ENHANCED NUMERICAL CAPABILITY THROUGH NAG DLLS

SECTION 1 **INTRODUCTION TO NAG DYNAMIC LINK LIBRARIES** **3**

SECTION 2 **WHY A DYNAMIC LINK LIBRARY?** **4**

SECTION 3 **THE NAG FORTRAN DLLS** **6**

SUPPLIED MATERIALS 6
USING THE BROWSER 6
USING THE INDIVIDUAL HELP FILES 6

SECTION 4 **DOCUMENTATION AND SUPPORT** **7**

DOCUMENTATION 7
TECHNICAL SUPPORT 7
NEW RELEASES 8
FURTHER INFORMATION ON THE WEB 8

SECTION 5 **CALLING THE DLLS FROM OTHER LANGUAGES** **9**

MICROSOFT EXCEL AND MICROSOFT VISUAL BASIC FOR APPLICATIONS 9
MICROSOFT VISUAL BASIC 10
MICROSOFT C++ 10
BORLAND C++ 10
BORLAND DELPHI 11
POWERBUILDER 11
WATCOM C/C++ 12

SECTION 6 **THE NUMERICAL AND STATISTICAL ROUTINES AVAILABLE** **13**

Chapter A00 – Library Identification 13
Chapter A02 – Complex Arithmetic 13
Chapter C02 – Zeros of Polynomials 13
Chapter C05 – Roots of One or More Transcendental Equations 13
Chapter C06 – Summation of Series 13
Chapter D01 – Quadrature 14
Chapter D02 – Ordinary Differential Equations 15
Chapter D03 – Partial Differential Equations 17
Chapter D04 – Numerical Differentiation 18
Chapter D05 – Integral Equations 18
Chapter E01 – Interpolation 18
Chapter E02 – Curve and Surface Fitting 18
Chapter E04 – Minimizing or Maximizing a Function 19
Chapter F01 – Matrix Factorizations 20
Chapter F02 – Eigenvalues and Eigenvectors 21
Chapter F03 – Determinants 21
Chapter F04 – Simultaneous Linear Equations 22
Chapter F05 – Orthogonalisation 23
Chapter F06 – Linear Algebra Support Routines 23
Chapter F07 – Linear Equations (LAPACK) 27
Chapter F08 – Least-squares and Eigenvalue Problems (LAPACK) 30
Chapter F11 – Sparse Linear Algebra 32
Chapter G01 – Simple Calculations and Statistical Data 33
Chapter G02 – Correlation and Regression Analysis 34
Chapter G03 – Multivariate Methods 36
Chapter G04 – Analysis of Variance 36
Chapter G05 – Random Number Generators 36
Chapter G07 – Univariate Estimation 37
Chapter G08 – Nonparametric Statistics 38
Chapter G10 – Smoothing in Statistics 38
Chapter G11 – Contingency Table Analysis 38
Chapter G12 – Survival Analysis 38

Chapter G13 – Time Series Analysis 39
Chapter H – Operations Research 40
Chapter M0 – Sorting 40
Chapter P01 – Error Trapping 40
Chapter S – Approximations of Special Functions 40
Chapter X01 – Mathematical Constants 42
Chapter X02 – Machine Constants 42
Chapter X03 – Inner Products 42
Chapter X04 – Input/Output Utilities 42
Chapter X05 – Date and Time Utilities 43

Section 1 INTRODUCTION TO NAG DYNAMIC LINK LIBRARIES

NAG has a **longstanding worldwide reputation** for the excellence of its numerical and statistical algorithms. Industry, universities and research institutes rely on the **accuracy, reliability and robustness** of these to solve complex problems, typically in research, engineering, life and earth sciences, and financial analysis.

A "Dynamic Link Library"(or DLL) is a term used in the Windows PC World to describe a type of library that in other environments is described as "shareable". The main feature of these libraries is that a program linking to these libraries does not copy the library code into the final executable. Instead it copies in a pointer to the library, so that at run time the application actually runs the code in the library rather than a copy of it.

This software provides a broad range of numerical and statistical routines, typically optimization, PDEs, ODEs, FFTs, correlation and regression, and multivariate methods, to name but a few areas. For a complete listing please see Section 6 of this booklet.

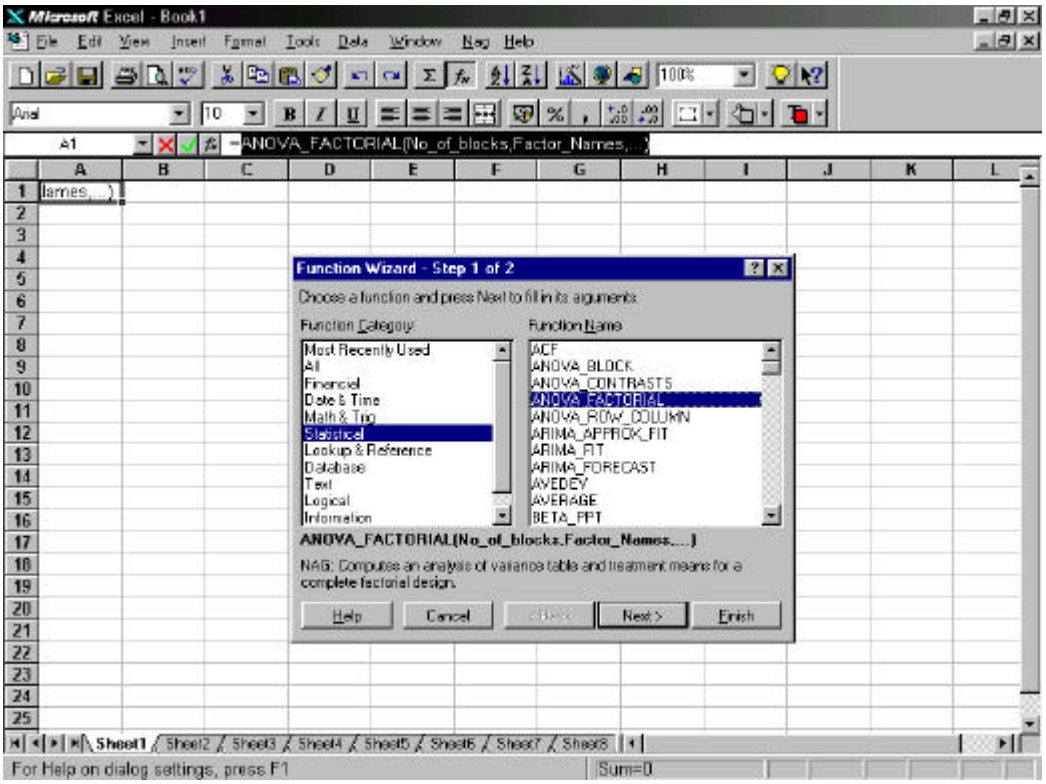
Using NAG software has regularly been proven to be the most cost effective way of solving numerical and statistical problems, reducing programming time, whilst increasing productivity and confidence.

Many organisations integrate our software seamlessly with a variety of applications, typically **Visual Basic, Visual Basic Applications, Excel, Microsoft C++, Borland C++, Borland Delphi, PowerBuilder** and **Watcom C/C++.** They know that by taking this approach they save development time and protect their investment in application development by receiving regular updates and full technical support from NAG. For users who need advice on development or on numerical techniques, we offer software development and numerical computation expertise through the **NAG Consultancy Service**. For more information please contact your local NAG office.

Companies rely on our software for mission-critical applications, so we operate very **stringent quality assurance standards**. No algorithm enters our software until it has been fully tested, validated and verified. Underpinning the quality of all NAG software is our renowned technical support. Customers have comprehensive documentation, as well as access to the actual developers of the software to ensure that their questions are answered to the highest standards.

Section 2 WHY A DYNAMIC LINK LIBRARY?

The NAG Fortran and C Libraries are both available as Dynamic Link Libraries (DLLs) for use on Windows systems. These DLLs provide an easy way to include powerful numerical functionality in Windows applications.



This Picture shows a snapshot of the Microsoft Excel spreadsheet after it has been extended to include several NAG routines

This booklet explains how the Fortran-based DLLs are easily called from Excel, Visual Basic (VB), Visual Basic for Applications (VBA), Microsoft C++, Borland Delphi, Fortran 77, Fortran 90, PowerBuilder and Watcom C/C++.

NAG has, historically, found that the majority of its customers call the DLLs from VB, VBA or C. If the DLLs are going to be called from VB or VBA, the Fortran-based DLL is normally preferred. VB stores its arrays using the same convention as Fortran and therefore the underlying numerical library and code would be invisible to the user.

All the NAG DLLs are provided complete with VB 'Declare' statements. These VB instructions introduce the routines to VB, VBA and also act as useful templates for the user calling the routines. Fortran and VB have a good one to one matching of types, for example VB **double** corresponds to Fortran DOUBLE PRECISION, VB **Long** corresponds to Fortran's INTEGER. Therefore it is straightforward to call the majority of the Fortran DLL routines from VB.

For users who prefer to use C, the NAG C Library is also available as a DLL. This library provides users with nearly 400 routines covering a broad range of numerical and statistical areas. For further information on using these C-based DLLs please contact your local NAG

office or visit our web site at www.nag.co.uk/numeric/CL/html. For enhanced coverage the NAG Fortran Library, with over 1000 routines, can be used within C programs through the C Header File interface. This interface is provided with the Fortran Library DLLs.

The Users' Notes provided with the NAG Fortran and C Libraries give examples and explain how to call the DLL routines.

Section 3 THE NAG FORTRAN DLLS

Supplied Materials

When you purchase the DLLs you will also receive:

- A small selection of test programs to test successful installation
- C Header files, provided in a Windows Help file for easy access to assist you in mapping and for 'cut and paste' usage
- Visual Basic (VB) Declare statements, provided in a Windows Help file to facilitate the usage from VB and Microsoft Excel
- Fortran 90 Interface blocks, also provided in a Windows Help file, to enable Fortran 90 programmers to conveniently insert interface blocks for the NAG routines into their programs
- A Windows Help version of the NAG routine documentation
- Skeletal 'jacket' routines, written in Fortran, again provided in a Windows Help format; these provide a mechanism for providing subroutine or function arguments to a NAG routine from the VB or Excel environment
- A VB 'Browser' program which may be used to conveniently launch the various Help files

Please note: the Fortran-based DLLs are, in general, not thread-safe and the user should check with NAG if planning to use the routines in a multi-threaded environment.

Using the Browser

After installation the Browser may be activated from the Start menu of Windows 95/98 or Windows NT version 4.0. From Windows version 3.5 it may be started from the Group Icon associated with the DLLs.

The Browser consists of a form containing various NAG Chapter titles. As the mouse pointer is moved over the Chapter titles a brief description of the chapter contents is automatically displayed in a text box. In this way a user unfamiliar with the NAG Library Contents may rapidly search for an appropriate chapter.

Using the Individual Help Files

The user may by-pass the Browser to call the individual Help files directly, either from the Start menu for Windows 95/98 and Windows NT version 4.0 or from the Group Icon for Windows NT version 3.5.

The Help files are relatively straightforward. It is envisaged that users copy information of interest to the Clipboard and then paste this information into their own application.

System developers may wish to use the NAG Help files within their own applications. To do this effectively they will need to be aware of the Context IDs used within the help files. Please see www.nag.co.uk/numeric/NAGFortranDLLs.asp for more details.

Section 4 DOCUMENTATION AND SUPPORT

We endeavour to ensure that all our customers gain maximum benefit from the investment they make when they license our products and services. We achieve this by continually enhancing our products and services to ensure that they fulfil the demands and expectations of the modern computing community. This requires careful engineering of our software and rigorous preparation and testing of our routines. Control records are kept of our servicing and support details and these are reviewed annually to ensure that our tight standards are maintained.

When you have licensed software from us, we want to ensure that you:
- Receive it in a timely manner
- Can install it simply and satisfactorily
- Can exploit the functionality of the product for your applications.

At the same time, we offer an extensive range of services and support facilities, from information and advice available to all users to specialised technical provision for those customers paying specifically for support and maintenance. In addition, the NAG web sites provide a valuable resource for product information, implementation availability, technical documentation and demonstrations, as well as general information about NAG.

NAG is proud of the quality of the products and services that we provide, and we always welcome comments on improvements you require. Please feel free to contact us to discuss such matters at any time.

Documentation

When you purchase the NAG DLLs from us, you are provided with a copy of the full Fortran Library manual in PDF format; printed documentation can be purchased separately. The manual is the principal form of documentation for the library. It has the same chapter structure as the library; each chapter of routines in the library has a corresponding chapter (of the same name) in the PDF manual. General introductory documents and indexes can be found at the beginning of the manual.

Each chapter has an introduction, giving a general background to the types of problem that can be solved, and guidance on selecting the most suitable routines. There is also a list of contents and then a routine document for each documented routine in the chapter.

For users of the DLLs Windows Help documentation is also provided. This is useful for package builders who can link directly to the help file. Additionally help files are provided to allow Visual Basic Declare statements and C Header file information and Fortran 90 interface blocks to be cut and pasted into user programs.

Technical Support

Full product support can be purchased for 18% of the perpetual licence fee. This service provides you with technical support via the NAG Response Centres. When you contact the Response Centre details of your call will be taken and a log number (a unique reference to identify your query within NAG) will be given. Your details will be passed onto the relevant experts within NAG who can then fully investigate your problem or query before replying. You can be assured that the answers you receive will have been thoroughly researched, usually by the very professionals involved in the development of the product. The NAG

Response Centres can also be used as a general enquiry service. Please see www.nag.co.uk/contact_us.asp for the contact details of each centre.

New Releases

New releases are issued on a regular basis and sent out free of charge to customers subscribing to our support service. The releases are designed primarily to enhance functionality, but also to include any necessary error corrections. A copy of the full manual is supplied with every new software release.

Further information on the web

Main Web Sites

The NAG web sites can be found at www.nag.co.uk and www.nag.com

General DLL Information **www.nag.co.uk/numeric/num_DLLhelp.asp**

The Fortran DLLs

Fortran DLL Installer's Note — Windows 95/NT Dynamic Link Libraries (32-bit)
www.nag.co.uk/numeric/FLOLCH.html

Fortran DLL Users' Note — Windows 95/NT Dynamic Link Libraries (32-bit)
www.nag.co.uk/numeric/FLOLCH.html

Fortran Library on-line documentation
www.nag.co.uk/Local/registry/register_FL_doc.html

The C DLLs

C Library DLL Installer's Note — Windows 95/NT Dynamic Link Library (32-bit) (Microsoft Visual C++)
www.nag.co.uk/numeric/CL.html

C Library DLL Users' Note — Windows 95/NT Dynamic Link Library (32-bit) (Microsoft Visual C++)
www.nag.co.uk/numeric/CL.html

C Library on-line documentation
www.nag.co.uk/Local/registry/register_CL_doc.html

Using 32-bit NAG C DLL functions from Microsoft Visual Basic and Microsoft Office 97
www.nag.co.uk/numeric/VB_SUPPORT.HTML

Section 5 CALLING THE DLLS FROM OTHER LANGUAGES

Microsoft Excel and Microsoft Visual Basic for Applications

Microsoft Excel spreadsheet users can use the power of NAG DLLs to boost the capabilities of the Excel spreadsheet. The secret lies in inserting the appropriate Declare statements in a module sheet of an Excel workbook.

The Fortran-based DLLs are especially suited for use with Excel 7/97. The basic Fortran types of INTEGER, LOGICAL and DOUBLE PRECISION map directly on to the Excel types of Long, Long and Double respectively. In addition Fortran array storage is directly comparable to the array storage conventions used in Excel. So, in general, the NAG routines may be interfaced readily to Excel.

Programming Excel 7

If you were to open a fresh workbook in Excel, the worksheet displayed will, by default, be a standard worksheet consisting of cells into which the user puts values or formulae. All you would need to do is click on the insert menu and choose a module from the macro sub-menu. You will then have opened a module sheet, which allows you to enter code to program Excel. The language used is Visual Basic for Applications (VBA).

To take a simple example, suppose we wish to use the NAG routine S14AAF using Excel 7. By using the NAG Browser or the vbheader.hlp file, copy the Declare statement for S14AAF onto the Clipboard. Open the Excel workbook and turn to a module sheet. If one does not exist, click on the insert Menu and choose macro then module from the resulting options. When the module sheet appears, paste in the Declare statement from the Clipboard. Amend the Declare statement as necessary. In particular the Lib component may need changing if the Libraries have been installed to a directory not on the search path, or if the DLL names have been changed. At this point it is worth making sure that that the command 'Option Base 1' is at the top of the module sheet. This ensures that any VBA arrays declared start their indices at 1, making them compatible with the Fortran routines.

Unlike in C, VBA arrays are stored by column and are thus compatible with Fortran. Experienced VBA programmers may now use the Fortran routine as though it had been written in VBA, subject to the conventions contained in the Declare statement.

The simplest of the NAG Functions may be used directly. To see this, you would move to an ordinary worksheet in this workbook and select a cell before clicking on the Function Wizard – fx on the Excel Ribbon. In the ‘User Defined’ function category, you would find the NAG S14AAF routine. Proceed as prompted by the Wizard, putting the dummy value 0 for IFAIL when finally prompted for this. If you have typed in valid input for the parameters, the function is now evaluated and placed in the cell selected. (You might wish to type in the value 1.25 for X, 0 for IFAIL and verify that the cell value is now 0.9064.) Care has to be taken with string parameters in VB since these and the Fortran CHARACTER types are not entirely compatible. Note the importance of the ByVal qualifier in the Declare statement and the mandatory extra argument immediately following the string. The extra argument defines the length of the string and is of type Long. It too is a ByVal parameter. Fortunately NAG routines by and large use CHARACTER arguments for input rather than output, so the most awkward problem is avoided. COMPLEX arguments may be handled using the user-defined

type:

```
Type COMPLEX
  REAL_PART As Double
  IMAG_PART As Double
End Type
```

To use NAG routines with subroutine or function names in their arguments see the discussion under the section on 'Microsoft Visual Basic'.

Microsoft Visual Basic

Microsoft Visual Basic (VB) versions 4/5/6 and Microsoft Visual Basic for Applications (VBA) have many similarities, so much of the VBA-specific information above applies directly to VB. Note especially the remarks about array conventions and string handling. Neither VBA nor VB 4 provide a mechanism for passing procedure arguments to a DLL. Please see our web site at www.nag.co.uk/numeric/MSVB.asp for more information.

Microsoft C++

A great number of systems allow the C programmer to call other language routines. Indeed the ANSI standard definition of C provides a powerful argument checking facility that, given the correct definition of function prototypes, can facilitate cross-language communication between C and, say, Fortran.

The NAG Fortran DLLs may be used with care from within a C or C++ environment. To assist the user make the mapping between Fortran and C types, a set of C Header files is provided in the user's program to allow the C compiler to check argument passage. Such a Header file has been created for the NAG Fortran Library. This was done automatically from the library source code in order to ensure its correctness. A document explaining how to call Fortran routines from C using the NAG C Header files can be found at: www.nag.co.uk/numeric/FLOLCH/CHW3206DA.html

To see examples illustrating the use of arrays, character strings and functions please refer to our web site at www.nag.co.uk/numeric/MSC++.asp

Alternatively the NAG C Library provides over 400 user-callable functions. As with all NAG products, you can rely on the fact that these functions are of the highest quality, fully tested and certified. For further information on using the C-based DLLs contact your local NAG office or visit our web site at www.nag.co.uk/numeric/CL.html

Borland C++

NAG has carried out selected tests calling the Fortran DLLs from Borland C++ version 5.0. From these it has been concluded that the DLLs may be used in conjunction with this Borland C++ compiler. The discussion in the section on 'Microsoft C' applies equally to Borland. Borland import libraries are not supplied, but may easily be constructed from the DLLs as follows:

```
Imp def name.def name.dll
Imp lib name.lib name.def
```

Where name denotes the name of the NAG DLL, e.g. NAGAC. The first statement constructs a module definitions file, name.def, and the second takes this module definition file and constructs an import library, name.lib.

The C Library can also be called from Borland C/C++. Functions in the NAG C DLL can be called from C/C++ code compiled with the Borland C++ compiler, version 5.2. The following conditions must be met:

- nagd_bc.lib must be specified as the import library
- The function which calls the NAG C DLL must be compiled with _stdcall calling convention
- The linker option allowing call by ordinal number must be set
- The non-incremental linker must be used

Borland Delphi

One important point to bear in mind when calling the NAG DLLs from Delphi is that the actual parameters must be of type var. This is because the Fortran calling convention requires parameters to be passed by reference and not by value. It is not necessary to include the library itself in the compilation linker list in Delphi; the DLL can be called straight from the code itself and the compiler will link it automatically.

The reference to the DLL is as a procedure or function, defined as external in the Delphi code. This procedure needs to have the same name as the DLL routine called. Delphi is case sensitive, so the NAG name must be in capital letters (the Delphi name construct may be used to change this if desired).

Further information on how to call the NAG Fortran DLLs from Borland Delphi can be found at www.nag.co.uk/numeric/BorlandDelphi.asp and at www.nag.co.uk/numeric/BorlandDelphi.html

PowerBuilder

The following text has been supplied by Sybase.

Description

External functions are functions that are written in languages other than PowerScript and stored in DLLs, known as shared libraries on Macintosh and UNIX. You can use external functions that are written in any language that supports dynamic libraries. Before you can use an external function in a script, you must declare it. You can declare two types of external functions.

1) **Global external functions:** These are available anywhere in the application.

2) **Local external functions:** These are defined for a particular type of window, menu, user object, or user-defined function. These functions are part of the object's definition and can always be used in scripts for the object itself. You can also choose to make these functions accessible to other scripts.

PowerBuilder for Unicode

If you call external functions in an application in PowerBuilder for Unicode, the functions must be defined and compiled with Unicode support. All strings must be passed as Unicode strings. If you call Windows API functions, use the Unicode version of the function name. For example, use FindWindowW (W for wide) instead of FindWindowA (A for ANSI).

To understand how to declare and call an external function, please see our web site at www.nag.co.uk/numeric/PowerBuilder.asp

Watcom C/C++

Functions in the NAG C Library DLL can be called from C/C++ code compiled with the Watcom C/C++ compiler, version 11.0. The nagc.lib or nagcd.lib import libraries can be added to your Watcom IDE project or can be used with the Watcom wlink command.
The 32-bit Fortran DLLs can also be used with the Watcom Fortran Compiler if the SC switch is used to alter the default calling convention of the Watcom compiler.

Section 6 THE NUMERICAL AND STATISTICAL ROUTINES AVAILABLE

Chapter A00 - Library Identification

A00AAF Prints details of the NAG Fortran Library implementation

Chapter A02 - Complex Arithmetic

A02AAF Square root of complex number

A02ABF Modulus of complex number

A02ACF Quotient of two complex numbers

Chapter C02 - Zeros of Polynomials

C02AFF All zeros of complex polynomial, modified Laguerre method

C02AGF All zeros of real polynomial, modified Laguerre method

C02AHF All zeros of complex quadratic

C02AJF All zeros of real quadratic

Chapter C05 - Roots of One or More Transcendental Equations

C05ADF Zero of continuous function in given interval, Bus and Dekker algorithm

C05AGF Zero of continuous function, Bus and Dekker algorithm, from given starting value, binary search for interval

C05AJF Zero of continuous function, continuation method, from a given starting value

C05AVF Binary search for interval containing zero of continuous function (reverse communication)

C05AXF Zero of continuous function by continuation method, from given starting value (reverse communication)

C05AZF Zero in given interval of continuous function by Bus and Dekker algorithm (reverse communication)

C05NBF Solution of system of nonlinear equations using function values only (easy-to-use)

C05NCF Solution of system of nonlinear equations using function values only (comprehensive)

C05NDF Solution of system of nonlinear equations using function values only (reverse communication)

C05PBF Solution of system of nonlinear equations using first derivatives (easy-to-use)

C05PCF Solution of system of nonlinear equations using first derivatives (comprehensive)

C05PDF Solution of system of nonlinear equations using first derivatives (reverse communication)

C05ZAF Check user's routine for calculating first derivatives

Chapter C06 - Summation of Series

C06BAF Acceleration of convergence of sequence, Shanks' transformation and epsilon algorithm

C06DBF Sum of a Chebyshev series

C06EAF Single one-dimensional real discrete Fourier transform, no extra workspace

C06EBF Single one-dimensional Hermitian discrete Fourier transform, no extra workspace

C06ECF Single one-dimensional complex discrete Fourier transform, no extra workspace

C06EKF Circular convolution or correlation of two real vectors, no extra workspace

C06FAF Single one-dimensional real discrete Fourier transform, extra workspace for greater speed

C06FBF Single one-dimensional Hermitian discrete Fourier transform, extra workspace for greater speed

C06FCF Single one-dimensional complex discrete Fourier transform, extra workspace for greater speed

C06FFF One-dimensional complex discrete Fourier transform of multi-dimensional data

C06FJF Multi-dimensional complex discrete Fourier transform of multi-dimensional data

C06FKF Circular convolution or correlation of two real vectors, extra workspace for greater speed

C06FPF Multiple one-dimensional real discrete Fourier transforms

C06FQF	Multiple one-dimensional Hermitian discrete Fourier transforms
C06FRF	Multiple one-dimensional complex discrete Fourier transforms
C06FUF	Two-dimensional complex discrete Fourier transform
C06FXF	Three-dimensional complex discrete Fourier transform
C06GBF	Complex conjugate of Hermitian sequence
C06GCF	Complex conjugate of complex sequence
C06GQF	Complex conjugate of multiple Hermitian sequences
C06GSF	Convert Hermitian sequences to general complex sequences
C06HAF	Discrete sine transform
C06HBF	Discrete cosine transform
C06HCF	Discrete quarter-wave sine transform
C06HDF	Discrete quarter-wave cosine transform
C06LAF	Inverse Laplace transform, Crump's method
C06LBF	Inverse Laplace transform, modified Weeks' method
C06LCF	Evaluate inverse Laplace transform as computed by C06LBF
C06PAF	Single one-dimensional real and Hermitian complex discrete Fourier transform, using complex data format for Hermitian sequences
C06PCF	Single one-dimensional complex discrete Fourier transform, complex data format
C06PFF	One-dimensional complex discrete Fourier transform of multi-dimensional data (using complex data type)
C06PJF	Multi-dimensional complex discrete Fourier transform of multi-dimensional data (using complex data type)
C06PKF	Circular convolution or correlation of two complex vectors
C06PPF	Multiple one-dimensional real and Hermitian complex discrete Fourier transforms, using complex data format for Hermitian sequences
C06PQF	Multiple one-dimensional real and Hermitian complex discrete Fourier transforms, using complex data format for Hermitian sequences and sequences stored as columns
C06PRF	Multiple one-dimensional complex discrete Fourier transforms using complex data format
C06PSF	Multiple one-dimensional complex discrete Fourier transforms using complex data format and sequences stored as columns
C06PUF	Two-dimensional complex discrete Fourier transform, complex data format
C06PXF	Three-dimensional complex discrete Fourier transform, complex data format
C06RAF	Discrete sine transform (easy-to-use)
C06RBF	Discrete cosine transform (easy-to-use)
C06RCF	Discrete quarter-wave sine transform (easy-to-use)
C06RDF	Discrete quarter-wave cosine transform (easy-to-use)

Chapter D01 - Quadrature

D01AHF	One-dimensional quadrature, adaptive, finite interval, strategy due to Patterson, suitable for well-behaved integrands
D01AJF	One-dimensional quadrature, adaptive, finite interval, strategy due to Piessens and de Doncker, allowing for badly-behaved integrands
D01AKF	One-dimensional quadrature, adaptive, finite interval, method suitable for oscillating functions
D01ALF	One-dimensional quadrature, adaptive, finite interval, allowing for singularities at user-specified break-points
D01AMF	One-dimensional quadrature, adaptive, infinite or semi-infinite interval
D01ANF	One-dimensional quadrature, adaptive, finite interval, weight function cos (ωx) or sin (ωx)
D01APF	One-dimensional quadrature, adaptive, finite interval, weight function with end-point singularities of algebraico-logarithmic type

D01AQF	One-dimensional quadrature, adaptive, finite interval, weight function $1/(x - c)$, Cauchy principal value (Hilbert transform)
D01ARF	One-dimensional quadrature, non-adaptive, finite interval with provision for indefinite integrals
D01ASF	One-dimensional quadrature, adaptive, semi-infinite interval, weight function $\cos(\omega x)$ or $\sin(\omega x)$.
D01ATF	One-dimensional quadrature, adaptive, finite interval, variant of D01AJF efficient on vector machines
D01AUF	One-dimensional quadrature, adaptive, finite interval, variant of D01AKF efficient on vector machines
D01BAF	One-dimensional Gaussian quadrature
D01BBF	Pre-computed weights and abscissae for Gaussian quadrature rules, restricted choice of rule
D01BCF	Calculation of weights and abscissae for Gaussian quadrature rules, general choice of rule
D01BDF	One-dimensional quadrature, non-adaptive, finite interval
D01DAF	Two-dimensional quadrature, finite region
D01EAF	Multi-dimensional adaptive quadrature over hyper-rectangle, multiple integrands
D01FBF	Multi-dimensional Gaussian quadrature over hyper-rectangle
D01FCF	Multi-dimensional adaptive quadrature over hyper-rectangle
D01FDF	Multi-dimensional quadrature, Sag Szekeres method, general product region or *n*-sphere
D01GAF	One-dimensional quadrature, integration of function defined by data values, Gill Miller method
D01GBF	Multi-dimensional quadrature over hyper-rectangle, Monte Carlo method
D01GCF	Multi-dimensional quadrature, general product region, number-theoretic method
D01GDF	Multi-dimensional quadrature, general product region, number-theoretic method, variant of D01GCF efficient on vector machines
D01GYF	Korobov optimal coefficients for use in D01GCF or D01GDF, when number of points is prime
D01GZF	Korobov optimal coefficients for use in D01GCF or D01GDF, when number of points is product of two primes
D01JAF	Multi-dimensional quadrature over an *n*-sphere, allowing for badly-behaved integrands
D01PAF	Multi-dimensional quadrature over an *n*-simplex

Chapter D02 - Ordinary Differential Equations

D02AGF	ODEs, boundary value problem, shooting and matching technique, allowing interior matching point, general parameters to be determined
D02BGF	ODEs, IVP, Runge–Kutta–Merson method, until a component attains given value (simple driver)
D02BHF	ODEs, IVP, Runge–Kutta–Merson method, until function of solution is zero (simple driver)
D02BJF	ODEs, IVP, Runge–Kutta method, until function of solution is zero, integration over range with intermediate output (simple driver)
D02CJF	ODEs, IVP, Adams method, until function of solution is zero, intermediate output (simple driver)
D02EJF	ODEs, stiff IVP, BDF method, until function of solution is zero, intermediate output (simple driver)
D02GAF	ODEs, boundary value problem, finite difference technique with deferred correction, simple nonlinear problem
D02GBF	ODEs, boundary value problem, finite difference technique with deferred correction, general linear problem
D02HAF	ODEs, boundary value problem, shooting and matching, boundary values to be determined
D02HBF	ODEs, boundary value problem, shooting and matching, general parameters to be determined
D02JAF	ODEs, boundary value problem, collocation and least-squares, single *n*th-order linear equation
D02JBF	ODEs, boundary value problem, collocation and least-squares, system of first-order linear equations
D02KAF	Second-order Sturm–Liouville problem, regular system, finite range, eigenvalue only
D02KDF	Second-order Sturm–Liouville problem, regular/singular system, finite/infinite range, eigenvalue only, user-specified break-points
D02KEF	Second-order Sturm–Liouville problem, regular/singular system, finite/infinite range, eigenvalue and eigenfunction, user-specified break-points

D02LAF	Second-order ODEs, IVP, Runge–Kutta–Nystrom method
D02LXF	Second-order ODEs, IVP, set-up for D02LAF
D02LYF	Second-order ODEs, IVP, diagnostics for D02LAF
D02LZF	Second-order ODEs, IVP, interpolation for D02LAF
D02MVF	ODEs, IVP, DASSL method, set-up for D02M–N routines
D02MZF	ODEs, IVP, interpolation for D02M–N routines, natural interpolant
D02NBF	Explicit ODEs, stiff IVP, full Jacobian (comprehensive)
D02NCF	Explicit ODEs, stiff IVP, banded Jacobian (comprehensive)
D02NDF	Explicit ODEs, stiff IVP, sparse Jacobian (comprehensive)
D02NGF	Implicit/algebraic ODEs, stiff IVP, full Jacobian (comprehensive)
D02NHF	Implicit/algebraic ODEs, stiff IVP, banded Jacobian (comprehensive)
D02NJF	Implicit/algebraic ODEs, stiff IVP, sparse Jacobian (comprehensive)
D02NMF	Explicit ODEs, stiff IVP (reverse communication, comprehensive)
D02NNF	Implicit/algebraic ODEs, stiff IVP (reverse communication, comprehensive)
D02NRF	ODEs, IVP, for use with D02M–N routines, sparse Jacobian, enquiry routine
D02NSF	ODEs, IVP, for use with D02M–N routines, full Jacobian, linear algebra set-up
D02NTF	ODEs, IVP, for use with D02M–N routines, banded Jacobian, linear algebra set-up
D02NUF	ODEs, IVP, for use with D02M–N routines, sparse Jacobian, linear algebra set-up
D02NVF	ODEs, IVP, BDF method, set-up for D02M–N routines
D02NWF	ODEs, IVP, Blend method, set-up for D02M–N routines
D02NXF	ODEs, IVP, sparse Jacobian, linear algebra diagnostics, for use with D02M–N routines
D02NYF	ODEs, IVP, integrator diagnostics, for use with D02M–N routines
D02NZF	ODEs, IVP, set-up for continuation calls to integrator, for use with D02M–N routines
D02PCF	ODEs, IVP, Runge–Kutta method, integration over range with output
D02PDF	ODEs, IVP, Runge–Kutta method, integration over one step
D02PVF	ODEs, IVP, set-up for D02PCF and D02PDF
D02PWF	ODEs, IVP, resets end of range for D02PDF
D02PXF	ODEs, IVP, interpolation for D02PDF
D02PYF	ODEs, IVP, integration diagnostics for D02PCF and D02PDF
D02PZF	ODEs, IVP, error assessment diagnostics for D02PCF and D02PDF
D02QFF	ODEs, IVP, Adams method with root-finding (forward communication, comprehensive)
D02QGF	ODEs, IVP, Adams method with root-finding (reverse communication, comprehensive)
D02QWF	ODEs, IVP, set-up for D02QFF and D02QGF
D02QXF	ODEs, IVP, diagnostics for D02QFF and D02QGF
D02QYF	ODEs, IVP, root-finding diagnostics for D02QFF and D02QGF
D02QZF	ODEs, IVP, interpolation for D02QFF or D02QGF
D02RAF	ODEs, general nonlinear boundary value problem, finite difference technique with deferred correction, continuation facility
D02SAF	ODEs, boundary value problem, shooting and matching technique, subject to extra algebraic equations, general parameters to be determined
D02TGF	nth-order linear ODEs, boundary value problem, collocation and least-squares
D02TKF	ODEs, general nonlinear boundary value problem, collocation technique
D02TVF	ODEs, general nonlinear boundary value problem, set-up for D02TKF
D02TXF	ODEs, general nonlinear boundary value problem, continuation facility for D02TKF
D02TYF	ODEs, general nonlinear boundary value problem, interpolation for D02TKF
D02TZF	ODEs, general nonlinear boundary value problem, diagnostics for D02TKF

D02XJF ODEs, IVP, interpolation for D02M–N routines, natural interpolant

D02XKF ODEs, IVP, interpolation for D02M–N routines, C_1 interpolant

D02ZAF ODEs, IVP, weighted norm of local error estimate for D02M–N routines

Chapter D03 - Partial Differential Equations

D03EAF Elliptic PDE, Laplace's equation, two-dimensional arbitrary domain

D03EBF Elliptic PDE, solution of finite difference equations by SIP, five-point two-dimensional molecule, iterate to convergence

D03ECF Elliptic PDE, solution of finite difference equations by SIP for seven-point three-dimensional molecule, iterate to convergence

D03EDF Elliptic PDE, solution of finite difference equations by a multigrid technique

D03EEF Discretize a second-order elliptic PDE on a rectangle

D03FAF Elliptic PDE, Helmholtz equation, three-dimensional Cartesian co-ordinates

D03MAF Triangulation of plane region

D03PCF General system of parabolic PDEs, method of lines, finite differences, one space variable

D03PDF General system of parabolic PDEs, method of lines, Chebyshev C^0 collocation, one space variable

D03PEF General system of first-order PDEs, method of lines, Keller box discretisation, one space variable

D03PFF General system of convection-diffusion PDEs with source terms in conservative form, method of lines, upwind scheme using numerical flux function based on Riemann solver, one space variable

D03PHF General system of parabolic PDEs, coupled DAEs, method of lines, finite differences, one space variable

D03PJF General system of parabolic PDEs, coupled DAEs, method of lines, Chebyshev C^0 collocation, one space variable

D03PKF General system of first-order PDEs, coupled DAEs, method of lines, Keller box discretisation, one space variable

D03PLF General system of convection-diffusion PDEs with source terms in conservative form, coupled DAEs, method of lines, upwind scheme using numerical flux function based on Riemann solver, one space variable

D03PPF General system of parabolic PDEs, coupled DAEs, method of lines, finite differences, remeshing, one space variable

D03PRF General system of first-order PDEs, coupled DAEs, method of lines, Keller box discretisation, remeshing, one space variable

D03PSF General system of convection-diffusion PDEs with source terms in conservative form, coupled DAEs, method of lines, upwind scheme using numerical flux function based on Riemann solver, remeshing, one space variable

D03PUF Roe's approximate Riemann solver for Euler equations in conservative form, for use with D03PFF, D03PLF and D03PSF

D03PVF Osher's approximate Riemann solver for Euler equations in conservative form, for use with D03PFF, D03PLF and D03PSF

D03PWF Modified HLL Riemann solver for Euler equations in conservative form, for use with D03PFF, D03PLF and D03PSF

D03PXF Exact Riemann Solver for Euler equations in conservative form, for use with D03PFF, D03PLF and D03PSF

D03PYF PDEs, spatial interpolation with D03PDF or D03PJF

D03PZF PDEs, spatial interpolation with D03PCF, D03PEF, D03PFF, D03PHF, D03PKF, D03PLF, D03PPF, D03PRF or D03PSF

D03RAF General system of second-order PDEs, method of lines, finite differences, remeshing, two space variables, rectangular region

D03RBF General system of second-order PDEs, method of lines, finite differences, remeshing, two space variables, rectilinear region

D03RYF Check initial grid data in D03RBF

D03RZF Extract grid data from D03RBF

D03UAF Elliptic PDE, solution of finite difference equations by SIP, five-point two-dimensional molecule, one iteration

D03UBF Elliptic PDE, solution of finite difference equations by SIP, seven-point three-dimensional molecule, one iteration

Chapter D04 - Numerical Differentiation

D04AAF Numerical differentiation, derivatives up to order 14, function of one real variable

Chapter D05 - Integral Equations

D05AAF Linear non-singular Fredholm integral equation, second kind, split kernel

D05ABF Linear non-singular Fredholm integral equation, second kind, smooth kernel

D05BAF Nonlinear Volterra convolution equation, second kind

D05BDF Nonlinear convolution Volterra–Abel equation, second kind, weakly singular

D05BEF Nonlinear convolution Volterra–Abel equation, first kind, weakly singular

D05BWF Generate weights for use in solving Volterra equations

D05BYF Generate weights for use in solving weakly singular Abel-type equations

Chapter E01 - Interpolation

E01AAF Interpolated values, Aitken's technique, unequally spaced data, one variable

E01ABF Interpolated values, Everett's formula, equally spaced data, one variable

E01AEF Interpolating functions, polynomial interpolant, data may include derivative values, one variable

E01BAF Interpolating functions, cubic spline interpolant, one variable

E01BEF Interpolating functions, monotonicity-preserving, piecewise cubic Hermite, one variable

E01BFF Interpolated values, interpolant computed by E01BEF, function only, one variable

E01BGF Interpolated values, interpolant computed by E01BEF, function and first derivative, one variable

E01BHF Interpolated values, interpolant computed by E01BEF, definite integral, one variable

E01DAF Interpolating functions, fitting bicubic spline, data on rectangular grid

E01RAF Interpolating functions, rational interpolant, one variable

E01RBF Interpolated values, evaluate rational interpolant computed by E01RAF, one variable

E01SAF Interpolating functions, method of Renka and Cline, two variables

E01SBF Interpolated values, evaluate interpolant computed by E01SAF, two variables

E01SEF Interpolating functions, modified Shepard's method, two variables

E01SFF Interpolated values, evaluate interpolant computed by E01SEF, two variables

E01SGF Interpolating functions, modified Shepard's method, two variables

E01SHF Interpolated values, evaluate interpolant computed by E01SGF, function and first derivatives, two variables

E01TGF Interpolating functions, modified Shepard's method, three variables

E01THF Interpolated values, evaluate interpolant computed by E01TGF, function and first derivatives, three variables

Chapter E02 - Curve and Surface Fitting

E02ACF Minimax curve fit by polynomials

E02ADF Least-squares curve fit, by polynomials, arbitrary data points

E02AEF Evaluation of fitted polynomial in one variable from Chebyshev series form (simplified parameter list)

E02AFF Least-squares polynomial fit, special data points (including interpolation)

E02AGF Least-squares polynomial fit, values and derivatives may be constrained, arbitrary data points

E02AHF Derivative of fitted polynomial in Chebyshev series form

E02AJF Integral of fitted polynomial in Chebyshev series form

E02AKF Evaluation of fitted polynomial in one variable from Chebyshev series form

E02BAF Least-squares curve cubic spline fit (including interpolation)
E02BBF Evaluation of fitted cubic spline, function only
E02BCF Evaluation of fitted cubic spline, function and derivatives
E02BDF Evaluation of fitted cubic spline, definite integral
E02BEF Least-squares cubic spline curve fit, automatic knot placement
E02CAF Least-squares surface fit by polynomials, data on lines
E02CBF Evaluation of fitted polynomial in two variables
E02DAF Least-squares surface fit, bicubic splines
E02DCF Least-squares surface fit by bicubic splines with automatic knot placement, data on rectangular grid
E02DDF Least-squares surface fit by bicubic splines with automatic knot placement, scattered data
E02DEF Evaluation of fitted bicubic spline at a vector of points
E02DFF Evaluation of fitted bicubic spline at a mesh of points
E02GAF L_1-approximation by general linear function
E02GBF L_1-approximation by general linear function subject to linear inequality constraints
E02GCF L_∞-approximation by general linear function
E02RAF Padé-approximants
E02RBF Evaluation of fitted rational function as computed by E02RAF
E02ZAF Sort two-dimensional data into panels for fitting bicubic splines

Chapter E04 - Minimizing or Maximizing a Function

E04ABF Minimum, function of one variable using function values only
E04BBF Minimum, function of one variable, using first derivative
E04CCF Unconstrained minimum, simplex algorithm, function of several variables using function values only (comprehensive)
E04DGF Unconstrained minimum, preconditioned conjugate gradient algorithm, function of several variables using first derivatives (comprehensive)
E04DJF Read optional parameter values for E04DGF from external file
E04DKF Supply optional parameter values to E04DGF
E04FCF Unconstrained minimum of a sum of squares, combined Gauss–Newton and modified Newton algorithm using function values only (comprehensive)
E04FYF Unconstrained minimum of a sum of squares, combined Gauss–Newton and modified Newton algorithm using function values only (easy-to-use)
E04GBF Unconstrained minimum of a sum of squares, combined Gauss–Newton and quasi-Newton algorithm using first derivatives (comprehensive)
E04GDF Unconstrained minimum of a sum of squares, combined Gauss–Newton and modified Newton algorithm using first derivatives (comprehensive)
E04GYF Unconstrained minimum of a sum of squares, combined Gauss–Newton and quasi-Newton algorithm, using first derivatives (easy-to-use)
E04GZF Unconstrained minimum of a sum of squares, combined Gauss–Newton and modified Newton algorithm using first derivatives (easy-to-use)
E04HCF Check user's routine for calculating first derivatives of function
E04HDF Check user's routine for calculating second derivatives of function
E04HEF Unconstrained minimum of a sum of squares, combined Gauss–Newton and modified Newton algorithm, using second derivatives (comprehensive)
E04HYF Unconstrained minimum of a sum of squares, combined Gauss–Newton and modified Newton algorithm, using second derivatives (easy-to-use)
E04JYF Minimum, function of several variables, quasi-Newton algorithm, simple bounds, using function values only (easy-to-use)
E04KDF Minimum, function of several variables, modified Newton algorithm, simple bounds, using first derivatives (comprehensive)

E04KYF	Minimum, function of several variables, quasi-Newton algorithm, simple bounds, using first derivatives (easy-to-use)
E04KZF	Minimum, function of several variables, modified Newton algorithm, simple bounds, using first derivatives (easy-to-use)
E04LBF	Minimum, function of several variables, modified Newton algorithm, simple bounds, using first and second derivatives (comprehensive)
E04LYF	Minimum, function of several variables, modified Newton algorithm, simple bounds, using first and second derivatives (easy-to-use)
E04MFF	LP problem (dense)
E04MGF	Read optional parameter values for E04MFF from external file
E04MHF	Supply optional parameter values to E04MFF
E04MZF	Converts MPSX data file defining LP or QP problem to format required by E04NKF
E04NCF	Convex QP problem or linearly-constrained linear least-squares problem (dense)
E04NDF	Read optional parameter values for E04NCF from external file
E04NEF	Supply optional parameter values to E04NCF
E04NFF	QP problem (dense)
E04NGF	Read optional parameter values for E04NFF from external file
E04NHF	Supply optional parameter values to E04NFF
E04NKF	LP or QP problem (sparse)
E04NLF	Read optional parameter values for E04NKF from external file
E04NMF	Supply optional parameter values to E04NKF
E04UCF	Minimum, function of several variables, sequential QP method, nonlinear constraints, using function values and optionally first derivatives (forward communication, comprehensive)
E04UDF	Read optional parameter values for E04UCF or E04UFF from external file
E04UEF	Supply optional parameter values to E04UCF or E04UFF
E04UFF	Minimum, function of several variables, sequential QP method, nonlinear constraints, using function values and optionally first derivatives (reverse communication, comprehensive)
E04UGF	NLP problem (sparse)
E04UHF	Read optional parameter values for E04UGF from external file
E04UJF	Supply optional parameter values to E04UGF
E04UNF	Minimum of a sum of squares, nonlinear constraints, sequential QP method, using function values and optionally first derivatives (comprehensive)
E04UQF	Read optional parameter values for E04UNF from external file
E04URF	Supply optional parameter values to E04UNF
E04XAF	Estimate (using numerical differentiation) gradient and/or Hessian of a function
E04YAF	Check user's routine for calculating Jacobian of first derivatives
E04YBF	Check user's routine for calculating Hessian of a sum of squares
E04YCF	Covariance matrix for nonlinear least-squares problem (unconstrained)
E04ZCF	Check user's routines for calculating first derivatives of function and constraints

Chapter F01 - Matrix Factorizations

F01ABF	Inverse of real symmetric positive-definite matrix using iterative refinement
F01ADF	Inverse of real symmetric positive-definite matrix
F01BLF	Pseudo-inverse and rank of real m by n matrix ($m \geq n$)
F01BRF	LU factorization of real sparse matrix
F01BSF	LU factorization of real sparse matrix with known sparsity pattern
F01BUF	$ULDL^TU^T$ factorization of real symmetric positive-definite band matrix
F01BVF	Reduction to standard form, generalized real symmetric-definite banded eigenproblem
F01CKF	Matrix multiplication

F01CRF Matrix transposition
F01CTF Sum or difference of two real matrices, optional scaling and transposition
F01CWF Sum or difference of two complex matrices, optional scaling and transposition
F01LEF LU factorization of real tridiagonal matrix
F01LHF LU factorization of real almost block diagonal matrix
F01MCF LDL^T factorization of real symmetric positive-definite variable-bandwidth matrix
F01QGF RQ factorization of real m by n upper trapezoidal matrix ($m \leq n$)
F01QJF RQ factorization of real m by n matrix ($m \leq n$)
F01QKF Operations with orthogonal matrices, form rows of Q, after RQ factorization by F01QJF
F01RGF RQ factorization of complex m by n upper trapezoidal matrix ($m \leq n$)
F01RJF RQ factorization of complex m by n matrix ($m \leq n$)
F01RKF Operations with unitary matrices, form rows of Q, after RQ factorization by F01RJF
F01ZAF Convert real matrix between packed triangular and square storage schemes
F01ZBF Convert complex matrix between packed triangular and square storage schemes
F01ZCF Convert real matrix between packed banded and rectangular storage schemes
F01ZDF Convert complex matrix between packed banded and rectangular storage schemes

Chapter F02 - Eigenvalues and Eigenvectors

F02BJF All eigenvalues and optionally eigenvectors of generalized eigenproblem by QZ algorithm, real matrices (Black Box)
F02EAF All eigenvalues and Schur factorization of real general matrix (Black Box)
F02EBF All eigenvalues and eigenvectors of real general matrix (Black Box)
F02ECF Selected eigenvalues and eigenvectors of real nonsymmetric matrix (Black Box)
F02FAF All eigenvalues and eigenvectors of real symmetric matrix (Black Box)
F02FCF Selected eigenvalues and eigenvectors of real symmetric matrix (Black Box)
F02FDF All eigenvalues and eigenvectors of real symmetric-definite generalized problem (Black Box)
F02FHF All eigenvalues of generalized banded real symmetric-definite eigenproblem (Black Box)
F02FJF Selected eigenvalues and eigenvectors of sparse symmetric eigenproblem (Black Box)
F02GAF All eigenvalues and Schur factorization of complex general matrix (Black Box)
F02GBF All eigenvalues and eigenvectors of complex general matrix (Black Box)
F02GCF Selected eigenvalues and eigenvectors of complex nonsymmetric matrix (Black Box)
F02GJF All eigenvalues and optionally eigenvectors of generalized complex eigenproblem by QZ algorithm (Black Box)
F02HAF All eigenvalues and eigenvectors of complex Hermitian matrix (Black Box)
F02HCF Selected eigenvalues and eigenvectors of complex Hermitian matrix (Black Box)
F02HDF All eigenvalues and eigenvectors of complex Hermitian-definite generalized problem (Black Box)
F02SDF Eigenvector of generalized real banded eigenproblem by inverse iteration
F02WDF QR factorization, possibly followed by SVD
F02WEF SVD of real matrix (Black Box)
F02WUF SVD of real upper triangular matrix (Black Box)
F02XEF SVD of complex matrix (Black Box)
F02XUF SVD of complex upper triangular matrix (Black Box)

Chapter F03 - Determinants

F03AAF Determinant of real matrix (Black Box)
F03ABF Determinant of real symmetric positive-definite matrix (Black Box)
F03ACF Determinant of real symmetric positive-definite band matrix (Black Box)
F03ADF Determinant of complex matrix (Black Box)

F03AEF LL^T factorization and determinant of real symmetric positive-definite matrix

F03AFF LU factorization and determinant of real matrix

Chapter F04 - Simultaneous Linear Equations

F04AAF Solution of real simultaneous linear equations with multiple right-hand sides (Black Box)

F04ABF Solution of real symmetric positive-definite simultaneous linear equations with multiple right-hand sides using iterative refinement (Black Box)

F04ACF Solution of real symmetric positive-definite banded simultaneous linear equations with multiple right-hand sides (Black Box)

F04ADF Solution of complex simultaneous linear equations with multiple right-hand sides (Black Box)

F04AEF Solution of real simultaneous linear equations with multiple right-hand sides using iterative refinement (Black Box)

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)

F04AMF Least-squares solution of m real equations in n unknowns, rank = n, $m \geq n$ using iterative refinement (Black Box)

F04ARF Solution of real simultaneous linear equations, one right-hand side (Black Box)

F04ASF Solution of real symmetric positive-definite simultaneous linear equations, one right-hand side using iterative refinement (Black Box)

F04ATF Solution of real simultaneous linear equations, one right-hand side using iterative refinement (Black Box)

F04AXF Solution of real sparse simultaneous linear equations (coefficient matrix already factorized)

F04EAF Solution of real tridiagonal simultaneous linear equations, one right-hand side (Black Box)

F04FAF Solution of real symmetric positive-definite tridiagonal simultaneous linear equations, one right-hand side (Black Box)

F04FEF Solution of the Yule–Walker equations for real symmetric positive-definite Toeplitz matrix, one right-hand side

F04FFF Solution of real symmetric positive-definite Toeplitz system, one right-hand side

F04JAF Minimal least-squares solution of m real equations in n unknowns, rank $\leq n$, $m \geq n$

F04JDF Minimal least-squares solution of m real equations in n unknowns, rank $\leq n$, $m \geq n$

F04JGF Least-squares (if rank = n) or minimal least-squares (if rank < n) solution of m real equations in n unknowns, rank $\leq n$, $m \geq n$

F04JLF Real general Gauss–Markov linear model (including weighted least-squares)

F04JMF Equality-constrained real linear least-squares problem

F04KLF Complex general Gauss–Markov linear model (including weighted least-squares)

F04KMF Equality-constrained complex linear least-squares problem

F04LEF Solution of real tridiagonal simultaneous linear equations (coefficient matrix already factorized by F01LEF)

F04LHF Solution of real almost block diagonal simultaneous linear equations (coefficient matrix already factorized by F01LHF)

F04MCF Solution of real symmetric positive-definite variable-bandwidth simultaneous linear equations (coefficient matrix already factorized by F01MCF)

F04MEF Update solution of the Yule–Walker equations for real symmetric positive-definite Toeplitz matrix

F04MFF Update solution of real symmetric positive-definite Toeplitz system

F04QAF Sparse linear least-squares problem, m real equations in n unknowns

F04YAF Covariance matrix for linear least-squares problems, m real equations in n unknowns

F04YCF Norm estimation (for use in condition estimation), real matrix
F04ZCF Norm estimation (for use in condition estimation), complex matrix

Chapter F05 - Orthogonalisation

F05AAF Gram–Schmidt orthogonalisation of n vectors of order m

Chapter F06 - Linear Algebra Support Routines

F06AAF (DROTG) Generate real plane rotation
F06BAF Generate real plane rotation, storing tangent
F06BCF Recover cosine and sine from given real tangent
F06BEF Generate real Jacobi plane rotation
F06BHF Apply real similarity rotation to 2 by 2 symmetric matrix
F06BLF Compute quotient of two real scalars, with overflow flag
F06BMF Compute Euclidean norm from scaled form
F06BNF Compute square root of $(a^2 + b^2)$, real a and b
F06BPF Compute eigenvalue of 2 by 2 real symmetric matrix
F06CAF Generate complex plane rotation, storing tangent, real cosine
F06CBF Generate complex plane rotation, storing tangent, real sine
F06CCF Recover cosine and sine from given complex tangent, real cosine
F06CDF Recover cosine and sine from given complex tangent, real sine
F06CHF Apply complex similarity rotation to 2 by 2 Hermitian matrix
F06CLF Compute quotient of two complex scalars, with overflow flag
F06DBF Broadcast scalar into integer vector
F06DFF Copy integer vector
F06EAF (DDOT) Dot product of two real vectors
F06ECF (DAXPY) Add scalar times real vector to real vector
F06EDF (DSCAL) Multiply real vector by scalar
F06EFF (DCOPY) Copy real vector
F06EGF (DSWAP) Swap two real vectors
F06EJF (DNRM2) Compute Euclidean norm of real vector
F06EKF (DASUM) Sum absolute values of real vector elements
F06EPF (DROT) Apply real plane rotation
F06ERF (DDOTI) Dot product of two real sparse vectors
F06ETF (DAXPYI) Add scalar times real sparse vector to real sparse vector
F06EUF (DGTHR) Gather real sparse vector
F06EVF (DGTHRZ) Gather and set to zero real sparse vector
F06EWF (DSCTR) Scatter real sparse vector
F06EXF (DROTI) Apply plane rotation to two real sparse vectors
F06FAF Compute cosine of angle between two real vectors
F06FBF Broadcast scalar into real vector
F06FCF Multiply real vector by diagonal matrix
F06FDF Multiply real vector by scalar, preserving input vector
F06FGF Negate real vector
F06FJF Update Euclidean norm of real vector in scaled form
F06FKF Compute weighted Euclidean norm of real vector
F06FLF Elements of real vector with largest and smallest absolute value
F06FPF Apply real symmetric plane rotation to two vectors
F06FQF Generate sequence of real plane rotations

F06FRF	Generate real elementary reflection, NAG style
F06FSF	Generate real elementary reflection, LINPACK style
F06FTF	Apply real elementary reflection, NAG style
F06FUF	Apply real elementary reflection, LINPACK style
F06GAF	(ZDOTU) Dot product of two complex vectors, unconjugated
F06GBF	(ZDOTC) Dot product of two complex vectors, conjugated
F06GCF	(ZAXPY) Add scalar times complex vector to complex vector
F06GDF	(ZSCAL) Multiply complex vector by complex scalar
F06GFF	(ZCOPY) Copy complex vector
F06GGF	(ZSWAP) Swap two complex vectors
F06GRF	(ZDOTUI) Dot product of two complex sparse vector, unconjugated
F06GSF	(ZDOTCI) Dot product of two complex sparse vector, conjugated
F06GTF	(ZAXPYI) Add scalar times complex sparse vector to complex sparse vector
F06GUF	(ZGTHR) Gather complex sparse vector
F06GVF	(ZGTHRZ) Gather and set to zero complex sparse vector
F06GWF	(ZSCTR) Scatter complex sparse vector
F06HBF	Broadcast scalar into complex vector
F06HCF	Multiply complex vector by complex diagonal matrix
F06HDF	Multiply complex vector by complex scalar, preserving input vector
F06HGF	Negate complex vector
F06HPF	Apply complex plane rotation
F06HQF	Generate sequence of complex plane rotations
F06HRF	Generate complex elementary reflection
F06HTF	Apply complex elementary reflection
F06JDF	(ZDSCAL) Multiply complex vector by real scalar
F06JJF	(DZNRM2) Compute Euclidean norm of complex vector
F06JKF	(DZASUM) Sum absolute values of complex vector elements
F06JLF	(IDAMAX) Index, real vector element with largest absolute value
F06JMF	(IZAMAX) Index, complex vector element with largest absolute value
F06KCF	Multiply complex vector by real diagonal matrix
F06KDF	Multiply complex vector by real scalar, preserving input vector
F06KFF	Copy real vector to complex vector
F06KJF	Update Euclidean norm of complex vector in scaled form
F06KLF	Last non-negligible element of real vector
F06KPF	Apply real plane rotation to two complex vectors
F06PAF	(DGEMV) Matrix-vector product, real rectangular matrix
F06PBF	(DGBMV) Matrix-vector product, real rectangular band matrix
F06PCF	(DSYMV) Matrix-vector product, real symmetric matrix
F06PDF	(DSBMV) Matrix-vector product, real symmetric band matrix
F06PEF	(DSPMV) Matrix-vector product, real symmetric packed matrix
F06PFF	(DTRMV) Matrix-vector product, real triangular matrix
F06PGF	(DTBMV) Matrix-vector product, real triangular band matrix
F06PHF	(DTPMV) Matrix-vector product, real triangular packed matrix
F06PJF	(DTRSV) System of equations, real triangular matrix
F06PKF	(DTBSV) System of equations, real triangular band matrix
F06PLF	(DTPSV) System of equations, real triangular packed matrix
F06PMF	(DGER) Rank-1 update, real rectangular matrix

F06PPF	(DSYR) Rank-1 update, real symmetric matrix
F06PQF	(DSPR) Rank-1 update, real symmetric packed matrix
F06PRF	(DSYR2) Rank-2 update, real symmetric matrix
F06PSF	(DSPR2) Rank-2 update, real symmetric packed matrix
F06QFF	Matrix copy, real rectangular or trapezoidal matrix
F06QHF	Matrix initialisation, real rectangular matrix
F06QJF	Permute rows or columns, real rectangular matrix, permutations represented by an integer array
F06QKF	Permute rows or columns, real rectangular matrix, permutations represented by a real array
F06QMF	Orthogonal similarity transformation of real symmetric matrix as a sequence of plane rotations
F06QPF	*QR* factorization by sequence of plane rotations, rank-1 update of real upper triangular matrix
F06QQF	*QR* factorization by sequence of plane rotations, real upper triangular matrix augmented by a full row
F06QRF	*QR* or *RQ* factorization by sequence of plane rotations, real upper Hessenberg matrix
F06QSF	*QR* or *RQ* factorization by sequence of plane rotations, real upper spiked matrix
F06QTF	*QR* factorization of *UZ* or *RQ* factorization of *ZU*, *U* real upper triangular, *Z* a sequence of plane rotations
F06QVF	Compute upper Hessenberg matrix by sequence of plane rotations, real upper triangular matrix
F06QWF	Compute upper spiked matrix by sequence of plane rotations, real upper triangular matrix
F06QXF	Apply sequence of plane rotations, real rectangular matrix
F06RAF	1-norm, ∞-norm, Frobenius norm, largest absolute element, real general matrix
F06RBF	1-norm, ∞-norm, Frobenius norm, largest absolute element, real band matrix
F06RCF	1-norm, ∞-norm, Frobenius norm, largest absolute element, real symmetric matrix
F06RDF	1-norm, ∞-norm, Frobenius norm, largest absolute element, real symmetric matrix, packed storage
F06REF	1-norm, ∞-norm, Frobenius norm, largest absolute element, real symmetric band matrix
F06RJF	1-norm, ∞-norm, Frobenius norm, largest absolute element, real trapezoidal/triangular matrix
F06RKF	1-norm, ∞-norm, Frobenius norm, largest absolute element, real triangular matrix, packed storage
F06RLF	1-norm, ∞-norm, Frobenius norm, largest absolute element, real triangular band matrix
F06RMF	1-norm, ∞-norm, Frobenius norm, largest absolute element, real Hessenberg matrix
F06SAF	(ZGEMV) Matrix-vector product, complex rectangular matrix
F06SBF	(ZGBMV) Matrix-vector product, complex rectangular band matrix
F06SCF	(ZHEMV) Matrix-vector product, complex Hermitian matrix
F06SDF	(ZHBMV) Matrix-vector product, complex Hermitian band matrix
F06SEF	(ZHPMV) Matrix-vector product, complex Hermitian packed matrix
F06SFF	(ZTRMV) Matrix-vector product, complex triangular matrix
F06SGF	(ZTBMV) Matrix-vector product, complex triangular band matrix
F06SHF	(ZTPMV) Matrix-vector product, complex triangular packed matrix
F06SJF	(ZTRSV) System of equations, complex triangular matrix
F06SKF	(ZTBSV) System of equations, complex triangular band matrix
F06SLF	(ZTPSV) System of equations, complex triangular packed matrix
F06SMF	(ZGERU) Rank-1 update, complex rectangular matrix, unconjugated vector
F06SNF	(ZGERC) Rank-1 update, complex rectangular matrix, conjugated vector
F06SPF	(ZHER) Rank-1 update, complex Hermitian matrix
F06SQF	(ZHPR) Rank-1 update, complex Hermitian packed matrix
F06SRF	(ZHER2) Rank-2 update, complex Hermitian matrix
F06SSF	(ZHPR2) Rank-2 update, complex Hermitian packed matrix
F06TFF	Matrix copy, complex rectangular or trapezoidal matrix
F06THF	Matrix initialisation, complex rectangular matrix

F06TMF	Unitary similarity transformation of Hermitian matrix as a sequence of plane rotations
F06TPF	*QR* factorization by sequence of plane rotations, rank-1 update of complex upper triangular matrix
F06TQF	*QRxk* factorization by sequence of plane rotations, complex upper triangular matrix augmented by a full row
F06TRF	*QR* or *RQ* factorization by sequence of plane rotations, complex upper Hessenberg matrix
F06TSF	*QR* or *RQ* factorization by sequence of plane rotations, complex upper spiked matrix
F06TTF	*QR* factorization of *UZ* or *RQ* factorization of *ZU*, *U* complex upper triangular, *Z* a sequence of plane rotations
F06TVF	Compute upper Hessenberg matrix by sequence of plane rotations, complex upper triangular matrix
F06TWF	Compute upper spiked matrix by sequence of plane rotations, complex upper triangular matrix
F06TXF	Apply sequence of plane rotations, complex rectangular matrix, real cosine and complex sine
F06TYF	Apply sequence of plane rotations, complex rectangular matrix, complex cosine and real sine
F06UAF	1-norm, ∞-norm, Frobenius norm, largest absolute element, complex general matrix
F06UBF	1-norm, ∞-norm, Frobenius norm, largest absolute element, complex band matrix
F06UCF	1-norm, ∞-norm, Frobenius norm, largest absolute element, complex Hermitian matrix
F06UDF	1-norm, ∞-norm, Frobenius norm, largest absolute element, complex Hermitian matrix, packed storage
F06UEF	1-norm, ∞-norm, Frobenius norm, largest absolute element, complex Hermitian band matrix
F06UFF	1-norm, ∞-norm, Frobenius norm, largest absolute element, complex symmetric matrix
F06UGF	1-norm, ∞-norm, Frobenius norm, largest absolute element, complex symmetric matrix, packed storage
F06UHF	1-norm, ∞-norm, Frobenius norm, largest absolute element, complex symmetric band matrix
F06UJF	1-norm, ∞-norm, Frobenius norm, largest absolute element, complex trapezoidal/triangular matrix
F06UKF	1-norm, ∞-norm, Frobenius norm, largest absolute element, complex triangular matrix, packed storage
F06ULF	1-norm, ∞-norm, Frobenius norm, largest absolute element, complex triangular band matrix
F06UMF	1-norm, ∞-norm, Frobenius norm, largest absolute element, complex Hessenberg matrix
F06VJF	Permute rows or columns, complex rectangular matrix, permutations represented by an integer array
F06VKF	Permute rows or columns, complex rectangular matrix, permutations represented by a real array
F06VXF	Apply sequence of plane rotations, complex rectangular matrix, real cosine and sine
F06YAF	(DGEMM) Matrix-matrix product, two real rectangular matrices
F06YCF	(DSYMM) Matrix-matrix product, one real symmetric matrix, one real rectangular matrix
F06YFF	(DTRMM) Matrix-matrix product, one real triangular matrix, one real rectangular matrix
F06YJF	(DTRSM) Solves system of equations with multiple right-hand sides, real triangular coefficient matrix
F06YPF	(DSYRK) Rank-*k* update of real symmetric matrix
F06YRF	(DSYR2K) Rank-2*k* update of real symmetric matrix
F06ZAF	(ZGEMM) Matrix-matrix product, two complex rectangular matrices
F06ZCF	(ZHEMM) Matrix-matrix product, one complex Hermitian matrix, one complex rectangular matrix
F06ZFF	(ZTRMM) Matrix-matrix product, one complex triangular matrix, one complex rectangular matrix
F06ZJF	(ZTRSM) Solves system of equations with multiple right-hand sides, complex triangular coefficient matrix
F06ZPF	(ZHERK) Rank-*k* update of complex Hermitian matrix
F06ZRF	(ZHER2K) Rank-2*k* update of complex Hermitian matrix
F06ZTF	(ZSYMM) Matrix-matrix product, one complex symmetric matrix, one complex rectangular matrix
F06ZUF	(ZSYRK) Rank-*k* update of complex symmetric matrix
F06ZWF	(ZHER2K) Rank-2*k* update of complex symmetric matrix

Chapter F07 - Linear Equations (LAPACK)

F07ADF (DGETRF) *LU* factorization of real *m* by *n* matrix

F07AEF (DGETRS) Solution of real system of linear equations, multiple right-hand sides, matrix already factorized by F07ADF

F07AGF (DGECON) Estimate condition number of real matrix, matrix already factorized by F07ADF

F07AHF (DGERFS) Refined solution with error bounds of real system of linear equations, multiple right-hand sides

F07AJF (DGETRI) Inverse of real matrix, matrix already factorized by F07ADF

F07ARF (ZGETRF) *LU* factorization of complex *m* by *n* matrix

F07ASF (ZGETRS) Solution of complex system of linear equations, multiple right-hand sides, matrix already factorized by F07ARF

F07AUF (ZGECON) Estimate condition number of complex matrix, matrix already factorized by F07ARF

F07AVF (ZGERFS) Refined solution with error bounds of complex system of linear equations, multiple right-hand sides

F07AWF (ZGETRI) Inverse of complex matrix, matrix already factorized by F07ARF

F07BDF (DGBTRF) *LU* factorization of real *m* by *n* band matrix

F07BEF (DGBTRS) Solution of real band system of linear equations, multiple right-hand sides, matrix already factorized by F07BDF

F07BGF (DGBCON) Estimate condition number of real band matrix, matrix already factorized by F07BDF

F07BHF (DGBRFS) Refined solution with error bounds of real band system of linear equations, multiple right-hand sides

F07BRF (ZGBTRF) *LU* factorization of complex *m* by *n* band matrix

F07BSF (ZGBTRS) Solution of complex band system of linear equations, multiple right-hand sides, matrix already factorized by F07BRF

F07BUF (ZGBCON) Estimate condition number of complex band matrix, matrix already factorized by F07BRF

F07BVF (ZGBRFS) Refined solution with error bounds of complex band system of linear equations, multiple right-hand sides

F07FDF (DPOTRF) Cholesky factorization of real symmetric positive-definite matrix

F07FEF (DPOTRS) Solution of real symmetric positive-definite system of linear equations, multiple right-hand sides, matrix already factorized by F07FDF

F07FGF (DPOCON) Estimate condition number of real symmetric positive-definite matrix, matrix already factorized by F07FDF

F07FHF (DPORFS) Refined solution with error bounds of real symmetric positive-definite system of linear equations, multiple right-hand sides

F07FJF (DPOTRI) Inverse of real symmetric positive-definite matrix, matrix already factorized by F07FDF

F07FRF (ZPOTRF) Cholesky factorization of complex Hermitian positive-definite matrix

F07FSF (ZPOTRS) Solution of complex Hermitian positive-definite system of linear equations, multiple right-hand sides, matrix already factorized by F07FRF

F07FUF (ZPOCON) Estimate condition number of complex Hermitian positive-definite matrix, matrix already factorized by F07FRF

F07FVF (ZPORFS) Refined solution with error bounds of complex Hermitian positive-definite system of linear equations, multiple right-hand sides

F07FWF (ZPOTRI) Inverse of complex Hermitian positive-definite matrix, matrix already factorized by F07FRF

F07GDF (DPPTRF) Cholesky factorization of real symmetric positive-definite matrix, packed storage

F07GEF (DPPTRS) Solution of real symmetric positive-definite system of linear equations, multiple right-hand sides, matrix already factorized by F07GDF, packed storage

F07GGF (DPPCON) Estimate condition number of real symmetric positive-definite matrix, matrix already factorized by F07GDF, packed storage

F07GHF (DPPRFS) Refined solution with error bounds of real symmetric positive-definite system of linear equations, multiple right-hand sides, packed storage

F07GJF	(DPPTRI) Inverse of real symmetric positive-definite matrix, matrix already factorized by F07GDF, packed storage
F07GRF	(ZPPTRF) Cholesky factorization of complex Hermitian positive-definite matrix, packed storage
F07GSF	(ZPPTRS) Solution of complex Hermitian positive-definite system of linear equations, multiple right-hand sides, matrix already factorized by F07GRF, packed storage
F07GUF	(ZPPCON) Estimate condition number of complex Hermitian positive-definite matrix, matrix already factorized by F07GRF, packed storage
F07GVF	(ZPPRFS) Refined solution with error bounds of complex Hermitian positive-definite system of linear equations, multiple right-hand sides, packed storage
F07GWF	(ZPPTRI) Inverse of complex Hermitian positive-definite matrix, matrix already factorized by F07GRF, packed storage
F07HDF	(DPBTRF) Cholesky factorization of real symmetric positive-definite band matrix
F07HEF	(DPBTRS) Solution of real symmetric positive-definite band system of linear equations, multiple right-hand sides, matrix already factorized by F07HDF
F07HGF	(DPBCON) Estimate condition number of real symmetric positive-definite band matrix, matrix already factorized by F07HDF
F07HHF	(DPBRFS) Refined solution with error bounds of real symmetric positive-definite band system of linear equations, multiple right-hand sides
F07HRF	(ZPBTRF) Cholesky factorization of complex Hermitian positive-definite band matrix
F07HSF	(ZPBTRS) Solution of complex Hermitian positive-definite band system of linear equations, multiple right-hand sides, matrix already factorized by F07HRF
F07HUF	(ZPBCON) Estimate condition number of complex Hermitian positive-definite band matrix, matrix already factorized by F07HRF
F07HVF	(ZPBRFS) Refined solution with error bounds of complex Hermitian positive-definite band system of linear equations, multiple right-hand sides
F07MDF	(DSYTRF) Bunch–Kaufman factorization of real symmetric indefinite matrix
F07MEF	(DSYTRS) Solution of real symmetric indefinite system of linear equations, multiple right-hand sides, matrix already factorized by F07MDF
F07MGF	(DSYCON) Estimate condition number of real symmetric indefinite matrix, matrix already factorized by F07MDF
F07MHF	(DSYRFS) Refined solution with error bounds of real symmetric indefinite system of linear equations, multiple right-hand sides
F07MJF	(DSYTRI) Inverse of real symmetric indefinite matrix, matrix already factorized by F07MDF
F07MRF	(ZHETRF) Bunch–Kaufman factorization of complex Hermitian indefinite matrix
F07MSF	(ZHETRS) Solution of complex Hermitian indefinite system of linear equations, multiple right-hand sides, matrix already factorized by F07MRF
F07MUF	(ZHECON) Estimate condition number of complex Hermitian indefinite matrix, matrix already factorized by F07MRF
F07MVF	(ZHERFS) Refined solution with error bounds of complex Hermitian indefinite system of linear equations, multiple right-hand sides
F07MWF	(ZHETRI) Inverse of complex Hermitian indefinite matrix, matrix already factorized by F07MRF
F07NRF	(ZSYTRF) Bunch–Kaufman factorization of complex symmetric matrix
F07NSF	(ZSYTRS) Solution of complex symmetric system of linear equations, multiple right-hand sides, matrix already factorized by F07NRF
F07NUF	(ZSYCON) Estimate condition number of complex symmetric matrix, matrix already factorized by F07NRF
F07NVF	(ZSYRFS) Refined solution with error bounds of complex symmetric system of linear equations, multiple right-hand sides
F07NWF	(ZSYTRI) Inverse of complex symmetric matrix, matrix already factorized by F07NRF
F07PDF	(DSPTRF) Bunch–Kaufman factorization of real symmetric indefinite matrix, packed storage
F07PEF	(DSPTRS) Solution of real symmetric indefinite system of linear equations, multiple right-hand sides, matrix already factorized by F07PDF, packed storage

F07PGF	(DSPCON) Estimate condition number of real symmetric indefinite matrix, matrix already factorized by F07PDF, packed storage
F07PHF	(DSPRFS) Refined solution with error bounds of real symmetric indefinite system of linear equations, multiple right-hand sides, packed storage
F07PJF	(DSPTRI) Inverse of real symmetric indefinite matrix, matrix already factorized by F07PDF, packed storage
F07PRF	(ZHPTRF) Bunch–Kaufman factorization of complex Hermitian indefinite matrix, packed storage
F07PSF	(ZHPTRS) Solution of complex Hermitian indefinite system of linear equations, multiple right-hand sides, matrix already factorized by F07PRF, packed storage
F07PUF	(ZHPCON) Estimate condition number of complex Hermitian indefinite matrix, matrix already factorized by F07PRF, packed storage
F07PVF	(ZHPRFS) Refined solution with error bounds of complex Hermitian indefinite system of linear equations, multiple right-hand sides, packed storage
F07PWF	(ZHPTRI) Inverse of complex Hermitian indefinite matrix, matrix already factorized by F07PRF, packed storage
F07QRF	(ZSPTRF) Bunch–Kaufman factorization of complex symmetric matrix, packed storage
F07QSF	(ZSPTRS) Solution of complex symmetric system of linear equations, multiple right-hand sides, matrix already factorized by F07QRF, packed storage
F07QUF	(ZSPCON) Estimate condition number of complex symmetric matrix, matrix already factorized by F07QRF, packed storage
F07QVF	(ZSPRFS) Refined solution with error bounds of complex symmetric system of linear equations, multiple right-hand sides, packed storage
F07QWF	(ZSPTRI) Inverse of complex symmetric matrix, matrix already factorized by F07QRF, packed storage
F07TEF	(DTRTRS) Solution of real triangular system of linear equations, multiple right-hand sides
F07TGF	(DTRCON) Estimate condition number of real triangular matrix
F07THF	(DTRRFS) Error bounds for solution of real triangular system of linear equations, multiple right-hand sides
F07TJF	(DTRTRI) Inverse of real triangular matrix
F07TSF	(ZTRTRS) Solution of complex triangular system of linear equations, multiple right-hand sides
F07TUF	(ZTRCON) Estimate condition number of complex triangular matrix
F07TVF	(ZTRRFS) Error bounds for solution of complex triangular system of linear equations, multiple right-hand sides
F07TWF	(ZTRTRI) Inverse of complex triangular matrix
F07UEF	(DTPTRS) Solution of real triangular system of linear equations, multiple right-hand sides, packed storage
F07UGF	(DTPCON) Estimate condition number of real triangular matrix, packed storage
F07UHF	(DTPRFS) Error bounds for solution of real triangular system of linear equations, multiple right-hand sides, packed storage
F07UJF	(DTPTRI) Inverse of real triangular matrix, packed storage
F07USF	(ZTPTRS) Solution of complex triangular system of linear equations, multiple right-hand sides, packed storage
F07UUF	(ZTPCON) Estimate condition number of complex triangular matrix, packed storage
F07UVF	(ZTPRFS) Error bounds for solution of complex triangular system of linear equations, multiple right-hand sides, packed storage
F07UWF	(ZTPTRI) Inverse of complex triangular matrix, packed storage
F07VEF	(DTBTRS) Solution of real band triangular system of linear equations, multiple right-hand sides
F07VGF	(DTBCON) Estimate condition number of real band triangular matrix
F07VHF	(DTBRFS) Error bounds for solution of real band triangular system of linear equations, multiple right-hand sides
F07VSF	(ZTBTRS) Solution of complex band triangular system of linear equations, multiple right-hand sides
F07VUF	(ZTBCON) Estimate condition number of complex band triangular matrix

F07VVF (ZTBRFS) Error bounds for solution of complex band triangular system of linear equations, multiple right-hand sides

Chapter F08 - Least-squares and Eigenvalue Problems (LAPACK)

F08AEF (DGEQRF) *QR* factorization of real general rectangular matrix

F08AFF (DORGQR) Form all or part of orthogonal *Q* from *QR* factorization determined by F08AEF or F08BE

F08AGF (DORMQR) Apply orthogonal transformation determined by F08AEF or F08BEF

F08AHF (DGELQF) *LQ* factorization of real general rectangular matrix

F08AJF (DORGLQ) Form all or part of orthogonal *Q* from *LQ* factorization determined by F08AHF

F08AKF (DORMLQ) Apply orthogonal transformation determined by F08AHF

F08ASF (ZGEQRF) *QR* factorization of complex general rectangular matrix

F08ATF (ZUNGQR) Form all or part of unitary *Q* from *QR* factorization determined by F08ASF or F08BSF

F08AUF (ZUNMQR) Apply unitary transformation determined by F08ASF or F08BSF

F08AVF (ZGELQF) *LQ* factorization of complex general rectangular matrix

F08AWF (ZUNGLQ) Form all or part of unitary *Q* from *LQ* factorization determined by F08AVF

F08AXF (ZUNMLQ) Apply unitary transformation determined by F08AVF

F08BEF (DGEQPF) *QR* factorization of real general rectangular matrix with column pivoting

F08BSF (ZGEQPF) *QR* factorization of complex general rectangular matrix with column pivoting

F08FCF (DSYEVD) All eigenvalues and optionally all eigenvectors of real symmetric matrix, using divide and conquer

F08FEF (DSYTRD) Orthogonal reduction of real symmetric matrix to symmetric tridiagonal form

F08FFF (DORGTR) Generate orthogonal transformation matrix from reduction to tridiagonal form determined by F08FEF

F08FGF (DORMTR) Apply orthogonal transformation determined by F08FEF

F08FQF (ZHEEVD) All eigenvalues and optionally all eigenvectors of complex Hermitian matrix, using divide and conquer

F08FSF (ZHETRD) Unitary reduction of complex Hermitian matrix to real symmetric tridiagonal form

F08FTF (ZUNGTR) Generate unitary transformation matrix from reduction to tridiagonal form determined by F08FSF

F08FUF (ZUNMTR) Apply unitary transformation matrix determined by F08FSF

F08GCF (DSPEVD) All eigenvalues and optionally all eigenvectors of real symmetric matrix, packed storage, using divide and conquer

F08GEF (DSPTRD) Orthogonal reduction of real symmetric matrix to symmetric tridiagonal form, packed storage

F08GFF (DOPGTR) Generate orthogonal transformation matrix from reduction to tridiagonal form determined by F08GEF

F08GGF (DOPMTR) Apply orthogonal transformation determined by F08GEF

F08GQF (ZHPEVD) All eigenvalues and optionally all eigenvectors of complex Hermitian matrix, packed storage, using divide and conquer

F08GSF (ZHPTRD) Unitary reduction of complex Hermitian matrix to real symmetric tridiagonal form, packed storage

F08GTF (ZUPGTR) Generate unitary transformation matrix from reduction to tridiagonal form determined by F08GSF

F08GUF (ZUPMTR) Apply unitary transformation matrix determined by F08GSF

F08HCF (DSBEVD) All eigenvalues and optionally all eigenvectors of real symmetric band matrix, using divide and conquer

F08HEF (DSBTRD) Orthogonal reduction of real symmetric band matrix to symmetric tridiagonal form

F08HQF (ZHBEVD) All eigenvalues and optionally all eigenvectors of complex Hermitian band matrix, using divide and conquer

F08HSF (ZHBTRD) Unitary reduction of complex Hermitian band matrix to real symmetric tridiagonal form

F08JCF (DSTEVD) All eigenvalues and optionally all eigenvectors of real symmetric tridiagonal matrix, using divide and conquer

F08JEF (DSTEQR) All eigenvalues and eigenvectors of real symmetric tridiagonal matrix, reduced from real symmetric matrix using implicit *QL* or *QR*

F08JFF (DSTERF) All eigenvalues of real symmetric tridiagonal matrix, root-free variant of *QL* or *QR*

F08JGF (DPTEQR) All eigenvalues and eigenvectors of real symmetric positive-definite tridiagonal matrix, reduced from real symmetric positive-definite matrix

F08JJF (DSTEBZ) Selected eigenvalues of real symmetric tridiagonal matrix by bisection

F08JKF (DSTEIN) Selected eigenvectors of real symmetric tridiagonal matrix by inverse iteration, storing eigenvectors in real array

F08JSF (ZSTEQR) All eigenvalues and eigenvectors of real symmetric tridiagonal matrix, reduced from complex Hermitian matrix, using implicit *QL* or *QR*

F08JUF (ZPTEQR) All eigenvalues and eigenvectors of real symmetric positive-definite tridiagonal matrix, reduced from complex Hermitian positive-definite matrix

F08JXF (ZSTEIN) Selected eigenvectors of real symmetric tridiagonal matrix by inverse iteration, storing eigenvectors in complex array

F08KEF (DGEBRD) Orthogonal reduction of real general rectangular matrix to bidiagonal form

F08KFF (DORGBR) Generate orthogonal transformation matrices from reduction to bidiagonal form determined by F08KEF

F08KGF (DORMBR) Apply orthogonal transformations from reduction to bidiagonal form determined by F08KEF

F08KSF (ZGEBRD) Unitary reduction of complex general rectangular matrix to bidiagonal form

F08KTF (ZUNGBR) Generate unitary transformation matrices from reduction to bidiagonal form determined by F08KSF

F08KUF (ZUNMBR) Apply unitary transformations from reduction to bidiagonal form determined by F08KSF

F08LEF (DGBBRD) Reduction of real rectangular band matrix to upper bidiagonal form

F08LSF (ZGBBRD) Reduction of complex rectangular band matrix to upper bidiagonal form

F08MEF (DBDSQR) SVD of real bidiagonal matrix reduced from real general matrix

F08MSF (ZBDSQR) SVD of real bidiagonal matrix reduced from complex general matrix

F08NEF (DGEHRD) Orthogonal reduction of real general matrix to upper Hessenberg form

F08NFF (DORGHR) Generate orthogonal transformation matrix from reduction to Hessenberg form determined by F08NEF

F08NGF (DORMHR) Apply orthogonal transformation matrix from reduction to Hessenberg form determined by F08NEF

F08NHF (DGEBAL) Balance real general matrix

F08NJF (DGEBAK) Transform eigenvectors of real balanced matrix to those of original matrix supplied to F08NHF

F08NSF (ZGEHRD) Unitary reduction of complex general matrix to upper Hessenberg form

F08NTF (ZUNGHR) Generate unitary transformation matrix from reduction to Hessenberg form determined by F08NSF

F08NUF (ZUNMHR) Apply unitary transformation matrix from reduction to Hessenberg form determined by F08NSF

F08NVF (ZGEBAL) Balance complex general matrix

F08NWF (ZGEBAK) Transform eigenvectors of complex balanced matrix to those of original matrix supplied to F08NVF

F08PEF (DHSEQR) Eigenvalues and Schur factorization of real upper Hessenberg matrix reduced from real general matrix

F08PKF (DHSEIN) Selected right and/or left eigenvectors of real upper Hessenberg matrix by inverse iteration

F08PSF (ZHSEQR) Eigenvalues and Schur factorization of complex upper Hessenberg matrix reduced from complex general matrix

F08PXF (ZHSEIN) Selected right and/or left eigenvectors of complex upper Hessenberg matrix by inverse iteration

F08QFF (DTREXC) Reorder Schur factorization of real matrix using orthogonal similarity transformation

F08QGF (DTRSEN) Reorder Schur factorization of real matrix, form orthonormal basis of right invariant subspace for selected eigenvalues, with estimates of sensitivities

F08QHF (DTRSYL) Solve real Sylvester matrix equation $AX + XB = C$, A and B are upper quasi-triangular or transposes

F08QKF (DTREVC) Left and right eigenvectors of real upper quasi-triangular matrix

F08QLF (DTRSNA) Estimates of sensitivities of selected eigenvalues and eigenvectors of real upper quasi-triangular matrix

F08QTF (ZTREXC) Reorder Schur factorization of complex matrix using unitary similarity transformation

F08QUF (ZTRSEN) Reorder Schur factorization of complex matrix, form orthonormal basis of right invariant subspace for selected eigenvalues, with estimates of sensitivities

F08QVF (ZTRSYL) Solve complex Sylvester matrix equation $AX + XB = C$, A and B are upper triangular or conjugate-transposes

F08QXF (ZTREVC) Left and right eigenvectors of complex upper triangular matrix

F08QYF (ZTRSNA) Estimates of sensitivities of selected eigenvalues and eigenvectors of complex upper triangular matrix

F08SEF (DSYGST) Reduction to standard form of real symmetric-definite generalized eigenproblem $Ax = \lambda Bx$, $ABx = \lambda x$ or $BAx = \lambda x$, B factorized by F07FDF

F08SSF (ZHEGST) Reduction to standard form of complex Hermitian-definite generalized eigenproblem $Ax = \lambda Bx$, $ABx = \lambda x$ or $BAx = \lambda x$, B factorized by F07FRF

F08TEF (DSPGST) Reduction to standard form of real symmetric-definite generalized eigenproblem $Ax = \lambda Bx$, $ABx = \lambda x$ or $BAx = \lambda x$, packed storage, B factorized by F07GDF

F08TSF (ZHPGST) Reduction to standard form of complex Hermitian-definite generalized eigenproblem $Ax = \lambda Bx$, $ABx = \lambda x$ or $BAx = \lambda x$, packed storage, B factorized by F07GRF

F08UEF (DSBGST) Reduction of real symmetric-definite banded generalized eigenproblem $Ax = \lambda Bx$ to standard form $Cy = \lambda y$, such that C has the same bandwidth as A

F08UFF (DPBSTF) Computes a split Cholesky factorization of real symmetric positive-definite band matrix A

F08USF (ZHBGST) Reduction of complex Hermitian-definite banded generalized eigenproblem $Ax = \lambda Bx$ to standard form $Cy = \lambda y$, such that C has the same bandwidth as A

F08UTF (ZPBSTF) Computes a split Cholesky factorization of complex Hermitian positive-definite band matrix A

Chapter F11 - Sparse Linear Algebra

F11BAF Real sparse nonsymmetric linear systems, set-up for F11BBF

F11BBF Real sparse nonsymmetric linear systems, preconditioned RGMRES, CGS or Bi-CGSTAB

F11BCF Real sparse nonsymmetric linear systems, diagnostic for F11BBF

F11BDF Real sparse nonsymmetric linear systems, set-up for F11BEF

F11BEF Real sparse nonsymmetric linear systems, preconditioned RGMRES, CGS, Bi-CGSTAB or TFQMR method

F11BFF Real sparse nonsymmetric linear systems, diagnostic for F11BEF

F11BRF Complex sparse non-Hermitian linear systems, set-up for F11BSF

F11BSF Complex sparse non-Hermitian linear systems, preconditioned RGMRES, CGS, Bi-CGSTAB or TFQMR method

F11BTF Complex sparse non-Hermitian linear systems, diagnostic for F11BSF

F11DAF Real sparse nonsymmetric linear systems, incomplete LU factorization

F11DBF Solution of linear system involving incomplete LU preconditioning matrix generated by F11DAF

F11DCF Solution of real sparse nonsymmetric linear system, RGMRES, CGS or Bi-CGSTAB method, preconditioner computed by F11DAF (Black Box)

F11DDF Solution of linear system involving preconditioning matrix generated by applying SSOR to real sparse nonsymmetric matrix

F11DEF	Solution of real sparse nonsymmetric linear system, RGMRES, CGS or Bi-CGSTAB method, Jacobi or SSOR preconditioner (Black Box)
F11DNF	Complex sparse non-Hermitian linear systems, incomplete *LU* factorization
F11DPF	Solution of complex linear system involving incomplete *LU* preconditioning matrix generated by F11DNF
F11DQF	Solution of complex sparse non-Hermitian linear system, RGMRES, CGS, Bi-CGSTAB or TFQMR method, preconditioner computed by F11DNF (Black Box)
F11DRF	Solution of linear system involving preconditioning matrix generated by applying SSOR to complex sparse non-Hermitian matrix
F11DSF	Solution of complex sparse non-Hermitian linear system, RGMRES, CGS, Bi-CGSTAB or TFQMR method, Jacobi or SSOR preconditioner (Black Box)
F11GAF	Real sparse symmetric linear systems, set-up for F11GBF
F11GBF	Real sparse symmetric linear systems, preconditioned conjugate gradient or Lanczos
F11GCF	Real sparse symmetric linear systems, diagnostic for F11GBF
F11JAF	Real sparse symmetric matrix, incomplete Cholesky factorization
F11JBF	Solution of linear system involving incomplete Cholesky preconditioning matrix generated by F11JAF
F11JCF	Solution of real sparse symmetric linear system, conjugate gradient/Lanczos method, preconditioner computed by F11JAF (Black Box)
F11JDF	Solution of linear system involving preconditioning matrix generated by applying SSOR to real sparse symmetric matrix
F11JEF	Solution of real sparse symmetric linear system, conjugate gradient/Lanczos method, Jacobi or SSOR preconditioner (Black Box)
F11JNF	Complex sparse Hermitian matrix, incomplete Cholesky factorization
F11JPF	Solution of complex linear system involving incomplete Cholesky preconditioning matrix generated by F11JNF
F11JQF	Solution of complex sparse Hermitian linear system, conjugate gradient/Lanczos method, preconditioner computed by F11JNF (Black Box)
F11JRF	Solution of linear system involving preconditioning matrix generated by applying SSOR to complex sparse Hermitian matrix
F11JSF	Solution of complex sparse Hermitian linear system, conjugate gradient/Lanczos method, Jacobi or SSOR preconditioner (Black Box)
F11XAF	Real sparse nonsymmetric matrix vector multiply
F11XEF	Real sparse symmetric matrix vector multiply
F11XNF	Complex sparse non-Hermitian matrix vector multiply
F11XSF	Complex sparse Hermitian matrix vector multiply
F11ZAF	Real sparse nonsymmetric matrix reorder routine
F11ZBF	Real sparse symmetric matrix reorder routine
F11ZNF	Complex sparse non-Hermitian matrix reorder routine
F11ZPF	Complex sparse Hermitian matrix reorder routine

Chapter G01 - Simple Calculations and Statistical Data

G01AAF	Mean, variance, skewness, kurtosis, etc, one variable, from raw data
G01ABF	Mean, variance, skewness, kurtosis, etc, two variables, from raw data
G01ADF	Mean, variance, skewness, kurtosis, etc, one variable, from frequency table
G01AEF	Frequency table from raw data
G01AFF	Two-way contingency table analysis, with χ^2 /Fisher's exact test
G01AGF	Lineprinter scatterplot of two variables
G01AHF	Lineprinter scatterplot of one variable against Normal scores
G01AJF	Lineprinter histogram of one variable
G01ALF	Computes a five-point summary (median, hinges and extremes)

G01ARF Constructs a stem and leaf plot
G01ASF Constructs a box and whisker plot
G01BJF Binomial distribution function
G01BKF Poisson distribution function
G01BLF Hypergeometric distribution function
G01DAF Normal scores, accurate values
G01DBF Normal scores, approximate values
G01DCF Normal scores, approximate variance-covariance matrix
G01DDF Shapiro and Wilk's *W* test for Normality
G01DHF Ranks, Normal scores, approximate Normal scores or exponential (Savage) scores
G01EAF Computes probabilities for the standard Normal distribution
G01EBF Computes probabilities for Student's *t*-distribution
G01ECF Computes probabilities for χ^2 distribution
G01EDF Computes probabilities for *F*-distribution
G01EEF Computes upper and lower tail probabilities and probability density function for the beta distribution
G01EFF Computes probabilities for the gamma distribution
G01EMF Computes probability for the Studentized range statistic
G01EPF Computes bounds for the significance of a Durbin–Watson statistic
G01ERF Computes probability for von Mises distribution
G01EYF Computes probabilities for the one-sample Kolmogorov–Smirnov distribution
G01EZF Computes probabilities for the two-sample Kolmogorov–Smirnov distribution
G01FAF Computes deviates for the standard Normal distribution
G01FBF Computes deviates for Student's *t*-distribution
G01FCF Computes deviates for the χ^2 distribution
G01FDF Computes deviates for the *F*-distribution
G01FEF Computes deviates for the beta distribution
G01FFF Computes deviates for the gamma distribution
G01FMF Computes deviates for the Studentized range statistic
G01GBF Computes probabilities for the non-central Student's *t*-distribution
G01GCF Computes probabilities for the non-central χ^2 distribution
G01GDF Computes probabilities for the non-central *F*-distribution
G01GEF Computes probabilities for the non-central beta distribution
G01HAF Computes probability for the bivariate Normal distribution
G01HBF Computes probabilities for the multivariate Normal distribution
G01JCF Computes probability for a positive linear combination of χ^2 variables
G01JDF Computes lower tail probability for a linear combination of (central) χ^2 variables
G01MBF Computes reciprocal of Mills' Ratio
G01NAF Cumulants and moments of quadratic forms in Normal variables
G01NBF Moments of ratios of quadratic forms in Normal variables, and related statistics

Chapter G02 - Correlation and Regression Analysis

G02BAF Pearson product-moment correlation coefficients, all variables, no missing values
G02BBF Pearson product-moment correlation coefficients, all variables, casewise treatment of missing values
G02BCF Pearson product-moment correlation coefficients, all variables, pairwise treatment of missing values
G02BDF Correlation-like coefficients (about zero), all variables, no missing values
G02BEF Correlation-like coefficients (about zero), all variables, casewise treatment of missing values
G02BFF Correlation-like coefficients (about zero), all variables, pairwise treatment of missing values

G02BGF	Pearson product-moment correlation coefficients, subset of variables, no missing values
G02BHF	Pearson product-moment correlation coefficients, subset of variables, casewise treatment of missing values
G02BJF	Pearson product-moment correlation coefficients, subset of variables, pairwise treatment of missing values
G02BKF	Correlation-like coefficients (about zero), subset of variables, no missing values
G02BLF	Correlation-like coefficients (about zero), subset of variables, casewise treatment of missing values
G02BMF	Correlation-like coefficients (about zero), subset of variables, pairwise treatment of missing values
G02BNF	Kendall/Spearman non-parametric rank correlation coefficients, no missing values, overwriting input data
G02BPF	Kendall/Spearman non-parametric rank correlation coefficients, casewise treatment of missing values, overwriting input data
G02BQF	Kendall/Spearman non-parametric rank correlation coefficients, no missing values, preserving input data
G02BRF	Kendall/Spearman non-parametric rank correlation coefficients, casewise treatment of missing values, preserving input data
G02BSF	Kendall/Spearman non-parametric rank correlation coefficients, pairwise treatment of missing values
G02BTF	Update a weighted sum of squares matrix with a new observation
G02BUF	Computes a weighted sum of squares matrix
G02BWF	Computes a correlation matrix from a sum of squares matrix
G02BXF	Computes (optionally weighted) correlation and covariance matrices
G02BYF	Computes partial correlation/variance-covariance matrix from correlation/variance-covariance matrix computed by G02BXF
G02CAF	Simple linear regression with constant term, no missing values
G02CBF	Simple linear regression without constant term, no missing values
G02CCF	Simple linear regression with constant term, missing values
G02CDF	Simple linear regression without constant term, missing values
G02CEF	Service routines for multiple linear regression, select elements from vectors and matrices
G02CFF	Service routines for multiple linear regression, re-order elements of vectors and matrices
G02CGF	Multiple linear regression, from correlation coefficients, with constant term
G02CHF	Multiple linear regression, from correlation-like coefficients, without constant term
G02DAF	Fits a general (multiple) linear regression model
G02DCF	Add/delete an observation to/from a general linear regression model
G02DDF	Estimates of linear parameters and general linear regression model from updated model
G02DEF	Add a new variable to a general linear regression model
G02DFF	Delete a variable from a general linear regression model
G02DGF	Fits a general linear regression model for new dependent variable
G02DKF	Estimates and standard errors of parameters of a general linear regression model for given constraints
G02DNF	Computes estimable function of a general linear regression model and its standard error
G02EAF	Computes residual sums of squares for all possible linear regressions for a set of independent variables
G02ECF	Calculates R^2 and C_P values from residual sums of squares
G02EEF	Fits a linear regression model by forward selection
G02FAF	Calculates standardized residuals and influence statistics
G02FCF	Computes Durbin–Watson test statistic
G02GAF	Fits a generalized linear model with Normal errors
G02GBF	Fits a generalized linear model with binomial errors
G02GCF	Fits a generalized linear model with Poisson errors
G02GDF	Fits a generalized linear model with gamma errors

G02GKF Estimates and standard errors of parameters of a general linear model for given constraints
G02GNF Computes estimable function of a generalized linear model and its standard error
G02HAF Robust regression, standard *M*-estimates
G02HBF Robust regression, compute weights for use with G02HDF
G02HDF Robust regression, compute regression with user-supplied functions and weights
G02HFF Robust regression, variance-covariance matrix following G02HDF
G02HKF Calculates a robust estimation of a correlation matrix, Huber's weight function
G02HLF Calculates a robust estimation of a correlation matrix, user-supplied weight function plus derivatives
G02HMF Calculates a robust estimation of a correlation matrix, user-supplied weight function

Chapter G03 - Multivariate Methods

G03AAF Performs principal component analysis
G03ACF Performs canonical variate analysis
G03ADF Performs canonical correlation analysis
G03BAF Computes orthogonal rotations for loading matrix, generalized orthomax criterion
G03BCF Computes Procrustes rotations
G03CAF Computes maximum likelihood estimates of the parameters of a factor analysis model, factor loadings, communalities and residual correlations
G03CCF Computes factor score coefficients (for use after G03CAF)
G03DAF Computes test statistic for equality of within-group covariance matrices and matrices for discriminant analysis
G03DBF Computes Mahalanobis squared distances for group or pooled variance-covariance matrices (for use after G03DAF)
G03DCF Allocates observations to groups according to selected rules (for use after G03DAF)
G03EAF Computes distance matrix
G03ECF Hierarchical cluster analysis
G03EFF *K*-means cluster analysis
G03EHF Constructs dendrogram (for use after G03ECF)
G03EJF Computes cluster indicator variable (for use after G03ECF)
G03FAF Performs principal co-ordinate analysis, classical metric scaling
G03FCF Performs non-metric (ordinal) multidimensional scaling
G03ZAF Produces standardized values (*z*-scores) for a data matrix

Chapter G04 - Analysis of Variance

G04AGF Two-way analysis of variance, hierarchical classification, subgroups of unequal size
G04BBF Analysis of variance, randomized block or completely randomized design, treatment means and standard errors
G04BCF Analysis of variance, general row and column design, treatment means and standard errors
G04CAF Analysis of variance, complete factorial design, treatment means and standard errors
G04DAF Computes sum of squares for contrast between means
G04DBF Computes confidence intervals for differences between means computed by G04BBF or G04BCF
G04EAF Computes orthogonal polynomials or dummy variables for factor/classification variable

Chapter G05 - Random Number Generators

G05CAF Pseudo-random real numbers, uniform distribution over (0,1)
G05CBF Initialise random number generating routines to give repeatable sequence
G05CCF Initialise random number generating routines to give non-repeatable sequence
G05CFF Save state of random number generating routines
G05CGF Restore state of random number generating routines

G05DAF Pseudo-random real numbers, uniform distribution over (*a*,*b*)
G05DBF Pseudo-random real numbers, (negative) exponential distribution
G05DCF Pseudo-random real numbers, logistic distribution
G05DDF Pseudo-random real numbers, Normal distribution
G05DEF Pseudo-random real numbers, log-normal distribution
G05DFF Pseudo-random real numbers, Cauchy distribution
G05DHF Pseudo-random real numbers, χ^2 distribution
G05DJF Pseudo-random real numbers, Student's *t*-distribution
G05DKF Pseudo-random real numbers, *F*-distribution
G05DPF Pseudo-random real numbers, Weibull distribution
G05DRF Pseudo-random integer, Poisson distribution
G05DYF Pseudo-random integer from uniform distribution
G05DZF Pseudo-random logical (boolean) value
G05EAF Set up reference vector for multivariate Normal distribution
G05EBF Set up reference vector for generating pseudo-random integers, uniform distribution
G05ECF Set up reference vector for generating pseudo-random integers, Poisson distribution
G05EDF Set up reference vector for generating pseudo-random integers, binomial distribution
G05EEF Set up reference vector for generating pseudo-random integers, negative binomial distribution
G05EFF Set up reference vector for generating pseudo-random integers, hypergeometric distribution
G05EGF Set up reference vector for univariate ARMA time series model
G05EHF Pseudo-random permutation of an integer vector
G05EJF Pseudo-random sample from an integer vector
G05EWF Generate next term from reference vector for ARMA time series model
G05EXF Set up reference vector from supplied cumulative distribution function or probability distribution function
G05EYF Pseudo-random integer from reference vector
G05EZF Pseudo-random multivariate Normal vector from reference vector
G05FAF Generates a vector of random numbers from a uniform distribution
G05FBF Generates a vector of random numbers from an (negative) exponential distribution
G05FDF Generates a vector of random numbers from a Normal distribution
G05FEF Generates a vector of pseudo-random numbers from a beta distribution
G05FFF Generates a vector of pseudo-random numbers from a gamma distribution
G05FSF Generates a vector of pseudo-random variates from von Mises distribution
G05GAF Computes random orthogonal matrix
G05GBF Computes random correlation matrix
G05HDF Generates a realisation of a multivariate time series from a VARMA model

Chapter G07 - Univariate Estimation

G07AAF Computes confidence interval for the parameter of a binomial distribution
G07ABF Computes confidence interval for the parameter of a Poisson distribution
G07BBF Computes maximum likelihood estimates for parameters of the Normal distribution from grouped and/or censored data
G07BEF Computes maximum likelihood estimates for parameters of the Weibull distribution
G07CAF Computes *t*-test statistic for a difference in means between two Normal populations, confidence interval
G07DAF Robust estimation, median, median absolute deviation, robust standard deviation
G07DBF Robust estimation, *M*-estimates for location and scale parameters, standard weight functions
G07DCF Robust estimation, *M*-estimates for location and scale parameters, user-defined weight functions

G07DDF Computes a trimmed and winsorized mean of a single sample with estimates of their variance
G07EAF Robust confidence intervals, one-sample
G07EBF Robust confidence intervals, two-sample

Chapter G08 - Nonparametric Statistics

G08AAF Sign test on two paired samples
G08ACF Median test on two samples of unequal size
G08AEF Friedman two-way analysis of variance on k matched samples
G08AFF Kruskal–Wallis one-way analysis of variance on k samples of unequal size
G08AGF Performs the Wilcoxon one-sample (matched pairs) signed rank test
G08AHF Performs the Mann–Whitney U test on two independent samples
G08AJF Computes the exact probabilities for the Mann–Whitney U statistic, no ties in pooled sample
G08AKF Computes the exact probabilities for the Mann–Whitney U statistic, ties in pooled sample
G08ALF Performs the Cochran Q test on cross-classified binary data
G08BAF Mood's and David's tests on two samples of unequal size
G08CBF Performs the one-sample Kolmogorov–Smirnov test for standard distributions
G08CCF Performs the one-sample Kolmogorov–Smirnov test for a user-supplied distribution
G08CDF Performs the two-sample Kolmogorov–Smirnov test
G08CGF Performs the cc goodness of fit test, for standard continuous distributions
G08DAF Kendall's coefficient of concordance
G08EAF Performs the runs up or runs down test for randomness
G08EBF Performs the pairs (serial) test for randomness
G08ECF Performs the triplets test for randomness
G08EDF Performs the gaps test for randomness
G08RAF Regression using ranks, uncensored data
G08RBF Regression using ranks, right-censored data

Chapter G10 - Smoothing in Statistics

G10ABF Fit cubic smoothing spline, smoothing parameter given
G10ACF Fit cubic smoothing spline, smoothing parameter estimated
G10BAF Kernel density estimate using Gaussian kernel
G10CAF Compute smoothed data sequence using running median smoothers
G10ZAF Reorder data to give ordered distinct observations

Chapter G11 - Contingency Table Analysis

G11AAF χ^2 statistics for two-way contingency table
G11BAF Computes multiway table from set of classification factors using selected statistic
G11BBF Computes multiway table from set of classification factors using given percentile/quantile
G11BCF Computes marginal tables for multiway table computed by G11BAF or G11BBF
G11CAF Returns parameter estimates for the conditional analysis of stratified data
G11SAF Contingency table, latent variable model for binary data
G11SBF Frequency count for G11SAF

Chapter G12 - Survival Analysis

G12AAF Computes Kaplan–Meier (product-limit) estimates of survival probabilities
G12BAF Fits Cox's proportional hazard model
G12ZAF Creates the risk sets associated with the Cox proportional hazards model for fixed covariates

Chapter G13 - Time Series Analysis

G13AAF	Univariate time series, seasonal and non-seasonal differencing
G13ABF	Univariate time series, sample autocorrelation function
G13ACF	Univariate time series, partial autocorrelations from autocorrelations
G13ADF	Univariate time series, preliminary estimation, seasonal ARIMA model
G13AEF	Univariate time series, estimation, seasonal ARIMA model (comprehensive)
G13AFF	Univariate time series, estimation, seasonal ARIMA model (easy-to-use)
G13AGF	Univariate time series, update state set for forecasting
G13AHF	Univariate time series, forecasting from state set
G13AJF	Univariate time series, state set and forecasts, from fully specified seasonal ARIMA model
G13ASF	Univariate time series, diagnostic checking of residuals, following G13AEF or G13AFF
G13AUF	Computes quantities needed for range-mean or standard deviation-mean plot
G13BAF	Multivariate time series, filtering (pre-whitening) by an ARIMA model
G13BBF	Multivariate time series, filtering by a transfer function model
G13BCF	Multivariate time series, cross-correlations
G13BDF	Multivariate time series, preliminary estimation of transfer function model
G13BEF	Multivariate time series, estimation of multi-input model
G13BGF	Multivariate time series, update state set for forecasting from multi-input model
G13BHF	Multivariate time series, forecasting from state set of multi-input model
G13BJF	Multivariate time series, state set and forecasts from fully specified multi-input model
G13CAF	Univariate time series, smoothed sample spectrum using rectangular, Bartlett, Tukey or Parzen lag window
G13CBF	Univariate time series, smoothed sample spectrum using spectral smoothing by the trapezium frequency (Daniell) window
G13CCF	Multivariate time series, smoothed sample cross spectrum using rectangular, Bartlett, Tukey or Parzen lag window
G13CDF	Multivariate time series, smoothed sample cross spectrum using spectral smoothing by the trapezium frequency (Daniell) window
G13CEF	Multivariate time series, cross amplitude spectrum, squared coherency, bounds, univariate and bivariate (cross) spectra
G13CFF	Multivariate time series, gain, phase, bounds, univariate and bivariate (cross) spectra
G13CGF	Multivariate time series, noise spectrum, bounds, impulse response function and its standard error
G13DBF	Multivariate time series, multiple squared partial autocorrelations
G13DCF	Multivariate time series, estimation of VARMA model
G13DJF	Multivariate time series, forecasts and their standard errors
G13DKF	Multivariate time series, updates forecasts and their standard errors
G13DLF	Multivariate time series, differences and/or transforms (for use before G13DCF)
G13DMF	Multivariate time series, sample cross-correlation or cross-covariance matrices
G13DNF	Multivariate time series, sample partial lag correlation matrices, χ^2 statistics and significance levels
G13DPF	Multivariate time series, partial autoregression matrices
G13DSF	Multivariate time series, diagnostic checking of residuals, following G13DCF
G13DXF	Calculates the zeros of a vector autoregressive (or moving average) operator
G13EAF	Combined measurement and time update, one iteration of Kalman filter, time-varying, square root covariance filter
G13EBF	Combined measurement and time update, one iteration of Kalman filter, time-invariant, square root covariance filter

Chapter H - Operations Research

H02BBF Integer LP problem (dense)
H02BFF Interpret MPSX data file defining IP or LP problem, optimize and print solution
H02BUF Convert MPSX data file defining IP or LP problem to format required by H02BBF or E04MFF
H02BVF Print IP or LP solutions with user specified names for rows and columns
H02BZF Integer programming solution, supplies further information on solution obtained by H02BBF
H02CBF Integer QP problem (dense)
H02CCF Read optional parameter values for H02CBF from external file
H02CDF Supply optional parameter values to H02CBF
H02CEF Integer LP or QP problem (sparse)
H02CFF Read optional parameter values for H02CEF from external file
H02CGF Supply optional parameter values to H02CEF
H03ABF Transportation problem, modified stepping stone method
H03ADF Shortest path problem, Dijkstra's algorithm

Chapter M01 - Sorting

M01CAF Sort a vector, real numbers
M01CBF Sort a vector, integer numbers
M01CCF Sort a vector, character data
M01DAF Rank a vector, real numbers
M01DBF Rank a vector, integer numbers
M01DCF Rank a vector, character data
M01DEF Rank rows of a matrix, real numbers
M01DFF Rank rows of a matrix, integer numbers
M01DJF Rank columns of a matrix, real numbers
M01DKF Rank columns of a matrix, integer numbers
M01DZF Rank arbitrary data
M01EAF Rearrange a vector according to given ranks, real numbers
M01EBF Rearrange a vector according to given ranks, integer numbers
M01ECF Rearrange a vector according to given ranks, character data
M01EDF Rearrange a vector according to given ranks, complex numbers
M01ZAF Invert a permutation
M01ZBF Check validity of a permutation
M01ZCF Decompose a permutation into cycles

Chapter P01 - Error Trapping

P01ABF Return value of error indicator/terminate with error message

Chapter S - Approximations of Special Functions

S01BAF $\ln(1 + x)$
S01EAF Complex exponential, e^z
S07AAF $\tan x$
S09AAF $\arcsin x$
S09ABF $\arccos x$
S10AAF $\tanh x$
S10ABF $\sinh x$
S10ACF $\cosh x$
S11AAF $\text{arctanh } x$

S11ABF arcsinh x
S11ACF arccosh x
S13AAF Exponential integral $E_1(x)$
S13ACF Cosine integral Ci(x)
S13ADF Sine integral Si(x)
S14AAF Gamma function
S14ABF Log Gamma function
S14ACF $y(x) - \ln x$
S14ADF Scaled derivatives of $y(x)$
S14BAF Incomplete Gamma functions $P(a, x)$ and $Q(a, x)$
S15ABF Cumulative normal distribution function P(x)
S15ACF Complement of cumulative normal distribution function $Q(x)$
S15ADF Complement of error function erfc(x)
S15AEF Error function erf(x)
S15AFF Dawson's integral
S15DDF Scaled complex complement of error function, $\exp(-z^2)\text{erfc}(-iz)$
S17ACF Bessel function $Y_0(x)$
S17ADF Bessel function $Y_1(x)$
S17AEF Bessel function $J_0(x)$
S17AFF Bessel function $J_1(x)$
S17AGF Airy function Ai(x)
S17AHF Airy function Bi(x)
S17AJF Airy function Ai'(x)
S17AKF Airy function Bi'(x)
S17DCF Bessel functions $Y_{v+a}(z)$, real $a \geq 0$, complex z, $v = 0,1, 2,...$
S17DEF Bessel functions $J_{v+a}(z)$, real $a \geq 0$, complex z, $v = 0,1, 2,...$
S17DGF Airy functions Ai(z) and Ai'(z), complex z
S17DHF Airy functions Bi(z) and Bi'(z), complex z
S17DLF Hankel functions $H^{(j)}_{v+a}(z)$, j= 1,2, real $a \geq 0$, complex z, $v = 0,1, 2,...$
S18ACF Modified Bessel function $K_0(x)$
S18ADF Modified Bessel function $K_1(x)$
S18AEF Modified Bessel function $I_0(x)$
S18AFF Modified Bessel function $I_1(x)$
S18CCF Modified Bessel function $e^x K_0(x)$
S18CDF Modified Bessel function $e^x K_1(x)$
S18CEF Modified Bessel function $e^{-|x|} I_0(x)$
S18CFF Modified Bessel function $e^{-|x|} I_1(x)$
S18DCF Modified Bessel functions $k_{v+a}(z)$, real $a \geq 0$, complex z, $v = 0,1, 2,...$
S18DEF Modified Bessel functions $I_{v+a}(z)$, real $a \geq 0$, complex z, $v = 0,1, 2,...$
S19AAF Kelvin function ber x
S19ABF Kelvin function bei x
S19ACF Kelvin function ker x
S19ADF Kelvin function kei x
S20ACF Fresnel integral $S(x)$
S20ADF Fresnel integral $C(x)$
S21BAF Degenerate symmetrised elliptic integral of 1st kind $R_C(x,y)$

S21BBF Symmetrised elliptic integral of 1st kind $R_F(x,y,z)$
S21BCF Symmetrised elliptic integral of 2nd kind $R_D(x,y,z)$
S21BDF Symmetrised elliptic integral of 3rd kind $R_J(x,y,z,r)$
S21CAF Jacobian elliptic functions sn, cn and dn

Chapter X01 - Mathematical Constants

X01AAF Provides the mathematical constant π
X01ABF Provides the mathematical constant γ (Euler's Constant)

Chapter X02 - Machine Constants

X02AHF The largest permissible argument for sin and cos
X02AJF The machine precision
X02AKF The smallest positive model number
X02ALF The largest positive model number
X02AMF The safe range parameter
X02ANF The safe range parameter for complex floating-point arithmetic
X02BBF The largest representable integer
X02BEF The maximum number of decimal digits that can be represented
X02BHF The floating-point model parameter, b
X02BJF The floating-point model parameter, p
X02BKF The floating-point model parameter e_{min}
X02BLF The floating-point model parameter e_{max}
X02DAF Switch for taking precautions to avoid underflow
X02DJF The floating-point model parameter ROUNDS

Chapter X03 - Inner Products

X03AAF Real inner product added to initial value, basic/additional precision
X03ABF Complex inner product added to initial value, basic/additional precision

Chapter X04 - Input/Output Utilities

X04AAF Return or set unit number for error messages
X04ABF Return or set unit number for advisory messages
X04ACF Open unit number for reading, writing or appending, and associate unit with named file
X04ADF Close file associated with given unit number
X04BAF Write formatted record to external file
X04BBF Read formatted record from external file
X04CAF Print real general matrix (easy-to-use)
X04CBF Print real general matrix (comprehensive)
X04CCF Print real packed triangular matrix (easy-to-use)
X04CDF Print real packed triangular matrix (comprehensive)
X04CEF Print real packed banded matrix (easy-to-use)
X04CFF Print real packed banded matrix (comprehensive)
X04DAF Print complex general matrix (easy-to-use)
X04DBF Print complex general matrix (comprehensive)
X04DCF Print complex packed triangular matrix (easy-to-use)
X04DDF Print complex packed triangular matrix (comprehensive)
X04DEF Print complex packed banded matrix (easy-to-use)
X04DFF Print complex packed banded matrix (comprehensive)

X04EAF Print integer matrix (easy-to-use)
X04EBF Print integer matrix (comprehensive)

Chapter X05 - Date and Time Utilities

X05AAF Return date and time as an array of integers
X05ABF Convert array of integers representing date and time to character string
X05ACF Compare two character strings representing date and time
X05BAF Return the CPU time

Published November 1999