NAG Fortran Library, Mark 20

FLSG620DA

Silicon Graphics (IRIX 6) Double Precision

Users' Note

Contents

1. Introduction

This document is essential reading for every user of the NAG Fortran Library Implementation specified in the title. It provides implementation-specific detail that augments the information provided in the NAG Fortran Library Manual and Introductory Guide. Wherever those manuals refer to the "Users' Note for your implementation", you should consult this note.

In addition, NAG recommends that before calling any library routine you should read the following reference material (see Section 5):

(a) Essential Introduction
(b) Chapter Introduction

(c) Routine Document

The libraries supplied with this implementation have **not** been compiled in a manner that facilitates the use of multiple threads.

2. Availability of Routines

All routines listed in the chapter contents documents of the NAG Fortran Library Manual, Mark 20 are available in this implementation. At Mark 20, 95 new primary ("user-callable") routines and 36 thread safe equivalents of existing routines have been introduced. Two routines have been deleted. Please consult Mark 20 News (see Section 5) for lists of these routines and for a list of routines scheduled for withdrawal at Mark 21 or later. Your suggestions for new algorithms for future releases of the Library are welcomed (see Section 7).

3. General Information

This implementation includes libraries in both archive and dynamic share object (DSO) forms for the following combinations of Application Binary Interfaces (ABI) and Instruction Set Architectures (ISA):

high performance 32-bit (-n32) mode using -mips3
high performance 32-bit (-n32) mode using -mips4
64-bit mode using -mips3
64-bit mode using -mips4

3.1. Accessing the Library

You may link to the NAG Fortran Library in the following manner:

```
f77 driver.f -lnag
```

or

```
f90 driver.f -lnag
```

where driver.f is the user's application program.

The specific library used by the linker depends on the compilation options, if any, specified by the user and on the system processor. On systems such as the Silicon Graphics Power Indigo2, based on a R8000 processor, the default, if the SGI_ABI environment variable has not been set, is 64-bit and MIPS IV.

On systems such as the Silicon Graphics Origin 2000, based on a R10000 processor, the default, if the SGI_ABI environment variable has not been set, is high performance 32-bit (-n32) and MIPS IV.

To use a library that requires the BLAS and LAPACK routines from the SGI Scientific Computing Software Library (SCSL), the compilation command above should be modified to

```
f77 driver.f -lnag_scs -lscs
```

or

```
f90 driver.f -lnag_scs -lscs
```

This links to the NAG library as well as the Silicon Graphics BLAS and LAPACK routines in SCSL.

Note that the choice of the library linked to can be overridden by specifying the application binary interface (ABI), that is -n32 or -64, and the instruction set architecture (ISA), that is -mips3 or -mips4.

Gaining access to the libraries is a simple process, loading the nag module (`module load nag`) is the only step that is required.

3.2. Example Programs

The example programs are most easily accessed by the command nagexample, which will provide you with a copy of an example program (and its data, if any), compile the program and link it with the library (showing you the compile command so that you can recompile your own version of the program). Finally, the executable program will be run, presenting its output to stdout. The example program concerned is specified by the argument to nagexample, e.g.

```
nagexample c06eaf
```

will copy the example program and its data into the files c06eafe.f and c06eafe.d in the current directory and process them to produce the example program results.

In the NAG Fortran Library Manual, routine documents that have been typeset since Mark 12 present the example programs in a generalised form, using bold italicised terms as described in Section 3.3.

In other routine documents, the example programs are in single precision and require modification for use with double precision routines. This conversion can entail:

- Changing REAL or COMPLEX type specifications to REAL*8 or COMPLEX*16
- Changing certain intrinsic function references, e.g. REAL or FLOAT to DBLE, ALOG to DLOG, CMPLX to DCMPLX, and so on
- Changing real constants to double precision form, e.g. 0.1 or 0.1E0 to 0.1D0

The example programs supplied to a site in machine-readable form have been modified as necessary so that they are suitable for immediate execution. Note that the distributed example programs are those used in this implementation and may not correspond exactly with the programs published in the manual. The distributed example programs should be used in preference wherever possible.

3.3. Interpretation of Bold Italicised Terms

For this double precision implementation, the bold italicised terms used in the NAG Fortran Library Manual should be interpreted as:

```
real                 - DOUBLE PRECISION (REAL*8)
basic precision      - double precision
complex              - COMPLEX*16
additional precision - quadruple precision (REAL*16)
machine precision    - the machine precision, see the value
```

```
                      returned by X02AJF in Section 4
```

Thus a parameter described as ***real*** should be declared as DOUBLE PRECISION in your program. If a routine accumulates an inner product in ***additional precision***, it is using software to simulate quadruple precision.

In routine documents that have been newly typeset since Mark 12 additional bold italicised terms are used in the published example programs and they must be interpreted as follows:

```
real as an intrinsic function name - DBLE
imag                               - DIMAG
cmplx                              - DCMPLX
conjg                              - DCONJG
e in constants, e.g. 1.0e-4        - D, e.g. 1.0D-4
e in formats, e.g. e12.4           - D, e.g. D12.4
```

All references to routines in Chapter F07 - Linear Equations (LAPACK) and Chapter F08 - Least-squares and Eigenvalue Problems (LAPACK) use the LAPACK name, not the NAG F07/F08 name. The LAPACK name is precision dependent, and hence the name appears in a bold italicised typeface.

The typeset examples use the single precision form of the LAPACK name. To convert this name to its double precision form, change the first character either from S to D or C to Z as appropriate.
For example:

```
sgetrf refers to the LAPACK routine name - DGETRF
cpotrs                                   - ZPOTRS
```

3.4. Explicit Output from NAG Routines

Certain routines produce explicit error messages and advisory messages via output units which either have default values or can be reset by using X04AAF for error messages and X04ABF for advisory messages. (The default values are given in Section 4). The maximum record lengths of error messages and advisory messages (including carriage control characters) are 80 characters, except where otherwise specified. These routines are potentially not thread safe and in general output is not recommended in a multithreaded environment.

3.5. Interface Blocks

The NAG Fortran Library Interface Blocks define the type and arguments of each user callable NAG Fortran Library routine. These are not essential to calling the NAG Fortran Library from Fortran 90 programs. Their purpose is to allow the Fortran 90 compiler to check that NAG Fortran Library routines are called correctly. The interface blocks enable the compiler to check that:

(a) Subroutines are called as such
(b) Functions are declared with the right type
(c) The correct number of arguments are passed
(d) All arguments match in type and structure

These interface blocks have been generated automatically by analysing the source code for the NAG Fortran Library. As a consequence, and because these files have been thoroughly tested, their use is recommended in preference to writing your own declarations.

The NAG Fortran Library Interface Block files are organised by Library chapter. The module names are:

```
nag_f77_a_chapter
nag_f77_c_chapter
nag_f77_d_chapter
nag_f77_e_chapter
nag_f77_f_chapter
nag_f77_g_chapter
nag_f77_h_chapter
nag_f77_m_chapter
nag_f77_p_chapter
nag_f77_s_chapter
nag_f77_x_chapter
```

These are supplied in pre-compiled form (.mod files) and they can be accessed by specifying the `-I`"pathname" option on each f90 invocation, where "pathname" is the path of the directory containing the .mod files.

In order to make use of these modules from existing Fortran 77 code the following changes need to be made:

- Add a USE statement for each of the module files for the chapters of the NAG Fortran Library that your program calls directly. Often only one USE statement will be required.
- Delete all EXTERNAL statements for NAG Fortran Library routines. These are now declared in the module(s).
- Delete the type declarations for any NAG Fortran Library functions. These are now declared in the module(s).

These changes are illustrated by showing the conversion of the Fortran 77 version of the example program for NAG Fortran Library routine S18DEF. Please note that this is not exactly the same as the example program that is distributed with this implementation. Each change is surrounded by comments boxed with asterisks.

```
*     S18DEF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*******************************************************************
* Add USE statement for relevant chapters                         *
      USE NAG_F77_S_CHAPTER
*                                                                 *
*******************************************************************
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          N
      PARAMETER        (N=2)
*     .. Local Scalars ..
      COMPLEX*16       Z
      DOUBLE PRECISION FNU
      INTEGER          IFAIL, NZ
      CHARACTER*1      SCALE
*     .. Local Arrays ..
      COMPLEX*16       CY(N)
*     .. External Subroutines ..
*******************************************************************
* EXTERNAL declarations need to be removed (and type declarations *
*  for functions).                                                *
C      EXTERNAL         S18DEF
*                                                                 *
```

```
*******************************************************************
*     .. Executable Statements ..
      WRITE (NOUT,*) 'S18DEF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      WRITE (NOUT,*)
      WRITE (NOUT,99999) 'Calling with N =', N
      WRITE (NOUT,*)
      WRITE (NOUT,*)
     +'   FNU            Z        SCALE       CY(1)              CY(2)
     +   NZ IFAIL'
      WRITE (NOUT,*)
   20 READ (NIN,*,END=40) FNU, Z, SCALE
      IFAIL = 0
*
      CALL S18DEF(FNU,Z,N,SCALE,CY,NZ,IFAIL)
*
      WRITE (NOUT,99998) FNU, Z, SCALE, CY(1), CY(2), NZ, IFAIL
      GO TO 20
   40 STOP
*
99999 FORMAT (1X,A,I2)
99998 FORMAT (1X,F7.4,'  (',F7.3,',',F7.3,')   ',A,
     +       2('  (',F7.3,',',F7.3,')'),I4,I4)
      END
```

4. Routine-specific Information

Any further information which applies to one or more routines in this implementation is listed below, chapter by chapter.

(a) D03

The example programs for D03RAF and D03RBF take much longer to run than other examples.

(b) F06, F07 and F08

In this implementation calls to the Basic Linear Algebra Subprograms (BLAS) and linear algebra routines (LAPACK) can be resolved by calls to SCSL.

(c) G02

The value of ACC, the machine-dependent constant mentioned in several documents in the chapter, is 1.0D-13.

(d) P01

On hard failure, P01ABF writes the error message to the error message unit specified by X04AAF and then stops.

(e) S07 - S21

The constants referred to in the NAG Fortran Library Manual have the following values in this implementation:

```
S07AAF  F(1)   = 1.0D+13
```

```
                F(2)   = 1.0D-14

        S10AAF  E(1)   = 18.50
        S10ABF  E(1)   = 708.0
        S10ACF  E(1)   = 708.0

        S13AAF  x(hi)  = 708.3
        S13ACF  x(hi)  = 5.6D+14
        S13ADF  x(hi)  = 5.6D+14

        S14AAF  IFAIL  = 1 if X > 170.0
                IFAIL  = 2 if X < -170.0
                IFAIL  = 3 if abs(X) < 2.23D-308
        S14ABF  IFAIL  = 2 if X > 2.55D+305

        S15ADF  x(hi)  = 26.6
                x(low) = -6.25
        S15AEF  x(hi)  = 6.25

        S17ACF  IFAIL  = 1 if X > 5.6D+14
        S17ADF  IFAIL  = 1 if X > 5.6D+14
                IFAIL  = 3 if 0.0 < X <= 2.23D-308
        S17AEF  IFAIL  = 1 if abs(X) > 5.6D+14
        S17AFF  IFAIL  = 1 if abs(X) > 5.6D+14
        S17AGF  IFAIL  = 1 if X > 1.038D+2
                IFAIL  = 2 if X < -8.9D+9
        S17AHF  IFAIL  = 1 if X > 1.041D+2
                IFAIL  = 2 if X < -8.9D+9
        S17AJF  IFAIL  = 1 if X > 104.1
                IFAIL  = 2 if X < -1.8D+9
        S17AKF  IFAIL  = 1 if X > 104.1
                IFAIL  = 2 if X < -1.8D+9
        S17DCF  IFAIL  = 2 if abs (Z) < 3.93D-305
                IFAIL  = 4 if abs (Z) or FNU+N-1 > 3.27D+4
                IFAIL  = 5 if abs (Z) or FNU+N-1 > 1.07D+9
        S17DEF  IFAIL  = 2 if imag (Z) > 700.0
                IFAIL  = 3 if abs (Z) or FNU+N-1 > 3.27D+4
                IFAIL  = 4 if abs (Z) or FNU+N-1 > 1.07D+9
        S17DGF  IFAIL  = 3 if abs (Z) > 1.02D+3
                IFAIL  = 4 if abs (Z) > 1.04D+6
        S17DHF  IFAIL  = 3 if abs (Z) > 1.02D+3
                IFAIL  = 4 if abs (Z) > 1.04D+6
        S17DLF  IFAIL  = 2 if abs (Z) < 3.93D-305
                IFAIL  = 4 if abs (Z) or FNU+N-1 > 3.27D+4
                IFAIL  = 5 if abs (Z) or FNU+N-1 > 1.07D+9

        S18ADF  IFAIL  = 2 if 0.0 < X <= 2.23D-308
        S18AEF  IFAIL  = 1 if abs(X) > 711.6
        S18AFF  IFAIL  = 1 if abs(X) > 711.6
        S18CDF  IFAIL  = 2 if 0.0 < X <= 2.23D-308
        S18DCF  IFAIL  = 2 if abs (Z) < 3.93D-305
                IFAIL  = 4 if abs (Z) or FNU+N-1 > 3.27D+4
                IFAIL  = 5 if abs (Z) or FNU+N-1 > 1.07D+9
        S18DEF  IFAIL  = 2 if real (Z) > 700.0
                IFAIL  = 3 if abs (Z) or FNU+N-1 > 3.27D+4
                IFAIL  = 4 if abs (Z) or FNU+N-1 > 1.07D+9

        S19AAF  IFAIL  = 1 if abs(x) >= 49.50
        S19ABF  IFAIL  = 1 if abs(x) >= 49.50
        S19ACF  IFAIL  = 1 if X > 997.26
        S19ADF  IFAIL  = 1 if X > 997.26

        S21BCF  IFAIL  = 3 if an argument < 1.579D-205
                IFAIL  = 4 if an argument >= 3.774D+202
        S21BDF  IFAIL  = 3 if an argument < 2.820D-103
                IFAIL  = 4 if an argument >= 1.404D+102
```

(f) X01

The values of the mathematical constants are:

```
X01AAF (PI)    = 3.1415926535897932
X01ABF (GAMMA) = 0.5772156649015329
```

(g) X02

The values of the machine constants are:

The basic parameters of the model

```
X02BHF =     2
X02BJF =    53
X02BKF =  -1021
X02BLF =  1024
X02DJF =  .TRUE.
```

Derived parameters of the floating-point arithmetic

```
X02AJF = Z'3CA0000000000001' ( 1.11022302462516D-16 )
X02AKF = Z'0010000000000000' ( 2.22507385850721D-308 )
X02ALF = Z'7FEFFFFFFFFFFFFF' ( 1.79769313486231D+308 )
X02AMF = Z'0010000000000000' ( 2.22507385850721D-308 )
X02ANF = Z'0020000000000000' ( 4.45014771701441D-308 )
```

Parameters of other aspects of the computing environment

```
X02AHF = Z'4300000000000000' ( 5.62949953421312D+14 )
X02BBF = 2147483647
X02BEF = 15
X02DAF = .FALSE.
```

(h) X04

The default output units for error and advisory messages for those routines which can produce explicit output are both Fortran Unit 6.

It is not possible to use the routines X04ACF and X04ADF with the f90 compiler.

(i) X05

The finest granularity of wall-clock time available on this system is one second, so the seventh element of the integer array passed as a parameter to X05AAF will always be returned with the value 0.

5. Documentation

Each supported NAG Fortran Library site is ordinarily provided with a printed copy of the NAG Fortran Library Manual (or Update) and Introductory Guide. Additional copies are available for purchase; please refer to the NAG websites or contact your local NAG Response Centre for current prices (see Section 6).

A full on-line version of the NAG Fortran Library Manual is supplied in the form of Portable

Document Format (PDF) files, with an HTML index, in the NAGdoc directory. The introductory material is also provided as HTML files in the NAGdoc directory.

A main index file has been provided (NAGdoc/fl/html/mark20.html) which contains a fully linked contents document pointing to all the available PDF (and where available HTML) files. Use your HTML browser to navigate from here.

In addition the following are provided in the doc directory:

- `un.html` - Users' Note (this document)
- `blas_lapack_to_nag` - BLAS/F06, LAPACK/F07 and LAPACK/F08 listing
- `nag_to_blas_lapack` - F06/BLAS, F07/LAPACK and F08/LAPACK listing

6. Support from NAG

(a) Contact with NAG

Queries concerning this document or the implementation generally should be directed initially to your local Advisory Service. If you have difficulty in making contact locally, you can contact NAG directly at one of the addresses given in the Appendix. Users subscribing to the support service are encouraged to contact one of the NAG Response Centres (see below).

(b) NAG Response Centres

The NAG Response Centres are available for general enquiries from all users and also for technical queries from sites with an annually licensed product or support service.

The Response Centres are open during office hours, but contact is possible by fax, email and phone (answering machine) at all times.

When contacting a Response Centre it helps us deal with your enquiry quickly if you can quote your NAG site reference and NAG product code (in this case CHANGE FLSOL20DA).

(c) NAG Websites

The NAG websites provide information about implementation availability, descriptions of products, downloadable software, product documentation and technical reports. The NAG websites can be accessed at

http://www.nag.co.uk/, http://www.nag.com/ (in North America) or http://www.nag-j.co.jp/ (in Japan)

(d) NAG Electronic Newsletter

If you would like to be kept up to date with news from NAG then please register to receive our free electronic newsletter, which will alert you to special offers, announcements about new products or product/service enhancements, customer stories and NAG's event diary. To register simply visit one of our websites or contact us at nagnews@nag.co.uk.

7. User Feedback

Many factors influence the way NAG's products and services evolve and your ideas are invaluable in helping us to ensure that we meet your needs. If you would like to contribute to this process we would be delighted to receive your comments. Please contact your local NAG Response Centre (shown below).

Appendix - Contact Addresses

NAG Ltd
Wilkinson House
Jordan Hill Road
OXFORD OX2 8DR
United Kingdom

Tel: +44 (0)1865 511245
Fax: +44 (0)1865 310139

NAG Ltd Response Centre
email: infodesk@nag.co.uk

Tel: +44 (0)1865 311744
Fax: +44 (0)1865 311755

NAG Inc
1400 Opus Place, Suite 200
Downers Grove
IL 60515-5702
USA

Tel: +1 630 971 2337
Fax: +1 630 971 2706

NAG Inc Response Center
email: infodesk@nag.com

Tel: +1 630 971 2345
Fax: +1 630 971 2346

NAG GmbH
Schleißheimerstraße 5
85748 Garching
Deutschland
email: naggmbh@nag.co.uk

Tel: +49 (0)89 320 7395
Fax: +49 (0)89 320 7396

Nihon NAG KK
Yaesu Nagaoka Building No. 6
1-9-8 Minato
Chuo-ku
Tokyo
Japan
email: help@nag-j.co.jp

Tel: +81 (0)3 5542 6311
Fax: +81 (0)3 5542 6312

[NP3631/UN]