SANDIA REPORT SAND82-8767 • Unlimited Release
Printed June 1982

MATHDOC and SLATEC, A User's Guide

(To be Presented at *NASIG '82,* June 23-25, 1982, Los Alamos, NM)

T. H. Jefferson

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-76DP00789

SF2900Q(8-81)

Issued by Sandia Laboratories, operated for the United States Department of Energy by Sandia Corporation.

SAND82-8767
Unlimited Release
June 10, 1982

MATHDOC AND SLATEC, A USER'S GUIDE

Thomas H. Jefferson Jr.
Scientific Computing Division
Sandia National Laboratories
Livermore, California 94550

ABSTRACT

MATHDOC is an interactive documentation program for the VAX. MATHDOC was written to provide quick on-line access to documentation and is presently being used with the SLATEC FORTRAN Library. Capabilities of MATHDOC include retrieval of documentation specified explicitly by name, or implicitly by keywords or category. This document describes how to use MATHDOC and also contains the SLATEC categorization list and the SLATEC Version 1.0 table of contents.

TABLE OF CONTENTS

I.	INTRODUCTION	5
II.	DESIGN OF MATHDOC	6
III.	HOW TO USE MATHDOC	7
IV.	MORE ON GETTING HELP	16
V.	THE SLATEC CATEGORIZATION SCHEME	18
VI	THE SLATEC TABLE OF CONTENTS.	27

I. INTRODUCTION

The MATHDOC documentation program was written to provide the SNLL computer user with on-line documentation on various subjects. The immediate use for MATHDOC is with the SLATEC Mathematical Subroutine Library, but MATHDOC is designed to be able to provide documentation on multiple computers and libraries.

The SNLL computer user now has access to three major computers--the Cray, the VAX, and the 6600. The Cray has no interactive capability. The VAX is the newer timesharing computer, so MATHDOC was written to run on the VAX only, although documentation on all the SNLL computers is available through MATHDOC.

The SLATEC Mathematical Subroutine Library is a result of the continuing cooperation of the participating laboratories. These agencies include Sandia National Laboratories, Los Alamos National Laboratory, the Air Force Weapons Laboratory, Lawrence Livermore National Laboratory, the National Magnetic Energy Fusion Energy Computing Center, Oak Ridge National Laboratory, and the National Bureau of Standards. Version 1.0 of the SLATEC library was released in April 1982.

The SLATEC library is a large collection of FORTRAN subprograms, with 492 user-callable routines in Version 1.0. No hardcopy documentation is distributed with SLATEC. Instead, the documentation for SLATEC is entirely computer readable and is more than 45,000 lines long. MATHDOC gives on-line access to that documentation and relieves the computer user of keeping a document that would be more than 900 pages in length.

Via MATHDOC one can ask for SLATEC documentation by explicitly naming a subroutine, or by implicitly specifying all routines that have certain keywords (actually keyphrases) or are in particular categories. The documentation retrieved is that which matches all the selection criteria. An especially useful feature of the documentation selection process is capability of using the wildcard characters "*" and "%" in specifying the selections.

Although the immediate use of MATHDOC is to give on-line access to the SLATEC documentation, long range plans are to include documentation on other libraries and subjects.

II. DESIGN OF MATHDOC

The goals in designing MATHDOC included the following:

1. It should be self-contained. That is, a user should be able to use MATHDOC with no supporting documentation.

2. It should be easy-to-use.

3. It should be interactive.

4. It should be fast.

5. It should be easy to implement.

6. It should be versatile. That is, it should not be limited to one particular library and should be able to provide documentation for all the SNLL computers.

The only quantitative information applies to goal 4. For the SLATEC library, on the SNLL VAX system, a keyword search takes about two seconds. The time for writing out the appropriate documentation depends on the length of the documentation.

The MATHDOC program is written in FORTRAN, but is not portable. It runs under the VMS operating system and makes use of the VAX VMS Librarian utilities both to store and access the documentation and for the built-in documentation program HELP facility. The Librarian provides random access and eases the task of the implementor.

Additional speed in doing keyword and category searches is acquired by maintaining an index for each text library to be accessed via MATHDOC. The index contains one record for each document. That record holds the document name, the keywords, and the categories for that document.

MATHDOC has an extensive built-in help facility that gives the user information on using the program. The character "?" may be entered at any time input is requested and is always interpreted as a request for help. This help facility also uses the VAX Librarian. The help facility was designed to be, as nearly as possible, independent of the documentation that MATHDOC presents. This was done so that MATHDOC could be used to provide documentation on a variety of libraries with no change in the program help facility.

III. HOW TO USE MATHDOC

The documentation accessible via MATHDOC is stored as blocks of text called MODULEs. MODULEs are grouped together into collections identified by COMPUTER and LIBRARY. The documentation selection process consists of specifying the three attributes, COMPUTER, LIBRARY, and MODULE, either explicitly or implicitly, and then retrieving all documentation modules that match the selection criteria.

For example, in the SLATEC library each subroutine has a corresponding documentation module of the same name. AVINT is the name of a particular routine in the SLATEC library for integrating tabulated data. To get the documentation for the SLATEC subroutine AVINT for the Cray computer, you could explicitly specify the module name AVINT. Or you could implicitly select the AVINT documentation by asking for all modules that have the same keywords as AVINT--"tabulated", for example.

The document selection process uses character string matching to retrieve all those documentation modules that match your selection criteria. The following examples illustrate how to use MATHDOC to select the appropriate documentation. In the examples, the entry of the keystroke "return" is indicated by <ret>. In addition, the computer prompts are in UPPER CASE and user responses are in lower case. The examples are shortened for brevity and may not depict exactly what would appear on your terminal.

****

EXAMPLE: Start the MATHDOC program.

Login to the VAX and at the $ prompt, enter mathdoc<ret> . You will then be at the documentation program command level, which is indicated by the "COMMAND =" prompt.

****

The documentation program starts at command level and returns to command level if you enter <control/c> anytime. This is helpful if you have made a mistake in input, or if your terminal is filling with unexpected output.

As you use the program you might need help on the input for a particular command, or you might want other information about the documentation program itself. Use the "?" for help. At any time input is requested, if you enter the single character "?" you will get a display explaining more about that particular input. At any time you can also enter "? topic" to get information about a particular topic. The "?" is always a request for help on the documentation program and never satisfies a prompt.

****

EXAMPLE: At command level, get help on the "COMMAND =" input.

    COMMAND = ?<ret>

A DISPLAY OF THE AVAILABLE COMMANDS APPEARS FOLLOWED BY A REPEAT OF THE PREVIOUS "COMMAND =" PROMPT.

EXAMPLE: Enter the following at any time input is requested to get help on getting help.

    ? help<ret>

INFORMATION ON GETTING HELP ON USING THE DOCUMENTATION PROGRAM APPEARS, FOLLOWED BY A REPEAT OF THE PREVIOUS PROMPT.

****

From documentation program command level, three commands are accepted in response to the "COMMAND =" prompt. They are the following.

1. SELECT initiate the documentation selection process.
2. COMMENT make comments about the documentation program.
3. END end the documentation program.

These commands may be abbreviated, as may most documentation program entries. This is an important point and can save you much typing time.

The END command is self-explanatory. The COMMENT command gives you the opportunity to make comments on the documentation program. Your comments are written to a special file to be read later by the implementor. Your comments are most welcome.

At the "COMMAND =" prompt enter the command COMMENT, or C, for short. You will then be asked to enter your comments. The entry of your comments is terminated by the <control/z> character and you will be returned to documentation program command level.

****

EXAMPLE: Make comments to the MATHDOC implementor.

COMMAND = comment<ret>

ENTER YOUR COMMENTS AND TERMINATE BY A <CONTROL/Z>

this is a comment on the documentation<ret>

program that will be stored and read<ret>

later by the implementor. <control/z>

****

The SELECT command is the one to use to start the documentation selection process. Because SELECT is the major program command, the following example shows how to get help on SELECT.

****

EXAMPLE: Get help on the SELECT command.

COMMAND = ?select<ret>

A DISPLAY OF INFORMATION ON SELECT APPEARS, FOLLOWED BY A REPEAT OF THE PREVIOUS PROMPT.

****

After entering the documentation program command SELECT, or S for short, you will be shown the available COMPUTER\LIBRARY collections and be prompted for your choices for COMPUTER, LIBRARY, MODULE, KEYWORDS, and CATEGORY. If you make a KEYWORDS entry you will be prompted for another. If you do not want to make a selection entry for MODULE or KEYWORDS or CATEGORY, just enter <ret> when prompted.

Each of the selection entries is a list, delimited by commas. The commas mean "or", as will be illustrated in the next example. The documentation retrieved will be all documentation modules that match at least one component of each selection list entered.

You will always be prompted for COMPUTER, LIBRARY, MODULE, KEYWORDS, and CATEGORY. Your SELECTion entries are not complete until you have responded to each of the five prompts with either a list or just <ret>.

The following example shows how to retrieve a documentation module if you know the name of the module. Each SLATEC subroutine has a corresponding documentation module of the same name.

****

EXAMPLE: Start selection process and get the documentation for the SLATEC routine AVINT for the Cray computer.

```
COMMAND = s<ret>

THE AVAILABLE COMPUTER/LIBRARY CHOICES ARE THE FOLLOWING

CRAY,6600,VAX\SLATEC

COMPUTER  :cray<ret>

LIBRARY   :slatec<ret>

MODULE    :avint<ret>

KEYWORDS  :<ret>

CATEGORY  :<ret>
```

The documentation program will then search for AVINT, tell you that it has found one module, and ask where you want the documentation written.

****

If you want the documentation for more than one routine, list the routines separated by commas. For example, "avint,fzero,isort".

Remember that even if you explicitly specify a module name, you will always be prompted for KEYWORDS and CATEGORY too. Just enter <ret> in response to a prompt if you do not want to make a selection.

Often you do not know the name of the documentation module you are seeking. Rather you want documentation on a routine to solve a particular problem. In that case you can seek documentation by specifying keywords that a module must have. Or you can specify a category, or categories, in which the module must be classified. MATHDOC will then retrieve all documentation modules that match your selection criteria.

The next example shows how this is done. The example also illustrates that each selection entry list further limits the number of documentation modules that will satisfy the selection criteria.

****

EXAMPLE: SELECTion entries to retrieve all documents for computer either VAX or Cray, in the SLATEC library, with module name starting with either the letter "D" or the letter "G", with keywords (actually keyphrases) containing either "linear equations", "solve", or "decompose", and with keyword containing "double".

```
COMPUTER   :cray,vax<ret>

LIBRARY    :slatec<ret>

MODULE     :d*,g*<ret>

KEYWORDS   :linear equations,solve,decompose<ret>

KEYWORDS 2:double<ret>

KEYWORDS 3:<ret>

CATEGORY   :<ret>
```

****

The preceding example illustrates the use of the wildcard character "*" and that each of the selection entries can be a list with components delimited by commas. In the selection matching process the commas are interpreted as "or" and each list entered is qualified by "and". That is, in the previous example, the documentation modules selected must be for the Cray "or" VAX computer, "and" in the SLATEC library, "and" must begin with either D "or" G, "and" must contain keywords linear equations "or" solve "or" decompose, "and" must contain keyword double. Note that if a KEYWORDS entry is made you will be prompted for another.

Use of the wildcard characters * and % in making your selection entries is quite powerful. In fact, after you have made your selection entries, they are displayed, and usually have the wildcard character * inserted automatically in specific places.

*   matches any character string of length 0 or greater.

%   matches any character string of length 1.

The entire selection process is based on matching one character string by another.

****

EXAMPLE: character string matching.

LINEAR matches LINEAR, but not NONLINEAR.

LINEAR matches LINEAR and also NONLINEAR EQUATIONS.

%LINEAR* matches neither LINEAR nor NONLINEAR.

%%%LIN* matches NONLINEAR, but not LINEAR.

****

As mentioned above, if you do not instruct otherwise (otherwise means by preceding the component of your selection list by the underscore character "_") the documentation program will automatically insert the wildcard character * in the following places.

1. After each component of the COMPUTER list.

2. After each component of the LIBRARY list.

3. Both before and after each component of each KEYWORDS list.

4. After each component of the CATEGORY list.

This automatic insertion of the * makes your selections easier. For example, if Cray, VAX, and 6600 are the only available choices for COMPUTER, the entry of "C" will be converted automatically to "C*" and will thus match Cray, but not VAX nor 6600. Therefore, you can shorten your entries and save keystrokes.

****

EXAMPLE: Abbreviated SELECTion entries to retrieve all documents for computer either VAX or Cray, in the SLATEC library, with module name starting with either the letter "S" or the letter "G", with keywords of either "linear equations", "solve", or "decompose", and with keyword "positive definite". Make liberal use of wildcard characters.

COMPUTER :c,v<ret>

LIBRARY :sl<ret>

MODULE :s*,g*<ret>

KEYWORDS :lin*ions,solve,decomp<ret>

KEYWORDS 2:pos*def<ret>

KEYWORDS 3:<ret>

```
CATEGORY  :<ret>
```

****

The insertion of the * after each component of your CATEGORY selection is appropriate because of the structure of the categorization scheme used for SLATEC, in particular. A subcategory begins with the same letters as its parent category, but has a longer name. See the SLATEC category list contained in this document.

If you do not want the automatic wildcard insertion to take place for any component in a selection list, prefix that component with the underscore ("_") character. This is not usually needed nor desirable, but could be helpful occasionally to eliminate extraneous documentation.

****

EXAMPLE: A KEYWORDS selection list that will match LINEAR EQUATIONS, but will not match NONLINEAR EQUATIONS.

```
KEYWORDS   :_linear*<ret>
```

EXAMPLE: A KEYWORDS selection list that will not match the sub-phrase NONLINEAR in any keyword, but will probably match all occurrences of the word LINEAR.

```
KEYWORDS   :_linear*,_* linear*,_*(linear*<ret>
```

****

In making your SELECTion entries, take care not to be too restrictive. Each SELECTion list (MODULE, KEYWORDS, CATEGORY) that you enter will further limit the number of modules meeting your selection criteria. Adding more components to any selection entry can only expand the number of modules meeting your selection criteria. If a module might have the keyword "DECOMPOSE" try the keyword entry of "decomp", which, because of the addition of the implicit wildcards, will match both DECOMPOSE and DECOMPOSITION.

If you are not sure how to spell a keyword, or if you are not sure how the implementor spelled a keyword, try a list of variants. For example, instead of "non-linear", try "nonlinear,non linear,non-linear" or "non*linear".

Once the documentation modules that match all your selection criteria have been located you will be prompted for your output instructions.

You can direct the documentation to your terminal or to another VAX file for later perusal or printing. In addition you can ask that each module written be preceded by a form feed (convenient when the documentation is to be printed, so that each module starts on a new page), or be preceded by an identifying header (the header identifies the documentation module by name and date). The number of lines of each module written can be all the lines, or some fixed number of lines. Limiting the number of lines written is handy when browsing and when the items of interest are located at the start of the documentation modules.

The output options are specified by adding "/option" switches at the end of your input line.

Options

/[NO]Formfeed	[do not] precede each module output by a form feed
/[NO]Header	[do not] precede each module by an identifying header
/Lines=nnn	write out only nnn lines of each module
/Lines	write out all lines of each module

The current output file and options are displayed when you are prompted for your output instructions. The options and the output file remain as is until you explicitly change them.

Abbreviations for the options are allowed, as in the following examples.

****

EXAMPLE: use current output file and current output options.

    YOUR ENTRY?_<ret>

EXAMPLE: Send documentation output to file doc.dat using current options.

    YOUR ENTRY?_doc.dat<ret>

EXAMPLE: Send output to file doc.dat with a form feed and no header.

    YOUR ENTRY?_doc.dat/f/noh<ret>

EXAMPLE: Use current output file and have just the first 10 lines of each documentation module output.

```
YOUR ENTRY?_/lin=10<ret>
```

EXAMPLE: Send output to terminal with no form feed, no header and the first 15 lines only.

```
YOUR ENTRY?_tt:/noform/nohead/line=15<ret>
```

EXAMPLE: Use current output file and current options, except write out all lines of every module.

```
YOUR ENTRY?_/lines<ret>
```

****

Note, in the preceding examples, that TT: is the entry to use to direct output to your terminal.

Recall that you can enter <control/c> to terminate output prematurely and return to the program command level. In any case, when the documentation output is completed you will be returned to command level where you can again SELECT, COMMENT, or END.

IV. MORE ON GETTING HELP

The documentation program has a built-in help facility that provides information on using the documentation program. At any time input is requested, entering the character "?" is treated as a request for help on the documentation program and will not satisfy the current request for input.

The help facility was designed to be independent of the COMPUTER\LIBRARY collections of documentation. This was done so that the help facility would not have to be changed with the addition of a new COMPUTER\LIBRARY collection.

If, when input is requested, just the single character ? is entered, help on the current input is displayed. However, at any time input is requested, help on any topic can be obtained by entering "? topic".

The documentation program help facility is arranged in a tree-like fashion and uses the VAX VMS Librarian utilities. The list of all topics available is shown if "? help" is entered. If help is given on any of the topics displayed, e.g. SELECT, then subtopics are displayed. Help on a subtopic can be obtained by entering "? topic subtopic". If there is help available on the subtopic, then subsubtopics are shown. In general, then the format of the request for help is "? topic subtopic subsubtopic ...".

Abbreviations are accepted.

****

EXAMPLE: Two forms to request help on using wildcards in selecting.

```
? select wildcards

? sel wil
```

****

The help facility allows a limited use of wildcard characters. If you want to have displayed all the help text, or all the help text on a particular topic and its subtopics, then use the help wildcard characters. However, you will have to control the text as it fills your screen.

*	may appear by itself and matches everything
*...	returns all help text in the library.

topic... returns all help text associated with the topic and its subtopics.

****

EXAMPLE: Entry to list all documentation program help text.

? *...

ALL DOCUMENTATION PROGRAM HELP TEXT WILL STREAM ACROSS YOUR TERMINAL. YOU MUST CONTROL THE SCROLLING. THE HELP TEXT CAN NOT BE INTERRUPTED BY ENTERING <CONTROL/C>.

EXAMPLE: Entry to list all help text on the topic SELECT and its subtopics.

?select...

ALL DOCUMENTATION HELP TEXT ON "SELECT" AND ITS SUBTOPICS WILL STREAM ACROSS YOUR TERMINAL. YOU MUST CONTROL THE SCROLLING. THE HELP TEXT CAN NOT BE INTERRUPTED BY <CONTROL/C>.

EXAMPLE: Entry to list all the first level help text.

? *

****

V. THE SLATEC CATEGORIZATION SCHEME

When you SELECT documentation, MATHDOC will prompt for an entry for CATEGORY. The appropriate categorization scheme could apply to each COMPUTER\LIBRARY collection separately. Because MATHDOC now provides documentation only for SLATEC, there is only one categorization scheme. That scheme is reproduced in this section.

If you are using MATHDOC with SLATEC and have not memorized the SLATEC categorization list or do not have this document handy, you can retrieve a copy of the category list for yourself. It has been stored with the SLATEC library documentation as the documentation module named CATEGORIES.

****

EXAMPLE: Retrieve the SLATEC Categorization list.

COMMAND = s<ret>

THE AVAILABLE COMPUTER/LIBRARY CHOICES ARE THE FOLLOWING

CRAY,6600,VAX\SLATEC

COMPUTER :*<ret>

LIBRARY :slatec<ret>

MODULE :categories<ret>

KEYWORDS :<ret>

CATEGORY :<ret>

The documentation program will then search for CATEGORIES, tell you that it has found one module, and ask where you want the documentation written.

****

When you are prompted for your CATEGORY selection input, enter only a <ret> if you have no category selection to make.

Otherwise, enter a list (delimited by commas) of category selections. Any documentation retrieved will be in one of the categories of the list.

****

EXAMPLE: Retrieve all documentation modules for the VAX computer, in the SLATEC library, in category C4 (Zeros/Solutions/Roots of a Single Nonlinear Equation).

```
COMPUTER  :vax<ret>

LIBRARY   :slatec<ret>

MODULE    :<ret>

KEYWORDS  :<ret>

CATEGORY  :c4<ret>
```

****

The following categorization scheme is used to categorize the user-callable routines in the SLATEC library. Note, however, that the scheme is more general than SLATEC, so many categories may have no corresponding SLATEC routines.

SLATEC Categorization System

Major Category listing

A. Arithmetic, Elementary Operations on Polynomials
B. Evaluation of Elementary and Special Functions
C. Roots/Zeros of Functions, Simultaneous Nonlinear Equations
D. Operations Involving Derivatives and Integrals
E. Interpolation and Approximation
F. Operations on Matrices and Vectors (Simultaneous Linear Equations, Inversion, Canonical Forms)
G. Statistical Analysis and Probability
H. Operations Research Techniques, Simulation and Management Science (see also Optimization, I)
I. Optimization: Minimizing or Maximizing a Function--with or without Constraints
J. Input/Output
K. Internal File Manipulations
L. Language Processors
M. Data Handling
N. Debugging
O. Simulation of Computers and Data Processors, Emulators
Q. Service Routines, Programming Aids
R. Logical and Symbolic

S. Information Retrieval
T. Applications and Application-Oriented Programs
Z. All Others

Complete Category listing

A. Arithmetic, Elementary Operations on Polynomials
A1. Real (i.e., Floating Point) Arithmetic (Includes Interval Arithmetic)
A2. Complex Arithmetic
A3. Decimal Arithmetic
A4. Integer Arithmetic (e.g., Multiple Precision)
A5. Number Theory: Prime Numbers, Greatest Common Divisor, Partitions, etc.
A6. Algebraic Operations on Polynomials and Power Series. This excludes non-algebraic operations, such as the sum of an infinite series, but includes Horner's scheme, add two polynomials, divide two polynomials, exponentiate power series, etc.

B. Evaluation of Elementary and Special Functions
B1. Trigonometric and Inverse Trigonometric Functions
B2. Hyperbolic and Inverse Hyperbolic Functions
B3. Exponential and Logarithm
B4. Roots and Powers (e.g., Square Root, A**X, where A and X are real)
B5. Higher Transcendental Functions (Special Functions)
B5C. Factorials, Binomial Coefficients
B5E. Exponential and Logarithmic Integrals, Sine and Cosine Integrals, etc.
B5F. Factorial or Gamma Function, Psi Function, Polygamma Functions, Beta Function, Incomplete Gamma and Beta Functions
B5G. Error Integral, Higher Integrals, Derivatives, Normal Distribution Function, Hermite Functions, Moments
B5H. Legendre Functions
B5I. Bessel Functions of Real Argument
B5J. Bessel Functions of Pure Imaginary Argument, or Modified Bessel Functions
B5K. Bessel Functions of Complex Argument. Kelvin Functions
B5L. Miscellaneous Bessel and Related Functions, Weber's Function
B5M. Elliptic Integrals,Elliptic Functions, Theta Functions
B5N. Hypergeometric and Confluent Hypergeometric Functions
B5O. Mathieu Functions
B5P. Coulomb Wave Functions
B5Q. Spheroidal Wave Functions
B5R. Orthogonal Functions,Including Polynomials
B5Z. Miscellaneous Higher Mathematical Functions

C. Roots/Zeros of Functions, Simultaneous Nonlinear Equations
C2. Roots/Zeros of Polynomials
C4. Zeros/Solutions/Roots of a Single Nonlinear Equation
C4A. Derivatives Not Required
C4B. Derivatives Required
C5. Zeros/Solutions/Roots of more than One Nonlinear Equation
C5A. Derivatives Not Required
C5B. Derivatives Required

D. Operations Involving Derivatives and Integrals
D1. Quadrature (Numerical Integration)
D1A. Background Calculation: Calculate Gaussian Weights and Abscissas for a Given Weight Function
D1B. Finite Interval, Unit Weight Function
D1B1. Ordinary--User Specifies How Many Subdivisions
D1B1A. Derivatives Not Required
D1B1B. Derivatives Required
D1B2. Automatic--User Specifies Error Tolerance, Routine Automatically Subdivides
D1B2A. Derivatives Not Required
D1B2B. Derivatives Required
D1C. Finite Interval, Built-in Weight Function
D1C1. Ordinary
D1C2. Automatic
D1D. Semi-Infinite Interval (a to infinity), (with Weight Function)
D1D1. Ordinary
D1D2. Automatic
D1E. Infinite Interval (with Weight Function)
D1E1. Ordinary
D1E2. Automatic
D1I. Multiple Integration
D2. Ordinary Differential Equations
D2A. Initial Value Problems
D2A1. Runge-Kutta Methods
D2A2. Predictor-Corrector Methods (e.g., Adams' Methods)
D2A3. Extrapolation Methods (e.g., Bulirsch-Stoer)
D2A9. Others
D2B. Two-Point Boundary Value Problems
D2B1. Shooting Methods
D2B2. Finite Difference Methods (Non-Shooting)
D3. Partial Differential Equations
D3A. Initial Boundary Value Problems (Hyperbolic, Parabolic)
D3D. Boundary Value Problems (Elliptic)
D3D1. Iterative Methods
D3D2. Direct Methods
D3E. Eigenvalue Problems
D4. Numerical Differentiation
D5. Integral Equations
D6. Integral Transforms and Their Discrete Analogues (Laplace Transforms, Fourier Transforms, etc.)

E. Interpolation and Approximation
E1. Table Look-Up and Interpolation
E1A. Splines/Piecewise Polynomial Interpolation
E1B. Polynomial Interpolation
E1C. Rational Interpolation
E2. Curve and Surface Fitting/Approximation
E2A. Spline Fits (All Norms)
E2B. General L sub p Approximation
E2C. L sub infinity (Minimax) Approximations
E2C1. "True" Minimax Approximation (Remez Schemes, etc.)
E2C2. Near-Minimax Approximations (Chebyshev Polynomial Expansions, Generalized Pade, etc.)
E2D. Other Analytic Approximations (Pade, QD, etc.)
E2G. Least Squares Approximation (see also Regression in the Statistics Section)
E2G1. Unconstrained Least Squares
E2G1A. Linear Least Squares Approximation
E2G1B. Nonlinear Least Squares Approximation
E2G1B1. Derivatives Not Required
E2G1B2. Derivatives Required
E2G2. Constrained Least Squares
E2G2A. Linear Function with Linear Equality or Inequality Constraints, Quadratic Programming
E2G2B. Nonlinear Function with Linear Equality or Inequality Constraints
E2G2B1. Derivatives (of the Function) Not Required
E2G2B2. Derivatives Required
E2I. L sub 1 Approximation
E3. Smoothing
E5. Summation of Series, Convergence Acceleration (This does not include the Fast Fourier Transform but does include Aitken's del squared Algorithm, Wynn's & Algorithm, the Euler-Maclaurin Summation Formula, etc.)

F. Operations on Matrices and Vectors (Simultaneous Linear Equations, Inversion, Canonical Forms)
F1. Elementary Vector/Matrix Operations (e.g., Addition, Multiplication, Transpose, Dot Product). Excluded are Inverse, Triangular Factors, Singular Value Decomposition, etc.
F1A. Elementary Vector Operations
F1B. Elementary Matrix Operations
F2. Eigenvalues and Eigenvectors of Matrices
F2A. Reductions to Canonical Form (Givens, Householder)
F2A1. Reduction to Symmetric Tridiagonal Form
F2A2. Reduction to Hessenberg (Almost Triangular) Form
F2C. Ordinary Eigenvalue Problems (Ax = Lambda x)
F2C1. For Tridiagonal Matrices
F2C2. For Real Symmetric Matrices
F2C3. For Real Unsymmetric Matrices

F2C4. For Hessenberg Matrices
F2C5. For Complex Hermitian Matrices
F2C6. For Complex Non-Hermitian Matrices
F2C8. For Banded Matrices
F2C9. For Sparse Matrices
F2D. Generalized Eigenvalue Problems (Ax = Lambda Bx, etc.)
F3. Determinants
F4. Simultaneous Linear Equations, Inversion, LU Decomposition, Crout, Cholesky Decomposition
F4A. For Arbitrary Real (Dense) Matrices
F4B. For Positive Definite Symmetric Matrices
F4C. For Symmetric Indefinite Matrices
F4F. For Complex Matrices
F4H. For Band Matrices
F4K. For Sparse Matrices
F6. Other Matrix Factorizations (Singular Value Decomposition)
F7. Orthogonalization (Modified Gram-Schmidt)

G. Statistical Analysis and Probability
G1. Data Reduction
G1A. Frequency Distributions (Histograms)
G1B. Descriptive Statistics
G2. Multivariate Analysis Including Correlation and Regression Analysis
G2A. Partial Correlation
G2B. Bivariate Correlation Analysis
G2C. Stepwise Regression
G2D. Multiple Regression
G2E. Nonlinear Regression
G2F. Discriminant Analysis
G2G. Factor Analysis
G2H. Canonical Correlation
G3. Sequential Analysis
G4. Analysis of Variance and Covariance
G4A. One-Way Analysis of Variance
G4B. Two- to N-way Factorial Analysis of Variance
G4C. Non-Factorial Designs
G4D. Analysis of Covariance
G4E. Multiple Comparisons
G5. Random Number Generators & Probability Distributions
G5B. Normal Distribution
G5C. Other Distributions (t-, f-, Beta-, Chi-square-, Binomial, Exponential, Poisson, Gamma, etc.)
G5E. Uniform random Number Generators
G5F. Normal Random Number Generators
G5G. Other Random Number Generators (as in G5C above)
G5H. Tests on Random Numbers
G6. Permutations and Combinations
G7. Subset Generators
G8. Non-Parametric Statistics (Goodness of Fit)
G9. Hypothesis Testing

G10. Non-Metric Techniques
G10A. Scalogram Analysis
G10B. Cluster analysis
G10C. Multidimensional Scaling
G11. Table Analysis
G12. Description of Subpopulations
G13. Time Series Analysis, Autocorrelation

H. Operations Research Techniques, Simulation and Management Science (see also Optimization, I)
H3. Transportation and Network Codes
H4. Simulation Modeling and Models (GPSS, Simscript, etc.)
H6. Critical Path Programs (PERT)
H8. Auxiliary Programs

I. Optimization: Minimizing or Maximizing a Function--with or without Constraints
I1. Unconstrained Optimization (for sum of squares, see "Least Squares")
I1A. Minimize or Maximize a Function of One Variable
I1A1. Derivatives Not Required
I1A2. First Derivatives Required
I1A3. Second Derivatives Required
I1B. Minimize or Maximize a Function of Several Variables
I1B1. Derivatives Not Required
I1B2. First Derivatives Required
I1B3. Second Derivatives Required
I2. Constrained Optimization, Mathematical Programming
I2A. Linear Programming: Minimize a Linear Function Subject to Linear Equality or Inequality Constraints
I2A1. Dense, In-Core Matrix of Constraints
I2A2. Structured, Sparse Matrix of Constraints
I2A3. Auxiliary Storage Required
I2B. Nonlinear Programming (for quadratic programming, see Constrained Least Squares)
I2B1. Linear Equality or Inequality Constraints
I2B1A. Derivatives (of the Objective Function) Not Required
I2B1B. First Derivatives Required
I2B1C. Second Derivatives Required
I2B2. Nonlinear Constraints
I2B2A. Equality Constraints Only
I2B2A1. Derivatives Not Required
I2B2A2. First Derivatives of Function and Constraints Required
I2B2A3. Second Derivatives of Function and First Derivatives of Constraints Required
I2B2A4. Second Derivatives of Both Function and Constraints Required
I2B2B. Equality and Inequality Constraints
I2B2B1. Derivatives No Required

I2B2B2. First Derivatives of Function and Constraints Required
I2B2B3. Second Derivatives of Function and First Derivatives of Constraints Required
I2B2B4. Second Derivatives of Both Function and Constraints Required
I2D. Integer Programming

J. Input/Output
J1. Binary
J2. Octal/Hexadecimal
J3. Decimal
J4. Character String
J5. Graphics, Plotting
J5A. Batch
J5B. Interactive

K. Internal File Manipulations
K1. Copy or Move File(s)
K2. Create a File
K2A. Sequential
K2B. Library ("Partitioned Data Set")
K2Z. Other
K3. Destroy a File
K4. Compare Two Files
K5. Update a File
K7. File Maintenance
K8. Program Library Maintenance

L. Language Processors
L1. Assemblers
L1A. Macro Assemblers
L1B. Other Assemblers (Which Produce Code for the Same Machine)
L1C. Cross-Assemblers (i.e., One Which Runs on One Computer But Produces Code for Another Computer)
L2. Compilers for Algebraic Languages (Excludes LISP,SNOBOL,etc.)
L2A. Batch
L2B. Debugging (e.g., WATFIV)
L2C. Interactive or Incremental
L2D. Cross-Compilers
L4. Preprocessors
L5. Disassembly and De-Relativizing
L6. Loaders and Linkers
L7. Computer Language to Computer Language Translators
L8. Interpreters (But Not Emulators)

M. Data Handling
M1. Sorting
M2. Conversion and/or Scaling
M3. Merging
M4. Character Manipulation

M5. Searching, Seeking, Locating

N. Debugging
N1. Tracing, Trapping, Breakpoints
N2. Dumping
N2A. Memory Dump
N2B. Tape Dump
N2C. Disk/Drum Dump

O. Simulation of Computers and Data Processors, Emulators
O1. Off-Line Equipment
O3. Computers
O4. Pseudo-Computers
O9. Other or Composite

Q. Service Routines, Programming Aids
Q1. Program Timers
Q2. Interrupt Processors
Q5. Report Generator Routines

R. Logical and Symbolic
R1. Shifting, Masking, Logical Bit Operations
R2. Symbol Manipulation (e.g., SNOBOL, REDUCE)
R3. List Processing
R4. Text Editors

S. Information Retrieval

T. Applications and Application-Oriented Programs
T1. Physics (Including Nuclear)
T2. Chemistry
T3. Other Physical Sciences (Geology, Astronomy, etc.)
T4. Engineering
T5. Business Data Processing
T6. Manufacturing, (Non-Data) Processing, and Process Control
T7. Mathematics and Applied Mathematics
T8. Social and Behavioral Sciences and Psychology
T9. Biological Sciences

Z. All Others

VI. THE SLATEC TABLE OF CONTENTS.

This section contains the table of contents for Version 1.0 of the SLATEC FORTRAN Library. The table is ordered by category. Following the table of contents is an alphabetical listing of the user-callable routines.

If you are using MATHDOC with SLATEC and have not memorized the SLATEC table of contents or do not have this document handy, you can retrieve a copy of the table of contents for yourself. It has been stored with the SLATEC library documentation as the documentation module named CONTENTS. If you retrieve the CONTENTS using MATHDOC you can be assured of getting the table of contents for the latest version of SLATEC.

****

EXAMPLE: Retrieve the SLATEC table of contents.

```
COMPUTER  :*<ret>

LIBRARY   :slatec<ret>

MODULE    :contents<ret>

KEYWORDS  :<ret>

CATEGORY  :<ret>
```

The documentation program will then search for CONTENTS, tell you that it has found one module, and ask where you want the documentation written.

****

SLATEC Table of Contents

Category	Name	Purpose
B	CARG	Computes the argument of a complex number.
B1	CACOS	Computes the complex arc cosine.
B1	CASIN	Computes the complex arc sine.
B1	CATAN	Computes the complex arc tangent.
B1	CATAN2	Computes the complex arc tangent in the proper quadrant.
B1	CCOT	Computes the complex cotangent.
B1	COSDG	Computes the cosine of an argument in degrees.
B1	COT	Computes the cotangent.
B1	CTAN	Computes the complex tangent.

B1	D9ATN1	Evaluates datan(x) from first order relative accuracy so that datan(x) = x + x**3*D9ATN1(x).
B1	DCOSDG	Computes the double precision cosine for degree arguments.
B1	DCOT	Computes the double precision cotangent.
B1	DSINDG	Computes the double precision sine for degree arguments.
B1	R9ATN1	Evaluate atan(x) from first order relative accuracy so that atan(x) = x + x**3*R9ATN1(x).
B1	SINDG	Computes the sine of an argument in degrees.
B2	ACOSH	Compute the arc hyperbolic cosine.
B2	ASINH	Computes the arc hyperbolic sine.
B2	ATANH	Compute the arc hyperbolic tangent.
B2	CACOSH	Computes the complex arc hyperbolic cosine.
B2	CASINH	Computes the complex arc hyperbolic sine.
B2	CATANH	Computes the complex arc hyperbolic tangent.
B2	CCOSH	Computes the complex hyperbolic cosine.
B2	CSINH	Computes the complex hyperbolic sine.
B2	CTANH	Computes the complex hyperbolic tangent.
B2	DACOSH	Computes the arc hyperbolic cosine.
B2	DASINH	Computes the arc hyperbolic sine.
B2	DATANH	Computes the double precision arc hyperbolic tangent.
B3	ALNREL	Evaluates ln(1+x) accurate in the sense of relative error.
B3	C9LN2R	Evaluates clog(1+z) from second order with relative error so that clog(1+z) = z - z**2/2 + z**3*C9LN2R(z).
B3	CEXPRL	Evaluates (cexp(z)-1)/z so that exp(z) = 1 + z*CEXPRL(z).
B3	CLNREL	Computes the principal value of the complex natural logarithm of 1+z with relative error accuracy for small cabs(z).
B3	CLOG10	Computes the principal value of the complex base 10 logarithm.
B3	D9LN2R	Evaluate dlog(1+x) from second order relative accuracy so that dlog(1+x) = x - x**2/2 + x**3*D9LN2R(x).
B3	DEXPRL	Calculates the double precision relative error exponential (dexp(x)-1)/x.
B3	DLNREL	Evaluates double precision ln(1+x) accurate in the relative error sense.
B3	EXPREL	Evaluates EXPREL(x) = (exp(x)-1)/x).
B3	R9LN2R	Evaluates alog(1+x) from second order relative accuracy so that alog(1+x) = x - x**2/2 + x**3*R9LN2R(x).
B4	CBRT	Computes the cube root of x.
B4	CCBRT	Computes the complex cube root of z.
B4	DCBRT	Computes the double precision cube root.
B5	R9AIMP	Evaluate the Airy modulus and phase.
B5C	BINOM	Computes the binomial coefficients.
B5C	DBINOM	Computes the double precision binomial coefficients.
B5C	DFAC	Computes the double precision factorial of n.
B5C	FAC	Computes the factorial of n.
B5E	ALI	Computes the logarithmic integral.
B5E	DE1	Computes the double precision exponential integral E1(x).
B5E	DEI	Computes the double precision exponential integral EI(x).

B5E	DEXINT	DEXINT computes m member sequences of exponential integrals e(n+K,x), k=0,1,...,m-1 for n.ge.1 and x.ge.0.
B5E	DLI	Computes the double precision logarithmic integral.
B5E	E1	Computes the exponential integral E1(x).
B5E	EI	Computes the exponential integral EI(x).
B5E	EXINT	EXINT computes m member sequences of exponential integrals e(n+K,x), k=0,1,...,m-1 for n.ge.1 and x.ge.0.
B5F	ALBETA	Computes the natural log of the complete BETA function.
B5F	ALGAMS	Computes log(abs(GAMMA(x))).
B5F	ALNGAM	Computes the log of the absolute value of the GAMMA function.
B5F	BETA	Computes the complete BETA function.
B5F	BETAI	Computes the incomplete BETA function.
B5F	COLGMC	Evaluates (z+0.5)*clog((z+1.)/z) - 1.0 with relative accuracy.
B5F	C9LGMC	Computes the log GAMMA correction term for most z so that clog(CGAMMA(z)) = 0.5*alog(2.*pi) + (z-0.5)*clog(z) - z +C9LGMC(z).
B5F	CBETA	Computes the complete BETA function of complex parameters a and c.
B5F	CGAMMA	Computes the GAMMA function of complex argument.
B5F	CGAMR	Computes the reciprocal GAMMA function of complex argument.
B5F	CLBETA	CLBETA computes the natural log of the complex valued complete BETA function of complex parameters a and B.
B5F	CLNGAM	CLNGAM computes the natural log of the complex valued GAMMA function at zin, where zin is a complex number.
B5F	CPSI	Computes the PSI function of complex argument.
B5F	D9GMIC	Calculates the double precision incomplete GAMMA function for a near a negative integer and x small.
B5F	D9GMIT	Computes double precision Tricomi-s incomplete GAMMA function for small x.
B5F	D9LGIC	Computes the double precision log incomplete GAMMA function for large x and for a .le. x.
B5F	D9LGIT	Computes the log of Tricomi-s incomplete GAMMA function with Perron-s continued fraction for large x and a .ge. x.
B5F	D9LGMC	Computes the double precision log GAMMA correction factor for x .ge. 10. so that dlog(DGAMMA(x)) = dlog(dsqrt(2*pi)) + (x-5.)*dlog(x) - x + D9LGMC(x).
B5F	DBETA	Computes the double precision complete BETA function.
B5F	DBETAI	Calculates the double precision incomplete BETA function.
B5F	DGAMI	Calculates the double precision incomplete GAMMA function.
B5F	DGAMIC	Calculates the double precision complementary incomplete GAMMA function.
B5F	DGAMIT	Calculates Tricomi-s form of the incomplete GAMMA function.
B5F	DGAMLM	Computes the double precision minimum and maximum bounds for x in GAMMA(x).
B5F	DGAMMA	Computes the double precision complete GAMMA function.

B5F	DGAMR	Calculates double precision reciprocal GAMMA function.
B5F	DLBETA	Computes the double precision natural logarithm of the complete BETA function.
B5F	DLGAMS	Calculates the log of the absolute value of the GAMMA function.
B5F	DLNGAM	Computes the double precision logarithm of the absolute value of the GAMMA fn.
B5F	DPSI	Computes the double precision PSI (or digamma) function.
B5F	GAMI	Computes the incomplete GAMMA function.
B5F	GAMIC	Computes the complementary incomplete GAMMA function.
B5F	GAMIT	Computes Tricomi-s form of the incomplete GAMMA function.
B5F	GAMLIM	Computes the minimum and maximum bounds for x in GAMMA(x).
B5F	GAMMA	Computes the GAMMA function.
B5F	GAMR	Computes the reciprocal of the GAMMA function.
B5F	PSI	Computes the PSI (or digamma) function.
B5F	R9GMIC	Compute the complementary incomplete GAMMA function for a near a negative integer and for small x.
B5F	R9GMIT	Computes Tricomi-s incomplete GAMMA function for small x.
B5F	R9LGIC	Computes the log complementary incomplete GAMMA function for large x and for a .le. x.
B5F	R9LGIT	Computes the log of Tricomi-s incomplete GAMMA function with Perron-s continued fraction for large x and for a .ge. x.
B5F	R9LGMC	Computes the log GAMMA correction factor so that alog(GAMMA(x)) = alog(sqrt(2*pi)) + (x-.5)*alog(x) - x + R9LGMC(x).
B5G	DERF	Computes the double precision error function, ERF, of x.
B5G	DERFC	Computes the double precision complementary error function, ERFC.
B5G	ERF	Computes the error function, ERF.
B5G	ERFC	Computes the complementary error function.
B5I	AI	Computes the Airy function.
B5I	AIE	Computes the exponentially scaled Airy function.
B5I	BESI	BESI computes an n member sequence of i Bessel functions i/sub(alpha+K-1)/(x), k=1,...,n or scaled Bessel functions exp(-x)*i/sub(alpha+K-1)/(x), k=1,...,n for non-negative alpha and x.
B5I	BESJ	BESJ computes an n member sequence of j Bessel functions j/sub(alpha+K-1)/(x), k=1,...,n for non-negative alpha and x.
B5I	BESJ0	Computes the Bessel function of the first kind of order zero.
B5I	BESJ1	Computes the Bessel function of the first kind of order one.
B5I	BESK	BESK implements forward recursion on the three term recursion relation for a sequence of non-negative order Bessel functions k/sub(fnu+i-1)/(x), or scaled Bessel functions exp(x)*k/sub(fnu+i-1)/(x), i=1,...,n for real x.gt.0.0e0 and non-negative orders fnu.
B5I	BESY	BESY implements forward recursion on the three term

		recursion relation for a sequence of non-negative order Bessel functions y/sub(fnu+i-1)/(x), i=1,n for real x.gt.0.0e0 and non-negative orders fnu.
B5I	BESY0	Computes the Bessel function of the second kind of order zero.
B5I	BESY1	Computes the Bessel function of the second kind of order one.
B5I	BI	Computes the bairy function.
B5I	BIE	Computes the exponentially scaled bairy function.
B5I	D9AIMP	Evaluates the Airy modulus and phase for x .le. -1.0.
B5I	DAI	Calculates the double precision Airy function.
B5I	DAIE	Calculates the double precision Airy function for negative x and an exponentially scaled Airy function for positive x.
B5I	DBESI	DBESI computes an n member sequence of i Bessel functions i/sub(alpha+K-1)/(x), k=1,...,n or scaled Bessel functions dexp(-x)*i/sub(alpha+K-1)/(x), k=1,...,n for nonnegative alpha and x.
B5I	DBESJ	DBESJ computes an n member sequence of j Bessel functions j/sub(alpha+K-1)/(x), k=1,...,n for non-negative alpha and x.
B5I	DBESJ0	Computes the double precision Bessel function of the first kind of order zero.
B5I	DBESJ1	Computes the double precision Bessel function of the first kind of order one.
B5I	DBESK	DBESK implements forward recursion on the three term recursion relation for a sequence of non-negative order Bessel functions k/sub(fnu+i-1)/(x), or scaled Bessel functions dexp(x)*k/sub(fnu+i-1)/(x), i=1,...,n for real x.gt.0.0d0 and non-negative orders fnu.
B5I	DBESY	DBESY implements forward recursion on the three term recursion relation for a sequence of non-negative order Bessel functions y/sub(fnu+i-1)/(x), i=1,n for real x.gt.0.0d0 and non-negative orders fnu.
B5I	DBESY0	Computes the double precision Bessel function of the second kind of order zero.
B5I	DBESY1	Computes the double precision Bessel function of the second kind of order one.
B5I	DBI	Calculates the double precision bairy function (Airy of second kind).
B5I	DBIE	Calculates the double precision bairy function for negative x and an exponentially scaled bairy function for positive x.
B5J	BESI0	Computes the hyperbolic Bessel function of the first kind of order 0.
B5J	BESI0E	Computes the exponentially scaled hyperbolic Bessel function of the first kind of order zero.
B5J	BESI1	Computes the hyperbolic Bessel function of first kind of order one.
B5J	BESK0	Computes the hyperbolic Bessel function of the third kind of order 0.

B5J	BESK0E	Computes the exponentially scaled hyperbolic Bessel function of the third kind of order zero.
B5J	BESK1	Computes the hyperbolic Bessel function of the third kind of order one.
B5J	BESK1E	Computes the exponentially scaled hyperbolic Bessel function of the third kind of order one.
B5J	BESKES	Computes a sequence of exponentially scaled modified Bessel functions of the third kind of fractional order.
B5J	BESKS	Computes a sequence of modified Bessel functions of the third kind of fractional order.
B5J	DBESI0	Computes the double precision hyperbolic Bessel function of the first kind of order zero.
B5J	DBESI1	Computes the double precision modified (hyperbolic) Bessel function of the first kind of order one.
B5J	DBESK0	Computes the double precision modified (hyperbolic) Bessel function of the third kind of order zero.
B5J	DBESK1	Computes the double precision modified Bessel function of the third kind.
B5J	DBESKS	Computes a double precision sequence of modified Bessel functions of the third kind of fractional order.
B5J	DBSI0E	Computes the double precision exponentially scaled hyperbolic Bessel function of the first kind of order zero.
B5J	DBSK0E	Computes the double precision exponentially scaled modified (hyperbolic) Bessel function of the third kind of order zero.
B5J	DBSK1E	Computes the exponentially scaled,modified (hyperbolic) Bessel function of the third kind of order one (double precision).
B5J	DBSKES	Computes a double precision sequence of exponentially scaled modified Bessel functions of the third kind of fractional order.
B5K	BESI1E	Computes the exponentially scaled hyperbolic Bessel function of the first kind of order one.
B5K	DBSI1E	Computes the double precision exponentially scaled modified (hyperbolic) Bessel function of the first kind of order one.
B5L	D9B0MP	Evaluates the double precision modulus and phase for j0 and y0 Bessel functions.
B5L	D9B1MP	Evaluate the double precision modulus and phase for the j1 and y1 Bessel fns.
B5L	D9KNUS	Computes Bessel functions exp(x)*K-sub-xnu(x) and exp(x)* K-sub-xnu+1(x) for 0. .le. xnu .lt. 1.
B5L	R9KNUS	Computes Bessel functions exp(x)*K-sub-xnu(x) and exp(x)* K-sub-xnu+1(x) for 0.0 .le. xnu .lt. 1.0.
B5N	CHU	Computes the logarithmic confluent hypergeometric function.
B5N	D9CHU	Evaluates for large z z**a * u(a,B,z) where u is the logarithmic confluent hypergeometric function. (double precision).
B5N	DCHU	Calculates the double precision logarithmic confluent

		hypergeometric function.
B5N	R9CHU	Evaluates for large z z**a * CHU(a,B,z).
B5R	CSEVL	Evaluate the n-term chebyshev series cs at x.
B5R	DCSEVL	Evaluate the double precision n-term chebyshev series a at x.
B5R	INITDS	Initializes the double precision properly normalized orthogonal polynomial series to determine the number of terms needed for specific accuracy.
B5R	INITS	Initializes an orthogonal series so that it defines the number of terms to carry in the series to meet a specified error.
B5Z	DAWS	Computes dawson-s function.
B5Z	DDAWS	Computes the double precision dawson-s function.
B5Z	DPOCH	Computes double precision generalized Pochhammer-s symbol.
B5Z	DPOCH1	Calculates a generalization of pochhammers symbol starting from first order. (double precision).
B5Z	DSPENC	Computes a double precision form of Spence-s integral due to K. Mitchell.
B5Z	POCH	Evaluates a generalization of Pochhammer-s symbol.
B5Z	POCH1	Computes Pochhammer-s symbol from first order.
B5Z	SPENC	Compute Spence-s function.
C2	CPQR79	Find the zeros of a polynomial with complex coefficients.
C2	CPZERO	Find the zeros of a polynomial with complex coefficients.
C2	RPQR79	To find the zeros of a polynomial with real coefficients.
C2	RPZERO	Find the zeros of a polynomial with real coefficients.
C4A	FZERO	FZERO searches for a zero of a function f(x) in a given interval (B,c). It is designed primarily for problems where f(B) and f(c) have opposite signs.
C5A	SNSQ	SNSQ finds to find a zero of a system of n nonlinear functions in n variables by a modification of the Powell hybrid method. this code is the combination of the MINPACK codes (Argonne) hybrd and hybrdj.
C5A	SNSQE	SNSQE is the easy-to-use version of SNSQ which finds a zero of a system of n nonlinear functions in n variables by a modification of Powell hybrid method. This code is the combination of the MINPACK codes (Argonne) hybrd1 and hybrj1.
C5B	SOS	SOS solves a system of nonlinear equations.
D1	QAGP	To evaluate a given definite integral over a finite interval with user supplied break points.
D1B	AVINT	Integrate a function tabulated at arbitrarily spaced abscissas using overlapping parabolas.
D1B1A	QNG	To estimate the value of a given definite integral over a finite interval using a non-adaptive scheme.
D1B2A	DGAUS8	DGAUS8 integrates double precision functions of one variable over finite intervals using an adaptive 8-point Legendre-Gauss algorithm. DGAUS8 is intended primarily for high accuracy integration or integration of smooth functions.
D1B2A	GAUS8	GAUS8 integrates real functions of one variable over

		finite intervals using an adaptive 8-point Legendre-Gauss algorithm. GAUS8 is intended primarily for high accuracy integration or integration of smooth functions.
D1B2A	QAG	To evaluate a given definite integral over a finite interval.
D1B2A	QAGS	To evaluate a given definite integral over a finite interval.
D1B2A	QNC79	Integrate a user defined function by a 7-point adaptive Newton-Cotes quadrature rule.
D1C2	QAWC	To evaluate a given definite integral over a finite interval with a Cauchy weight function.
D1C2	QAWO	To estimate the value of a given definite integral over a finite interval with an oscillatory weight function.
D1C2	QAWS	To estimate the value of a given definite integral over a finite interval with an algebraic or logarithmic weight function.
D1D2	QAGI	To estimate the value of a given definite integral over a semi-infinite or infinite interval.
D1D2	QAWF	To estimate the value of a definite integral over a semi-infinite interval with a fourier weight function.
D2A1	DERKF	DERKF solves initial value problems in ordinary differential equations using the Runga-Kutta-Fehlberg fifth order formula. It is variable step.
D2A2	DEABM	DEABM solves initial value problems in ordinary differential equations using an adams method. It is variable order(1-12) and variable step.
D2A2	INTRP	Approximates the solution at xout by evaluating the polynomial computed in STEP2 at xout. Must be used in conjunction with STEP2.
D2A2	STEP2	Integrates a system of first order odes one step.
D2A9	DEBDF	DEBDF solves initial value problems in ordinary differential equations using backward differentiation formulas. It is both variable order (1-5) and variable step.
D2B	BVSUP	Subroutine BVSUP solves a linear two-point boundary value problem using superposition coupled with an orthonormalization procedure and a variable-step integration scheme.
D3D2	BLKTRI	Solves a block tridiagonal system of linear equations (usually resulting from the discretization of separable elliptic equations).
D3D2	POIS	Solves Poisson-s equation using a standard difference approximation.
D3D2	PWSCRT	Solves the Helmholtz equation in cartesian coordinates using the standard 5-point difference approximation.
D3D2	PWSCSP	Solves the modified Helmholtz equation in spherical coordinates (assuming axisymmetry) using a finite difference approximation.
D3D2	PWSCYL	Solves the modified Helmholtz equation in cylindrical coordinates using a finite difference approximation.
D3D2	PWSPLR	Solves the Helmholtz equation in polar coordinates using

		a finite difference approximation.
D3D2	PWSSSP	Solves the Helmholtz equation in spherical coordinates and on the unit sphere using a finite difference approximation.
D6	CFFTB	Inverse fft of a complex periodic sequence.
D6	CFFTF	Forward fft of a complex periodic sequence.
D6	CFFTI	Initialize for CFFTF and CFFTB.
D6	COSQB	Inverse cosine transform with odd wave numbers (inverse of COSQF).
D6	COSQF	Forward cosine transform with odd wave numbers.
D6	COSQI	Initialize for COSQF and COSQB.
D6	COST	Cosine transform of a real even sequence.
D6	COSTI	Initialize for COST.
D6	EZFFTB	Simplified version of inverse fft of a real periodic sequence.
D6	EZFFTF	Forward fft of a real periodic sequence (simplifed version).
D6	EZFFTI	Initialize for EZFFTF and EZFFTB.
D6	RFFTB	Inverse fft of a real periodic sequence.
D6	RFFTF	Forward fft of a real periodic sequence.
D6	RFFTI	Initialize for RFFTF and RFFTB.
D6	SINQB	Inverse sine transform with odd wave numbers (inverse of SINQF).
D6	SINQF	Forward sine transform with odd wave numbers.
D6	SINQI	Initialize for SINQF and SINQB.
D6	SINT	Sine transform of a real odd sequence.
D6	SINTI	Initialize for SINT.
E1	POLCOF	Converts POLINT coefficients to Taylor series form.
E1A	BFQAD	Computes the integral on (x1,x2) of a product of a function f and the id-th derivative of a K-th order B-spline (B-representation).
E1A	BINT4	Computes the B-representation of a cubic spline which interpolates data (x(i),y(i)),i=1,ndata.
E1A	BINTK	Produces the B-spline coefficients, bcoef, of the B-spline of order K with knots t(i), i=1,...,n+K, which takes on the value y(i) at x(i), i=1,...,n.
E1A	BSPDR	Uses the B-representation to construct a divided difference table preparatory to a (right) derivative calculation in BSPEV.
E1A	BSPEV	Calculates the value of the spline and its derivatives at x from the B-representation .
E1A	BSPPP	Converts the B-representation to the piecewise polynomial (PP) form for use with PPVAL.
E1A	BSPVD	Calculates the value and all derivatives of order less than nderiv of all basis functions which do not vanish at x.
E1A	BSPVN	Calculates the value of all (possibly) nonzero basis functions at x.
E1A	BSQAD	Computes the integral on (x1,x2) of a K-th order B-spline using the B-representation.
E1A	BVALU	Evaluates the B-representation of a B-spline at x for the

		function value or any of its derivatives.
E1A	DBFQAD	Computes the integral on (x1,x2) of a product of a function f and the id-th derivative of a K-th order B-spline (B-representation).
E1A	DBINT4	Computes the B-representation of a cubic spline which interpolates data (x(i),y(i)),i=1,ndata.
E1A	DBINTK	Produces the B-spline coefficients, bcoef, of the B-spline of order K with knots t(i), i=1,...,n+K, which.
E1A	DBSPDR	Uses the B-representation to construct a divided difference table adif preparatory to a (right) derivative calculation in BSPEV.
E1A	DBSPEV	Calculates the value of the spline and its derivatives at x from the B-representation .
E1A	DBSPPP	Converts the B-representation to the piecewise polynomial (PP) form for use with PPVAL.
E1A	DBSPVD	Calculates the value and all derivatives of order less than nderiv of all basis functions which do not vanish at x.
E1A	DBSPVN	Calculates the value of all (possibly) nonzero basis functions at x.
E1A	DBSQAD	Computes the integral on (x1,x2) of a K-th order B-spline using the B-representation.
E1A	DBVALU	Evaluates the B-representation of a B-spline at x for the func value or any of its derivatives.
E1A	DINTRV	Computes the largest integer ileft in 1.le.ileft.le.lxt such that xt(ileft).le.x where xt(*) is a subdivision of the x interval.
E1A	DPFQAD	Computes the integral on (x1,x2) of a product of a function f and the id-th derivative of a B-spline, (PP-representation).
E1A	DPPQAD	Computes the integral on (x1,x2) of a K-th order B-spline using the piecewise polynomial representation.
E1A	DPPVAL	Calculates (at x) the value of the ideriv-th derivative of the B-spline from the PP-representation.
E1A	INTRV	Computes the largest integer ileft in 1.le.ileft.le.lxt such that xt(ileft).le.x where xt(*) is a subdivision of the x interval.
E1A	PFQAD	Computes the integral on (x1,x2) of a product of a function f and the id-th derivative of a B-spline, (PP-representation).
E1A	PPQAD	Computes the integral on (x1,x2) of a K-th order B-spline using the piecewise polynomial representation.
E1A	PPVAL	Calculates (at x) the value of the ideriv-th derivative of the B-spline from the PP-representation.
E1B	POLINT	To produce the polynomial which interpolates a set of discrete data points.
E1B	POLYVL	Calculates the value of the polynomial and its first nder derivatives where the polynomial was produced by a previous call to POLINT.
E1B	PVALUE	To use the coefficients generated by POLFIT to evaluate the polynomial fit of degree l, along with the first nder

		of its derivatives, at a specified point.
E2	CHKDER	Checks user supplied jacobians for SNSQ and SNSQE.
E2	CV	Covariance companion to least squares code FC( ).
E2	PCOEF	Converts POLFIT coefficients to Taylor series form.
E2A	EFC	Fits a piece-wise polynomial curve to discrete data. The piece-wise polynomials are represented as B-splines. The fitting is done in a weighted least squares sense.
E2A	FC	Fits a piece-wise polynomial curve to discrete data. The piece-wise polynomials are represented as B-splines. The fitting is done in a least squares sense. Equality and inequality constraints can be imposed on the fitted curve.
E2G1A	POLFIT	To fit data in a least squares sense by polynomials in one variable.
E2G1A	SODS	Solves an overdetermined system of linear equations. For full rank matrices the unique least squares solution is provided.
E2G1A	SUDS	Solves an underdetermined system of linear equations. For full rank matrices the least squares solution is provided.
E2G1B1	SNLS	SNLS minimizes the sum of the squares of m nonlinear functions in n variables by a modification of the Levenberg-Marquardt algorithm. This code is the combination of the MINPACK codes (Argonne) lmder, lmdif, and lmstr.
E2G1B1	SNLSE	SNLSE is the easy-to-use version of SNLS which minimizes the sum of the squares of m nonlinear functions in n variables by a modification of the Levenberg-Marquardt algorithm. This code is the combination of the MINPACK codes (Argonne) lmder1, lmdif1, and lmstr.
E2G2A	LSEI	Solve a linearly constrained least squares problem with equality and inequality constraints, and optionally compute a covariance matrix.
E2G2A	WNNLS	Solves a linearly constrained linear least squares problem where the constraints can be equality or nonnegativity constraints.
F1A	CAXPY	Complex computation y = a*x + y.
F1A	CCOPY	Complex vector copy y = x.
F1A	CDCDOT	Complex inner product --see SDSDOT.
F1A	CDOTC	Dot product of complex vectors, uses complex conjugate of first vector.
F1A	CDOTU	Inner product of complex vectors.
F1A	CROTG	Constructs complex Givens rotation --see SROTG.
F1A	CSCAL	Complex vector scale x = a*x.
F1A	CSROT	Applies a plane rotation to complex vectors.
F1A	CSSCAL	Scale a complex vector.
F1A	CSWAP	Interchange complex vectors.
F1A	DASUM	Sum of magnitudes of double precision vector components.
F1A	DAXPY	Double precision computation y = a*x + y.
F1A	DCDOT	Complex inner product --see DSDOT.
F1A	DCOPY	Double precision vector copy y = x.

F1A	DDOT	Double precision inner product of double precision vectors.
F1A	DNRM2	Euclidean length (l2 norm) of double precision vector.
F1A	DQDOTA	Double precision inner product with extended precision accumulation and result.
F1A	DQDOTI	Double precision inner product with extended precision accumulation and result.
F1A	DROT	Apply double precision Givens rotation.
F1A	DROTG	Construct double precision plane Givens rotation.
F1A	DROTM	Apply double precision modified Givens transformation.
F1A	DROTMG	Construct double precision modified Givens transformation.
F1A	DSCAL	Double precision vector scale x = a*x.
F1A	DSDOT	Double precision inner product of single precision vectors.
F1A	DSWAP	Interchange double precision vectors.
F1A	ICAMAX	Find largest component of complex vector.
F1A	IDAMAX	Find largest component of double precision vector.
F1A	ISAMAX	Find largest component of single precision vector.
F1A	SASUM	Sum of magnitudes of single precision vector components.
F1A	SAXPY	Single precision computation y = a*x + y.
F1A	SCASUM	Sum of magnitudes of real and imaginary components of complex vector.
F1A	SCNRM2	Unitary norm of complex vector.
F1A	SCOPY	Copy single precision vector y = x.
F1A	SCOPYM	Copy negative of real sx to real sy.
F1A	SDOT	Single precision inner product of single precision vectors.
F1A	SDSDOT	Single precision result with inner product accumulated in double precision.
F1A	SNRM2	Euclidean length (l2 norm) of single precision vector.
F1A	SROT	Apply single precision Givens rotation.
F1A	SROTG	Construct single precision plane Givens rotation.
F1A	SROTM	Apply single precision modified Givens transformation.
F1A	SROTMG	Construct single precision modified Givens transformation.
F1A	SSCAL	Single precision vector scale x = a*x.
F1A	SSWAP	Interchange single precision vectors.
F2C2	SSIEV	To compute the eigenvalues and, optionally, the eigenvectors of a real symmetric matrix.
F2C2	SSPEV	To compute the eigenvalues and, optionally, the eigenvectors of a real symmetric matrix stored in packed form.
F2C3	SGEEV	To compute the eigenvalues and, optionally, the eigenvectors of a general real matrix.
F2C5	CHIEV	To compute the eigenvalues and, optionally, the eigenvectors of a complex Hermitian matrix.
F2C6	CGEEV	To compute the eigenvalues and, optionally, the eigenvectors of a general complex matrix.
F4A	DGECO	Factors a double precision matrix by Gaussian elimination and estimates the condition of the matrix.

F4A	DGEDI	Computes the determinant and inverse of a matrix using factors computed by DGECO or DGEFA.
F4A	DGEFA	Factors a double precision matrix by Gaussian elimination.
F4A	DGEFS	DGEFS solves a general double precision nxn system of linear equations.
F4A	DGESL	Solves the double precision system a*x=b or trans(a)*x=b using the factors computed by DGECO or DGEFA.
F4A	DGTSL	Solves the system t*x=b where t is a general tridiagonal matrix.
F4A	DTRCO	Estimates the condition of a double precision triangular matrix.
F4A	DTRDI	Computes the determinant and inverse of a double precision triangular matrix.
F4A	DTRSL	Solves systems of the form t*x=b or trans(t)*x=b where t is a triangular matrix of order n.
F4A	SGECO	Factors a real matrix by Gaussian elimination and estimates the condition number of the matrix.
F4A	SGEDI	Computes the determinant and inverse of a matrix using the factors computed by SGECO or SGEFA.
F4A	SGEFA	Factors a real matrix by Gaussian elimination.
F4A	SGEFS	SGEFS solves a general single precision real nxn system of linear equations.
F4A	SGEIR	SGEIR solves a general single precision real nxn system of linear equations. Iterative refinement is used to obtain an error estimate.
F4A	SGESL	Solves the real system a*x=b or trans(a)*x=b using the factors of SGECO or SGEFA.
F4A	SGTSL	Solves the system a*x=b where a is tridiagonal.
F4A	STRCO	Estimates the condition of a real triangular matrix.
F4A	STRDI	Computes the determinant and inverse of a real triangular matrix.
F4A	STRSL	Solves systems of the form t*x=b or trans(t)*x=b where t is a triangular matrix of order n.
F4B	CCHDC	Computes the Cholesky decomposition of a positive definite matrix. a pivoting option allows the user to estimate the condition of a positive definite matrix or determine the rank of a positive definite matrix.
F4B	CCHDD	Downdates an augmented Cholesky decomposition or the triangular factor of an augmented QR decomposition.
F4B	CCHEX	CCHEX updates the Cholesky factorization a=ctrans(r)*r of a positive definite matrix a of order p under diagonal permutations of the form u*r*e=rr, where e is a permutation matrix.
F4B	CCHUD	Updates an augmented Cholesky decomposition of the triangular part of an augmented QR decomposition.
F4B	CPBCO	Factors a complex Hermitian positive definite matrix stored in band form and estimates the condition of the matrix.
F4B	CPBDI	Computes the determinant of a complex Hermitian positive

		definite band matrix using factors from CPBCO or CPBFA. Use CPBSL for inverse. using the factors computed by CPBCO or CPBFA.
F4B	CPBFA	Factors a complex Hermitian positive definite matrix (band form).
F4B	CPBSL	Solves the complex Hermitian positive definite band system a*x=b using factors from CPBCO or CPBFA.
F4B	CPOCO	Factors a complex Hermitian positive definite matrix and estimates the condition of the matrix.
F4B	CPODI	Computes the determinant and inverse of a certain complex Hermitian positive definite matrix using factors of CPOCO, CPOFA, or CQRDC.
F4B	CPOFA	Factors a complex Hermitian positive definite matrix.
F4B	CPOFS	CPOFS solves a positive definite symmetric complex nxn system of linear equations.
F4B	CPOIR	CPOIR solves a positive definite Hermitian complex nxn system of linear equations. iterative refinement is used to obtain an error estimate.
F4B	CPOSL	Solves the complex Hermitian positive definite system a*x=b using the factors computed by CPOCO or CPOFA.
F4B	CPPCO	Factors a complex Hermitian positive definite matrix stored in packed form and estimates the condition of the matrix.
F4B	CPPDI	Computes the determinant and inverse of a complex Hermitian positive definite matrix using factors from CPPCO or CPPFA.
F4B	CPPFA	Factors a complex Hermitian positive definite matrix (packed form).
F4B	CPPSL	Solves the complex Hermitian positive definite system a*x=b using the factors computed by CPPCO or CPPFA.
F4B	CPTSL	Solves a complex positive definite tridiagonal system of equations.
F4B	DCHDC	Computes the Cholesky decomposition of a positive definite matrix. A pivoting option allows the user to estimate the condition of a positive definite matrix or determine the rank of a positive semidefinite matrix.
F4B	DCHDD	Downdates an augmented Cholesky decomposition or the triangular factor of an augmented QR decomposition.
F4B	DCHEX	Updates the Cholesky factorization a=trans(r)*r of a positive definite matrix a of order p under diagonal permutations of the form trans(e)*a*e where e is a permutation matrix.
F4B	DCHUD	Updates an augmented Cholesky decomposition of the triangular part of an augmented QR decomposition.
F4B	DPBCO	Factors a double precision symmetric positive definite matrix stored in band form and estimates the condition of the matrix.
F4B	DPBDI	Computes the determinant of a double precision symmetric positive definite band matrix using factors of DPBCO or DPBFA. Use DPBSL n times for inverse.
F4B	DPBFA	Factors a double precision symmetric positive definite

		matrix stored in band form.
F4B	DPBSL	Solves the double precision symmetric positive definite band system a*x=b using the factors computed by DPBCO or DPBFA.
F4B	DPOCO	Factors a double precision symmetric positive definite matrix and estimates the condition of the matrix.
F4B	DPODI	Computes the determinant and inverse of a certain double precision symmetric positive definite matrix (see abstract) using the factors computed by DPOCO, DPOFA or DQRDC.
F4B	DPOFA	Factors a double precision symmetric positive definite matrix.
F4B	DPOFS	DPOFS solves a positive definite symmetric double precision nxn system of linear equations.
F4B	DPOSL	Solves the double precision symmetric positive definite system a*x=b using the factors computed by DPOCO or DPOFA.
F4B	DPPCO	Factors a double precision symmetric positive definite matrix stored in packed form and estimates the condition of the matrix.
F4B	DPPDI	Computes the determinant and inverse of a double precision symmetric positive definite matrix using factors from DPPCO or DPPFA.
F4B	DPPFA	Factors a double precision symmetric positive definite matrix (packed form).
F4B	DPPSL	Solves the double precision symmetric positive definite system a*x=b.
F4B	DPTSL	Solves the system t*x=b where t is positive definite tridiagonal.
F4B	SCHDC	Computes the Cholesky decomposition of a positive definite matrix. a pivoting option allows the user to estimate the condition of a positive definite matrix or determine the rank of a positive semidefinite matrix.
F4B	SCHDD	Downdates an augmented Cholesky decomposition or the triangular factor of an augmented QR decomposition.
F4B	SCHEX	Updates the Cholesky factorization a=trans(r)*r of a positive definite matrix a of order p under diagonal permutations of the form trans(e)*a*e where e is a permutation matrix.
F4B	SCHUD	Updates an augmented Cholesky decomposition of the triangular part of an augmented QR decomposition.
F4B	SPBCO	Factors a real symmetric positive definite matrix (band form) and estimates the condition of the matrix.
F4B	SPBDI	Computes the determinant of a real symmetric positive definite band matrix using the factors computed by SPBCO or SPBFA. if the inverse is needed, use SPBSL n times.
F4B	SPBFA	Factors a real symmetric positive definite matrix (band form).
F4B	SPBSL	Solves the real symmetric positive definite band system a*x=b using the factors computed by SPBCO or SPBFA.
F4B	SPOCO	Factors a real symmetric positive definite matrix and

		estimates the condition number of the matrix.
F4B	SPODI	Computes the determinant and inverse of a certain real symmetric positive definite matrix (see abstract) using the factors computed by SPOCO, SPOFA or SQRDC.
F4B	SPOFA	Factors a real symmetric positive definite matrix.
F4B	SPOFS	SPOFS solves a positive definite symmetric real nxn system of linear equations.
F4B	SPOIR	SPOIR solves a positive definite symmetric real nxn system of linear equations. Iterative refinement is used to obtain an error estimate.
F4B	SPOSL	Solves the real symmetric positive definite system a*x=b using the factors computed by SPOCO or SPOFA.
F4B	SPPCO	Factors a real symmetric positive definite matrix stored in packed form and estimates the condition of the matrix.
F4B	SPPDI	Computes the determinant and inverse of a real symmetric positive definite matrix using factors from SPPCO or SPPFA.
F4B	SPPFA	SPPFA factors a real symmetric positive definite matrix (packed form).
F4B	SPPSL	Solves the real symmetric positive definite system a*x=b using the factors computed by SPPCO or SPPFA.
F4B	SPTSL	Solves the system a*x=b where a is positive definite tridiagonal.
F4C	CHICO	Factors a complex Hermitian matrix by elimination with symmetric pivoting and estimates the condition of the matrix.
F4C	CHIDI	Computes the determinant, inertia and inverse of a complex Hermitian matrix using the factors from CHIFA.
F4C	CHIFA	Factors a complex Hermitian matrix by elimination (symmetric pivoting).
F4C	CHISL	CHISL solves the complex Hermitian system a*x=b using factors of CHIFA.
F4C	CHPCO	Factors a complex Hermitian matrix (packed form) by elimination with symmetric pivoting and estimates the condition of the matrix.
F4C	CHPDI	Computes the determinant, inertia and inverse of a complex Hermitian matrix (packed form) using the factors from CHPFA.
F4C	CHPFA	Factors a complex Hermitian matrix (packed form) by elimination with symmetric pivoting.
F4C	CHPSL	Solves the complex Hermitian system a*x=b using factors of CHPFA.
F4C	CSICO	Factors a complex symmetric matrix by elimination with symmetric pivoting and estimates the condition of the matrix.
F4C	CSIDI	Computes the determinant and inverse of a complex symmetric matrix using the factors from CSIFA.
F4C	CSIFA	Factors a complex symmetric matrix by elimination (symmetric pivoting).
F4C	CSISL	Solves the complex symmetric system a*x=b using factors from CSIFA.

F4C	CSPCO	Factors a complex symmetric matrix stored in packed form by elimination with symmetric pivoting and estimates the condition of the matrix.
F4C	CSPDI	Computes the determinant and inverse of a complex symmetric matrix stored in packed form using factors from CSPFA.
F4C	CSPFA	Factors a complex symmetric matrix stored in packed form by elimination with symmetric pivoting.
F4C	CSPSL	Solves the complex symmetric system a*x=b using factors from CSPFA.
F4C	DSICO	Factors a double precision symmetric matrix by elimination with symmetric pivoting and estimates the condition of the matrix.
F4C	DSIDI	Computes the determinant, inertia and inverse of a double precision symmetric matrix using the factors from DSIFA.
F4C	DSIFA	Factors a double precision symmetric matrix by elimination with symmetric pivoting.
F4C	DSISL	Solves the double precision symmetric system a*x=b using the factors computed by DSIFA.
F4C	DSPCO	Factors a double precision symmetric matrix stored in packed form by elimination with symmetric pivoting and estimates the condition of the matrix.
F4C	DSPDI	Computes the determinant, inertia and inverse of a double precision symmetric matrix using the factors from DSPFA, where the matrix is stored in packed form.
F4C	DSPFA	Factors a double precision symmetric matrix stored in packed form by elimination with symmetric pivoting.
F4C	DSPSL	Solves the double precision symmetric system a*x=b using the factors computed by DSPFA.
F4C	SSICO	Factors a real symmetric matrix by elimination with symmetric pivoting and estimates the condition of the matrix.
F4C	SSIDI	Computes the determinant, inertia and inverse of a real symmetric matrix using the factors from SSIFA.
F4C	SSIFA	Factors a real symmetric matrix by elimination with symmetric pivoting.
F4C	SSISL	Solves the real symmetric system a*x=b using the factors of SSIFA.
F4C	SSPCO	Factors a real symmetric matrix stored in packed form by elimination with symmetric pivoting and estimates the condition of the matrix.
F4C	SSPDI	Computes the determinant, inertia and inverse of a real symmetric matrix (packed form) using the factors from SSPFA.
F4C	SSPFA	Factors a real symmetric matrix stored in packed form by elimination with symmetric pivoting.
F4C	SSPSL	Solves the real symmetric system a*x=b using factors from SSPFA.
F4F	CGECO	Factors a complex matrix by Gaussian elimination and estimates the condition of the matrix.
F4F	CGEDI	Computes the determinant and inverse of a complex matrix

		using the factors computed by CGECO or CGEFA.
F4F	CGEFA	Factors a complex matrix by Gaussian elimination.
F4F	CGEFS	CGEFS solves a general complex nxn system of linear equations.
F4F	CGEIR	CGEIR solves a general complex nxn system of linear equations. Iterative refinement is used to obtain an error estimate.
F4F	CGESL	Solves the complex system a*x=b or ctrans(a)*x=b using the factors computed by CGECO or CGEFA.
F4F	CGTSL	Solves a general complex tridiagonal system of equations.
F4F	CTRCO	Estimates the condition of a complex triangular matrix.
F4F	CTRDI	Computes the determinant and inverse of a complex triangular matrix.
F4F	CTRSL	Solves systems of the form t*x=b or ctrans(t)*x=b where t is a triangular matrix. Here ctrans(t) is conjugate transpose.
F4H	CGBCO	Factors a complex band matrix by Gaussian elimination and estimates the condition of the matrix.
F4H	CGBDI	Computes the determinant of a complex band matrix using factors from CGBCO or CGBFA. If the inverse is needed, use CGBSL n times.
F4H	CGBFA	Factors a complex band matrix by elimination.
F4H	CGBSL	Solves the complex band system a*x=b or ctrans(a)*x=b using the factors computed by CGBCO or CGBFA.
F4H	CNBCO	Factors a complex band matrix by Gaussian elimination and estimates the condition of the matrix.
F4H	CNBDI	Computes the determinant of a complex band matrix using factors computed by CNBCO or CNBFA.
F4H	CNBFA	Factors a complex band matrix by elimination.
F4H	CNBFS	CNBFS solves a general nonsymmetric banded complex nxn system of linear equations.
F4H	CNBIR	CNBIR solves a general nonsymmetric banded complex nxn system of linear equations. iterative refinement is used to obtain an error estimate.
F4H	CNBSL	Solves a complex band system using factors computed by CNBCO or CNBFA.
F4H	DGBCO	Factors a double precision band matrix by Gaussian elimination and estimates the condition of the matrix.
F4H	DGBDI	Computes the determinant of a band matrix using factors computed by DGBCO or DGBFA. Use DGBSL n times for the inverse.
F4H	DGBFA	Factors a double precision band matrix by elimination.
F4H	DGBSL	Solves the double precision band system a*x=b or trans(a)*x=b using the factors computed by DGBCO or DGBFA.
F4H	DNBCO	Factors a double precision band matrix by Gaussian elimination and estimates the condition of the matrix.
F4H	DNBDI	Computes the determinant of a band matrix using the factors computed by DNBCO or DNBFA.
F4H	DNBFA	Factors a double precision band matrix by elimination.
F4H	DNBFS	DNBFS solves a general nonsymmetric banded double

		precision nxn system of linear equations.
F4H	DNBSL	Solves a double precision band system using the factors computed by DNBCO or DNBFA.
F4H	SGBCO	Factors a real band matrix by Gaussian elimination and estimates the condition number of the matrix.
F4H	SGBDI	Computes the determinant of a band matrix using the factors computed by SGBCO or SGBFA. if the inverse is needed, use SGBSL n times.
F4H	SGBFA	Factors a real band matrix by elimination.
F4H	SGBSL	Solves the real band system a*x=b or trans(a)*x=b using the factors computed by SGBCO or SGBFA.
F4H	SNBCO	SNBCO factors a real band matrix by Gaussian elimination and estimates the condition number.
F4H	SNBDI	Computes the determinant of a band matrix using the factors computed by SNBCO or SNBFA.
F4H	SNBFA	SNBFA factors a real band matrix by elimination. nxn system of linear equations.
F4H	SNBFS	SNBFS solves a general nonsymmetric banded nxn system of linear equations.
F4H	SNBIR	SNBIR solves a general nonsymmetric banded nxn system of linear equations. Iterative refinement is used to obtain an error estimate.
F4H	SNBSL	SNBSL solves a real band system using factors computed by SNBCO or SNBFA.
F6	CQRDC	Uses Householder transformations to compute the QR factorization of an n by p matrix x. Column pivoting is optional.
F6	CQRSL	Applies the output of CQRDC to compute coordinate transformations, projections, and least squares solutions.
F6	CSVDC	Perform the singular value decomposition of a complex nxp matrix.
F6	DQRDC	Uses Householder transformations to compute the QR factorization of n by p matrix x. Column pivoting is optional.
F6	DQRSL	Applies the output of DQRDC to compute coordinate transformations, projections, and least squares solutions.
F6	DSVDC	Perform the singular value decomposition of a double precision nxp matrix.
F6	SQRDC	Uses Householder transformations to compute the QR factorization of an n by p matrix x. Column pivoting is a users option.
F6	SQRSL	Applies the output of SQRDC to compute coordinate transformations projections, and least squares solutions.
F6	SSVDC	Perform the singular value decomposition of a real nxp matrix.
G5E	RAND	Generates a uniformly distributed random number.
G5E	RUNIF	A portable random number generator.
G5F	RGAUSS	Generates a normally distributed (Gaussian) random number.

M1	SSORT	SSORT sorts array x and optionally makes the same interchanges in array y. The array x may be sorted in increasing order or decreasing order. A slightly modified quicksort algorithm is used.
M1A1	ISORT	ISORT sorts integer array x and optionally makes the same interchanges in integer array y. The array x may be sorted in increasing order or decreasing order. A slightly modified quicksort algorithm is used.
M2	D9PAK	Pack a base 2 exponent into a double precision floating point no.
M2	D9UPAK	Unpack a double precision floating point no. x so that x=y*2**n.
M2	R9PAK	Pack a base 2 exponent into a floating point number.
M2	R9UPAK	Unpack a floating point number x so that x=y*2**n.
Q	D1MACH	Returns double precision machine dependent constants.
Q	I1MACH	Returns integer machine dependent constants.
Q	R1MACH	Returns single precision machine dependent constants.
Q	XERABT	Aborts program execution and prints error message.
Q	XERCLR	Resets current error number to zero.
Q	XERCTL	Allows user control over handling of individual errors.
Q	XERDMP	Prints the error tables and then clears them.
Q	XERMAX	Sets maximum number of times any error message is to be printed.
Q	XERPRT	Prints error messages.
Q	XERROR	Processes an error (diagnostic) message.
Q	XERRWV	Processes error message allowing 2 integer and two real values to be included in the message.
Q	XERSAV	Records that an error occurred.
Q	XGETF	Returns current value of error control flag.
Q	XGETUA	Returns unit number(s) to which error messages are being sent.
Q	XGETUN	Returns the (first) output file to which messages are being sent.
Q	XSETF	Sets the error control flag.
Q	XSETUA	Sets up to 5 unit numbers to which messages are to be sent.
Q	XSETUN	Sets output file to which error messages are to be sent.
Q2	FDUMP	Symbolic dump (should be locally written).

Below is the list of SLATEC user-callable routines ordered by subroutine name.

	Name	Category
1	ACOSH	B2
2	AI	B5I
3	AIE	B5I
4	ALBETA	B5F

5	ALGAMS	B5F
6	ALI	B5E
7	ALNGAM	B5F
8	ALNREL	B3
9	ASINH	B2
10	ATANH	B2
11	AVINT	D1B
12	BESI	B5I
13	BESI0	B5J
14	BESI0E	B5J
15	BESI1	B5J
16	BESI1E	B5K
17	BESJ	B5I
18	BESJ0	B5I
19	BESJ1	B5I
20	BESK	B5I
21	BESK0	B5J
22	BESK0E	B5J
23	BESK1	B5J
24	BESK1E	B5J
25	BESKES	B5J
26	BESKS	B5J
27	BESY	B5I
28	BESY0	B5I
29	BESY1	B5I
30	BETA	B5F
31	BETAI	B5F
32	BFQAD	E1A
33	BI	B5I
34	BIE	B5I
35	BINOM	B5C
36	BINT4	E1A
37	BINTK	E1A
38	BLKTRI	D3D2
39	BSPDR	E1A
40	BSPEV	E1A
41	BSPPP	E1A
42	BSPVD	E1A
43	BSPVN	E1A
44	BSQAD	E1A
45	BVALU	E1A
46	BVSUP	D2B
47	C0LGMC	B5F
48	C9LGMC	B5F
49	C9LN2R	B3
50	CACOS	B1
51	CACOSH	B2
52	CARG	B
53	CASIN	B1
54	CASINH	B2
55	CATAN	B1
56	CATAN2	B1
57	CATANH	B2
58	CAXPY	F1A
59	CBETA	B5F
60	CBRT	B4
61	CCBRT	B4
62	CCHDC	F4B
63	CCHDD	F4B
64	CCHEX	F4B
65	CCHUD	F4B
66	CCOPY	F1A
67	CCOSH	B2
68	CCOT	B1
69	CDCDOT	F1A
70	CDOTC	F1A
71	CDOTU	F1A
72	CEXPRL	B3
73	CFFTB	D6
74	CFFTF	D6
75	CFFTI	D6
76	CGAMMA	B5F
77	CGAMR	B5F
78	CGBCO	F4H
79	CGBDI	F4H
80	CGBFA	F4H
81	CGBSL	F4H
82	CGECO	F4F
83	CGEDI	F4F
84	CGEEV	F2C6
85	CGEFA	F4F
86	CGEFS	F4F
87	CGEIR	F4F
88	CGESL	F4F
89	CGTSL	F4F
90	CHICO	F4C
91	CHIDI	F4C
92	CHIEV	F2C5
93	CHIFA	F4C
94	CHISL	F4C
95	CHKDER	E2
96	CHPCO	F4C
97	CHPDI	F4C
98	CHPFA	F4C
99	CHPSL	F4C
100	CHU	B5N
101	CLBETA	B5F
102	CLNGAM	B5F
103	CLNREL	B3
104	CLOG10	B3
105	CNBCO	F4H
106	CNBDI	F4H
107	CNBFA	F4H
108	CNBFS	F4H
109	CNBIR	F4H
110	CNBSL	F4H
111	COSDG	B1
112	COSQB	D6
113	COSQF	D6
114	COSQI	D6
115	COST	D6
116	COSTI	D6
117	COT	B1
118	CPBCO	F4B
119	CPBDI	F4B
120	CPBFA	F4B
121	CPBSL	F4B
122	CPOCO	F4B
123	CPODI	F4B
124	CPOFA	F4B
125	CPOFS	F4B
126	CPOIR	F4B
127	CPOSL	F4B
128	CPPCO	F4B
129	CPPDI	F4B
130	CPPFA	F4B
131	CPPSL	F4B
132	CPQR79	C2
133	CPSI	B5F
134	CPTSL	F4B
135	CPZERO	C2
136	CQRDC	F6
137	CQRSL	F6
138	CROTG	F1A
139	CSCAL	F1A
140	CSEVL	B5R
141	CSICO	F4C
142	CSIDI	F4C
143	CSIFA	F4C
144	CSINH	B2
145	CSISL	F4C
146	CSPCO	F4C
147	CSPDI	F4C
148	CSPFA	F4C
149	CSPSL	F4C
150	CSROT	F1A
151	CSSCAL	F1A
152	CSVDC	F6
153	CSWAP	F1A
154	CTAN	B1
155	CTANH	B2
156	CTRCO	F4F
157	CTRDI	F4F

158	CTRSL	F4F
159	CV	E2
160	D1MACH	Q
161	D9AIMP	B5I
162	D9ATN1	B1
163	D9B0MP	B5L
164	D9B1MP	B5L
165	D9CHU	B5N
166	D9GMIC	B5F
167	D9GMIT	B5F
168	D9KNUS	B5L
169	D9LGIC	B5F
170	D9LGIT	B5F
171	D9LGMC	B5F
172	D9LN2R	B3
173	D9PAK	M2
174	D9UPAK	M2
175	DACOSH	B2
176	DAI	B5I
177	DAIE	B5I
178	DASINH	B2
179	DASUM	F1A
180	DATANH	B2
181	DAWS	B5Z
182	DAXPY	F1A
183	DBESI	B5I
184	DBESI0	B5J
185	DBESI1	B5J
186	DBESJ	B5I
187	DBESJ0	B5I
188	DBESJ1	B5I
189	DBESK	B5I
190	DBESK0	B5J
191	DBESK1	B5J
192	DBESKS	B5J
193	DBESY	B5I
194	DBESY0	B5I
195	DBESY1	B5I
196	DBETA	B5F
197	DBETAI	B5F
198	DBFQAD	E1A
199	DBI	B5I
200	DBIE	B5I
201	DBINOM	B5C
202	DBINT4	E1A
203	DBINTK	E1A
204	DBSI0E	B5J
205	DBSI1E	B5K
206	DBSK0E	B5J
207	DBSK1E	B5J
208	DBSKES	B5J
209	DBSPDR	E1A
210	DBSPEV	E1A
211	DBSPPP	E1A
212	DBSPVD	E1A
213	DBSPVN	E1A
214	DBSQAD	E1A
215	DBVALU	E1A
216	DCBRT	B4
217	DCDOT	F1A
218	DCHDC	F4B
219	DCHDD	F4B
220	DCHEX	F4B
221	DCHU	B5N
222	DCHUD	F4B
223	DCOPY	F1A
224	DCOSDG	B1
225	DCOT	B1
226	DCSEVL	B5R
227	DDAWS	B5Z
228	DDOT	F1A
229	DE1	B5E
230	DEABM	D2A2
231	DEBDF	D2A9
232	DEI	B5E
233	DERF	B5G
234	DERFC	B5G
235	DERKF	D2A1
236	DEXINT	B5E
237	DEXPRL	B3
238	DFAC	B5C
239	DGAMI	B5F
240	DGAMIC	B5F
241	DGAMIT	B5F
242	DGAMLM	B5F
243	DGAMMA	B5F
244	DGAMR	B5F
245	DGAUS8	D1B2A
246	DGBCO	F4H
247	DGBDI	F4H
248	DGBFA	F4H
249	DGBSL	F4H
250	DGECO	F4A
251	DGEDI	F4A
252	DGEFA	F4A
253	DGEFS	F4A
254	DGESL	F4A
255	DGTSL	F4A
256	DINTRV	E1A
257	DLBETA	B5F
258	DLGAMS	B5F
259	DLI	B5E
260	DLNGAM	B5F
261	DLNREL	B3
262	DNBCO	F4H
263	DNBDI	F4H
264	DNBFA	F4H
265	DNBFS	F4H
266	DNBSL	F4H
267	DNRM2	F1A
268	DPBCO	F4B
269	DPBDI	F4B
270	DPBFA	F4B
271	DPBSL	F4B
272	DPFQAD	E1A
273	DPOCH	B5Z
274	DPOCH1	B5Z
275	DPOCO	F4B
276	DPODI	F4B
277	DPOFA	F4B
278	DPOFS	F4B
279	DPOSL	F4B
280	DPPCO	F4B
281	DPPDI	F4B
282	DPPFA	F4B
283	DPPQAD	E1A
284	DPPSL	F4B
285	DPPVAL	E1A
286	DPSI	B5F
287	DPTSL	F4B
288	DQDOTA	F1A
289	DQDOTI	F1A
290	DQRDC	F6
291	DQRSL	F6
292	DROT	F1A
293	DROTG	F1A
294	DROTM	F1A
295	DROTMG	F1A
296	DSCAL	F1A
297	DSDOT	F1A
298	DSICO	F4C
299	DSIDI	F4C
300	DSIFA	F4C
301	DSINDG	B1
302	DSISL	F4C
303	DSPCO	F4C
304	DSPDI	F4C
305	DSPENC	B5Z
306	DSPFA	F4C
307	DSPSL	F4C
308	DSVDC	F6
309	DSWAP	F1A
310	DTRCO	F4A

311	DTRDI	F4A
312	DTRSL	F4A
313	E1	B5E
314	EFC	E2A
315	EI	B5E
316	ERF	B5G
317	ERFC	B5G
318	EXINT	B5E
319	EXPREL	B3
320	EZFFTB	D6
321	EZFFTF	D6
322	EZFFTI	D6
323	FAC	B5C
324	FC	E2A
325	FDUMP	Q2
326	FZERO	C4A
327	GAMI	B5F
328	GAMIC	B5F
329	GAMIT	B5F
330	GAMLIM	B5F
331	GAMMA	B5F
332	GAMR	B5F
333	GAUS8	D1B2A
334	I1MACH	Q
335	ICAMAX	F1A
336	IDAMAX	F1A
337	INITDS	B5R
338	INITS	B5R
339	INTRP	D2A2
340	INTRV	E1A
341	ISAMAX	F1A
342	ISORT	M1A1
343	LSEI	E2G2A
344	PCOEF	E2
345	PFQAD	E1A
346	POCH	B5Z
347	POCH1	B5Z
348	POIS	D3D2
349	POLCOF	E1
350	POLFIT	E2G1A
351	POLINT	E1B
352	POLYVL	E1B
353	PPQAD	E1A
354	PPVAL	E1A
355	PSI	B5F
356	PVALUE	E1B
357	PWSCRT	D3D2
358	PWSCSP	D3D2
359	PWSCYL	D3D2
360	PWSPLR	D3D2
361	PWSSSP	D3D2
362	QAG	D1B2A
363	QAGI	D1D2
364	QAGP	D1
365	QAGS	D1B2A
366	QAWC	D1C2
367	QAWF	D1D2
368	QAWO	D1C2
369	QAWS	D1C2
370	QNC79	D1B2A
371	QNG	D1B1A
372	R1MACH	Q
373	R9AIMP	B5
374	R9ATN1	B1
375	R9CHU	B5N
376	R9GMIC	B5F
377	R9GMIT	B5F
378	R9KNUS	B5L
379	R9LGIC	B5F
380	R9LGIT	B5F
381	R9LGMC	B5F
382	R9LN2R	B3
383	R9PAK	M2
384	R9UPAK	M2
385	RAND	G5E
386	RFFTB	D6
387	RFFTF	D6
388	RFFTI	D6
389	RGAUSS	G5F
390	RPQR79	C2
391	RPZERO	C2
392	RUNIF	G5E
393	SASUM	F1A
394	SAXPY	F1A
395	SCASUM	F1A
396	SCHDC	F4B
397	SCHDD	F4B
398	SCHEX	F4B
399	SCHUD	F4B
400	SCNRM2	F1A
401	SCOPY	F1A
402	SCOPYM	F1A
403	SDOT	F1A
404	SDSDOT	F1A
405	SGBCO	F4H
406	SGBDI	F4H
407	SGBFA	F4H
408	SGBSL	F4H
409	SGECO	F4A
410	SGEDI	F4A
411	SGEEV	F2C3
412	SGEFA	F4A
413	SGEFS	F4A
414	SGEIR	F4A
415	SGESL	F4A
416	SGTSL	F4A
417	SINDG	B1
418	SINQB	D6
419	SINQF	D6
420	SINQI	D6
421	SINT	D6
422	SINTI	D6
423	SNBCO	F4H
424	SNBDI	F4H
425	SNBFA	F4H
426	SNBFS	F4H
427	SNBIR	F4H
428	SNBSL	F4H
429	SNLS	E2G1B1
430	SNLSE	E2G1B1
431	SNRM2	F1A
432	SNSQ	C5A
433	SNSQE	C5A
434	SODS	E2G1A
435	SOS	C5B
436	SPBCO	F4B
437	SPBDI	F4B
438	SPBFA	F4B
439	SPBSL	F4B
440	SPENC	B5Z
441	SPOCO	F4B
442	SPODI	F4B
443	SPOFA	F4B
444	SPOFS	F4B
445	SPOIR	F4B
446	SPOSL	F4B
447	SPPCO	F4B
448	SPPDI	F4B
449	SPPFA	F4B
450	SPPSL	F4B
451	SPTSL	F4B
452	SQRDC	F6
453	SQRSL	F6
454	SROT	F1A
455	SROTG	F1A
456	SROTM	F1A
457	SROTMG	F1A
458	SSCAL	F1A
459	SSICO	F4C
460	SSIDI	F4C
461	SSIEV	F2C2
462	SSIFA	F4C
463	SSISL	F4C

464	SSORT	M1
465	SSPCO	F4C
466	SSPDI	F4C
467	SSPEV	F2C2
468	SSPFA	F4C
469	SSPSL	F4C
470	SSVDC	F6
471	SSWAP	F1A
472	STEP2	D2A2
473	STRCO	F4A
474	STRDI	F4A
475	STRSL	F4A
476	SUDS	E2G1A
477	WNNLS	E2G2A
478	XERABT	Q
479	XERCLR	Q
480	XERCTL	Q
481	XERDMP	Q
482	XERMAX	Q
483	XERPRT	Q
484	XERROR	Q
485	XERRWV	Q
486	XERSAV	Q
487	XGETF	Q
488	XGETUA	Q
489	XGETUN	Q
490	XSETF	Q
491	XSETUA	Q
492	XSETUN	Q

ACKNOWLEDGMENTS

Many thanks go to Dona L. Crawford and Hilary D. Jones. These colleagues were always happy to provide constructive criticism and helpful suggestions.

REFERENCE

1. K. W. Fong, T. H. Jefferson, T. Suyehiro, "SLATEC Library Source File Format," informal SLATEC document, August 1981.

UNLIMITED RELEASE

INITIAL DISTRIBUTION

DR. DAVID KAHANER
NATIONAL BUREAU OF STANDARDS
SCIENTIFIC COMPUTING DIVISION 713
WASHINGTON, DC 20234

DR. MARTIN CORDES
MATHEMATICAL ANALYSIS DIVISION
BUILDING 101, ROOM A-302
NATIONAL BUREAU OF STANDARDS
WASHINGTON DC 20234

DR. WAYNE FULLERTON
COMPUTING SCIENCE RESEARCH CENTER
2C-459
BELL LABORATORIES
MURRAY HILL, NJ 07974

DR. RONALD PEIERLS
APPLIED MATHEMATICS DEPARTMENT
BROOKHAVEN NATIONAL LABORATORY
UPTON, NY 11973

DR. PATRICK W. GAFFNEY
UNION CARBIDE CORPORATION
NUCLEAR DIVISION, COMPUTER SCIENCE
P.O. BOX X, BUILDING 9704-1
OAK RIDGE, TN 37830

DR. ROBERT WARD
UNION CARBIDE CORPORATION
NUCLEAR DIVISION, COMPUTER SCIENCE
P.O. BOX X, BUILDING 9704-1
OAK RIDGE, TN 37830

DR. THEODORE HATCHER
EG&G IDAHO
COMPUTER SCIENCE CENTER
P.O. BOX 1625
IDAHO FALLS, IDAHO 83415

DEPARTMENT OF COMPUTER SCIENCES
REPORT SECTION
ROYAL INSTITUTE OF TECHNOLOGY
5-100 44
STOCKHOLM 70 SWEDEN

DR. C. L. LAWSON
MS 125/128
JET PROPULSION LABORATORY
4800 OAK GROVE DRIVE
PASADENA, CA 91103

DR. C. B. MOLER
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF NEW MEXICO
ALBUQUERQUE, NM 87131

DR. PAUL MESSINA
AMD
ARGONE NATIONAL LABORATORY
9700 SOUTH CASS AVENUE
ARGONNE, IL 60439

DR. MYRON GINSBERG
GENERAL MOTORS RESEARCH LABORATORY
COMPUTER SCIENCE DEPARTMENT
RAND-234
GENERAL MOTORS TECHNICAL CENTER
WARREN, MI 48090-9055

DR. JOHN RICE
DIVISION OF MATHEMATICAL SCIENCES
PURDUE UNIVEERSITY
WEST LAFAYETTE, IN 47907

DR. PHYLISS FOX
BELL LABORATORIES
600 MOUNTAIN HILL
MURRAY HILL, NJ 07974

MARGARET BUTLER
NATIONAL ENERGY SOFTWARE CENTER
ARGONNE NATIONAL LABORATORY
9700 SOUTH CASS AVENUE
ARGONNE, IL 60439

DR. BRIAN FORD
NAG, LTD.
MAYFIELD HOUSE
256 BANBURY ROAD
OXFORD, OX2 7DE
UNITED KINGDOM

AFWL/ADS LT. A. FUNK
AFWL/AD R. C. ALLEN

LANL K. BERKBIGLER, MS-B296
LANL R. BOLAND, MS-265
LANL B. BUZBEE, MS-265
LANL J. CHOW, MS-265
LANL P. FREDERICKSON, MS-265
LANL A. HAYES, MS-265
LANL T. JORDAN, MS-265
LANL T. MANTEUFFEL, MS-265
LANL E. VOORHEES, MS-265
LANL A. B. WHITE, MS-265

LLNL R. C. BASINGER, L 316
LLNL K. W. FONG, L 560
LLNL F. N. FRITSCH, L 316
LLNL A. C. HINDMARSH, L 316
LLNL N. G. SMIRIGA L 316
LLNL T. SUYEHIRO, L 316

2614 JONES R E
2614 SHEPHERD G C
2646 SCOTT M R
2646 HASKELL K H
2646 VANDEVENDER W H
2646 WALTON L
5642 SHAMPINE L F
5642 AMOS D E
8000 CLAASEN R S
8100 OLSON D M
8112 WITEK H M
8113 HILES L A
8115 WOODARD JAMES B JR
8116 BOLTON WILLARD R JR
8116 PURA CAROLYN A
8121 SCOTT CARLTON A
8121 KEILMAN JOHN C
8121 CALLABRESI MELVIN L
8121 VOELKER L EUGENE
8121 KAN YIH-RENN
8121 CHIESA MICHAEL L
8122 HOYLE CHARLES S
8122 ROGERS L ALICE
8122 GRANT JOSEPH E
8122 WEINGARTEN LAWRENCE I
8122 DUNDER VERA D
8122 CHANG ROSEMARY E
8122 NAPOLITANO LEONARD M
8122 MASON WILLIAM E JR
8123 ZINKE WILLIAM D

8123	JORGENSON WILBUR E
8123	PERRA MARK W
8123	KAWAHARA WENDELL A
8123	BIELER THOMAS R
8124	CHENOWETH DONALD R
8124	ABRAMS MARTIN
8124	GALLAGHER ROBERT J
8124	HACKETT COLIN E
8124	OWENS DOUGLAS D
8124	SIEBERS DENNIS L
8124	WINTERS WILLIAM S JR
8124	FISH MIRIAM JOHN
8124	KRAABEL JOHN S
8124	PAOLUCCI SAMUEL
8124	BELL DIANE M
8124	ORTEGA ARTHUR R
8124	STODDARD MARY C
8131	DEAN DAVID K
8131	HENDERSON GERALD W
8132	YOKOMIZO CLIFFORD T
8132	CULL EDWARD T
8132	HENSON DOUGLAS R
8132	MCDONALD ALFREDO
8132	HIRANO HOWARD H
8152	RYCHNOVSKY R E
8152	DREMALAS JAMES F
8152	NELSON DENNIS B
8152	EVERETT ROGER N
8152	LOLL MARVIN B
8152	LAPETINA NEIL A
8153	WRIGHT JAMES B
8153	DIGHTON LEONARD E
8153	SKOOG CLIFFORD D
8162	FURNBERG CARLTON M
8162	RIVENES ARNOLD S
8162	AMARAL RONALD J
8162	ROGERS MICHAEL H
8162	BURRIS KEITH L
8162	TILLEY ROGER S
8163	CHILDERS CARL W
8163	DIDLAKE JOHN E JR
8168	HICKS MEARLE G
8200	BLACKWELL A N
8214	CONNORS MATTHEW J
8215	INZERILLA JANET K
8215	CUPPS FRANK J
8215	YANO HESA
8215	JOINER EMILY A
8215	BARNCORD LINDA JO
8254	BIRNBAUM MICHAEL R
8265	DEAN PETER W

8271 EICHERT FRED F JR
8272 COOK RICHARD S JR
8274 FINN RICHARD W
8274 NEIGHBORS PAULA K
8274 HACHMAN JIM M
8274 KOOPMANN BRUCE E
8274 JONES JOHN F JR
8300 MURPHEY B F
8310 SCHUSTER D M
8312 SMUGERESKY JOHN E
8313 NICHOLS MONTE C
8313 BOEHME DALE R
8315 WEST LLOYD A
8315 MILLS BERNICE E
8320 RINNE R L
8322 DOLAN KENNETH W
8324 HANKINS JOE D
8324 ROGERS JAMES N
8324 BREAZEAL NORMAN L
8324 CZAPINSKI ROBERT H
8324 STRUVE JAMES E
8324 LOOK GEORGE W
8324 TOOMAN TIM P
8328 STRANDIN GERALD E
8328 BRANDT LARRY D
8328 IANNUCCI JOSEPH J
8328 VITKO JOHN JR
8329 KRIEGER HAROLD A
8329 DECARLI CHARLES J
8329 WAGNER NORMAN R
8330 ANDERSON G W JR
8331 GABRIELSON VERLAN K
8331 CLARK GARY L
8331 BISSON CHARLES L
8331 KEE ROBERT J JR
8331 LEARY PATRICIA L
8331 MARGOLIS STEPHEN B
8331 SMOOKE MITCHELL D
8331 PETZOLD LINDA R
8332 MILLER GORDON J
8332 LATHROP JAMES F
8332 MANSFIELD JUANITA
8332 HUDDLESTON ROBERT E
8332 JEFFERSON THOMAS H JR (15)
8332 CRAWFORD DONA L
8332 ELLIS RAYMOND W
8332 RINGLAND JAMES T
8334 BARKER BERTON E
8334 SHORT HAROLD G
8334 LEONARD CHARLES M JR
8334 LEE ROY Y

8334 JONES HILARY D
8335 SCHUKNECHT ARNOLD G
8335 WHITWORTH FREDDY L
8335 ANGVICK GENE L
8335 ISLER RICHARD E
8335 MITCHELL DONNA L
8335 PENDLEY MICHAEL H
8335 DRISCOLL PATRICIA A
8336 DRUMMOND G B JR
8340 BAUER W
8341 WILSON WILLIAM D
8341 THOMAS GEORGE J
8341 HAGGMARK LEROY G
8341 BASKES MICHAEL I
8341 BROWN LESLIE A
8341 STOLL MICHAEL E
8341 BINKLEY JOHN S
8342 VAN HOOK ARTHUR R
8342 GRAY STEPHEN C
8342 RAHN LARRY A
8342 GAY RICHARD D
8342 FARROW ROGER L
8342 SELBERG LARS A
8343 CORONADO PAUL R
8343 FOLK DANIEL R
8343 SWANSIGER WILLIAM A
8343 MELIUS CARL F
8343 KOSZYKOWSKI MICHAEL L
8343 NOELL J OAKEY
8347 CHRISMAN WAYNE L
8347 MORSE DANIEL H
8347 MALINOWSKI MICHAEL E
8347 GUTHRIE STEPHEN E
8347 WILSON KENNETH L
8347 VER BERKMOES ALFRED A
8347 STULEN RICHARD H
8347 BASTASZ ROBERT J
8347 PONTAU ARTHUR E
8347 KERST RONALD A
8400 GUTIERREZ L
8411 BENTON JOE D
8411 STOCKLEY C H JR
8411 SWAN HERBERT W
8411 HOLBROOK ELMOND D
8411 MACMILLAN DOUGLAS C
8411 GRISBY SYLVESTER
8411 CONVERSE LOUISE S
8411 MANROW BRITT MARIE
8411 BAUMAN JIMMIE L
8411 HUMPHREY ROBERT E
8411 MAYER ELIZABETH R

8415 ESTILL WESLEY B
8415 TALLERICO LOUIE N
8431 RADOSEVICH LEE G
8431 EICKER PATRICK J
8431 BATTLESON KIRK W
8441 SCHAFER CLIFFORD T
8441 BYROADS WILLIAM T JR
8441 BORELLO LAWRENCE A
8441 NG RAYMOND
8441 WEST ANTON J
8442 HARTWIG CHARLES M
8442 WHEELER JOHN C
8442 BEHRENS RICHARD JR
8442 HINCKLEY C MARTIN
8442 CAROTHERS KAREN A
8444 WILLIS ALEC R
8444 OIEN CHARLES T
8445 BOLEN JACK L
8445 HANSER HENRY
8445 PUTZ DONALD W
8445 DERICKSON LINN W
8452 BAKER ALVIN F
8452 RORKE WILLIAM S JR
8452 FAAS SCOTT E
8452 FALCONE PATRICIA K
8453 MAVIS CLAYTON L
8453 NORRIS HAROLD F JR
8453 WILSON WM G
8453 DELAMETER WILLIAM R
8453 DAWSON DANIEL B
8453 DELAQUILL PASCAL III
8453 YANG CHRISTINE L
8461 TOCKEY ROBERT J
8465 POTTHOFF C M
8466 BARR VERNON C
8466 STIMMELL KATHLEEN
8500 HARTLEY D M
8510 MATTERN PETER L
8511 SCHMIEDER ROBERT W
8511 CATTOLICA ROBERT J
8511 WERTENBERG RUSSELL F
8511 FLOWER WILLIAM L
8511 STEPHENSON DAVID A
8512 BENTHUSON D E
8512 MARION JOHN E
8512 LIBKIND MARCUS A
8513 MILLER JAMES A
8520 ROBINSON CLARENCE W
8521 TICHENOR DANIEL A
8521 MITCHELL REGINALD E
8521 WANG JAMES C F

8522 WITZE PETER O
8522 SMITH JAMES R
8522 GREEN ROBERT M
8522 JOHNSTON SHERIDAN C
8522 KEETON STEWART C
8522 WOODARD JOAN BRUNE
8522 RAMBACH GLENN D
8523 MARX KENNETH D
8523 ASHURST WILLIAM T
8523 SANDERS BILLY R
8523 KERSTEIN ALAN R
8523 DIBBLE ROBERT W
8523 BARR PAMELA K
PUBLICATIONS DIVISION, 8265/TECHNICAL LIBRARY PROCESSES DIVISION, 3141
TECHNICAL LIBRARY PROCESSES DIVISION, 3141 (3)
M. A. POUND, 8214, FOR CENTRAL TECHNICAL FILES (3)

Org.	Bldg.	Name	Rec'd by	Org.	Bldg.	Name	Rec'd by

Sandia National Laboratories