Using the IMSL & NAG Libraries

A handbook describing how to use the IMSL* and NAG** Libraries of numerical mathematical and statistical subroutines as installed at the JRC Computing Centre, Ispra.

Author: Martyn D. Dowell

Version 1, September 1981

* IMSL is the trademark of the International Mathematical and Statistical Library Inc. (Houston, USA)

** NAG is the trademark of the Numerical Algorithms Group Ltd. (Oxford, UK)
CONTENTS

1. INTRODUCTION
   1.1 General Introduction
   1.2 The Pitfalls in Writing Numerical Software!

2. THE LIBRARIES
   2.1 The IMSL Library
   2.2 The NAG Library

3. COMPARISON OF THE LIBRARIES
   3.1 Which Library Should I Use?
   3.2 Comparative Summary of the Content of the IMSL and NAG Libraries

4. DOCUMENTATION
   4.1 Overview
   4.2 IMSL Documentation
   4.3 NAG Documentation

5. IMPLEMENTATION SPECIFIC DETAILS
   5.1 IMSL Implementation Details
   5.2 NAG Implementation Details

6. USING THE LIBRARIES
   6.1 Library Data Set Definition
   6.2 Use of the Libraries in Batch
   6.3 Use of the Libraries from a TSO Session

7. INCLUSION OF THE LIBRARIES IN OTHER LANGUAGE PROGRAMS

APPENDIX A
   Examples of the Use of the Libraries in Batch Jobs

APPENDIX B
   Detailed Comparison of Content of the Libraries
1. INTRODUCTION

1.1 General Introduction

In the design and implementation of computer programs there is always a requirement for the inclusion of modules (procedures, subroutines, functions) which perform specific well defined tasks. The most obvious examples of this are modules for performing transfers from peripherals and generally handling input/output devices. The program writer would almost never consider writing his own routine to read a card from the card reader or write a record to a lineprinter. Similarly, basic trigonometric and mathematical functions such as \( \sin(x) \), \( \log(x) \) and \( e^x \) are always provided as standard. However, in the field of more advanced numerical mathematical and statistical calculations there has been a tradition of users writing their own subroutines to provide specific facilities. This has occurred for several reasons; the two most important are:

1) No good, comprehensive, well tested, well documented sets of routines have been available.
2) Users have always considered that they are capable of producing good routines suitable for their own needs.

In recent years the first of these reasons has become much less valid with the advent and subsequent development of two competing and yet complementary libraries of numerical mathematical and statistical subroutines (The International Mathematical and Statistical Library IMSL and the Numerical Algorithms Group NAG Library).

FORTRAN versions of both of these libraries are available for use on the JRC-Ispra Computing Centre Service. These libraries are rented from the organizations on an annual basis and are freely available for use to all of the local users of the JRC-Ispra Computing Centre Service. External and commercial users of the service should seek advice as to the conditions under which they may use these libraries from the Computer Manager (see the JRC Newsletter list of personnel for details).

Note. Users should note that single routines of IMSL and NAG may absolutely not be distributed outside the JRC, Ispra Establishment. However, complete programs or software systems which make use of the libraries may be distributed. For these cases users may request only object decks of the incorporated routines. The person who makes the request becomes responsible for any misuse of the requested deck.
1.2 The Pitfalls in Writing Numerical Software!

The second reason why users have habitually written their own numerical mathematical subroutines (as given in the previous section) is in almost all cases false! Perhaps a few program writers produce adequate numerical mathematical subroutines for their programs. However, very many more (by far the majority) produce subroutines which are inadequate and often produce results which are unnecessarily erroneous. This may be displayed by the following example (first described in the Newsletter of the Computer Center of Purdue University (USA)).

The object of this example is to illustrate the quality which has been built into the IMSL & NAG Libraries. We do this by solving a problem, using the algorithm many people would use, and then by comparing the results we obtain with those of the corresponding IMSL routine.

The problem we choose is to find the roots of a quadratic equation: given real numbers \( a, b, \) and \( c, \) find \( x \) such that \( ax^2 + bx + c = 0. \) For simplicity, we assume that \( a,b \) and \( c \) are such that the solution is also real. The two roots of a quadratic equation may be found by the well-known 'Quadratic Formula'

\[
x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
\]

where one root is obtained by using the "+" of the "\( \pm \)", and the other is obtained with the "-". The assumption that the roots are real means that \( b^2 - 4ac \geq 0. \) We can solve this problem with the following straightforward subroutine:

```fortran
SUBROUTINE QUAD (A,B,C,X1,X2)
  D = SQRT(B*B-4.0*A*C)
  X1 = 0.5*(-B+D)/A
  X2 = 0.5*(-B-D)/A
RETURN
END
```

When we use this subroutine to solve the rather difficult quadratic \( x^2 + (2^{13} + 2^{-13})x + 1 = 0 \) we obtain

\[
X1 = 0.0
X2 = -0.8192 \times 10^4
\]
This is not the correct solution, however. The corresponding IMSL routine ZQADR (single precision) does compute a much better approximation to the actual solution as follows:

\[
\begin{align*}
x_1 &= -0.12070 \times 10^{-3} \\
x_2 &= -0.81920 \times 10^4
\end{align*}
\]

(The these results are exact to 5 significant figures)

Where did QUAD go wrong? The second statement computes \( D \). Then the third statement forms the difference between \( D \) and \( B \), losing almost all significance in the process. ZQADR is more careful than QUAD and thus is able to retain full significance.

Now we touch briefly on several additional problems with QUAD. The first deals with problem scaling. If the quadratic equation above is multiplied by a constant, the solution is not changed mathematically, but it is changed computationally. For example, the quadratic

\[
10^{40}x^2 + 3 \times 10^{40}x + 2 \times 10^{40} = 0
\]

results in an overflow error because \( 10^{80} \) cannot be represented by the computer. Similarly, the quadratic

\[
10^{-40}x^2 + 3 \times 10^{-40}x + 2 \times 10^{-40} = 0
\]

produces an underflow and then gives results

\[
x_1 = x_2 = -1.5
\]

because \( 10^{-80} \) cannot be represented by the computer and is treated as zero. However, in both cases ZQADR still computes the same, correct solution.

Finally, consider QUAD's actions if the coefficient of \( X \) in the quadratic is zero. In this case QUAD returns an infinite value for one root and an indefinite value for the other. ZQADR returns the mathematically correct value \(-c/b\) for one root and infinity for the other.

The reader may ask: "What is the point of this if I never intend to solve difficult quadratic equations?". The answer is that this example shows the problems of trying to write a subroutine for solving a simple problem. It is much more difficult to write a good or even adequate subroutine to solve more complicated problems.
2. THE LIBRARIES

2.1 The IMSL Library

The International Mathematical and Statistical Library (IMSL) is produced by IMSL INC. of Houston, Texas (USA).

The library is a FORTRAN source library which contains over 400 user-callable subroutines. For IBM sites (such as the JRC-Ispra) two load module versions of the library are available, which contain sets of the subroutines with single and double precision real parameters.

The company was founded in the early 1970's and has now a very strong world-wide user base. It is especially strong in the North American continent. In total, the number of installations subscribing to the IMSL Library exceeds 1700 (located in 36 different countries). IMSL has an estimated user base of 106,000 persons.

A board of technical advisors to IMSL which consists of many famous experts in numerical mathematics, statistics and computer science, is responsible for ensuring that the library maintains a set of high quality subroutines which reflects the current state of the science.

The library is divided into a number of chapters, each of which covers an area of numerical mathematics or statistics. A brief list of the topics included follows:

Analysis of Experimental Design Data
Basic Statistics
  Data Screening: Transgeneration
  Elementary Classical Inference
  Elementary Bayesian Inference
Categorized Data Analysis
Differential Equations; Quadrature; Differentiation
Eigenanalysis
Forecasting; Econometrics; Time Series
Generation and Testing of Random Numbers; Goodness of Fit
Interpolation; Approximation; Smoothing
Linear Algebraic Equations
Mathematical and Statistical Functions
  Probability Distribution Functions
  Special Functions of Mathematical Physics
Non-Parametric Statistics
  Analyses of Variance
  Binomial or Multi-nomial Bases
  Hyper (or Multi-hyper) Geometric Bases
  Kolmogorov-Smirnov Tests
  Other Bases
  Randomization Bases
Observation Structure
   Canonical Analysis
   Cluster Analysis
   Discriminant Analysis
   Factor Analysis
   Principal Components Analysis
Regression Analysis
   Linear Models
Special Non-linear Models
Sampling
   Acceptance Sampling
   Preference Testing
   Survey Sampling
Utility Functions
   Error Detection
   Special I/O Routines
Vector, Matrix Arithmetic
   Zeros and Extrema; Linear Programming

Subroutines in the IMSL library have alphanumeric names of up to six characters in length. The first character of the name is always that of the chapter in which the subroutine is located.

2.2 The NAG Library
The NAG Library is produced and distributed by the British company NAG (Numerical Algorithms Group) Ltd. The NAG project began in 1970 when six British computer centres decided to jointly develop a library of mathematical routines. Later, many other universities and research organizations became involved in the development and use of the library. In 1975 the library distribution service to commercial subscribers began. The library now consists of approximately 400 user-callable FORTRAN subroutines (there are also Algol 60 and Algol 68 versions of the library available, but not at the JRC-Ispra). There are FORTRAN implementations of the library on approximately 50 machine ranges. The total number of installations using the library world-wide exceeds 300. The library is most popular in Europe although there is an increasing usage in the USA and Canada. The library is always distributed in compiled form (i.e. as a load module library for IBM users). Source copies of individual routines are available for inspection. For IBM users (such as the JRC-Ispra Computing Centre Service) there are two load module versions of the library using single and double precision real parameters.
The aim of NAG has always been to produce a comprehensive library of subroutines and to have as main criteria for selection of these routines the concepts of:

i) usefulness ii) robustness iii) numerical stability iv) accuracy v) speed

Contributors to the NAG Library are expert numerical mathematicians and computer scientists in the UK and throughout the world. These are backed by the NAG Central Office staff who co-ordinate and control the work of the contributions. The NAG Library is structured in chapters using the conventions adopted by the American A.C.M. (Association for Computing Machines) modified SHARE Classification Index.

Summary of the Chapters of the NAG FORTRAN Library

A02 - COMPLEX ARITHMETIC
C02 - ZEROS OF POLYNOMIALS
C05 - ROOTS OF ONE OR MORE TRASCENDENTAL EQUATIONS
C06 - SUMMATION OF SERIES
D01 - QUADRATURE
D02 - ORDINARY DIFFERENTIAL EQUATIONS
D04 - NUMERICAL DIFFERENTIATION
D05 - INTEGRAL EQUATIONS
E01 - INTERPOLATION
E02 - CURVE AND SURFACE FITTING
E04 - MINIMIZING OR MAXIMIZING A FUNCTION
F01 - MATRIX OPERATIONS INCLUDING INVERSION
F02 - EIGENVALUES AND EIGENVECTORS
F03 - DETERMINANTS
F04 - SIMULTANEOUS LINEAR EQUATIONS
F05 - ORTHOGONALISATION
G01 - SIMPLE CALCULATIONS ON STATISTICAL DATA
G02 - CORRELATION AND REGRESSION ANALYSIS
G04 - ANALYSIS OF VARIANCE
G05 - RANDOM NUMBER GENERATORS
H - OPERATIONS RESEARCH
M01 - SORTING
P01 - ERROR TRAPPING
S - MATHEMATICAL CONSTANTS
X01 - MACHINE CONSTANTS
X03 - INNERPRODUCTS
Subroutines in the NAG Library have names which are defined in the following manner:

* The name is 6 alphanumeric characters.
* The first 3 characters are the name of the chapter in which the subroutine is found (see previous page). For example C05 for a routine which is concerned with a technique in the subject area of roots of transcendental equations.
* The 4th and 5th characters are alphabetic and serve to distinguish between different subroutines in the same chapter.
* The 6th character defines the language type of the subroutine (A for Algol, F for FORTRAN etc.); see section 5.2 for more details.
3. COMPARISON OF THE LIBRARIES

3.1 Which Library Should I Use?
In some situations this question will be easy to answer. If a certain algorithm is only implemented in the NAG Library (for example) then the user must obviously make use of this subroutine.
If, however, equivalent subroutines are available in both the NAG Library and the IMSL Library, then other factors are involved in the choice.

* For cases in which the finished program or package is to be transported to another computer site or another computing system, the decision may depend on which of the two libraries will be available at the other site or system.

* For packages which are to be made generally available to a large number of users, the location of the users may be important. As previously stated, the NAG Library is widely available at centres in Europe, whilst the IMSL Library is more predominant in the USA and Canada.

* For cases in which such constraints do not apply, for example a small program to be run on the JRC-Isoia Computing Centre Service, then personal preference will be the deciding factor. The user should compare the specification and description of the equivalent routines and decide which is the more appropriate.

In the following section a brief summary and comparison of the content of the two libraries is given.

3.2 Comparative Summary of the Content of the IMSL and NAG Libraries
The following comparison gives a very basic idea of the strengths and weaknesses of the two libraries. It is very much the author's opinion on the subject and does not in any way express the views of either IMSL or NAG.
The list is ordered by the chapter of the IMSL Library. It gives details of each chapter with some idea of the coverage of the NAG Library in each subject area. At the end of this list details of NAG Library subject areas not covered by the IMSL chapters are given.
IMSL Chapter A - Analysis of Experimental Design Data
This subject area is well covered in the IMSL Library. In the NAG Library the G04 chapter contains only a subset of the material covered by IMSL.

IMSL Chapter B - Basic Statistics
IMSL Chapter C - Categorized Data Analysis
Are equivalent to the material covered in the NAG G01 and G02 chapters. In general, there is a more varied coverage in the IMSL Library, although the NAG Library implements some algorithms which are not included in the IMSL Library.

IMSL Chapter D - Differential Equations, Quadrature, Differentiation
This IMSL chapter is equivalent to the four NAG chapters:

D01 - Quadrature
D02 - Ordinary Differential Equations
D03 - Partial Differential Equations
D04 - Numerical Differentiation

Both libraries cover this area well. However, the NAG coverage is better.

IMSL Chapter E - Eigenanalysis
The IMSL chapter is equivalent to the NAG Chapter F02. Both of the libraries have good subroutines available for all the commonly required form of the eigenvalue problem. The NAG chapter is of an especially high quality.

IMSL Chapter F - Forecasting, Time Series Analysis, Fourier Transforms
The Fourier transform section is equivalent to the NAG C06 chapter. The forecasting and time series analysis subroutines are not available in the NAG Library (but are planned for a future release in the G13 chapter).

IMSL Chapter G - Generation and Testing of Pseudo-random Numbers
The NAG G05 chapter contains the equivalent subroutines. Both libraries give an excellent coverage of this area.

IMSL Chapter I - Interpolation, Approximation, Smoothing
The NAG E01 chapter covers the subject area of interpolation. The NAG E02 chapter covers the subject area of approximation and smoothing.
IMSL Chapter L - Linear Algebraic Equations
The NAG F04 chapter contains the equivalent subroutines. As with the chapter on eigenanalysis, this subject area is given excellent coverage by both libraries.

IMSL Chapter M - Mathematical and Statistical Special Functions
The NAG S chapter contains the equivalent subroutines. There is a wide diversity of the special functions which are covered. The user will generally need to consult both libraries to find an implementation of the subroutine required. IMSL has more statistical special functions and NAG has more mathematical special functions.

IMSL Chapter N - Non-parametric Statistics
Subroutines for this subject area are only available in the IMSL Library.

IMSL Chapter O - Observation Structure, Multivariate Statistics
The NAG G08 chapter covers this area. There is a much wider coverage in the IMSL Library.

IMSL Chapter R - Regression Analysis
The NAG G02 chapter contains some subroutines which cover part of the subject area. The NAG G02 chapter does not include subroutines for stepwise regression analysis or curvilinear regression analysis. For curvilinear regression analysis the NAG documentation suggest the choice of a model and then the use of a least squares fit subroutine for the E04 (minimization) chapter.

IMSL Chapter S - Sampling
Subroutines for this subject area are only available in the IMSL Library.

IMSL Chapter U - Utility Subroutines
This chapter has no equivalent chapter in the NAG Library. It consists of two separate subsets:
1) Subroutines for input/output in various special forms such as matrix input/output lineprinter histogram drawing.
2) HELP subroutines, to obtain information about various IMSL Library aspects.

IMSL Chapter V - Vector Arithmetic & Sorting
The NAG F01 chapter contains the equivalent subroutines for vector arithmetic. The coverage of the subject area in both libraries is good. The NAG M chapter contains the equivalent sorting routines.

- 10 -
IMSL Chapter Z Zeros and Extrema, Linear Programming

The NAG C02, C05 and E04 chapters contain the subroutines for zeros and extrema.
The NAG H Chapter contains the linear programming subroutines.
The NAG E04 Chapter gives a much wider coverage of the general problem of finding local maxima or minima of a function.
The NAG H Chapter (Operations Research) contains more than simply linear programming subroutines.

Chapters and Facilities in NAG which are not available in IMSL:

NAG has explicit chapters for mathematical and machine constants (X01 and X02).
Full chapters or determinants (F03) and orthonormalization (F05) are present in the NAG Library.
A chapter on integral equations (D05) is provided in the NAG Library.

In appendix B a full comparative list of the various subroutines in the NAG & IMSL libraries is given.

Both organizations actively encourage users to request inclusion of any algorithms which are not present. IMSL even provides a formal "RAI" request service (Request for Ability Inclusion) with a specific form for the users to complete.
4. DOCUMENTATION

4.1 Documentation Overview

Before using a subroutine from either of the libraries it is necessary to read the relevant documentation. This, and the following sections, give information which will assist the programmer to use the available documentation effectively. The documentation may be considered in five separate parts.

1) General introductory information.
   Information of a general introductory type may be found in this document and in the general introductions to the NAG Library Manual and the IMSL Library Manual. Both of these manuals are available for reference in the Computing Support Library (Room 1871 building A36)

2) Subject introduction and algorithm choice.
   Documentation which gives background information about a subject area in numerical mathematics or statistics, together with advice on choice of subroutines for different problems within the subject area. This information may be found in the introduction to the relevant chapter of the NAG Library Manual or the IMSL Library Manual. Also, in the case of the NAG Library only, there is a publication titled the NAG Mini-Manual, which is simply a collection of all the chapter introduction documents. Again this manual is available for reference in the Computing Support Library.

3) Individual Subroutines Documentation.
   Documentation about individual subroutines which explains in detail how to use the subroutines in a standard FORTRAN manner (i.e. without details of how to use it on the JRC-Ispra Computing Centre Service. This information may be found in the individual routine documents in the IMSL Library Manual and the NAG Library Manual.

4) Implementation Specific Documentation.
   Documentation which relates the documentation described in 2) and 3) to a particular implementation of the IMSL or NAG Library (e.g. documentation giving details of IBM specific features of certain aspects of the particular library).
This information is provided in the form of a short document by both IMSL and NAG. Copies of these documents have been included in the reference copies of both the IMSL Library Manual and the NAG Library Manual. Also, some of the more important information of this type is presented in the following sections of this document.

5) Use of the libraries at the JRC-Ispra Documentation giving details of how to use the library on the JRC-Ispra Computing Centre Service. This information is available in section 6. of this document.

4.2 IMSL Documentation
The IMSL documentation is in the form of the IMSL Library Manual. This is at present in three volumes. The manual structure is as follows:

Introduction
A general introduction which also gives some information about specific features relating to different computer-compiler environments.

Contents
A brief description of such subroutines, which is ordered by chapter.

KWIC Index
A keyword in Context (KWIC) index of the subroutines to help the user located the chapter/subroutine required.

Chapters A-Z
Each chapter is split into two parts:

a) The general introduction which contains:

Chapter Name
Quick Reference Guide to Chapter Facilities
Featured Abilities
Name Conventions for this Chapter
Special Instructions on Usage (Optional)
Subtleties to Note (Optional)
Pitfalls to Avoid (Optional)

The chapter introduction is followed by individual subroutine documentation.

b) Individual subroutine documents
Subroutine documentation consists of two parts. The first is a copy of comment lines that appear at the beginning of each subroutine source deck.
The comment lines are as follows:

**IMSL ROUTINE NAME** - routine name

**PURPOSE** - a statement of the purpose of the routine

**USAGE** - the form of the subprogram CALL with arguments listed

**ARGUMENTS** - a description of the arguments in the order of their occurrence in USAGE

**PRECISION/HARDWARE** - environment specific information giving the precision of the routine - SINGLE, or DOUBLE

**REQD. IMSL ROUTINES** - a list of all IMSL routines called (directly and indirectly) by this routine

**NOTATION** - reference to manual introduction and IMSL routine UHELP

**REMARKS** (optional) - details pertaining to code usage

The second part of the document (which does not appear in the source code) includes the following sections:

**ALGORITHM** - a brief statement of the algorithm and references to detailed information

**PROGRAMMING NOTES** (optional) - programming details not covered elsewhere

**ACCURACY** (optional) - a statement about the accuracy of the routine

**EXAMPLE** - an example showing subroutine input, required dimension and type statements and output

---

**4.3 NAG Documentation**

The NAG documentation for use at the JRC-Ispra Computing Centre Service is in the form of three publications:

1) the NAG Library Manual (at present in 6 volumes)
2) the NAG Mini-Manual (the chapter introduction for the NAG Library Manual)
3) the NAG IBM FORTRAN Implementation Documents (single and double precision)
The NAG Library Manual
The manual structure is as follows:

Foreword
Written by Professor L. Fox (Oxford University) and Dr. J.H. Wilkinson (N.P.L.) This is interesting and educational reading.

Introduction
Contains a great deal of important information.

Chapters A02-X04
Each is split into two parts:

a) the chapter introduction which contains:
   1. The scope of the chapter
   2. Background to the problems
   3. Recommendations on choice and use of routines

b) the individual routine documents which are structured as follows:

   All routine documents have 13 numbered sections with the following headings:
   1. Purpose
   2. Specification
   3. Description
   4. References
   5. Parameters
   6. Error Indicators
   7. Auxiliary Routines
   8. Timing
   9. Storage
  10. Accuracy
  11. Further Comments
  12. Keywords
  13. Example

The NAG Mini-Manual
This is formed from the foreword, introduction and all of the chapter introductions. This is a useful manual for an introduction to the NAG Library and for helping the user to find the routine which he requires. Actual routine specifications are, however, only found in the NAG Library Manual.

The NAG IBM FORTRAN·Implementation Documents
These documents contain details of how the IBM FORTRAN implementation of the NAG Library should be used (in a general sense) and how the implementation differs from the standard implementation (as defined by the NAG Library Manual). It is important that all prospective library users read the appropriate single or double precision documents. Both documents are inserted at the beginning of each NAG Library Manual.

- 15 -
5. IMPLEMENTATION SPECIFIC DETAILS

5.1 IMSL Implementation Details
The subroutine naming convention in the IMSL libraries is the same for both single and double precision versions. This implies that in one program it is not normally possible to include one routine from the single precision library and another routine from the double precision library.

Although most subroutines are available in both single and double precision versions, there are some exceptions. In each routine specification document there is a section "PRECISION/HARDWARE". The user must check this to make sure that the routine is implemented in the required environment.

SINGLE/H32 means that the subroutine is available in the IBM FORTRAN single-precision library.
DOUBLE/H32 means that the subroutine is available in the IBM FORTRAN double-precision library.

Examples given in the routine specification will normally require some modifications before being suitable for use with either the single or double precision libraries. Normally, the examples will have been written for the single precision version, and therefore the normal changes (of REAL to DOUBLE PRECISION etc.) will be necessary to run them with the double precision version of the subroutines.

5.2 NAG Implementation Details
The naming convention for NAG subroutines is given in Section 2.2. The sixth character of the NAG subroutine name is a letter which defines the language type of the subroutine. This letter is also used to create a difference between the single and double precision IBM FORTRAN subroutines.

For the single precision library the sixth character of the routine name is always E.
For the double precision library the sixth character of the routine name is always F.

Therefore, all subroutines in the single precision library and double precision library have different names. Thus, mixing of the use of different single and double precision NAG subroutines in one program is possible.
For double precision:

```plaintext
CALL E04CGF(N,X,F,IW,LIW,W,LW,IFAIL)
```

For single precision:

```plaintext
CALL E04CGE(N,X,F,IW,LIW,W,LW,IFAIL)
```

In the NAG Library Manual there are certain terms which are italicized. The implication is that these terms are implementation dependent and should be replaced by the appropriate actual term for the implementation being used (ie for the single precision IBM FORTRAN implementation or the double precision IBM FORTRAN implementation).

<table>
<thead>
<tr>
<th>Italicized term</th>
<th>IBM FORTRAN single precision</th>
<th>IBM FORTRAN double precision</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>real</code></td>
<td>(REAL*4)</td>
<td>DOUBLE PRECISION (REAL*8)</td>
</tr>
<tr>
<td><code>complex</code></td>
<td>COMPLEX*8</td>
<td>COMPLEX*16</td>
</tr>
<tr>
<td><code>basic precision</code></td>
<td>single precision</td>
<td>double precision</td>
</tr>
<tr>
<td><code>additional precision</code></td>
<td>double precision (REAL*8)</td>
<td>quadruple precision (REAL*16)</td>
</tr>
</tbody>
</table>

Example programs published in the Library Manual are in single precision. Therefore, for use with the single precision library they should seldom require any modification.

For use of example programs with the double precision library may require modifications in the following area:

1) Inserting, as appropriate REAL*8 and COMPLEX*16 statements
2) Changing any intrinsic functions to double precision version e.g. SORT to DSQRT
3) Specifying real constants in double precision (D) format.
4) Explicitly converting any implicit integer to real conversions using the DFLOAT function
5) Changing any E formats to D formats
6. USING THE LIBRARIES

The IMSL Library and NAG Library are available for use on the JRC-Ispra Computing Centre Service. The following sections describe their use in FORTRAN programs, both in batch and from a TSO foreground session. In section 7, a brief description is given of how to include NAG and IMSL subroutines in programs written in other languages.

6.1 Library data set definitions

There are four IMSL and NAG load module libraries available for use. These are the single and double precision versions of both libraries. The names of the data sets are given in the following table:

<table>
<thead>
<tr>
<th>Library</th>
<th>Data set name</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSL single precision</td>
<td>SYS1.LIBMASXS</td>
</tr>
<tr>
<td>IMSL double precision</td>
<td>SYS1.LIBMASXD</td>
</tr>
<tr>
<td>NAG single precision</td>
<td>SYS1.LIBNAGS</td>
</tr>
<tr>
<td>NAG double precision</td>
<td>SYS1.LIBNAGD</td>
</tr>
</tbody>
</table>

ALL OF THE DATA SETS IN THE ABOVE TABLE ARE CATALOGED

6.2 Use of the Libraries in Batch

Users may easily access one of the libraries by using one of the standard FORTRAN G1 compiler procedures in the following manner:

// EXEC FTG1CG,PRN=abcd

where abcd is replaced as follows:

NAGS - NAG single precision library
NAGD - NAG double precision library
MASXS - IMSL single precision library
MASXD - IMSL double precision library

So, for example:

// EXEC FG1CG,PRN=NAGS

implies a FORTRAN compilation, load and go with subroutines in the program from the NAG single-precision library.
Note. The use of the FG1CG procedure is only an example. The use of these libraries in this manner is possible for all of the following FORTRAN G1 & HE procedures.

<table>
<thead>
<tr>
<th>FG1CL</th>
<th>FTHECL</th>
</tr>
</thead>
<tbody>
<tr>
<td>FG1CLG</td>
<td>FTHECLG</td>
</tr>
<tr>
<td>FG1CG</td>
<td>FTHECG</td>
</tr>
<tr>
<td>FG1L</td>
<td>FTHEL</td>
</tr>
<tr>
<td>FG1LG</td>
<td>FTHELG</td>
</tr>
<tr>
<td>FG1G</td>
<td>FTHEG</td>
</tr>
</tbody>
</table>

In appendix A examples (with explanation) are given of the use of the above mentioned system.

Use of More than One Library in the Same Program
As has been stated in section 5.1, it is not normally possible to use different subroutines from both the IMSL single and double precision libraries in the same program. This is due to the fact that the naming conventions are the same for the two libraries. However, it is possible to mix subroutines from the two NAG library and also the mix subroutines from the NAG libraries with one of the IMSL libraries. In this case it is not possible to use the job control command specified in the previous section because it is necessary to include subroutines from more than one of the libraries.
In this case the libraries are included into the standard SYSLIB by concatenation.

e.g.

```
// EXEC FG1CLG
//CMP.SYSIN DD *
.
.
.
FORTAN source program
.
.
*/
//LKED.SYSLIB DD
// DD
// DD
// DD
// DD DSN=SYS1.LIBNAGS,DISP= SHR
// DD DSN=SYS1.LIBNAGD,DISP= SHR
```

In general, any 2 or 3 of the 4 libraries (except combinations including the two IMSL libraries) may be included by concatenation in this manner (see Appendix A for an example).
6.3 Use of Library from a TSO Session

There are two different ways in which subroutines from the mathematical libraries may be used in TSO test FORTRAN compilations.

a) Using the FG1CLG command procedure to perform the compilation, link edit and execution. With this procedure it is possible to obtain the necessary subroutines from one of the libraries by including a specific parameter.

b) Using the FORT TSO command followed by either LOADGO or LINK it is possible to include subroutines from one or more of the libraries.

Note. It is also possible to make use of the mathematical libraries by using the CONCAT TSO command. For further details see the IBM Manual TSO Command Reference Manual (GC28-6732).

a) Using FG1CLG
The TSO command procedure FG1CLG has a parameter PRN (---). This parameter is equivalent to the use of the PRN= parameter of FG1CLG in a batch mode. This allows the inclusion of subroutines from one of the mathematical libraries.

The PRN(---) parameter has as operand the xxxx part of the library name SYS1.LIBxxxx (e.g. for SYS1.LIBNAGS the user should include the parameter PRN(NAGS)). An example of the use of this technique is given in the following example (example 1).

b) Use of FORT followed by either LOADGO or LINK
Using this method the compilation of the FORTRAN program is split into two separate phases:

1) The compilation to produce an object module.

2) The use of either the link-editor or the loader to take the object module together with any appropriate subroutine from the various subroutine libraries and produce an executable program.

Note. If the link editor procedure is used then a load module version of the program is created. This must be executed using the TSO CALL command.

- 20 -
In the second stage of this operation it is possible to include subroutines from one or more of the mathematical libraries by using the LIB(---) parameter. Examples of the use of these techniques are given in examples 2, 3 and 4.

Note for NAG Users

Users are reminded that in the case of NAG library routines the routine names are different for single and double precision libraries.

E04CAF on the double precision library becomes:
E04CAE on the single precision library (i.e. the final character is changed from F to E)

However, for the IMSL library the routine names are the same for the single and double precision versions. Therefore, in general, it will not be possible to use subroutines from the single & double precision IMSL libraries in the same program.
Examples of TSO Usage of NAG & IMSL Subroutines

In the following examples lines typed by the user are shown in lower case. The carriage return/ENTER character at the end of each input line is marked by a CR.

The following type of examples are given:

Example 1:
This example shows the use of the FG1CLG TSO command procedure for a program which uses the IMSL single precision library.

Example 2:
This example shows the use of the NAG double precision library using FORT followed by LOADGO.

Example 3:
This example shows the compilation link edit and execution of a program which uses the NAG single precision library.

Example 4:
This example shows the compilation, load and execution of a program which uses both the NAG and IMSL single precision libraries.
Example 1

This example shows the use of the FG1CLG TSO command procedure for a program which uses the IMSL single precision library.

```fortran
qed newcomp1 forti
ew

INPUT
00010c example of imsl single precision library
00020c analysis of two-way classification design data
00030 integer i,ndf(5),ier
00040 real y(6),em(11),gm,s(5)
00050 data y/73.,90.,98.,107.,94.,99./
00060 call arcban(y,1,3,2,em,gm,s,ndf,ier)
00070 write(6,99999) (em(i),i=1,3)
00080 write(6,99998) (em(i),i=4,5)
00090 stop

0010099999 format(18h block means are : , 11f7.2)
0011099998 format(22h treatment means are : ,11f7.2)

end
end save

SAVED
READY

fg1clg newcomp1 prn(masxs)

DATA SET NEWCOMP1.LIST NOT IN CATALOG
DATA SET NEWCOMP1.OBJ NOT IN CATALOG
DATA SET NEWCOMP1.DECK NOT IN CATALOG
DATA SET NEWCOMP1.SYSUT1 NOT IN CATALOG
DATA SET NEWCOMP1.DUMP NOT IN CATALOG
DATA SET NEWCOMP1.LOAD NOT IN CATALOG

UTILITY DATA SET NOT FREED, IS NOT ALLOCATED
UTILITY DATA SET NOT FREED, IS NOT ALLOCATED
ENTER CONTROL STATEMENTS-

END OF CONTROL STATEMENTS

BLOCK MEANS ARE : 81.50 102.50 71.50
TREATMENT MEANS ARE : 88.33 82.00
READY

In stage A a FORTRAN program is typed by the user (using the QED editor). Note the use of the SCAN subcommand of QED to check the validity of the FORTRAN program.

In stage B the FG1CLG TSO command procedure is used to compile link edit and execute the program. Note, that link editor control statements may be input. Typing a CR without any other information ends these control statements.
Example 2

This example shows the use of the NAG double precision library using FORT followed by LOADGO.

```
list newcomp2.fort

NEWCOMP2.FORT

C EXAMPLE OF NAG DOUBLE PRECISION LIBRARY
C USES F03AAF - MATRIX DETERMINANT CALCULATION
C DOUBLE PRECISION DETERM,A(4,4),WKSPCE(18)
C INTEGER I,N,J,IA,IFAIL
C READ(5,99999) (WKSPCE(I),I=1,7)
N=3
C READ(5,99998) ((A(I,J),J=1,N),I=1,N)
IA=4
IFAIL=1
C CALL F03AAF(A,IA,N,DETERM,WKSPCE,IFAIL)
C WRITE(6,99997) (WKSPCE(I),I=1,6)
C IF(IFAIL.EQ.0) GOTO 20
C WRITE(6,99996) IFAIL
C STOP
C WRITE(6,99995) DETERM
C STOP
C99999 FORMAT(6A4,A3)
C99998 FORMAT(3F5.0)
C99997 FORMAT(4(1X/),1H5,5A4,A3,7RESULTS/1X)
C99996 FORMAT(25HOERROR IN F03AAF IFAIL = ,I2)
C99995 FORMAT(24HOVALUE OF DETERMINANT = ,F4.1)
END
```

fort newcomp2

G1 COMPILER ENTERED

SOURCE ANALYZED

PROGRAM NAME = MAIN

* NO DIAGNOSTICS GENERATED

READY

loadgo newcomp2.obj lib('sys.libname')

f03aaf example program data

```
C1
-24 -10 -57
-8 -4 -17
```

F03AAF EXAMPLE PROGRAM RESULTS

C2

VALUE OF DETERMINANT = 6.0

In stage A a previously created data set is listed.
In stage B the program stored in the data set is compiled.
In stage C the load and execution of the program is performed.
In C1 the data is input.
In C2 the output is produced.
Example 3

This example shows the compilation link edit and execution of a program which uses the NAG single precision library.

```
list newcomp3.fort

NEWCOMP3.FORT
00010 INTEGER MAXDIV,IFAIL,NOFUN
00020 REAL A,B,EPS,ACC,ANS,ERROR,FUN
00030 EXTERNAL FUN
00040 A=0.0
00050 B=1.0
00060 MAXDIV=20
00070 EPS=1.0E-8
00080 ACC=0.0
00090 IFAIL=1
00100 CALL D01AGE(A,B,FUN,MAXDIV,EPS,ACC,ANS,ERROR,NOFUN,
00110 * IFAIL)
00120 WRITE(6,99998)ANS,ERROR,NOFUN
00130 IF(IFAIL)20,40,20
00140 20
00140 WRITE(6,99997)
00150 40 STOP
00160 99998 FORMAT(/12H INTEGRAL = ,F11.4,3X,9H ERROR = ,E11.4,3X,
00170 * 20H NUMBER OF POINTS = ,I3)
00180 99997 FORMAT(43H METHOD WAS UNABLE TO EVALUATE THE INTEGRAL)
00190 END
00200 REAL FUNCTION FUN(X)
00210 REAL X
00220 FUN=4.0/(1.0+X*X)
00230 RETURN
00240 END

READY

fort newcomp3

G1 COMPILER ENTERED
SOURCE ANALYZED
PROGRAM NAME = MAIN
B * NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = FUN
* NO DIAGNOSTICS GENERATED
*STATISTICS* NO DIAGNOSTICS THIS STEP
READY

link newcomp3.obj lib('sys1.libnags') fortlib

READY

call newcomp3

TEMPPNAME ASSUMED AS A MEMBER NAME

D

INTEGRAL = 3.1416 , ERROR = 0.3052E-04 NUMBER OF POINTS = 9

In stage A a previously created data set is listed.
In stage B the program stored in the data set is compiled.
In stage C the link edit of the program is performed.
The output load module of the program is stored in
NEWCOMP3.LOAD(TEMPPNAME).
In stage D the library program is executed.
```
Example 4

This example shows the compilation, load and execution of a program which uses both the NAG and IMSL single precision libraries.

```
list newcomp4.fort (CR)
NEWCOMP4.FORT
00010 C  EXAMPLE OF THE USE OF TWO LIBRARIES
00020 C  NAG & IMSL (BOTH SINGLE PRECISION)
00030 C  THE PROGRAM FINDS THE ROOT OF A FUNCTION
00040 INTEGER MAXFN,IER
00050 REAL A,B,FUN
00060 EXTERNAL FUN
00070 A=-10.
00080 B=10.
00090 MAXFN=100
00100 CALL ZBRENT(FUN,0.0,8,A,B,MAXFN,IER)
00110 IF(IER.EQ.0)GOTO 20
A 00120 WRITE(6,99998)IER
00130 STOP
00140 20 WRITE(6,99999)B
00150 STOP
00160 99999 FORMAT(20HOESTIMATE OF ROOT = ,E10.3)
00170 99998 FORMAT(25HOERROR IN ZBRENT IER = ,I2)
00180 END
00190 REAL FUNCTION FUN(X)
00200 REAL X
00210 FUN=S15ACE(X,IFAIL)-0.75
00220 RETURN
00230 END
READY
fort newcomp4 (CR)
1 COMPILER ENTERED
SOURCE ANALYZED
PROGRAM NAME = MAIN
B  NO DIAGNOSTICS GENERATED
SOURCE ANALYZED
PROGRAM NAME = FUN
  NO DIAGNOSTICS GENERATED
  *STATISTICS* NO DIAGNOSTICS THIS STEP
READY
loadgo newcomp4.obj lib('sys1.libnags' 'sys1.libmasxs') fortlib (CR)
C  ESTIMATE OF ROOT = -0.674E+00
READY
```

In stage A a previously created data set is listed.
In stage B the program stored in the data set is compiled.
In stage C the load and execution of the program is performed.
(Note, in particular, the use of the LIB parameter with two libraries.)
7. INCLUSION OF LIBRARIES IN OTHER LANGUAGE PROGRAMS

Information regarding the use of the IMSL and NAG libraries in programs written in other programming languages (i.e. non-FORTRAN programs) will be included in a later version of this document.
APPENDIX A

Examples of the Use of the Libraries in Batch Jobs

1. Example of Use of the IMSL Library

The example shows the use of the IMSL single precision library using the IMSL subroutine ZX3LP which is an "easy-to-use" linear programming subroutine which uses the revised Simplex algorithm (see Hadley, G. "Linear Programming", Addison-Wesley, Reading, Massachusetts, 1962).

The problem is to maximise \( x_1 + 3x_2 = S \)

Subject to the constraints:

\[
\begin{align*}
x_1 & \geq 1 \\
x_2 & \leq 1 \\
x_1 + x_2 & \leq 1.5 \\
x_1 + x_2 & \geq 0.5 \\
x_1 & \geq 0 \\
x_2 & \geq 0
\end{align*}
\]

Results of Example

\[
\begin{align*}
\text{ZX3LP EXAMPLE PROGRAM RESULTS} \\
\text{VALUE OF OBJECTIVE FUNCTION} &= 3.500 \\
\text{SOLUTION VECTOR} &= 0.500 \quad 1.000
\end{align*}
\]

See next page for listing of example
Listing of Example 1 Job

// JOB (YOUR JOB CARD)
$ CLASS 2
// EXEC PTG1CG, PRN=MASXS
// CMP.SYSIN DD *
C ZX3LP EXAMPLE PROGRAM
C
INTEGER IA,N,M1,M2,IW(16),IER
REAL A(6,2),B(6),C(2),RW(52),PSOL(4),DSOL(6),S
C NUMBER OF UNKNOWNS
N=2
C M1=NUMBER OF INEQUALITY CONSTRAINTS
M1=4
C M2=NUMBER OF EQUALITY CONSTRAINTS
M2=0
C IA=FIRST DIMENSION OF A
IA=6
C SET UP MATRIX OF CONSTRAINTS
A(1,1)=1.0
A(1,2)=0.0
A(2,1)=0.0
A(2,2)=1.0
A(3,1)=1.0
A(3,2)=1.0
A(4,1)=-1.0
A(4,2)=-1.0
C VECTOR OF RIGHT-HAND SIDES OF CONSTRAINT EQUATIONS
B(1)=1.0
B(2)=1.0
B(3)=1.5
B(4)=-0.5
C COEFFICIENTS OF OBJECTIVE FUNCTIONS
C(1)=1.0
C(2)=3.0
CALL ZX3LP (A,IA,B,C,N,M1,M2,S,PSOL,DSOL,RW,IW,IER)
C CHECK IF ERROR (IER NOT EQUAL TO ZERO)
IF(IER.NE.0)WRITE(6,100)IER
IF(IER.NE.0)GOTO 20
C WRITE RESULTS
WRITE(6,1001)S,PSOL(1),PSOL(2)
20 STOP
100 FORMAT(I4,F8.2,F8.2,F8.2)
1000 FORMAT(' ERROR IN ZX3LP IER=',I5)
1001 FORMAT(' ZX3LP EXAMPLE PROGRAM RESULTS'/
1 VALUE OF OBJECTIVE FUNCTION=',F8.3/
2 SOLUTION VECTOR=',2F10.3)
END
2. Example of the Use of the NAG Library

The example shows the use of the NAG double precision library. The example shows the use of the NAG subroutine E04CGF which implements an easy-to-use "quasi-Newton" algorithm (see Gill P.E. & Murray W., "Quasi-Newton methods for unconstrained optimization", Journal of the Institute of Mathematics and its Applications, 1972, Vol. 9, 91-108) for finding an unconstrained minimum of a function $F(X_1, X_2, \ldots, X_n)$ of the N independent variables $X_1, X_2, \ldots, X_n$ using function values only.

In the example the function which is minimized is

$$F(X_1, X_2) = e^{X_2} \cdot (4X_1^2 + 2X_2^2 + 4X_1X_2 + 2X_1 + 1)$$

starting from an initial guess of $X_1 = -1$ and $X_2 = 1$.

Results of Example

E04CGF EXAMPLE PROGRAM RESULTS

<table>
<thead>
<tr>
<th>FUNCTION VALUE ON EXIT IS</th>
<th>0.0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT THE POINT</td>
<td>0.5000</td>
</tr>
</tbody>
</table>

See next page for listing of example
Listing of Example 2 Job

// JOB(Your Job Card)
$ CLASS 2
// EXEC FTG1CG,PRN=NAGD
//CMP.SYSIN DD *
C E04CGF EXAMPLE PROGRAM TEXT
C ..LOCAL SCALARS..
   DOUBLE PRECISION F
   INTEGER I, IFAIL, LIW, LW, N, NOUT
C ..LOCAL ARRAYS..
   DOUBLE PRECISION W(29), X(2)
   INTEGER IW(4)
C ..SUBROUTINE REFERENCES..
C E04CGF
C ..
DATA NOUT /6/
WRITE(NOUT,99999)
N=2
X(1)=-1.0D+0
X(2)= 1.0D+0
LIW=4
LW=29
IFAIL=1
CALL E04CGF(N,X,F,IW,LIW,W,LW,IFAIL)
C SINCE IFAIL WAS SET TO 1 BEFORE ENTERING E04CGF, IT IS
C ESSENTIAL TO TEST WHETHER IFAIL IS NON-ZERO ON EXIT
IF(IFAIL.NE.0) WRITE(NOUT,99999) IFAIL
IF(IFAIL.EQ.1) GO TO 20
WRITE(NOUT,99997) F
WRITE(NOUT,99996) (X(I),I:1,N)
20 STOP
C END OF E04CGF EXAMPLE MAIN PROGRAM
99999 FORMAT (///31H E04CGF EXAMPLE PROGRAM RESULTS/)
99998 FORMAT (16H ERROR EXIT TYPE,I3, 23H - SEE ROUTINE DOCUMENT)
99997 FORMAT (27H FUNCTION VALUE ON EXIT IS ,F12.4)
99996 FORMAT (13H AT THE POINT, 2F12.4)
END
C SUBROUTINE FUNCT1(N, XC, FC)
C FUNCTION EVALUATION ROUTINE FOR E04CGF EXAMPLE PROGRAM -
C THIS ROUTINE MUST BE CALLED FUNCT1
C ..SCALAR ARGUMENTS..
   DOUBLE PRECISION FC
   INTEGER N
C ..ARRAY ARGUMENTS..
   DOUBLE PRECISION XC(N)
C ..
C ..LOCAL SCALARS..
   DOUBLE PRECISION X1, X2
C ..FUNCTION REFERENCES..
   DOUBLE PRECISION DEXP
C ..
   X1=XC(1)
   X2=XC(2)
   FC=DEXP(X1)*(4.0D+0*X1*(X1+X2)+2.0D+0*X2*(X2+1.0D+0)+1.0D+0)
RETURN
C END OF FUNCTION EVALUATION ROUTINE
END
3. An Example of More than one Library in a Batch Job

The test program which is shown as example 4 in section 6.3 (for a TSO session) may be executed using the following batch job.

```plaintext
// JOB(YOUR JOB CARD)
$ CLASS 2
// EXEC FTG1CLG
//CMP.SYSIN DD *
C EXAMPLE OF THE USE OF TWO LIBRARIES
C NAG & IMSL (BOTH SINGLE PRECISION)
C THE PROGRAM FINDS THE ROOT OF A FUNCTION
INTEGER MAXFN,IER
REAL A,B,FUN
EXTERNAL FUN
A=-10.
B=10.
MAXFN=100
CALL ZBRENT(FUN,0.0,8,A,B,MAXFN,IER)
IF(IER.EQ.0)GOTO 20
WRITE(6,99998)IER
STOP
20 WRITE(6,99999)B
STOP
99999 FORMAT(20HORSTIMATE OF ROOT = ,E10.3)
99998 FORMAT(25HERROR IN ZBRENT IER = ,I2)
END
REAL FUNCTION FUN(X)
REAL X
IFAIL=0
FUN=SL5ACE(X,IFAIL)-0.75
RETURN
END

//LINKED.SYSLIB DD
// DD
// DD
// DD
// DD DSN=SYS1.LIBNAGS,DISP=SHR
// DD DSN=SYS1.LIBMASXS,DISP=SHR
END OF DATA
```
APPENDIX B

Detailed Comparison of Content of the Libraries

(based on a table produced by Dr. P. Kemp, University of Newcastle (U.K.))

<table>
<thead>
<tr>
<th>A02 COMPLEX ARITHMETIC</th>
<th>NAG</th>
<th>IMSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>square root</td>
<td>A02AAF</td>
<td>-</td>
</tr>
<tr>
<td>modulus</td>
<td>A02ABF</td>
<td>-</td>
</tr>
<tr>
<td>quotient</td>
<td>A02ACF</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C02 ZEROS OF POLYNOMIALS</th>
<th>NAG</th>
<th>IMSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>complex coefficients</td>
<td>C02ADF</td>
<td>ZCPOLY</td>
</tr>
<tr>
<td>real coefficients</td>
<td>C02AEF</td>
<td>ZRPOLY</td>
</tr>
<tr>
<td>quadratic, real coeffs</td>
<td>-</td>
<td>ZQADR</td>
</tr>
<tr>
<td>, complex coeffs</td>
<td>-</td>
<td>ZQADC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C05 ROOTS OF ONE OR MORE TRANSCENDENTAL EQUATIONS</th>
<th>NAG</th>
<th>IMSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>real function of one variable</td>
<td>C05AAF</td>
<td>ZBRENT</td>
</tr>
<tr>
<td>, Bus &amp; Dekker alg.</td>
<td>C05ABF</td>
<td>ZFALSF</td>
</tr>
<tr>
<td>, bin. search B &amp; D</td>
<td>C05ACF</td>
<td>ZREAL1</td>
</tr>
<tr>
<td>, continuation secant</td>
<td>C05AZF</td>
<td>ZREAL2</td>
</tr>
<tr>
<td>, bin. search, reverse comm.</td>
<td>C05ADF</td>
<td>-</td>
</tr>
<tr>
<td>, as C05AJF, reverse comm.</td>
<td>C05AGF</td>
<td>-</td>
</tr>
<tr>
<td>complex analytic function</td>
<td>C05AJF</td>
<td>-</td>
</tr>
<tr>
<td>n equations, n variables, functions</td>
<td>C05NAF</td>
<td>ZANLYT</td>
</tr>
<tr>
<td></td>
<td>ZSYSTM</td>
<td>ZSCNT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C06 SUMMATION OF SERIES, FOURIER TRANSFORMS</th>
<th>NAG</th>
<th>IMSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFT, $2^m$ real data values</td>
<td>C06AAF</td>
<td>-</td>
</tr>
<tr>
<td>FFT, real data values</td>
<td>C06EAF</td>
<td>FFTRC</td>
</tr>
<tr>
<td>(uses extra workspace)</td>
<td>C06FAF</td>
<td>-</td>
</tr>
<tr>
<td>FFT, Hermitian sequence</td>
<td>C06EBF</td>
<td>-</td>
</tr>
<tr>
<td>(uses extra workspace)</td>
<td>C06FBF</td>
<td>-</td>
</tr>
<tr>
<td>FFT, $2^m$ complex data values</td>
<td>C06ABF</td>
<td>FFT2C</td>
</tr>
<tr>
<td>FFT, complex data values</td>
<td>C06ADF</td>
<td>FFTCC</td>
</tr>
<tr>
<td>(uses extra workspace)</td>
<td>C06ECF</td>
<td>-</td>
</tr>
<tr>
<td>FFT estimates. power, cross spectra</td>
<td>-</td>
<td>FTFPS</td>
</tr>
<tr>
<td>real circular convolution, period 2m</td>
<td>C06ACF</td>
<td>-</td>
</tr>
<tr>
<td>sin, cos transforms. real series</td>
<td>-</td>
<td>FFTSC</td>
</tr>
</tbody>
</table>
sum of Chebyshev series C06DBF  -  
conjugate of Hermitian sequence C06GBF  -  
conjugate of complex sequence C06GCF  -  
inverse Laplace transform  - FLINV  
FFT of array (1,2 or 3 dim)  - FFTT3D  

**D01 QUADRATURE**

finite interval D01ACF DCADRE 
(Patterson algorithm) D01AGF DCSQDU 
(de Doncker algorithm) D01AHF - 
(for oscillating fns.) D01AJF - 
(user-specified singularities) D01ALF - 
(log-type end point sing.) D01APF - 
(Cauchy princ. value) D01AQF - 
(non-adaptive) D01BDF -  
Gauss. integral weights and abscissae D01BBF -  

infinite interval D01ANF - 
double integral D01DAF DBCQDU 
D01BCF - 

multiple integral, Monte Carlo, Gauss D01FAF -  
, adaptive D01FBF -  

trigonometric integral D01ANF -  
tabular function D01GAF -  
spline E02BDF -  

**D02 ORDINARY DIFFERENTIAL EQUATIONS**

IV problem, range D02BAF DREBS  
(stiff) D02CAF DVREK  
, range with output D02EBF -  
(stiff) D02CAF DGEAR  
, range, err. est. stiff chk D02EAF DBEFS  
, until soln comp. zero D02EBF -  
(stiff) D02EAF DBEFS  
, until fn. of soln. zero D02EBF -  
(stiff) D02EAF DBEFS  
, comprehensive control D02PAF -  
(stiff) D02QAF D02QBF  
interpolation for D02PAF, all comps. D02XAF -  
, one comp. D02XBF -  
D02QA/BF, all comp. D02XGF -  
, one comp. D02XHF -  

- 34 -
one step Runge Kutta
BV problem, 2 point
, 2-point, linear
, 2-point, non-linear
, generalized
collocation method

eigenvalues St-L., reg., finite range
general
eigenfns. St-L., general

D03 PARTIAL DIFFERENTIAL EQUATIONS

elliptic, Laplace 2-d
, soln f.d. eqs. 5pt 2-d mol.
, Stone's strongly imp. 5pt
, soln f.d. eqs. 7pt 3-d mol.
, Stone's strongly imp. 7pt
triangulation
parabolic, 2 point BV, non linear
, (single eq.)
, (general sys.)

D04 NUMERICAL DIFFERENTIATION

fn of single real variable
partial differentiation

D05 INTEGRAL EQUATIONS

Fredholm, 2nd kind, split kernel
, smooth kernel

E01 INTERPOLATION

1 variable, equal spacing
, unequal spacing
, cubic spline
, periodic cubic spline
, Chebyshev polynomial
2 variables

- 35 -
E02 CURVE AND SURFACE FITTING

1-s curve, cubic splines
, polynomial
, Chebyshev series
, user supplied basis
1-s surface fit

minimax curve fit
minimax minimax soln. over-deter. lin. eq.
L1 approx. linear fn
constraints
rational approx.

Pade approximant
evaluate Chebyshev series
cubic spline
ditto derivs.
ditto, definite integral
poly in 2 vers, from E02CAF
bi-cubic spline
rational fn. from E01RAF
poly. from Cheby. series
Chebyshev coeffs. of derivative
ditto of integral

sort 2-d data into panels
generate points in n-dim space
data smoothing, outlier detection
cubic spline
(easy to use)
, quasi Hermite

E04 MINIMISING OR MAXIMISING A FUNCTION

1 variable, fn values
, fn, 1 deriv.
1 var, easy use, fn values
, fn, 1 deriv.
, fn, 1 deriv.
1 var, comprehensive, fn values
, fn, 1 deriv.
bounded vars, easy use, fn values
, fn, 1 deriv.
bounded vars, comprehensive, fn values
, fn, 1 deriv.
, fn, 1 deriv.
constrained, fn values

- 36 -
, fn, 1 deriv.
, fn, 1 2 deriv.
, quadratic form
sum of squares, fn values
, fn, 1 deriv.
, fn, 1 2 deriv.
sum sqs., comprehensive, fn vals.
, fn, 1 deriv.
, fn, 1 2 deriv.

1st deriv estimation
check 1st deriv
check 2nd deriv
check Jacobian
check Hessian
check 1st deriv fn constraints
check 2nd deriv fn constraints

F01 MATRIX OPERATIONS, INCLUDING INVERSION

invert real matrix, approximate
, accurate
invert real, sym, pos def, approx
, accurate
, simplified
, sym def band, approx
, accurate
, pos def
pseudo inverse and rank
singular value decomposition
, bidiagonal
QR decomposition, real, rank = n
, rank = n
LU decomposition, real
LU decomposition, real, banded
LU decomposition, real, sparse
LL' decomps., real, sym, pos. def.
, band
, complex, herm., pos. def.
ULD L'U' decomps., real, sym, def, band
LDL' of A E, A symm, E diag.
QU of m by n matrix
UQ of m by n matrix
reduction of real, sym Ax=kBx. B def
ABx=xx. B def
real, band, sym Ax=kBx B def
real, general Ax=kBx
complex, general Ax=kBx
balance complex matrix

- 37 -
balance real matrix
reduction, complex- u. Hessenberg
complex Herm- real tridiag
real upp. Hessenberg
accumulate F01AKF products
reduction real sym tridiag
, accumulating product
, packed storage
, real sym band tridiag
(alt. storage)
, upper tri. bidiagonal

backtransformation of eigenvectors
-complex, after balancing
-real, after Hessenberg reduction
-real, after balancing
-real, after Hessenberg reductions
-real sym., after reduction
-real sym., after reduction, packed
-Az=kBx or ABx=kx
-BAx=kx
Householder trans., real
, real, zero 1 el.
, real, zero 2 els.
, complex
construct Givens rotation
apply Givens rotation
construct modified Givens rotation
apply modified Givens rotation

F02 EIGENVALUES AND EIGENVECTORS
blackbox, complex, all e-vals
, all e-vals -vecs
, selected e-vals -vecs
, complex Herm, all e-vals
, all e-vals -vecs
, complex generalised Ax=kBx
A,B band, 1 e-vec
, real, all e-vals
, all e-vals -vecs
, selected e-vals -vecs
, real symm., all e-vals
, all e-vals -vecs
, selected e-vals -vecs
, band, e-vals e-vecs
, generalised Ax=kBx
, symmetric Ax=kBx, e-vals
, e-vals -vecs
complex Hessenberg, all e-vals
, all e-vals -vecs
, selected e-vecs
reduced complex, all e-vals -vecs
reduced complex Herm, e-vals -vecs
| Reduced Real, All E-vals - Vecs | F02AQF | - |
| Real General, Complex | - | EQLVC |
| Real Hessenberg, All E-vals | F02APF | EQRH1F |
| Reduced Real Symm, All E-vals - Vecs | F02AMF | EHOBKS |
| Reduced Real Symm. Band, Selected E-vals | F02AVF | EQRT2S |
| Reduced Real Symm. Tri-Dia, All E-vals | F02BFF | EQRT1S |
| Reduced Real Symm. Tri-Dia, Selected E-vals | F02BEP | EQRT3S |
| SVD, Real Upper Bidiagonal | F02WAF | LSVDB |
| SVD, Real Upper Bidiagonal, Sing. Values Rt-Vecs. M by N(M=n) | F02WBF | - |
| SVD, Real Upper Bidiagonal, Sing. Values Vectors M by N | F02WCF | - |
| QU-Fact. Part of SVD | F02WDF | - |

### F03 Determinants

| Black Box, Complex | F03ADF | - |
| Real Symm. Pos. Def. | F03AAF | - |
| Real Symm. Pos. Def. Band | F03ABF | - |
| LU Det, Complex | F03AHF | - |
| LU Det, Real, Symm. Pos. Def. | F03AFF | - |
| Real Symm. Pos. Def. Band | F03AGF | LUDAPB |
| Complex Hermitian Pos. Def. | F03ALF | - |

### F04 Simultaneous Linear Equations

| Black Box, Complex, Approx. | F04ADF | LEQT1C |
| Accurate | F04ATF | LEQ2C |
| Real, Accurate, 1 Rhs | F04AEF | LEQT2F |
| Approx., 1 Rhs | F04ARF | LEQT1F |
| Real Sym. Def, Acc, 1 Rhs | F04ASF | LEQT2P |
| Approx. | F04ABF | LEQT1P |
| Band, Approx. | F04LDF | LEQT1B |
| Accurate | F04ACF | LEQ1PB |
| Real Sym. Def Band, Approx. (Variable Band) | F04MCF | LEQ2B |
| Acc. | F04AKF | LEQ2PB |
| Real Sym. Indef., Approx. | F04AFF | LEQ1S |
| Accurate | F04LDF | LEQ2S |
| Soln Inverse, Real | F04ZFF | LINV3F |
| Real Sym. Def. | F04AKF | LINV3P |

- 39 -
, complex Herm., approx
, real, accurate
, approx.
, real band
, real sparse, approx.
, real sym. def, accurate
, approx.
, real sym. def, band, approx.
, accurate
least squares, rank n, accurate
, approx.
, rank n
, m by n
m = n
, automatic treatment of rank def.
L1, rank n

F05 ORTHOGONALIZATION

Schmidt orthogonalization
2-norm of vector

G01 SIMPLE STATISTICAL CALCULATIONS

produce a letter-value summary
descriptive, 1 variable, from data
, from freq. table
, 2 vars, from data
frequency table from raw data
1-way analysis of variance
2-way conting. tab. reduction signif.

formation
median polish of two-way table
compute exact probs. for conting table
transgeneration matrix cols., in core
, out of core
var, co-var of linear fn., in core
, out of core
plot of 2 vars (scatter plot)
plot vector against normal scores
print a box plot
print stem and leaf display
minimum and maximum in vector
calculation of normal scores
general cts. prob. dist. fn.
ratio ordinate to normal upper tail
distribution fn., Students t

- 40 -
G02 CORRELATION AND REGRESSION ANALYSIS

Pearson product-moment corr coeffs
-all vars, no missing values
, casewise deletion
, pairwise deletion
-subset, no missing values
, casewise deletion
, pairwise deletion
"correlation-like" coeffs
-all vars, no missing values
, casewise deletion
, pairwise deletion
-subset, no missing values
, casewise deletion
, pairwise deletion
Kendall Spearman coeffs
-no missing vals, overwriting input
, preserving input
-casewise treatment, overwriting
, preserving
- pairwise treatment
linear regression, constant term,
, no missing values
, missing vals
linear regression, no constant term,
, no missing vals
, missing vals
mult.lin.reg, const term, from corrns.
select els. from vectors and matrices
re-order vectors and matrix elements
means, st devs, corrrns (out of score)

tetrachoric correlations
means, st devs, simple l.r. coeffs, st err
(missing values, in core)
(out of core)

means, st devs, 3rd 4th moments
(biserial/point biserial corrrns
biserial correlations
bivariate normal corrn. est.
from cont. table (ml method)
generate orthog. central comp. design
decode quadratic reg model
var est for decoded orth poly coeffs
coded
coeff decoder for orth. poly. model
leaps and bounds - best subsets reg.
replication err d.f. s.s. (in core)
(out of core)

univ. Curvilinear fit, orth poly
(easy use)

mean correction corrected s.s.s, in core
(out of core)
response control, simple lin reg model
inverse prediction
generate orth polys
certainty int. for responses, in core
(out of core)
residual anal for lin reg model
forward stepwise regression

fit y=a b*(c**x) by least squares
log-linear fit of conting. table
inverse pred., fitted lin. reg. model

G04 ANALYSIS OF VARIANCE

latin square design
one-way classification
two-way crossed classification
two-way hierarchical classification
balanced incomplete block/lattice
contrast estimate and sums of sqs.
full factorial plan

(balanced complete design (b.c.d.)
general linear model
interval est. variance component

- 42 -
expected ms. for b.c.d.
expected data by unweighted means
covariance anal. for 1-way classn.
completely nested design, equal subcl.
, unequal subcl.
reordering data from a b.c.d.
Student-Newman-Keuls test

G05 RANDOM NUMBER GENERATORS

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Code</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>uniform over (0,1)</td>
<td>G05CAF</td>
<td>GGUBFS</td>
</tr>
<tr>
<td>uniform over (a,b)</td>
<td>G05DAF</td>
<td></td>
</tr>
<tr>
<td>exponential</td>
<td>G05DBF</td>
<td>GGEXN</td>
</tr>
<tr>
<td>logistic</td>
<td>G05DFE</td>
<td></td>
</tr>
<tr>
<td>normal</td>
<td>G05DDF</td>
<td>GGNML</td>
</tr>
<tr>
<td>lognormal</td>
<td>G05DEF</td>
<td>GGNLG</td>
</tr>
<tr>
<td>Cauchy</td>
<td>G05EFG</td>
<td>GGCAY</td>
</tr>
<tr>
<td>gamma</td>
<td>G05EFL</td>
<td>GGAMR</td>
</tr>
<tr>
<td>chi-square</td>
<td>G05DFH</td>
<td>GGCHS</td>
</tr>
<tr>
<td>Student's t</td>
<td>G05DFF</td>
<td>GGAMR</td>
</tr>
<tr>
<td>Snedecor's f</td>
<td>G05DKF</td>
<td>GGAMR</td>
</tr>
<tr>
<td>beta, 1st kind</td>
<td>G05DGF</td>
<td>GGBTR</td>
</tr>
<tr>
<td>, 2nd kind</td>
<td>G05DMF</td>
<td></td>
</tr>
<tr>
<td>uniform integer</td>
<td>G05DYF</td>
<td>GGDUB</td>
</tr>
<tr>
<td>pseudo-random logical</td>
<td>G05DZL</td>
<td></td>
</tr>
<tr>
<td>Weibull generator</td>
<td>G05DPF</td>
<td>GGWIB</td>
</tr>
<tr>
<td>unif devs. from sphere surf 3,4-space</td>
<td>-</td>
<td>GGSPH</td>
</tr>
<tr>
<td>vector of uniform (0,1) devs.</td>
<td>-</td>
<td>GGUBS</td>
</tr>
<tr>
<td>uniform (0,1) with shuffling</td>
<td>-</td>
<td>GGUW</td>
</tr>
<tr>
<td>geometric deviate</td>
<td>-</td>
<td>GGEXN</td>
</tr>
<tr>
<td>Poisson gen.- frequent param changes</td>
<td>-</td>
<td>GGPON</td>
</tr>
<tr>
<td>vector of normal deviates</td>
<td>-</td>
<td>GGNML</td>
</tr>
<tr>
<td>triangular distn generator</td>
<td>-</td>
<td>GGTRA</td>
</tr>
<tr>
<td>general continuous distn.</td>
<td>-</td>
<td>GGVCR</td>
</tr>
<tr>
<td>multinomial deviate generator</td>
<td>-</td>
<td>GGMTN</td>
</tr>
<tr>
<td>integer from reference vector</td>
<td>G05EYF</td>
<td></td>
</tr>
<tr>
<td>set up reference vector, uniform</td>
<td>G05EBF</td>
<td></td>
</tr>
<tr>
<td>, Poisson</td>
<td>G05ECF</td>
<td>GGPOS</td>
</tr>
<tr>
<td>, binomial</td>
<td>G05EDF</td>
<td>GGBN</td>
</tr>
<tr>
<td>, negative binomial</td>
<td>G05EEF</td>
<td>GGBNR</td>
</tr>
<tr>
<td>, hypergeometric</td>
<td>G05EFF</td>
<td>GGHPR</td>
</tr>
<tr>
<td>reference vector from pdf or cdf</td>
<td>G05EXF</td>
<td></td>
</tr>
<tr>
<td>m.v. normal gen. using ref. vec.</td>
<td>G05EGF</td>
<td></td>
</tr>
<tr>
<td>time series ref. vect. init.</td>
<td>G05EWF</td>
<td>FTGEN</td>
</tr>
<tr>
<td>time series gen. using ref. vect.</td>
<td>G05CFF</td>
<td></td>
</tr>
<tr>
<td>initialise generator, repeatable</td>
<td>G05CBF</td>
<td></td>
</tr>
<tr>
<td>, non-repeatable</td>
<td>G05CCF</td>
<td></td>
</tr>
<tr>
<td>save state of generator</td>
<td>G05CGF</td>
<td></td>
</tr>
<tr>
<td>restore state of generator</td>
<td>-</td>
<td>GTDDU</td>
</tr>
<tr>
<td>D-squared tally</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
D-squared test
moments of uniform random numbers
test for normality
prob. dist. into 2 equi-prob. states
poker test tally
poker test
tally of co-ords. of pairs
pairs (Goods serial) test
runs test
tally of no of runs and down
tally for triplets test
triplets test
set of binomial random deviates
set of discrete random devs., alias,
table lookup
set of deviates, mixture 2 exponent'ls
set of stable dist. random deviates
non. hom. Poisson process gen.
random par of 1,...,k
simple random sample from finite pop.

G08 NONPARAMETRIC STATISTICS

sign test - G08AAF NBSIGN
Wilcoxon test - G08ABF NRYWMD
median test - G08ACF -
Mann-Whitney test - G08ADF NRYWRST
Friedman test - G08AEF -
Kruskal-Wallis test - G08AFF NAK1
Kolmogorov-Smirnov 1-sample test - G08CAF NKS1
Kendall coeff of concordance - G08DAF NMCC
Mood's and David's tests - G08BAF -
Wilson's anova, 2,3 way, no reps
, 1,2,3 way, equ.reps.
, uneq.reps. - NAWNRP - NAWRPE - NAWRPU
Noether's test for cyclical trend - NBCYC
Cochran q test - NBQT
Cox Stuart sign test for trends - NBSDL
nonparametric pdf estimation - NDMPLE
inclusion test - NHINC
Kolmogorov-Smirnov 2-sample test - NKS2
Kendall's test for correlation - NMKEN
significance of Kendall correlation - NMKSF
k sample trends test - NMKTS
Bhapkar v test - NRHBHA
ranking-a vector - NMRANK
computing tie statistics - NMTIE
evaluate p.d.f. - NDST
nonparametric p.d.f. estimation - NDKER

- 44 -
G09 PARAMETER ESTIMATION

- interval est. of p (binomial)
- lambda (poisson)
- mean inf., normal dist., known var.
- mean and var. inferences, normal
- var inf., normal sample, known mean
- mean and var inf., 2 norm., unequal var
- mean and var inf., known var
- mean and var inf., 2 norm., equal var
- ml est norm. params from censored data

G12 HYPOTHESIS TESTING

- Chi-squared goodness of fit
- sample size/no class interval, chi-sq

G13 TIME SERIES ANALYSIS

- Box-Jenkins univariate modelling
- mean, var, autocov, autocorr, par. corr.
- AR params preliminary estimation
- MA params preliminary estimation
- transforms, diff, seasonal diff.
- AR MA parameter estimation
- model analysis
- forecasting
- transfer functions
- cross correlation of 2 series
- preml est fir transfer fn model
- means, var, cross-cov - cor: 2 n-ch ser.
- FFT power and cross spectra
- single/multi chan tsa, time freq dom
- Kalman filtering
- Wiener forecasting
- multichannel Wiener forecast
- ML par est mult chan 1 o/p ts model

MULTIVARIATE TECHNIQUES

- cluster analysis
- discriminant anal, linear a la Fisher
- factor/pca, score coeffs
- unrot. factor loading
- factor rot., oblique axes
- unrot. fact. load (image)
- oblique trans fact loading
- communalities norm. fact. res. cor mx
- orthog fact rot. (qu-, var-, equ-max)
- target matrix
- pairwise dist. between cols of matrix
- fact scores form fact coeffs
principal component calculation
Wilks test for m.v. norm indep.

SAMPLING

simple random sampling, prop. data
, cont. data
, cont. data, ratio/reg
strat. random sampling, prop. data
, cont. data
, cont. data, ratio/reg
1-stage clust. sampling, cont. data
2-stage sampling, cont. data

H OPERATIONS RESEARCH

lin. prog., simplex, 1 iteration
, revised simplex
, contracted simplex
, find pt. given lin. constraints
quadratic programming
integer linear programming
transportation problem

M01 SORTING

vector, real, ascending
, descending
, absolute values
, integers, ascending
, integers, descending
, characters, alphanumeric
, reverse alphanumeric
vector index, real, ascending
, descending
, absolute values
, integer, ascending
, descending
index to sorted, real, ascending
, descending
, integers, ascending
, descending
matrix rows, real, ascending
, descending
, integers, ascending
, descending
matrix columns, character, alphanumeric
, reverse

- 46 -
### S APPROXIMATIONS OF SPECIAL FUNCTIONS

<table>
<thead>
<tr>
<th>Function</th>
<th>Routine(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tan</td>
<td>S07AAF</td>
</tr>
<tr>
<td>arcsin</td>
<td>S09AAF</td>
</tr>
<tr>
<td>arccos</td>
<td>S09ABF</td>
</tr>
<tr>
<td>tanh</td>
<td>S10AAF</td>
</tr>
<tr>
<td>sinh</td>
<td>S10ABF</td>
</tr>
<tr>
<td>cosh</td>
<td>S10ACF</td>
</tr>
<tr>
<td>arctanh</td>
<td>S11AAF</td>
</tr>
<tr>
<td>arcsinn</td>
<td>S11ABF</td>
</tr>
<tr>
<td>arccosh</td>
<td>S11ACF</td>
</tr>
<tr>
<td>Exponential integer</td>
<td>S13AAF MMDSI</td>
</tr>
<tr>
<td>Sine integral</td>
<td>S13ADF</td>
</tr>
<tr>
<td>Cosine integral</td>
<td>S13ACF</td>
</tr>
<tr>
<td>Gamma</td>
<td>S14AAF MGAMA</td>
</tr>
<tr>
<td>Log gamma</td>
<td>S14ABF MLGAMA</td>
</tr>
<tr>
<td>Logarithmic deriv of gamma fn.</td>
<td>MMPSI</td>
</tr>
<tr>
<td>Cumulative normal distribution</td>
<td>S15ABF MDNOR</td>
</tr>
<tr>
<td>Complement of cumulative normal dist</td>
<td>S15ACF -</td>
</tr>
<tr>
<td>Error function</td>
<td>S15AEF MERF</td>
</tr>
<tr>
<td>Complement of error fn</td>
<td>S15ADF MERRC</td>
</tr>
<tr>
<td>Inverse complement error fn.</td>
<td>- MERFC1</td>
</tr>
<tr>
<td>Inverse error function</td>
<td>- MERFI</td>
</tr>
<tr>
<td>Dawson's integral</td>
<td>S15AFF MMDAS</td>
</tr>
<tr>
<td>Bessel function j0</td>
<td>S17AEF MMBSJO</td>
</tr>
<tr>
<td>j1</td>
<td>S17AFF MMBSJ1</td>
</tr>
<tr>
<td>y0</td>
<td>S17ACF -</td>
</tr>
<tr>
<td>y1</td>
<td>S17ADF -</td>
</tr>
<tr>
<td>y, general order</td>
<td>- MMBSYN</td>
</tr>
<tr>
<td>Airy function ai</td>
<td>S17AGF -</td>
</tr>
<tr>
<td>bi</td>
<td>S17AHF -</td>
</tr>
<tr>
<td>Deriv. of Airy function ai</td>
<td>S17AJF -</td>
</tr>
<tr>
<td>bi</td>
<td>S17AKF -</td>
</tr>
<tr>
<td>Modified Bessel function i0</td>
<td>S18AEF MMBSIO</td>
</tr>
<tr>
<td>i1</td>
<td>S18AFF MMBSI1</td>
</tr>
<tr>
<td>k0</td>
<td>S18ACF MMBSKO</td>
</tr>
<tr>
<td>k1</td>
<td>S18ADF MMBSK1</td>
</tr>
<tr>
<td>Fresnel integrals s(x)</td>
<td>S20ACF -</td>
</tr>
<tr>
<td>c(x)</td>
<td>S20ADF -</td>
</tr>
<tr>
<td>Complete elliptic integral, 1st kind</td>
<td>S21BBF MMDELK</td>
</tr>
<tr>
<td>2nd kind</td>
<td>S21BCF MMDELE</td>
</tr>
<tr>
<td>3rd kind</td>
<td>S21BDF -</td>
</tr>
<tr>
<td>Kelvin fns., order zero</td>
<td>- MMKELO</td>
</tr>
<tr>
<td>, order one</td>
<td>- MMKE0L</td>
</tr>
<tr>
<td>, derivs.</td>
<td>- MMKELD</td>
</tr>
<tr>
<td>Decompose integer into prime factors</td>
<td>- VDCPS-</td>
</tr>
</tbody>
</table>
X01 MATHEMATICAL CONSTANTS

pi
Eulers const

X02 MACHINE CONSTANTS

smallreal
smallest positive real
maxreal
X02ABF/X02AAF
largest neg. argument for exp
smallest x; x, x, 1.0/x, -1.0/x repres.
floating point base
maxint
max n for 2**n representable
min n for 2**n representable
max decimal digits
active set size in paged environment

X03 VECTOR/MATRIX ARITHMETIC

null vector
null matrix
unit matrix
copy vector, real
    , complex
copy vector - matrix row
copy matrix
    , partial
interchange vectors, real
    , complex
interchange matrix row/column
const*vector, real
    , complex
    , real*complex
const*vector vector, real
    , complex
find. el. of max. magnitude, real
    , complex
max. absolute value, vector
    , matrix row/col.
sum of absolute values, real
    , complex
    , real matrix row/col
vector euclidean norm, real
    , complex
matrix 1-norm
matrix euclidean norm
matrix infinity norm
matrix addition

- 48 -
matrix subtraction
partial matrix additions
partial matrix subtraction
matrix transposition
matrix multiplication
Matrix multiplication by transpose
multi vector by symm.matrix (packed)
matrix polynomial
vector convolution
matrix storage mode conversion
scalar product, real

, complex unconjugated
conjugated
scalar prod. const, real, basic prec.
, additional prec.

extended precision add
multiply
store

**X04 INPUT/OUTPUT UTILITIES**

error message unit no.
advisory message unit no.
manipulate I/O unit numbers
set message level
print error message
plot cluster (from OCHIER)
input of matrix
input of vector
print histogram

print results of regression
print pdf information
plot 2 pdf's
print binary tree
"help" information

printer plot of functions
print matrix
print vector
MANUAL REGISTRATION

If you wish to receive updates for this green book please complete the attached form and send it to:

Computing Support Library
Ms. A. Cambon
Building 36
Div. 1 Dept. A
JRC Euratom
21020 Ispra
Italy

I wish to receive future updates for "Using the IMSL & NAG Libraries".

Name ....................................................

Address ..............................................

....................................................

....................................................

Signature ............................................
Please use this sheet to send to us information about errors and omissions you have noticed in this document. Please note any specific errors found giving page and paragraph reference.
If you wish to receive a reply to your comments please give your name and address.
Send your completed comment sheets to:

Computing Support Library
Ms. A. Cambon
Building 36
Div. 1 Dept. A
JRC Euratom
21020 Ispra
Italy

Comment sheet for green book: "Using the IMSL & NAG Libraries"

From: - Name ..............................................
Address ...........................................
......................................................
......................................................
Tele: ....................................................

Comments
......................................................
......................................................
......................................................
......................................................
......................................................
......................................................
......................................................
......................................................
......................................................
......................................................
......................................................
......................................................
......................................................