations, multiple right-hand sides, matrix already
  factorized by F07BDF (DGBTRF)
- F07BHF Refined solution with error bounds of real band system of linear equations, multiple right-hand sides
- F07BRF $LU$ factorization of complex $m$ by $n$ band matrix
- F07BSF Solution of complex band system of linear equations, multiple right-hand sides, matrix already
  factorized by F07BRF (ZGBTRF)
- F07BVF Refined solution with error bounds of complex band system of linear equations, multiple right-hand
  sides
- F07CHF Refined solution with error bounds of real tridiagonal system of linear equations, multiple
  right-hand sides
- F07CVF Refined solution with error bounds of complex tridiagonal system of linear equations, multiple
  right-hand sides
- F07FHF Refined solution with error bounds of real symmetric positive-definite system of linear
  equations, multiple right-hand sides
- F07FVF Refined solution with error bounds of complex Hermitian positive-definite system of linear
  equations, multiple right-hand sides
- F07GEF Solution of real symmetric positive-definite system of linear equations, multiple right-hand sides,
  matrix already factorized by F07GDF (DPPTRF), packed storage
- F07GHF Refined solution with error bounds of real symmetric positive-definite system of linear
  equations, multiple right-hand sides, packed storage
- F07GSF Solution of complex Hermitian positive-definite system of linear equations, multiple right-hand
  sides, matrix already factorized by F07GRF (ZPPTRF), packed storage
- F07GVF Refined solution with error bounds of complex Hermitian positive-definite system of linear
  equations, multiple right-hand sides, packed storage
<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>F07HEF</td>
</tr>
<tr>
<td>F07HHF</td>
</tr>
<tr>
<td>F07HSF</td>
</tr>
<tr>
<td>F07HVF</td>
</tr>
<tr>
<td>F07JHF</td>
</tr>
<tr>
<td>F07JVF</td>
</tr>
<tr>
<td>F07MHF</td>
</tr>
<tr>
<td>F07MVF</td>
</tr>
<tr>
<td>F07PHF</td>
</tr>
<tr>
<td>F07PVF</td>
</tr>
<tr>
<td>F07QVF</td>
</tr>
<tr>
<td>F07THF</td>
</tr>
<tr>
<td>F07TVF</td>
</tr>
<tr>
<td>F07UEF</td>
</tr>
<tr>
<td>F07UHF</td>
</tr>
<tr>
<td>F07USF</td>
</tr>
<tr>
<td>F07UVF</td>
</tr>
<tr>
<td>F07VEF</td>
</tr>
<tr>
<td>F07VHF</td>
</tr>
<tr>
<td>F07VVF</td>
</tr>
<tr>
<td>F07VVF</td>
</tr>
<tr>
<td>F08HEF</td>
</tr>
<tr>
<td>F08HSF</td>
</tr>
<tr>
<td>F08JRF</td>
</tr>
<tr>
<td>F08JKF</td>
</tr>
<tr>
<td>F08JXF</td>
</tr>
<tr>
<td>F08PEF</td>
</tr>
<tr>
<td>F08PKF</td>
</tr>
<tr>
<td>F08PSF</td>
</tr>
<tr>
<td>F08PXF</td>
</tr>
<tr>
<td>F08TAF</td>
</tr>
<tr>
<td>F08TBF</td>
</tr>
</tbody>
</table>
3 Enhanced Routines

These routines call one or more of the tuned routines as part of their core operations and may thereby exhibit improved performance and scalability. There are 159 newly enhanced routines at this mark; these include the areas of zeros of polynomials, PDEs, optimization, dense and sparse linear algebra, and multivariate statistics.

C02AKF All zeros of real cubic equation
C02ALF All zeros of real quartic equation
C02AMF All zeros of complex cubic equation
C02ANF All zeros of complex quartic equation
D03FAF Elliptic PDE, Helmholtz equation, three-dimensional Cartesian co-ordinates
D03NCF Finite difference solution of the Black–Scholes equations
E02RAF Padé approximants
E04USF Minimum of a sum of squares, nonlinear constraints, sequential QP method, using function values and optionally first derivatives (comprehensive)
E04YCF Covariance matrix for nonlinear least-squares problem (unconstrained)
F02ECF Selected eigenvalues and eigenvectors of real nonsymmetric matrix (Black Box)
F02WDF QR factorization, possibly followed by SVD
F02XUF SVD of complex upper triangular matrix (Black Box)
F04AEF Solution of real simultaneous linear equations with multiple right-hand sides using iterative refinement (Black Box)
F04ATF Solution of real simultaneous linear equations, one right-hand side using iterative refinement (Black Box)
F04RAF Computes the solution and error-bound to a real system of linear equations
F04BBF Computes the solution and error-bound to a real banded system of linear equations
F04BDF Computes the solution and error-bound to a real symmetric positive-definite system of linear equations
F04BEF Computes the solution and error-bound to a real symmetric positive-definite banded system of linear equations
F04BFF Computes the solution and error-bound to a real symmetric positive-definite banded system of linear equations
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F04CAF</td>
<td>Computes the solution and error-bound to a complex system of linear equations</td>
</tr>
<tr>
<td>F04CBF</td>
<td>Computes the solution and error-bound to a complex banded system of linear equations</td>
</tr>
<tr>
<td>F04CDF</td>
<td>Computes the solution and error-bound to a complex Hermitian positive-definite system of</td>
</tr>
<tr>
<td></td>
<td>linear equations</td>
</tr>
<tr>
<td>F04CEF</td>
<td>Computes the solution and error-bound to a complex Hermitian positive-definite system of</td>
</tr>
<tr>
<td></td>
<td>linear equations, packed storage</td>
</tr>
<tr>
<td>F04CF2</td>
<td>Computes the solution and error-bound to a complex Hermitian positive-definite banded</td>
</tr>
<tr>
<td></td>
<td>system of linear equations</td>
</tr>
<tr>
<td>F04JGF</td>
<td>Computes the solution to a real system of linear equations</td>
</tr>
<tr>
<td></td>
<td>Least-squares (if rank = n) or minimal least-squares (if rank &lt; n) solution of m real</td>
</tr>
<tr>
<td></td>
<td>equations in n unknowns, rank ≤ n, m ≥ n</td>
</tr>
<tr>
<td>F07AAF</td>
<td>Computes the solution to a real system of linear equations</td>
</tr>
<tr>
<td></td>
<td>Uses the LU factorization to compute the solution, error-bound and condition estimate for a</td>
</tr>
<tr>
<td></td>
<td>real system of linear equations</td>
</tr>
<tr>
<td>F07ANF</td>
<td>Computes the solution to a complex system of linear equations</td>
</tr>
<tr>
<td>F07APF</td>
<td>Uses the LU factorization to compute the solution, error-bound and condition estimate for a</td>
</tr>
<tr>
<td></td>
<td>complex system of linear equations</td>
</tr>
<tr>
<td>F07BAY</td>
<td>Computes the solution to a real banded system of linear equations</td>
</tr>
<tr>
<td>F07BBF</td>
<td>Uses the LU factorization to compute the solution, error-bound and condition estimate for a</td>
</tr>
<tr>
<td></td>
<td>real banded system of linear equations</td>
</tr>
<tr>
<td>F07BNF</td>
<td>Computes the solution to a complex banded system of linear equations</td>
</tr>
<tr>
<td>F07BPF</td>
<td>Uses the LU factorization to compute the solution, error-bound and condition estimate for a</td>
</tr>
<tr>
<td></td>
<td>complex banded system of linear equations</td>
</tr>
<tr>
<td>F07CBF</td>
<td>Uses the LU factorization to compute the solution, error-bound and condition estimate for a</td>
</tr>
<tr>
<td></td>
<td>complex tridiagonal system of linear equations</td>
</tr>
<tr>
<td>F07CPF</td>
<td>Uses the LU factorization to compute the solution, error-bound and condition estimate for a</td>
</tr>
<tr>
<td></td>
<td>complex tridiagonal system of linear equations</td>
</tr>
<tr>
<td>F07FAF</td>
<td>Computes the solution to a real symmetric positive-definite system of linear equations</td>
</tr>
<tr>
<td>F07FBF</td>
<td>Uses the Cholesky factorization to compute the solution, error-bound and condition estimate</td>
</tr>
<tr>
<td></td>
<td>for a real symmetric positive-definite system of linear equations</td>
</tr>
<tr>
<td>F07FNF</td>
<td>Computes the solution to a complex Hermitian positive-definite system of linear equations</td>
</tr>
<tr>
<td>F07FPF</td>
<td>Uses the Cholesky factorization to compute the solution, error-bound and condition estimate</td>
</tr>
<tr>
<td></td>
<td>for a complex Hermitian positive-definite system of linear equations</td>
</tr>
<tr>
<td>F07GAF</td>
<td>Computes the solution to a real symmetric positive-definite system of linear equations,</td>
</tr>
<tr>
<td></td>
<td>packed storage</td>
</tr>
<tr>
<td>F07GBF</td>
<td>Uses the Cholesky factorization to compute the solution, error-bound and condition estimate</td>
</tr>
<tr>
<td></td>
<td>for a real symmetric positive-definite system of linear equations, packed storage</td>
</tr>
<tr>
<td>F07GNF</td>
<td>Uses the solution to a complex Hermitian positive-definite system of linear equations,</td>
</tr>
<tr>
<td></td>
<td>packed storage</td>
</tr>
<tr>
<td>F07GPF</td>
<td>Uses the Cholesky factorization to compute the solution, error-bound and condition estimate</td>
</tr>
<tr>
<td></td>
<td>for a complex Hermitian positive-definite system of linear equations, packed storage</td>
</tr>
<tr>
<td>F07HAF</td>
<td>Computes the solution to a real symmetric positive-definite banded system of linear equations</td>
</tr>
<tr>
<td>F07HBF</td>
<td>Uses the Cholesky factorization to compute the solution, error-bound and condition estimate</td>
</tr>
<tr>
<td></td>
<td>for a real symmetric positive-definite banded system of linear equations</td>
</tr>
<tr>
<td>F07HNF</td>
<td>Computes the solution to a complex Hermitian positive-definite banded system of linear</td>
</tr>
<tr>
<td></td>
<td>equations</td>
</tr>
<tr>
<td>F07HPF</td>
<td>Uses the Cholesky factorization to compute the solution, error-bound and condition estimate</td>
</tr>
<tr>
<td></td>
<td>for a complex Hermitian positive-definite banded system of linear equations</td>
</tr>
<tr>
<td>F07JBF</td>
<td>Uses the modified Cholesky factorization to compute the solution, error-bound and condition</td>
</tr>
<tr>
<td></td>
<td>estimate for a real symmetric positive-definite tridiagonal system of linear equations</td>
</tr>
<tr>
<td>F07JPF</td>
<td>Uses the modified Cholesky factorization to compute the solution, error-bound and condition</td>
</tr>
<tr>
<td></td>
<td>estimate for a complex Hermitian positive-definite tridiagonal system of linear equations</td>
</tr>
<tr>
<td>F07MBF</td>
<td>Uses the diagonal pivoting factorization to compute the solution to a real symmetric system</td>
</tr>
<tr>
<td></td>
<td>of linear equations</td>
</tr>
<tr>
<td>F07MPF</td>
<td>Uses the diagonal pivoting factorization to compute the solution to a complex Hermitian</td>
</tr>
<tr>
<td></td>
<td>system of linear equations</td>
</tr>
<tr>
<td>F07NPF</td>
<td>Uses the diagonal pivoting factorization to compute the solution to a complex symmetric</td>
</tr>
<tr>
<td></td>
<td>system of linear equations</td>
</tr>
<tr>
<td>F07PBF</td>
<td>Uses the diagonal pivoting factorization to compute the solution to a real symmetric system</td>
</tr>
</tbody>
</table>
F07PPF Uses the diagonal pivoting factorization to compute the solution to a complex Hermitian system of linear equations, packed storage
F07QPF Uses the diagonal pivoting factorization to compute the solution to a complex symmetric system of linear equations, packed storage
F08AAF Solves an overdetermined or underdetermined real linear system
F08ANF Solves an overdetermined or underdetermined complex linear system
F08BAF Computes the minimum-norm solution to a real linear least-squares problem
F08BFF QR factorization of real general rectangular matrix with column pivoting, using BLAS-3
F08BNF Computes the minimum-norm solution to a complex linear least-squares problem
F08BTF QR factorization of complex general rectangular matrix with column pivoting, using BLAS-3
F08FAF Computes all eigenvalues and, optionally, eigenvectors of a real symmetric matrix
F08FBF Computes selected eigenvalues and, optionally, eigenvectors of a real symmetric matrix
F08FDF Computes selected eigenvalues and, optionally, eigenvectors of a real symmetric matrix (Relatively Robust Representations)
F08FNF Computes all eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix
F08FPF Computes selected eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix
F08FRF Computes selected eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix (Relatively Robust Representations)
F08GAF Computes all eigenvalues and, optionally, eigenvectors of a real symmetric matrix, packed storage
F08GBF Computes selected eigenvalues and, optionally, eigenvectors of a real symmetric matrix, packed storage
F08GNF Computes all eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix, packed storage
F08GPF Computes selected eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix, packed storage
F08HAF Computes all eigenvalues and, optionally, eigenvectors of a real symmetric band matrix
F08HBF Computes selected eigenvalues and, optionally, eigenvectors of a real symmetric band matrix
F08HNF Computes all eigenvalues and, optionally, eigenvectors of a complex Hermitian band matrix
F08HPF Computes selected eigenvalues and, optionally, eigenvectors of a complex Hermitian band matrix
F08JAF Computes all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix
F08JBF Computes selected eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix
F08JDF Computes selected eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix (Relatively Robust Representations)
F08JHF Computes all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix or a matrix reduced to this form (divide-and-conquer)
F08JLF Computes all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix or a symmetric matrix reduced to this form (Relatively Robust Representations)
F08JVF Computes all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix or a complex Hermitian matrix reduced to this form (divide-and-conquer)
F08JYF Computes all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix or a complex Hermitian matrix reduced to this form (Relatively Robust Representations)
F08KAF Computes the minimum-norm solution to a real linear least-squares problem using singular value decomposition
F08KBF Computes the singular value decomposition of a real matrix, optionally computing the left and/or right singular vectors
F08KCF Computes the minimum-norm solution to a real linear least-squares problem using singular value decomposition (divide-and-conquer)
F08KDF Computes the singular value decomposition of a real matrix, optionally computing the left and/or right singular vectors (divide-and-conquer)
F08KNF Computes the minimum-norm solution to a complex linear least-squares problem using singular value decomposition
F08KPF Computes the singular value decomposition of a complex matrix, optionally computing the left and/or right singular vectors
F08KQF Computes the minimum-norm solution to a complex linear least-squares problem using singular value decomposition (divide-and-conquer)
F08KRF Computes the singular value decomposition of a complex matrix, optionally computing the left and/or right singular vectors (divide-and-conquer)

F08MDF Computes the singular value decomposition of a real bidiagonal matrix, optionally computing the singular vectors (divide-and-conquer)

F08NAF Computes all eigenvalues and, optionally, left and/or right eigenvectors of a real nonsymmetric matrix

F08NBF Computes all eigenvalues and, optionally, left and/or right eigenvectors of a real nonsymmetric matrix; also, optionally, the balancing transformation, the reciprocal condition numbers for the eigenvalues and for the right eigenvectors

F08NNF Computes all eigenvalues and, optionally, left and/or right eigenvectors of a complex nonsymmetric matrix

F08NPF Computes all eigenvalues and, optionally, left and/or right eigenvectors of a complex nonsymmetric matrix; also, optionally, the balancing transformation, the reciprocal condition numbers for the eigenvalues and for the right eigenvectors

F08PAF Computes for real square nonsymmetric matrix, the eigenvalues, the real Schur form, and, optionally, the matrix of Schur vectors

F08PBF Computes for real square nonsymmetric matrix, the eigenvalues, the real Schur form, and, optionally, the matrix of Schur vectors; also, optionally, computes reciprocal condition numbers for selected eigenvalues

F08PNF Computes for complex square nonsymmetric matrix, the eigenvalues, the Schur form, and, optionally, the matrix of Schur vectors

F08PPF Computes for real square nonsymmetric matrix, the eigenvalues, the Schur form, and, optionally, the matrix of Schur vectors; also, optionally, computes reciprocal condition numbers for selected eigenvalues

F08SAF Computes all the eigenvalues, and optionally, the eigenvectors of a real generalized symmetric-definite eigenproblem

F08SBF Computes selected eigenvalues, and optionally, the eigenvectors of a real generalized symmetric-definite eigenproblem

F08SCF Computes all the eigenvalues, and optionally, the eigenvectors of a real generalized symmetric-definite eigenproblem (divide-and-conquer)

F08SNF Computes all the eigenvalues, and optionally, the eigenvectors of a complex generalized Hermitian-definite eigenproblem

F08SPF Computes selected eigenvalues, and optionally, the eigenvectors of a complex generalized Hermitian-definite eigenproblem

F08SQF Computes all the eigenvalues, and optionally, the eigenvectors of a complex generalized Hermitian-definite eigenproblem (divide-and-conquer)

F08UAF Computes all the eigenvalues, and optionally, the eigenvectors of a real banded generalized symmetric-definite eigenproblem

F08UBF Computes selected eigenvalues, and optionally, the eigenvectors of a real banded generalized symmetric-definite eigenproblem

F08UCF Computes all the eigenvalues, and optionally, the eigenvectors of a real banded generalized symmetric-definite eigenproblem (divide-and-conquer)

F08UNF Computes all the eigenvalues, and optionally, the eigenvectors of a complex banded generalized Hermitian-definite eigenproblem

F08UPF Computes selected eigenvalues, and optionally, the eigenvectors of a complex banded generalized Hermitian-definite eigenproblem

F08UQF Computes all the eigenvalues, and optionally, the eigenvectors of a complex banded generalized Hermitian-definite eigenproblem (divide-and-conquer)

F08WAF Computes, for a real nonsymmetric matrix pair, the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors

F08WBF Computes, for a real nonsymmetric matrix pair, the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors; also, optionally, the balancing transformation, the reciprocal condition numbers for the eigenvalues and for the right eigenvectors

F08WNF Computes, for a complex nonsymmetric matrix pair, the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors

F08WPF Computes, for a complex nonsymmetric matrix pair, the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors; also, optionally, the balancing transformation, the reciprocal condition numbers for the eigenvalues and for the right eigenvectors
F08XAF Computes, for a real nonsymmetric matrix pair, the generalized eigenvalues, the generalized
real Schur form and, optionally, the left and/or right matrices of Schur vectors
F08XBF Computes, for a real nonsymmetric matrix pair, the generalized eigenvalues, the generalized
real Schur form and, optionally, the left and/or right matrices of Schur vectors; also, optionally,
computes reciprocal condition numbers for selected eigenvalues
F08XNF Computes, for a complex nonsymmetric matrix pair, the generalized eigenvalues, the
generalized complex Schur form and, optionally, the left and/or right matrices of Schur vectors
F08XPF Computes, for a complex nonsymmetric matrix pair, the generalized eigenvalues, the
generalized complex Schur form and, optionally, the left and/or right matrices of Schur vectors;
also, optionally, computes reciprocal condition numbers for selected eigenvalues
F08ZAF Solves the real linear equality-constrained least-squares (LSE) problem
F08ZBF Solves a real general Gauss–Markov linear model (GLM) problem
F08ZEF Computes a generalized QR factorization of a real matrix pair
F08ZFF Computes a generalized RQ factorization of a real matrix pair
F08ZNF Solves the complex linear equality-constrained least-squares (LSE) problem
F08ZPF Solves a complex general Gauss–Markov linear model (GLM) problem
F08ZSF Computes a generalized QR factorization of a complex matrix pair
F08ZTF Computes a generalized RQ factorization of a complex matrix pair
F11DCF Solution of real sparse nonsymmetric linear system, RGMRES, CGS, Bi-CGSTAB or TFQMR
method, preconditioner computed by F11DAF
F11DEF Solution of real sparse nonsymmetric linear system, RGMRES, CGS, Bi-CGSTAB, or
TFQMR method, Jacobi or SSOR preconditioner (Black Box)
F11DKF Real sparse nonsymmetric linear systems, line Jacobi preconditioner
F11DQF Solution of complex sparse non-Hermitian linear system, RGMRES, CGS, Bi-CGSTAB or
TFQMR method, preconditioner computed by F11DNF (Black Box)
F11DSF Solution of complex sparse non-Hermitian linear system, RGMRES, CGS, Bi-CGSTAB or
TFQMR method, Jacobi or SSOR preconditioner Black Box
F11DXF Complex sparse nonsymmetric linear systems, line Jacobi preconditioner
F11JCF Solution of real sparse symmetric linear system, conjugate gradient/Lanczos method,
preconditioner computed by F11JAF (Black Box)
F11JEF Solution of real sparse symmetric linear system, conjugate gradient/Lanczos method, Jacobi or
SSOR preconditioner (Black Box)
F11JQF Solution of complex sparse Hermitian linear system, conjugate gradient/Lanczos method,
preconditioner computed by F11JNF (Black Box)
F11JSF Solution of complex sparse Hermitian linear system, conjugate gradient/Lanczos method,
Jacobi or SSOR preconditioner (Black Box)
F12FCF Returns the converged approximations (as determined by F12ABF) to eigenvalues of a real
symmetric sparse (standard or generalized) eigenproblem and, optionally, the corresponding
approximate eigenvectors and/or an orthonormal basis for the associated approximate invariant
subspace
F12FGF Computes approximations to selected eigenvalues of a real symmetric banded (standard or
generalized) eigenproblem and, optionally, the corresponding approximate eigenvectors and/or
an orthonormal basis for the associated approximate invariant subspace
G01HBF Computes probabilities for the multivariate Normal distribution
G02BYF Computes partial correlation/variance-covariance matrix from correlation/variance-covariance
matrix computed by G02BXF
G02DDF Estimates of linear parameters and general linear regression model from updated model
G02DKF Estimates and standard errors of parameters of a general linear regression model for given
constraints
G02GKF Estimates and standard errors of parameters of a general linear model for given constraints
G02HDF Robust regression, compute regression with user-supplied functions and weights
G02JAF Linear mixed effects regression using Restricted Maximum Likelihood (REML)
G03FAF Performs principal co-ordinate analysis, classical metric scaling
G05PCF Generates a realisation of a multivariate time series from a VARMA model
G11CAF Returns parameter estimates for the conditional analysis of stratified data
G12BAF Fits Cox’s proportional hazard model
G13ADF Univariate time series, preliminary estimation, seasonal ARIMA model
G13DXF Calculates the zeros of a vector autoregressive (or moving average) operator
Introduction

Mark 21 News – NAG Fortran SMP Library

G13FAF Univariate time series, parameter estimation for either a symmetric GARCH process or a GARCH process with asymmetry of the form \((\epsilon_{t-1} + \gamma)^2\).

G13FCF Univariate time series, parameter estimation for a GARCH process with asymmetry of the form \((|\epsilon_{t-1}| + \gamma\epsilon_{t-1})^2\).

G13FEF Univariate time series, parameter estimation for an asymmetric Glosten, Jagannathan and Runkle (GJR) GARCH process.

G13FGF Univariate time series, parameter estimation for an exponential GARCH (EGARCH) process.

4 New Routines

The 375 new user-callable routines included in the SMP Library at Mark 21 are as follows.

A00ACF Check availability of a valid licence key
C02AKF All zeros of real cubic equation
C02ALF All zeros of real quartic equation
C02AMF All zeros of complex cubic equation
C02ANF All zeros of complex quartic equation
D03NCF Finite difference solution of the Black–Scholes equations
D03NDF Analytic solution of the Black–Scholes equations
D03NEF Compute average values for D03NDF
D06AAF Generates a two-dimensional mesh using a simple incremental method
D06ABF Generates a two-dimensional mesh using a Delaunay–Voronoï process
D06ACF Generates a two-dimensional mesh using an Advancing-front method
D06BAF Generates a boundary mesh
D06CAF Uses a barycentric technique to smooth a given mesh
D06CBF Generates a sparsity pattern of a Finite Element matrix associated with a given mesh
D06CCF Renumbers a given mesh using Gibbs method
D06DAF Generates a mesh resulting from an affine transformation of a given mesh
D06DBF Joins together two given adjacent (possibly overlapping) meshes
E04NPF Initialization routine for E04NQF
E04NQF LP or QP problem (suitable for sparse problems)
E04NRF Supply optional parameter values for E04NQF from external file
E04NSF Set a single option for E04NQF from a character string
E04NTF Set a single option for E04NQF from an INTEGER argument
E04NUF Set a single option for E04NQF from a \texttt{double precision} argument
E04NYF Get the setting of a \texttt{double precision} valued option of E04NQF
E04USF Minimum of a sum of squares, nonlinear constraints, sequential QP method, using function values and optionally first derivatives (comprehensive)
E04VGF Initialization routine for E04VHF
E04VHF General sparse nonlinear optimizer
E04VJF Determine the pattern of nonzeros in the Jacobian matrix for E04VHF
E04VKF Supply optional parameter values for E04VHF from external file
E04VLF Set a single option for E04VHF from a character string
E04VMF Set a single option for E04VHF from an INTEGER argument
E04VNF Set a single option for E04VHF from a \texttt{double precision} argument
E04VRF Get the setting of an INTEGER valued option of E04VHF
E04VSF Get the setting of a \texttt{double precision} valued option of E04VHF
E04WBF Initialization routine for E04DGA E04MFA E04NCA E04NFA E04UFA E04UGA E04USA
E04WCF Initialization routine for E04WDF
E04WDF Solves the nonlinear programming (NP) problem
E04WEF Supply optional parameter values for E04WDF from external file
E04WFF Set a single option for E04WDF from a character string
E04WGF Set a single option for E04WDF from an INTEGER argument
E04WHF Set a single option for E04WDF from a \texttt{double precision} argument
E04WJF Determine whether an E04WDF option has been set or not
E04WKF Get the setting of an INTEGER valued option of E04WDF
E04WLF Get the setting of a \texttt{double precision} valued option of E04WDF
F04BAF Computes the solution and error-bound to a real system of linear equations
F04BBF Computes the solution and error-bound to a real banded system of linear equations
F04BCF Computes the solution and error-bound to a real tridiagonal system of linear equations
F04BDF Computes the solution and error-bound to a real symmetric positive-definite system of linear equations
F04BEF Computes the solution and error-bound to a real symmetric positive-definite system of linear equations, packed storage
F04BGF Computes the solution and error-bound to a real symmetric positive-definite tridiagonal system of linear equations
F04BHF Computes the solution and error-bound to a real symmetric system of linear equations
F04BJF Computes the solution and error-bound to a real symmetric system of linear equations, packed storage
F04CAF Computes the solution and error-bound to a complex system of linear equations
F04CBF Computes the solution and error-bound to a complex banded system of linear equations
F04CCF Computes the solution and error-bound to a complex tridiagonal system of linear equations
F04CDF Computes the solution and error-bound to a complex Hermitian positive-definite system of linear equations
F04CEF Computes the solution and error-bound to a complex Hermitian positive-definite system of linear equations, packed storage
F04CGF Computes the solution and error-bound to a complex Hermitian positive-definite tridiagonal system of linear equations
F04CHF Computes the solution and error-bound to a complex Hermitian system of linear equations
F04CJF Computes the solution and error-bound to a complex Hermitian system of linear equations, packed storage
F04DHF Computes the solution and error-bound to a complex symmetric system of linear equations
F04DJF Computes the solution and error-bound to a complex symmetric system of linear equations, packed storage.
F06FEF Multiply real vector by reciprocal of scalar
F06KEF Multiply complex vector by reciprocal of real scalar
F06RNF 1-norm, \text{\infty}-norm, Frobenius norm, largest absolute element, real tridiagonal matrix
F06RPF 1-norm, \text{\infty}-norm, Frobenius norm, largest absolute element, real symmetric tridiagonal matrix
F06TAF Matrix-vector product, complex symmetric matrix
F06TBF Rank-1 update, complex symmetric matrix
F06TCF Matrix-vector product, complex symmetric packed matrix
F06TDF Rank-1 update, complex symmetric packed matrix
F06UNF 1-norm, \text{\infty}-norm, Frobenius norm, largest absolute element, complex tridiagonal matrix
F06UPF 1-norm, \text{\infty}-norm, Frobenius norm, largest absolute element, complex Hermitian tridiagonal matrix
F07AAF Computes the solution to a real system of linear equations
F07ABF Uses the LU factorization to compute the solution, error-bound and condition estimate for a real system of linear equations
F07AFF Computes row and column scalings intended to equilibrate a general real matrix and reduce its condition number
F07ANF Computes the solution to a complex system of linear equations
F07APF Uses the LU factorization to compute the solution, error-bound and condition estimate for a complex system of linear equations
F07ATF Computes row and column scalings intended to equilibrate a general complex matrix and reduce its condition number
F07BAF Computes the solution to a real banded system of linear equations
F07BBF Uses the LU factorization to compute the solution, error-bound and condition estimate for a real banded system of linear equations
F07BFF Computes row and column scalings intended to equilibrate a real banded matrix and reduce its condition number
F07BNF Computes the solution to a complex banded system of linear equations

MARK 21 NEWS – NAG FORTRAN SMP LIBRARY.10

NAG Fortran Library Manual
F07BPF Uses the $LU$ factorization to compute the solution, error-bound and condition estimate for a complex banded system of linear equations

F07BTF Computes row and column scalings intended to equilibrate a complex banded matrix and reduce its condition number

F07CAF Computes the solution to a real tridiagonal system of linear equations

F07CBF Uses the $LU$ factorization to compute the solution, error-bound and condition estimate for a real tridiagonal system of linear equations

F07CDF $LU$ factorization of real tridiagonal matrix

F07CEF Solves a real tridiagonal system of linear equations using the $LU$ factorization computed by F07CDF (DGTTRF)

F07CGF Estimates the reciprocal of the condition number of a real tridiagonal matrix using the $LU$ factorization computed by F07CDF (DGTTRF)

F07CHF Refined solution with error bounds of real tridiagonal system of linear equations, multiple right-hand sides

F07CNF Computes the solution to a complex tridiagonal system of linear equations

F07CPF Uses the $LU$ factorization to compute the solution, error-bound and condition estimate for a complex tridiagonal system of linear equations

F07CRF $LU$ factorization of complex tridiagonal matrix

F07CSF Solves a complex tridiagonal system of linear equations using the $LU$ factorization computed by F07CDF (DGTTRF)

F07CUF Estimates the reciprocal of the condition number of a complex tridiagonal matrix using the $LU$ factorization computed by F07CDF (DGTTRF)

F07CVF Refined solution with error bounds of complex tridiagonal system of linear equations, multiple right-hand sides

F07FAF Computes the solution to a real symmetric positive-definite system of linear equations

F07FBF Uses the Cholesky factorization to compute the solution, error-bound and condition estimate for a real symmetric positive-definite system of linear equations

F07FFF Computes row and column scalings intended to equilibrate a real symmetric positive-definite matrix and reduce its condition number

F07FNF Computes the solution to a complex Hermitian positive-definite system of linear equations

F07FPF Uses the Cholesky factorization to compute the solution, error-bound and condition estimate for a complex Hermitian positive-definite system of linear equations

F07FTF Computes row and column scalings intended to equilibrate a complex Hermitian positive-definite matrix and reduce its condition number

F07GAF Computes the solution to a real symmetric positive-definite system of linear equations, packed storage

F07GBF Uses the Cholesky factorization to compute the solution, error-bound and condition estimate for a real symmetric positive-definite system of linear equations, packed storage

F07GFF Computes row and column scalings intended to equilibrate a real symmetric positive-definite matrix and reduce its condition number, packed storage

F07GNF Computes the solution to a complex Hermitian positive-definite system of linear equations, packed storage

F07GPF Computes the solution to a complex Hermitian positive-definite system of linear equations, packed storage

F07GTF Computes row and column scalings intended to equilibrate a complex Hermitian positive-definite matrix and reduce its condition number, packed storage

F07HAF Computes the solution to a real symmetric positive-definite banded system of linear equations

F07HBF Uses the Cholesky factorization to compute the solution, error-bound and condition estimate for a real symmetric positive-definite banded system of linear equations

F07HFF Computes row and column scalings intended to equilibrate a real symmetric positive-definite banded matrix and reduce its condition number

F07HNF Computes the solution to a complex Hermitian positive-definite banded system of linear equations

F07HPF Computes the solution to a complex Hermitian positive-definite banded system of linear equations

F07HTF Computes row and column scalings intended to equilibrate a complex Hermitian positive-definite banded matrix and reduce its condition number

F07JAF Computes the solution to a real symmetric positive-definite tridiagonal system of linear equations
F07JBF Uses the modified Cholesky factorization to compute the solution, error-bound and condition estimate for a real symmetric positive-definite tridiagonal system of linear equations
F07JDF Computes the modified Cholesky factorization of a real symmetric positive-definite tridiagonal matrix
F07JEF Solution of real symmetric tridiagonal linear system, matrix already factorized by F07JDF (DPTTRF)
F07JGF Computes the reciprocal of the condition number of a real symmetric positive-definite tridiagonal system using the modified Cholesky factorization computed by F07JDF (DPTTRF)
F07JHF Refined solution with error bounds of real symmetric positive-definite tridiagonal system of linear equations, multiple right-hand sides
F07JNF Computes the solution to a complex Hermitian positive-definite tridiagonal system of linear equations
F07JPF Uses the modified Cholesky factorization to compute the solution, error-bound and condition estimate for a complex Hermitian positive-definite tridiagonal system of linear equations
F07JRF Computes the modified Cholesky factorization of a complex Hermitian positive-definite tridiagonal matrix
F07JSF Solves a complex Hermitian positive-definite tridiagonal system using the modified Cholesky factorization computed by F07JRF (ZPTTRF)
F07JUF Computes the reciprocal of the condition number of a complex Hermitian positive-definite tridiagonal system using the modified Cholesky factorization computed by F07JRF (ZPTTRF)
F07JVF Refined solution with error bounds of complex Hermitian positive-definite tridiagonal system of linear equations, multiple right-hand sides
F07MAF Computes the solution to a real symmetric system of linear equations
F07MBF Uses the diagonal pivoting factorization to compute the solution to a real symmetric system of linear equations
F07MNF Computes the solution to a complex Hermitian system of linear equations
F07MPF Uses the diagonal pivoting factorization to compute the solution to a complex Hermitian system of linear equations
F07NNF Computes the solution to a complex symmetric system of linear equations
F07NPF Uses the diagonal pivoting factorization to compute the solution to a complex symmetric system of linear equations
F07PAF Computes the solution to a real symmetric system of linear equations, packed storage
F07PBF Uses the diagonal pivoting factorization to compute the solution to a real symmetric system of linear equations, packed storage
F07PNF Computes the solution to a complex Hermitian system of linear equations, packed storage
F07PPF Uses the diagonal pivoting factorization to compute the solution to a complex Hermitian system of linear equations, packed storage
F07QNF Computes the solution to a complex symmetric system of linear equations, packed storage
F07QPF Uses the diagonal pivoting factorization to compute the solution to a complex symmetric system of linear equations, packed storage
F08AAF Solves an overdetermined or underdetermined real linear system
F08ANF Solves an overdetermined or underdetermined complex linear system
F08BAF Computes the minimum-norm solution to a real linear least-squares problem
F08BFF QR factorization of real general rectangular matrix with column pivoting, using BLAS-3
F08BHF Reduces a real upper trapezoidal matrix to upper triangular form
F08BKF Apply orthogonal transformation determined by F08BHF (DTZRZF)
F08BNF Computes the minimum-norm solution to a complex linear least-squares problem
F08BTF QR factorization of complex general rectangular matrix with column pivoting, using BLAS-3
F08BVF Reduces a complex upper trapezoidal matrix to upper triangular form
F08BXF Apply unitary transformation determined by F08BVF (ZTZRZF)
F08CEF QL factorization of real general rectangular matrix
F08CCF Form all or part of orthogonal $Q$ from QL factorization determined by F08CEF (DGEQLF)
F08CGF Apply orthogonal transformation determined by F08CEF (DGEQLF)
F08CHF QR factorization of real general rectangular matrix
F08CFD Form all or part of orthogonal $Q$ from QR factorization determined by F08CHF (DGERQF)
F08CKF Apply orthogonal transformation determined by F08CHF (DGERQF)
F08CSF QL factorization of complex general rectangular matrix
F08CTF Form all or part of orthogonal $Q$ from QL factorization determined by F08CSF (ZGEQLF)
F08CWF Apply unitary transformation determined by F08CSF (ZGEQLF)
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F08CVF</td>
<td>$RQ$ factorization of complex general rectangular matrix</td>
</tr>
<tr>
<td>F08CWF</td>
<td>Form all or part of orthogonal $Q$ from $RQ$ factorization determined by F08CVF (ZGERQF)</td>
</tr>
<tr>
<td>F08CXF</td>
<td>Apply unitary transformation determined by F08CVF (ZGERQF)</td>
</tr>
<tr>
<td>F08FAF</td>
<td>Computes all eigenvalues and, optionally, eigenvectors of a real symmetric matrix</td>
</tr>
<tr>
<td>F08FBF</td>
<td>Computes selected eigenvalues and, optionally, eigenvectors of a real symmetric matrix</td>
</tr>
<tr>
<td>F08FDF</td>
<td>Computes selected eigenvalues and, optionally, eigenvectors of a real symmetric matrix (Relatively Robust Representations)</td>
</tr>
<tr>
<td>F08FLF</td>
<td>Computes the reciprocal condition numbers for the eigenvectors of a real symmetric or complex Hermitian matrix or for the left or right singular vectors of a general matrix</td>
</tr>
<tr>
<td>F08FNF</td>
<td>Computes all eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix</td>
</tr>
<tr>
<td>F08FPF</td>
<td>Computes selected eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix</td>
</tr>
<tr>
<td>F08FRF</td>
<td>Computes selected eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix (Relatively Robust Representations)</td>
</tr>
<tr>
<td>F08GAF</td>
<td>Computes all eigenvalues and, optionally, eigenvectors of a real symmetric matrix, packed storage</td>
</tr>
<tr>
<td>F08GBF</td>
<td>Computes selected eigenvalues and, optionally, eigenvectors of a real symmetric matrix, packed storage</td>
</tr>
<tr>
<td>F08GNF</td>
<td>Computes all eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix, packed storage</td>
</tr>
<tr>
<td>F08GPF</td>
<td>Computes selected eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix, packed storage</td>
</tr>
<tr>
<td>F08HAF</td>
<td>Computes all eigenvalues and, optionally, eigenvectors of a real symmetric band matrix</td>
</tr>
<tr>
<td>F08HBF</td>
<td>Computes selected eigenvalues and, optionally, eigenvectors of a real symmetric band matrix</td>
</tr>
<tr>
<td>F08HNF</td>
<td>Computes all eigenvalues and, optionally, eigenvectors of a complex Hermitian band matrix</td>
</tr>
<tr>
<td>F08HPF</td>
<td>Computes selected eigenvalues and, optionally, eigenvectors of a complex Hermitian band matrix</td>
</tr>
<tr>
<td>F08JAF</td>
<td>Computes all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix</td>
</tr>
<tr>
<td>F08JBF</td>
<td>Computes selected eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix</td>
</tr>
<tr>
<td>F08JDF</td>
<td>Computes selected eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix (Relatively Robust Representations)</td>
</tr>
<tr>
<td>F08JHF</td>
<td>Computes all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix or a matrix reduced to this form (divide-and-conquer)</td>
</tr>
<tr>
<td>F08JLF</td>
<td>Computes all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix or a matrix reduced to this form (Relatively Robust Representations)</td>
</tr>
<tr>
<td>F08JVF</td>
<td>Computes all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix or a complex Hermitian matrix reduced to this form (divide-and-conquer)</td>
</tr>
<tr>
<td>F08JYF</td>
<td>Computes all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix or a complex Hermitian matrix reduced to this form (Relatively Robust Representations)</td>
</tr>
<tr>
<td>F08KAF</td>
<td>Computes the minimum-norm solution to a real linear least-squares problem using singular value decomposition</td>
</tr>
<tr>
<td>F08KBF</td>
<td>Computes the singular value decomposition of a real matrix, optionally computing the left and/or right singular vectors</td>
</tr>
<tr>
<td>F08KCF</td>
<td>Computes the minimum-norm solution to a real linear least-squares problem using singular value decomposition (divide-and-conquer)</td>
</tr>
<tr>
<td>F08KDF</td>
<td>Computes the singular value decomposition of a real matrix, optionally computing the left and/or right singular vectors (divide-and-conquer)</td>
</tr>
<tr>
<td>F08KNF</td>
<td>Computes the minimum-norm solution to a complex linear least-squares problem using singular value decomposition</td>
</tr>
<tr>
<td>F08KPF</td>
<td>Computes the singular value decomposition of a complex matrix, optionally computing the left and/or right singular vectors</td>
</tr>
<tr>
<td>F08KQF</td>
<td>Computes the minimum-norm solution to a complex linear least-squares problem using singular value decomposition (divide-and-conquer)</td>
</tr>
<tr>
<td>F08KRF</td>
<td>Computes the singular value decomposition of a complex matrix, optionally computing the left and/or right singular vectors (divide-and-conquer)</td>
</tr>
<tr>
<td>F08MDF</td>
<td>Computes the singular value decomposition of a real bidiagonal matrix, optionally computing the singular vectors (divide-and-conquer)</td>
</tr>
<tr>
<td>F08NAF</td>
<td>Computes all eigenvalues and, optionally, left and/or right eigenvectors of a real nonsymmetric matrix</td>
</tr>
</tbody>
</table>
F08NBF Computes all eigenvalues and, optionally, left and/or right eigenvectors of a real
nonsymmetric matrix; also, optionally, the balancing transformation, the reciprocal condition
numbers for the eigenvalues and for the right eigenvectors
F08NNF Computes all eigenvalues and, optionally, left and/or right eigenvectors of a complex
nonsymmetric matrix
F08NPF Computes all eigenvalues and, optionally, left and/or right eigenvectors of a complex
nonsymmetric matrix; also, optionally, the balancing transformation, the reciprocal condition
numbers for the eigenvalues and for the right eigenvectors
F08PAF Computes for real square nonsymmetric matrix, the eigenvalues, the real Schur form, and,
on Optionally, the matrix of Schur vectors
F08PBF Computes for real square nonsymmetric matrix, the eigenvalues, the real Schur form, and,
on Optionally, the matrix of Schur vectors; also, optionally, computes reciprocal condition
numbers for selected eigenvalues
F08PNF Computes for complex square nonsymmetric matrix, the eigenvalues, the Schur form, and,
on Optionally, the matrix of Schur vectors
F08PPF Computes for real square nonsymmetric matrix, the eigenvalues, the Schur form, and,
on Optionally, the matrix of Schur vectors; also, optionally, computes reciprocal condition
numbers for selected eigenvalues
F08SAF Computes all the eigenvalues, and optionally, the eigenvectors of a real generalized
symmetric-definite eigenproblem
F08SBF Computes selected eigenvalues, and optionally, the eigenvectors of a real generalized
symmetric-definite eigenproblem
F08SCF Computes all the eigenvalues, and optionally, the eigenvectors of a real generalized
symmetric-definite eigenproblem (divide-and-conquer)
F08SNF Computes all the eigenvalues, and optionally, the eigenvectors of a complex generalized
Hermitian-definite eigenproblem
F08SPF Computes selected eigenvalues, and optionally, the eigenvectors of a complex generalized
Hermitian-definite eigenproblem
F08SQF Computes all the eigenvalues, and optionally, the eigenvectors of a complex generalized
Hermitian-definite eigenproblem (divide-and-conquer)
F08TAF Computes all the eigenvalues, and optionally, the eigenvectors of a real generalized
symmetric-definite eigenproblem, packed storage
F08TBF Computes selected eigenvalues, and optionally, the eigenvectors of a real generalized
symmetric-definite eigenproblem, packed storage
F08TCF Computes all the eigenvalues, and optionally, the eigenvectors of a real generalized
symmetric-definite eigenproblem, packed storage (divide-and-conquer)
F08TNF Computes all the eigenvalues, and optionally, the eigenvectors of a complex generalized
Hermitian-definite eigenproblem, packed storage
F08TPF Computes selected eigenvalues, and optionally, the eigenvectors of a complex generalized
Hermitian-definite eigenproblem, packed storage
F08TQF Computes selected eigenvalues, and optionally, the eigenvectors of a complex generalized
Hermitian-definite eigenproblem, packed storage (divide-and-conquer)
F08UAF Computes all the eigenvalues, and optionally, the eigenvectors of a real banded generalized
symmetric-definite eigenproblem
F08UBF Computes selected eigenvalues, and optionally, the eigenvectors of a real banded generalized
symmetric-definite eigenproblem
F08UCF Computes all the eigenvalues, and optionally, the eigenvectors of a real banded generalized
symmetric-definite eigenproblem (divide-and-conquer)
F08UNF Computes all the eigenvalues, and optionally, the eigenvectors of a complex banded
generalized Hermitian-definite eigenproblem
F08UPF Computes selected eigenvalues, and optionally, the eigenvectors of a complex banded
generalized Hermitian-definite eigenproblem
F08UQF Computes all the eigenvalues, and optionally, the eigenvectors of a complex banded
generalized Hermitian-definite eigenproblem (divide-and-conquer)
F08VAF Computes the generalized singular value decomposition of a real matrix pair
F08VEF Computes orthogonal matrices as processing steps for computing the generalized singular
value decomposition of a real matrix pair
F08VNF Computes the generalized singular value decomposition of a complex matrix pair
F08VSF Computes orthogonal matrices as processing steps for computing the generalized singular value decomposition of a complex matrix pair
F08WAF Computes, for a real nonsymmetric matrix pair, the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors
F08WBF Computes, for a real nonsymmetric matrix pair, the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors; also, optionally, the balancing transformation, the reciprocal condition numbers for the eigenvalues and for the right eigenvectors
F08WEF Orthogonal reduction of a pair of real general matrices to generalized upper Hessenberg form
F08WHF Balance a pair of real general matrices
F08WJF Transform eigenvectors of a pair of real balanced matrices to those of original matrix pair supplied to F08WHF (DGGBAL)
F08WNF Computes, for a complex nonsymmetric matrix pair, the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors
F08WPF Computes, for a complex nonsymmetric matrix pair, the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors; also, optionally, the balancing transformation, the reciprocal condition numbers for the eigenvalues and for the right eigenvectors
F08WSF Unitary reduction of a pair of complex general matrices to generalized upper Hessenberg form
F08WWF Transform eigenvectors of a pair of complex balanced matrices to those of original matrix pair supplied to F08WWF (ZGGBAL)
F08XAF Computes, for a real nonsymmetric matrix pair, the generalized eigenvalues, the generalized real Schur form and, optionally, the left and/or right matrices of Schur vectors
F08XBF Computes, for a real nonsymmetric matrix pair, the generalized eigenvalues, the generalized real Schur form and, optionally, the left and/or right matrices of Schur vectors; also, optionally, computes reciprocal condition numbers for selected eigenvalues
F08XEF Eigenvalues and generalized Schur factorization of real generalized upper Hessenberg form reduced from a pair of real general matrices
F08XNF Computes, for a complex nonsymmetric matrix pair, the generalized eigenvalues, the generalized complex Schur form and, optionally, the left and/or right matrices of Schur vectors
F08XPF Computes, for a complex nonsymmetric matrix pair, the generalized eigenvalues, the generalized complex Schur form and, optionally, the left and/or right matrices of Schur vectors; also, optionally, computes reciprocal condition numbers for selected eigenvalues
F08XSF Eigenvalues and generalized Schur factorization of complex generalized upper Hessenberg form reduced from a pair of complex general matrices
F08YEF Computes the generalized singular value decomposition of a real upper triangular (or trapezoidal) matrix pair
F08YFF Reorders the generalized real Schur decomposition of a real matrix pair using an orthogonal equivalence transformation
F08YGF Reorders the generalized real Schur decomposition of a real matrix pair using an orthogonal equivalence transformation, computes the generalized eigenvalues of the reordered pair and, optionally, computes the estimates of reciprocal condition numbers for eigenvalues and eigenspaces
F08YHF Solves the real-valued generalized Sylvester equation
F08YKF Left and right eigenvectors of a pair of real upper quasi-triangular matrices
F08YLF Estimates reciprocal condition numbers for specified eigenvalues and/or eigenvectors of a real matrix pair in generalized real Schur canonical form
F08YSF Computes the generalized singular value decomposition of a complex upper triangular (or trapezoidal) matrix pair
F08YTF Reorders the generalized Schur decomposition of a complex matrix pair using an unitary equivalence transformation
F08YUF Reorders the generalized Schur decomposition of a complex matrix pair using an unitary equivalence transformation, computes the generalized eigenvalues of the reordered pair and, optionally, computes the estimates of reciprocal condition numbers for eigenvalues and eigenspaces
F08YVF Solves the complex generalized Sylvester equation
F08YXF Left and right eigenvectors of a pair of complex upper triangular matrices
F08YYF Estimates reciprocal condition numbers for specified eigenvalues and/or eigenvectors of a complex matrix pair in generalized Schur canonical form
F08ZAF Solves the real linear equality-constrained least-squares (LSE) problem
F08ZBF Solves a real general Gauss–Markov linear model (GLM) problem
F08ZEF Computes a generalized $QR$ factorization of a real matrix pair
F08ZFF Computes a generalized $RQ$ factorization of a real matrix pair
F08ZNF Solves the complex linear equality-constrained least-squares (LSE) problem
F08ZPF Solves a complex general Gauss–Markov linear model (GLM) problem
F08ZSF Computes a generalized $QR$ factorization of a complex matrix pair
F08ZTF Computes a generalized $RQ$ factorization of a complex matrix pair
F11DXF Complex sparse nonsymmetric linear systems, line Jacobi preconditioner
F11GRF Complex sparse Hermitian linear systems, setup for F11GSF
F11GSF Complex sparse Hermitian linear systems, preconditioned conjugate gradient or Lanczos
F11GTF Complex sparse Hermitian linear systems, diagnostic for F11GSF
F11MDF Real sparse nonsymmetric linear systems, setup for F11MEF
F11MEF $LU$ factorization of real sparse matrix
F11MFF Solution of real sparse simultaneous linear equations (coefficient matrix already factorized)
F11MGF Estimate condition number of real matrix, matrix already factorized by F11MEF
F11MHF Refined solution with error bounds of real system of linear equations, multiple right-hand sides
F11MKF Real sparse nonsymmetric matrix matrix multiply, compressed column storage
F11MLF $1$-norm, $\infty$-norm, largest absolute element, real general matrix
F11MMF Real sparse nonsymmetric linear systems, diagnostic for F11MEF
F12AAF Initialization routine for (F12ABF) computing selected eigenvalues and, optionally, eigenvectors of a real nonsymmetric sparse (standard or generalized) eigenproblem
F12ABF Implements a reverse communication interface for the Implicitly Restarted Arnoldi iteration for computing selected eigenvalues and, optionally, eigenvectors of a real nonsymmetric sparse (standard or generalized) eigenproblem
F12ACF Returns the converged approximations (as determined by F12ABF) to eigenvalues of a real nonsymmetric sparse (standard or generalized) eigenproblem and, optionally, the corresponding approximate eigenvectors and/or an orthonormal basis for the associated approximate invariant subspace
F12ADF Set a single option from a string (F12ABF/F12ACF/F12AGF)
F12AEF Provides monitoring information for F12ABF
F12AFF Initialization routine for (F12AGF) computing selected eigenvalues and, optionally, eigenvectors of a real nonsymmetric banded (standard or generalized) eigenproblem
F12AGF Computes approximations to selected eigenvalues of a real nonsymmetric banded (standard or generalized) eigenproblem and, optionally, the corresponding approximate eigenvectors and/or an orthonormal basis for the associated approximate invariant subspace
F12ANF Initialization routine for (F12APF) computing selected eigenvalues and, optionally, eigenvectors of a complex sparse (standard or generalized) eigenproblem
F12APF Implements a reverse communication interface for the Implicitly Restarted Arnoldi iteration for computing selected eigenvalues and, optionally, eigenvectors of a complex sparse (standard or generalized) eigenproblem
F12AQF Returns the converged approximations (as determined by F12ABF) to eigenvalues of a complex sparse (standard or generalized) eigenproblem and, optionally, the corresponding approximate eigenvectors and/or an orthonormal basis for the associated approximate invariant subspace
F12ARF Set a single option from a string (F12APF/F12AQF)
F12ASF Provides monitoring information for F12APF
F12FAF Initialization routine for (F12FBF) computing selected eigenvalues and, optionally, eigenvectors of a real symmetric sparse (standard or generalized) eigenproblem
F12FBF Implements a reverse communication interface for the Implicitly Restarted Arnoldi iteration for computing selected eigenvalues and, optionally, eigenvectors of a real symmetric sparse (standard or generalized) eigenproblem
F12FCF Returns the converged approximations (as determined by F12ABF) to eigenvalues of a real symmetric sparse (standard or generalized) eigenproblem and, optionally, the corresponding approximate eigenvectors and/or an orthonormal basis for the associated approximate invariant subspace
F12FDF Set a single option from a string (F12FBF/F12FCF/F12FGF)
F12FEF Provides monitoring information for F12FBF
F12FFF Initialization routine for (F12FGF) computing selected eigenvalues and, optionally, eigenvectors of a real symmetric banded (standard or generalized) eigenproblem

F12FGF Computes approximations to selected eigenvalues of a real symmetric banded (standard or generalized) eigenproblem and, optionally, the corresponding approximate eigenvectors and/or an orthonormal basis for the associated approximate invariant subspace

G01ETF Landau distribution function $\Phi(\lambda)$
G01EUF Vavilov distribution function $\Phi_V(\lambda; \kappa, \beta^2)$
G01FTF Landau inverse function $\Psi(x)$
G01MTF Landau density function $\phi(\lambda)$
G01MUF Vavilov density function $\phi_V(\lambda; \kappa, \beta^2)$
G01PTF Landau first moment function $\Phi_1(x)$
G01QTF Landau second moment function $\Phi_2(x)$
G01RTF Landau derivative function $\phi'(\lambda)$
G01ZUF Initialization routine for G01MUF and G01EUF

G02EFP Stepwise linear regression
G02JAF Linear mixed effects regression using Restricted Maximum Likelihood (REML)
G02JBF Linear mixed effects regression using Maximum Likelihood (ML)
G05HKF Univariate time series, generate $n$ terms of either a symmetric GARCH process or a GARCH process with asymmetry of the form $(\epsilon_{t-1} + \gamma)^2$
G05HLF Univariate time series, generate $n$ terms of a GARCH process with asymmetry of the form $(\epsilon_{t-1} + \gamma \epsilon_{t-1})^2$
G05HMF Univariate time series, generate $n$ terms of an asymmetric Glosten, Jagannathan and Runkle (GJR) GARCH process
G05HNF Univariate time series, generate $n$ terms of an exponential GARCH (EGARCH) process
G05KAF Pseudo-random real numbers, uniform distribution over $(0,1)$, seeds and generator number passed explicitly
G05KBF Initialize seeds of a given generator for random number generating routines (that pass seeds explicitly) to give a repeatable sequence
G05KCF Initialize seeds of a given generator for random number generating routines (that pass seeds explicitly) to give non-repeatable sequence
G05KEF Pseudo-random logical (boolean) value, seeds and generator number passed explicitly
G05LAF Generates a vector of random numbers from a Normal distribution, seeds and generator number passed explicitly
G05LBF Generates a vector of random numbers from a Student’s $t$-distribution, seeds and generator number passed explicitly
G05LCF Generates a vector of random numbers from a $\chi^2$ distribution, seeds and generator number passed explicitly
G05LDF Generates a vector of random numbers from an $F$-distribution, seeds and generator number passed explicitly
G05LEF Generates a vector of random numbers from a $\beta$ distribution, seeds and generator number passed explicitly
G05LFF Generates a vector of random numbers from a $\gamma$ distribution, seeds and generator number passed explicitly
G05LGF Generates a vector of random numbers from a uniform distribution, seeds and generator number passed explicitly
G05LHF Generates a vector of random numbers from a triangular distribution, seeds and generator number passed explicitly
G05LJF Generates a vector of random numbers from an exponential distribution, seeds and generator number passed explicitly
G05LKF Generates a vector of random numbers from a lognormal distribution, seeds and generator number passed explicitly
G05LLF Generates a vector of random numbers from a Cauchy distribution, seeds and generator number passed explicitly
G05LMF Generates a vector of random numbers from a Weibull distribution, seeds and generator number passed explicitly
G05LNF Generates a vector of random numbers from a logistic distribution, seeds and generator number passed explicitly
G05LPF Generates a vector of random numbers from a von Mises distribution, seeds and generator number passed explicitly
G05LQF Generates a vector of random numbers from an exponential mixture distribution, seeds and generator number passed explicitly
G05LXF Generates a matrix of random numbers from a multivariate Student’s \( t \)-distribution, seeds and generator passed explicitly
G05LYF Generates a matrix of random numbers from a multivariate Normal distribution, seeds and generator passed explicitly
G05LZF Generates a vector of random numbers from a multivariate Normal distribution, seeds and generator number passed explicitly
G05MAF Generates a vector of random integers from a uniform distribution, seeds and generator number passed explicitly
G05MBF Generates a vector of random integers from a geometric distribution, seeds and generator number passed explicitly
G05MCF Generates a vector of random integers from a negative binomial distribution, seeds and generator number passed explicitly
G05MDF Generates a vector of random integers from a logarithmic distribution, seeds and generator number passed explicitly
G05MEF Generates a vector of random integers from a Poisson distribution with varying mean, seeds and generator number passed explicitly
G05MJF Generates a vector of random integers from a binomial distribution, seeds and generator number passed explicitly
G05MKF Generates a vector of random integers from a Poisson distribution, seeds and generator number passed explicitly
G05MLF Generates a vector of random integers from a hypergeometric distribution, seeds and generator number passed explicitly
G05MRF Generates a vector of random integers from a multinomial distribution, seeds and generator number passed explicitly
G05MZF Generates a vector of random integers from a general discrete distribution, seeds and generator number passed explicitly
G05NAF Pseudo-random permutation of an integer vector
G05NBF Pseudo-random sample from an integer vector
G05PAF Generates a realisation of a time series from an ARMA model
G05PCF Generates a realisation of a multivariate time series from a VARMA model
G05QAF Computes a random orthogonal matrix
G05QBF Computes a random correlation matrix
G05QDF Generates a random table matrix
G05RAF Generates a matrix of random numbers from a Gaussian Copula, seeds and generator passed explicitly
G05RBF Generates a matrix of random numbers from a Student’s \( t \)-Copula, seeds and generator passed explicitly
G05YAF Multi-dimensional quasi-random number generator with a uniform probability distribution
G05YBF Multi-dimensional quasi-random number generator with a Gaussian or log-normal probability distribution
G05YCF Initializes the Faure generator (G05YDF/G05YJF/G05YKF)
G05YDF Generates a sequence of quasi-random numbers using Faure’s method
G05YEF Generates the Sobol generator (G05YFF/G05YJF/G05YKF)
G05YFF Generates a sequence of quasi-random numbers using Sobol’s method
G05YGF Initializes the Neiderreiter generator (G05YHF/G05YJF/G05YKF)
G05YHF Generates a sequence of quasi-random numbers using Neiderreiter’s method
G05YJF Generates a Normal quasi-random number sequence using Faure’s, Sobol’s or Neiderreiter’s method
G05YKF Generates a log-Normal quasi-random number sequence using Faure’s, Sobol’s or Neiderreiter’s method
G13FAF Univariate time series, parameter estimation for either a symmetric GARCH process or a GARCH process with asymmetry of the form \((\epsilon_{t-1} + \gamma)^2\)
G13FBF Univariate time series, forecast function for either a symmetric GARCH process or a GARCH process with asymmetry of the form \((\epsilon_{t-1} + \gamma)^2\)
Univariate time series, parameter estimation for a GARCH process with asymmetry of the form 
\[ \left| r_{t-1} \right|^2 \]  
Univariate time series, forecast function for a GARCH process with asymmetry of the form 
\[ \left| r_{t-1} \right|^2 \]  
Univariate time series, parameter estimation for an asymmetric Glosten, Jagannathan and Runkle (GJR) GARCH process  
Univariate time series, forecast function for an asymmetric Glosten, Jagannathan and Runkle (GJR) GARCH process  
Univariate time series, parameter estimation for an exponential GARCH (EGARCH) process  
Univariate time series, forecast function for an exponential GARCH (EGARCH) process  
Polygamma function \( \psi^{(n)}(x) \) for real \( x \)  
Polygamma function \( \psi^{(n)}(z) \) for complex \( z \)  
Logarithm of the Gamma function \( \ln \Gamma(z) \)  
Zeros of Bessel functions \( J_\alpha(x) \), \( J_\alpha'(x) \), \( Y_\alpha(x) \) or \( Y_\alpha'(x) \)  
Bessel function of the 1st kind \( J_{\alpha,\nu}(z) \)  
Jacobian elliptic functions \( sn, cn \) and \( dn \) of complex argument  
Jacobian theta functions \( \theta_k(x, q) \) of real argument  
General elliptic integral of 2nd kind \( F(z; k^2, a, b) \) of complex argument  
Legendre functions of 1st kind \( P_n^m(x) \) or \( \overline{P}_n^m(x) \)  

5 Withdrawn SMP Functionality  
The random number generator routines parallelized in Release 2 have now been scheduled for withdrawal at Mark 22, as the interfaces used were not thread-safe due to the internal use of COMMON blocks. Therefore, the SMP functionality has not been included in this release, to encourage users to switch to the new thread-safe replacement routines. Users can easily take advantage of the new thread-safe replacement routines on multiple processors by calling the routines within their own OpenMP parallel region, using a different generator number and seed array for each thread. As there are 273 separate Wichmann–Hill generators available, statistically independent sequences of random numbers can be generated on up to 273 processors simultaneously.  

Routines Scheduled for Withdrawal Replacement Routine(s)  
G05FAF G05LGF  
G05FBF G05LJF  
G05FDF G05LAF  
G05FEF G05LEF  
G05FFF G05LFF  
G05FSF G05LPF  
G05GAF G05QAF  
G05GBF G05QBF  
G05HDF G05PCF  

6 Withdrawn Routines  
The following routines have been withdrawn from the NAG SMP Library at Mark 21. Warnings of their withdrawal were included in the NAG Fortran Library Manual at Mark 20, together with advice on which routines to use instead. See the document ‘Advice on Replacement Calls for Withdrawn/Superseded Routines’ for more detailed guidance.  

Withdrawn Routine Replacement Routine(s)  
E01SEF E01SGF  
E01SFF E01SHF  
F11BAF F11BDF  
F11BBF F11BEF  
F11BCF F11BFF
7 Routines Scheduled for Withdrawal

The routines listed below are scheduled for withdrawal from the SMP Library, because improved routines have now been included in the Library. Users are advised to stop using routines which are scheduled for withdrawal immediately and to use recommended replacement routines instead. See the document ‘Advice on Replacement Calls for Withdrawn/Superseded Routines’ for more detailed guidance, including advice on how to change a call to the old routine into a call to its recommended replacement.

The following routines will be withdrawn at Mark 22.

<table>
<thead>
<tr>
<th>Routine Scheduled for Withdrawal</th>
<th>Replacement Routine(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E04UNF</td>
<td>E04USF/E04USA</td>
</tr>
<tr>
<td>F11GAF</td>
<td>F11GDF</td>
</tr>
<tr>
<td>F11GBF</td>
<td>F11GEF</td>
</tr>
<tr>
<td>F11GCF</td>
<td>F11GFF</td>
</tr>
<tr>
<td>G05CAF</td>
<td>G05KAF</td>
</tr>
<tr>
<td>G05CBF</td>
<td>G05KBF</td>
</tr>
<tr>
<td>G05CCF</td>
<td>G05KCF</td>
</tr>
<tr>
<td>G05CFF</td>
<td>F06DFF</td>
</tr>
<tr>
<td>G05CGF</td>
<td>F06DFF</td>
</tr>
<tr>
<td>G05DAF</td>
<td>G05LGF</td>
</tr>
<tr>
<td>G05DBF</td>
<td>G05LJF</td>
</tr>
<tr>
<td>G05DCF</td>
<td>G05LNF</td>
</tr>
<tr>
<td>G05DDF</td>
<td>G05LAF</td>
</tr>
<tr>
<td>G05DEF</td>
<td>G05LKF</td>
</tr>
<tr>
<td>G05DFF</td>
<td>G05LLF</td>
</tr>
<tr>
<td>G05DHF</td>
<td>G05LCF</td>
</tr>
<tr>
<td>G05DJF</td>
<td>G05LBF</td>
</tr>
<tr>
<td>G05DKF</td>
<td>G05LDF</td>
</tr>
<tr>
<td>G05DPF</td>
<td>G05LMF</td>
</tr>
<tr>
<td>G05DRF</td>
<td>G05MEF</td>
</tr>
<tr>
<td>G05DYF</td>
<td>G05MAF</td>
</tr>
<tr>
<td>G05DZF</td>
<td>G05KEF</td>
</tr>
<tr>
<td>G05EAF</td>
<td>G05LZF</td>
</tr>
<tr>
<td>G05EBF</td>
<td>G05MAF</td>
</tr>
<tr>
<td>G05ECF</td>
<td>G05MKF</td>
</tr>
<tr>
<td>G05EDF</td>
<td>G05MJF</td>
</tr>
<tr>
<td>G05EEF</td>
<td>G05MCF</td>
</tr>
<tr>
<td>G05EFF</td>
<td>G05MLF</td>
</tr>
<tr>
<td>G05EGF</td>
<td>G05PAF</td>
</tr>
<tr>
<td>G05EHF</td>
<td>G05NAF</td>
</tr>
<tr>
<td>G05EJF</td>
<td>G05NBF</td>
</tr>
<tr>
<td>G05EWF</td>
<td>G05PAF</td>
</tr>
<tr>
<td>G05EXF</td>
<td>G05MZF</td>
</tr>
<tr>
<td>G05EYF</td>
<td>G05MZF</td>
</tr>
<tr>
<td>G05EZF</td>
<td>G05LZF</td>
</tr>
<tr>
<td>G05FAF</td>
<td>G05LGF</td>
</tr>
<tr>
<td>G05FBF</td>
<td>G05LJF</td>
</tr>
<tr>
<td>G05FDF</td>
<td>G05LAF</td>
</tr>
<tr>
<td>G05FEF</td>
<td>G05LEF</td>
</tr>
<tr>
<td>G05FFF</td>
<td>G05LFF</td>
</tr>
<tr>
<td>G05FSF</td>
<td>G05LPF</td>
</tr>
<tr>
<td>G05GAF</td>
<td>G05QAF</td>
</tr>
<tr>
<td>G05GBF</td>
<td>G05QBF</td>
</tr>
<tr>
<td>G05HDF</td>
<td>G05PCF</td>
</tr>
<tr>
<td>G05ZAF</td>
<td>No replacement document required</td>
</tr>
</tbody>
</table>

The following routines have been superseded, but will not be withdrawn from the Library until Mark 23 at the earliest.
<table>
<thead>
<tr>
<th>Superseded Routine</th>
<th>Replacement Routine(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E04NKF/E04NKA</td>
<td>E04NQF</td>
</tr>
<tr>
<td>E04NLF/E04NLA</td>
<td>E04NRF</td>
</tr>
<tr>
<td>E04UCF/E04UCA</td>
<td>E04WDF</td>
</tr>
<tr>
<td>F02BJF</td>
<td>F08WAF (DGGEV)</td>
</tr>
<tr>
<td>F02EAF</td>
<td>F08PAF (DGEES)</td>
</tr>
<tr>
<td>F02EBF</td>
<td>F08NAF (DGEEV)</td>
</tr>
<tr>
<td>F02FAF</td>
<td>F08FAF (DSYEV)</td>
</tr>
<tr>
<td>F02FCF</td>
<td>F08FBF (DSYEVX)</td>
</tr>
<tr>
<td>F02FDF</td>
<td>F08SAF (DSYGV)</td>
</tr>
<tr>
<td>F02FHF</td>
<td>F08UAF (DSBGV)</td>
</tr>
<tr>
<td>F02GAF</td>
<td>F08PFN (ZGEES)</td>
</tr>
<tr>
<td>F02GBF</td>
<td>F08NNF (ZGEEV)</td>
</tr>
<tr>
<td>F02HAF</td>
<td>F08FNF (ZHEEV)</td>
</tr>
<tr>
<td>F02HCF</td>
<td>F08FPF (ZHEEVX)</td>
</tr>
<tr>
<td>F02HDF</td>
<td>F08SNF (ZHEGV)</td>
</tr>
<tr>
<td>F02WEF</td>
<td>F08KBF (DGESVD)</td>
</tr>
<tr>
<td>F02XEF</td>
<td>F08KPF (ZGESVD)</td>
</tr>
<tr>
<td>F04AAF</td>
<td>F07AAF (DGESV)</td>
</tr>
<tr>
<td>F04ACF</td>
<td>F07HAF (DPBSV)</td>
</tr>
<tr>
<td>F04ADF</td>
<td>F07ANF (ZGESV)</td>
</tr>
<tr>
<td>F04ARF</td>
<td>F07AAL (DGESV)</td>
</tr>
<tr>
<td>F04EAF</td>
<td>F07CAF (DGTSV)</td>
</tr>
<tr>
<td>F04EAF</td>
<td>F07JAF (DPTSV), F07JDF (DPTTRF) and F07JEF (DPTTRS)</td>
</tr>
<tr>
<td>F04JAF</td>
<td>F08KAF (DGEOLSS)</td>
</tr>
<tr>
<td>F04JDF</td>
<td>F08KAF (DGEOLSS)</td>
</tr>
<tr>
<td>F04JLF</td>
<td>F08ZBF (DGGLLM)</td>
</tr>
<tr>
<td>F04JMF</td>
<td>F08ZAF (DGGLSE)</td>
</tr>
<tr>
<td>F04KLF</td>
<td>F08ZPF (ZGGGLM)</td>
</tr>
<tr>
<td>F04KMF</td>
<td>F08ZNF (ZGGLSE)</td>
</tr>
<tr>
<td>G05YAF</td>
<td>G05YCF, G05YDF, G05YEF, G05YFF, G05YGF, G05YHF, G05YJF and G05YKF</td>
</tr>
<tr>
<td>G05YBF</td>
<td>G05YCF, G05YDF, G05YEF, G05YFF, G05YGF, G05YHF, G05YJF and G05YKF</td>
</tr>
</tbody>
</table>