数 値 解 析 24－3
（１９８8. 3. １１）

Numerical Recipes と Example Book について

野寺 隆

慶應義塾大学理工学部数理科学科

近年，数学ソフトウエアは，その信頼性の向上とソフトウエア・パッケージの低価格化によって，数値計算の標準的なトゥールになりつつある。
今回，Numerical Recipes and Example bookについて外観することにするが，これは単に，アルゴリズムの観点からだけではなく，ソフトウエアの観点にも立って利用することをデザインされたものである。また，従来，メインフレーム計算機において利用可能であった多くの科学技術計算用のコードや手法が，パーソナル計算機でも利用できるようになった１例である。

Numerical Recipes and Example Book

Takashi NODERA
Department of Mathematics
Faculty of Science and Engineeing
Keio University
3-14-1 Hiyoshi Kohoku Yokohama 223
JAPAN

Mathematical Software has become a standard tool of numerical computation because of the greatly improved reliability and dramatically reduced costs of pakeged software.
In this talk, we briefly outline the software of "Numerical Recipes" and its "example book". Numerical Recipes are desinged to use not only in the algorithm sense but also in the software sense. Moreover, it is one of the examples that these codes and procedures in this software can be used on personal computers, bringing to these machines a capability for scientific computing previous available only in large mainframe computers.

1．はじめに

ほんの数年前といっても，一世代前といってもよいかもしれないが，数値計算の分野で数学的ソフトウエア（Mathematical Software)と言う言葉が持て囃された時代があった。この言葉は，当時，アメリカ，Purdue大学のJohn. R. Rice 教授が数回にわたって同名のシンポジウムを行って，数学ソフトウエアの開発，品質向上，利用促進を促したものであった。その内容は，基本アルゴリズムの再検討，ソフトウエアの評価方法，ソフトウエアの標準化，etc におよぶものであり，究極の目的は，利用者がいちいちこまかい指示を与えなくても，コンピュータが問題に応じて適切な解法を選択して，ある程度自動的に解を算出してくれることをめざすものであった。即ち，自動車で言うなれば，ソフトウエアのオートマ車を目指していた。当時，米国では，NSF の援助を得て様々な数値計算を行うソフトウエアの開発がおこなわれていた。とくに，BLAS，EISPACK や LINPACK に代表される基本数値計算向きのソフトウエアだけでなく，偏微分方程式の境界値問題を解くためのITPACKやElPACKの開発も行われていた。当時，わが国においても UMS(Universal Mathematical Software)と言う汎用数値計算パッケイジが開発されていたのであるが，残念ながらこの名前を知る人は少ないように思う。これらのソフトウエアは言うなれば，メインフレームのコンピュータを中心に設計されたソフトウエアであり，多少，小回りが効きにくい点もなきにしもあらずであった。そうそこしているうちに，段々と世の中から数学的ソフトウエアと言う言葉が聞かれなくなり，現在では皆無といってよいのではないかと思うのである。とはいっても，数学的ソフトウエアと言う名が無くなったのではなく，言うなれば一つの転換点にきているといってよいのではないか。

数値計算は，計算機の応用分野の中でもっとも早く世の中に現れた分野であり，計算機科学全体から見ると，多少，異質なFortran 文化と言うことができる。Fortran で記述され現在までに開発されたソフトウエアは，大切な文化遺産であるが，その反面，新天地の開発を妨げる傾向もなきにしもあらずではないか。

近年，非常に驚かされることの１つに，ファミコンや日本語ポータブルワープロの普及が上げられるが，それと同様に，NEC のPC-9800 や AppleのMacintosh の普及も目覚ましいものがある。通常，これらの小型計算機は，パソコンと呼ばれているのだが，ゲームをするだけでなく，いろいろな利用価値がある。一昔前のパソコンと言えば，アセンブラやBASIC 言語しか利用できないものだったが，現在では，立派に Fortran77 やTurbo-Pascal，さらに，C-言語まで利用できるのである。これらのパソコンの普及の原因の１つには，ハードウエアの低価格化が上げられる。それと同様に，ほんの数年前まではソフトウエアは非常に高額でなかなか手のとどくものではなかったが，最近，驚くほどの低価格のものが現れはじめたのである。いうなれば，品質の良い安いソフトウエアが出回りだしたと言っても過言ではなかろう。

1986 年の初頭に，Numerical Recpies(The Art of Scientific Computing, ISBN 0-521-30811-9) と言う本が，Cambrige University Press から発行された。これは，848 ページにおよぶ厚さ6cm 程度の本で，なんとなく数値計算に関するクッキングブックと言ったものであり，数値計算に関連する解法の大まかなもの全てを含んでいるのである。ただし，このクッキングブックの料理メニューは重厚なフランス料理とは，言え

ないかもしれない。単刀直入に言えば，英国の家庭料理程度のメニューで，その味は多少大まかではないかと思う。しかし，晩餐会程度の料理メニューも用意されていないわけではない。また，この料理の本には，料理のインストラクターとしてのソフトウエア(Numerical Recipies V1.0 Software) ばかりでなく，簡単なお料理の例題をこなせる例題集(Example Book, ISBN 0-521-30956-5, and Example Software)まで発売されているのである。この本の驚くべき特徴の１つはそのソフトウエアがびっくりするくらいの低価格(£15.00)で販売されていることである。ソフトウエア自身の値段は，ほぼ１枚のディスケットの正価程度で，別の言葉で言えば，教科書１冊程度の値段である。

近年，筆者はこのソフトウエアと例題集を入手し，このソフトウエアの特徴を色々分析したので報告する。

2. Numerical Recipes

Numerical Recipes(The Art of Scientific Computing) は，

William H. Press (Harvard-Smithsonian Center for Astrophysics)

Brian P. Flannery (EXXON Research and Engineering Company)

Saul A. Teukolsky (Dept. of Physics, Cornell University)

William T. Vetterling (Polaroid Corporation)

の４人によって書かれた数値計算のためのクッキング・ブックで，1986年の初頭にCambridge University Pressから発売された。また，そのソフトウエアも数箇月遅れて発売された。その内容は，特殊な数値解析に関する数値計算ではなく，ごく標準的な数値解析を勉強したことのある人ならだれでも知っていなければならない基本的な算法だけでなく，基本的な統計解析などについても言及していると思う。特に，前者に関連するものとして，線形計算，固有値問題，関数近似(内挿，外挿など)， 数値積分，非線形代数方程式の解法，常微分方程式，偏微分方程式の数値解法の入門などが上げられが，これらのトピックの中には，当然，標準的なレベル以上の，言うなれば，最先端の話題も含まれている。そのハイライトは次のようなものである。

(1)特殊な線形方程式の解法 (Vandermonde 行列，Toeprlitz 行列，疎行列の解法など)
(2)有理関数の内挿，外挿など
(3)特異点をもつ数値積分
(4)適応的にステップサイズをコントロールする微分方程式の Bulirsh-Stoer の数値積分

次に，一般的な数値解析の教科書には書かれていない，統計解析などを含むいくつかのトピックについてのべると次のようになる。即ち，特殊関数の評価，乱数，Monte Carlo 法，ソーティング，最適化法 (multidimensional法を含む)，Fourier変換(FFT 及

び maximum entropy法を含む)， 統計解析に関するもの，データのモデル化， 2点境界値問題 (shooting法と緩和法を含む) などが上げられる。これらのトピックのハイライトは，次のものが上げられる。

(1) working routine for statistical probability function
(2) Bessel functions and modified Bessel function
(3) random deviates from gamma, Poisson, and binomial distribution
(4) the DATA Encryption Standard
(5) sorting for determining equivalence classes of data
(6) a complete implementation of linear programming
(7) an introduction to annealing methods of optimization
(8) routines for real Fourier transformations, for sine and cosine transformation
(9) linear prediction and linear prediction cording
(10) digital filtering
(11) entropy measures of statistical dependency
(12) robust statistical fitting

“Numerical Recipes” では，これらのトピックを取り扱う算法がANSI-Standard FORTRAN-77とPASCALで記述されたものが掲載さており，単に，算法の説明書というだけでなく，そのマニュアルも兼ね備えていると言える (C－言語版は，後日発売される見込みである) 。またプログラムは，できるだけ構造(structured programming)を踏まえて書いてある。

3. Numerical Recipes Software

現在，Numerical Recipes のソフトウエアは，Fortran 及びPascalで書かれたものがあり，様々なコンピュータ，即ち，メインフレイムに代表されるマルチユーザーコンピュータだけでなく，パソコンでも使用可能なように設計されている。主に，DEC VAX-11/780 (o.s. VMS) とIBM-PCまたはXT(PC-DOS) 上で動いているのだが，NEC PC-9800 および Aplle Macintosh上でも使用可能である。提供されているソフトウエアは次の通りである。

(1) Numerical Recipes Pascal Diskette V 1.0
[Pascal procedures as listed in “Numerical Recipes”
in machine-readable form]
(2) Numerical Recipes Fortran Diskette V 1.0
[Fortran procedures as listed in “Numerical Recipes”
in machine readable form]

(3) Numerical Recipes UCSD p-System Diskette (UCSD Pascal) V 1.0
[Pascal procedures from "Numerical Recipes" for IBM/PC
using the UCSD p-system operating system]

(4) Numerical Recipes Macintosh Diskette (Pascal) V 1.0
[Pascal procedures from "Numerical Recipes" for Apple Macintosh]

(5) Numerical Recipes Object File Diskette (FORTRAN)
[FORTRAN subroutines from "Numerical Recipes" as .OBJ files]

(6) Numerical Recipes VAX INSTALLATION MAGNETIC TAPE (FORTRAN)
[FORTRAN subroutines from "Numerical Recipes" for multiuser
installation on DEC VAX]

このソフトウエアには，Numerical Recipes に記載されているすべてのアルゴリズム203個のプログラム・コードとして含まれている。さらに，これらのソフトウエアを有効に使用し評価するために，次のような基本的な例題集のディスケットもある。

(1) Numerical Recipes Example Diskette (Pascal)
[demonstration program in the Pascal language as listed in
"Numerical Recipes Example Book (Pascal)" in machine readable form]

(2) Numerical Recipes Example Diskette (FORTRAN)
[demonstration program in the FORTRAN language as listed in
"Numerical Recipes Example Book (FORTRAN)" in machine readable form]

例題デスケットには，Numerical Recipes ソフトウエアを使って仕事をする189 のコードが含まれている。このうち，154 個がNumerical Recipes に対応する個々のコードで残り35個のコードは5つの"omunibus"ファイルから構成されている。さらにこのディスケットには，デモンストレーション用に使用される14個のデータファイルも含まれている。

FORTRAN プログラムは，VAX-11では，ほぼ修正なしで動き，IBM PCでは，version 3.2 以上の Microsoft FORTRANで動く。

Pascalにおいては，Borland International のTurbo Pascalでも，DEC VAX 上のVAX-11 Pascal でも，他の多くのコンピュータ上で使用可能なUCSD Pascal においても，さらにIBM のメインフレイム上のPascal/VS でも実行可能である。

これらのソフトウエアは，下記のソフトウエア会社で製作され，ソフトウエアについての技術的な質問やプログラムのバグの修正についても，下記に連絡をとる体制がとられている。また，現在，C－言語版が製作途上にあるらしい。

Numerical Recipes Software
P.O. Box 243, Cambrige, MA 02238, U.S.A.

ソフトウエアの値段は，Numerical Recipes ディスケットおよび Exampleディスケットとともに，英国では£15.00, 米国では$19.95 でCambridge University Pressから発売されており，下記の住所から直接取り寄せることもできる。

[英国] Edinburgh Building
Shaftesbury Road,
Cambridge CB2 2RU, U.K.

[米国] 510 North Ave.
New Rochelle,
New York 10801, U.S.A.

4. 数值例

(例 1.) ニュートン法

〔問題〕

$$-x_1^2 - x_2^2 - x_3^2 + x_4 = 0$$
$$x_1^2 + x_2^2 + x_3^2 + x_4^2 - 1 = 0$$
$$x_1 - x_2 = 0$$
$$x_2 - x_3 = 0$$

```
PROGRAM d9r13(input,output);
(* driver for routine MNEWT *)
CONST
   ntrial=5;
   tolx=1.0e-6;
   n=4;
   np=n;
   tolf=1.0e-6;
TYPE
   glnarray = ARRAY [1..n] OF real;
   glnbyn = ARRAY [1..n,1..n] OF real;
   glindx = ARRAY [1..n] OF integer;
   glnpbynp = glnbyn;
VAR
   i,j,k,kk : integer;
   xx : real;
   x,beta : glnarray;
   alpha : glnbyn;

PROCEDURE usrfun(x: glnarray; n: integer; VAR alpha: glnbyn;
        VAR beta: glnarray);
(* Programs using routine USRFUN must define the types
TYPE
   glnarray = ARRAY [1..n] OF real;
   glnbyn = ARRAY [1..n,1..n] OF real;
in the main routine. *)
BEGIN
   alpha[1,1] := -2.0*x[1];
   alpha[1,2] := -2.0*x[2];
   alpha[1,3] := -2.0*x[3];
   alpha[1,4] := 1.0;
   alpha[2,1] := 2.0*x[1];
   alpha[2,2] := 2.0*x[2];
   alpha[2,3] := 2.0*x[3];
   alpha[2,4] := 2.0*x[4];
   alpha[3,1] := 1.0;
   alpha[3,2] := -1.0;
   alpha[3,3] := 0.0;
   alpha[3,4] := 0.0;
   alpha[4,1] := 0.0;
   alpha[4,2] := 1.0;
   alpha[4,3] := -1.0;
   alpha[4,4] := 0.0;
   beta[1]  := sqr(x[1])+sqr(x[2])+sqr(x[3])-x[4];
   beta[2]  := -sqr(x[1])-sqr(x[2])-sqr(x[3])-sqr(x[4])+1.0;
   beta[3]  := -x[1]+x[2];
   beta[4]  := -x[2]+x[3]
END;

(*$I MODFILE.PAS *)
(*$I ¥PAS¥LUBKSB.PAS *)

(*$I ¥PAS¥LUDCMP.PAS *)

(*$I ¥PAS¥MNEWT.PAS *)

BEGIN
   FOR kk := 1 to 2 DO BEGIN
      FOR k := 1 to 3 DO BEGIN
         xx := 0.2*k*(2*kk-3);
         writeln('Starting vector number',k:2);
         FOR i := 1 to 4 DO BEGIN
            x[i] := xx+0.2*i;
            writeln('x[':7,i:1,']   :=   ',x[i]:5:2)
         END;
         writeln;
         FOR j := 1 to ntrial DO BEGIN
            mnewt(1,x,n,tolx,tolf);
            usrfun(x,n,alpha,beta);
            writeln('i':5,'x[i]':13,'f':13);
            FOR i := 1 to n DO BEGIN
               writeln(i:5,x[i]:14:6,-beta[i]:15:6)
            END;
            writeln;
            writeln('press RETURN to continue...');
            readln
         END
      END
   END
END.
```

〔実行結果〕

```
Starting vector number 1
     x[1]   :=    0.00
     x[2]   :=    0.20
     x[3]   :=    0.40
     x[4]   :=    0.60

    i         x[i]            f
    1       0.681818     -0.776446
    2       0.681818      0.776777
    3       0.681818      0.000000
    4       0.618182      0.000000

press RETURN to continue...

    i         x[i]            f
    1       0.491984     -0.108111
    2       0.491984      0.108111
    3       0.491984      0.000000
    4       0.618034      0.000000

press RETURN to continue...

    i         x[i]            f
    1       0.455360     -0.004024
    2       0.455360      0.004024
    3       0.455360      0.000000
    4       0.618034      0.000000

press RETURN to continue...

    i         x[i]            f
    1       0.453887     -0.000007
    2       0.453887      0.000007
    3       0.453887      0.000000
    4       0.618034      0.000000

press RETURN to continue...

    i         x[i]            f
    1       0.453885     -0.000000
    2       0.453885      0.000000
    3       0.453885      0.000000
    4       0.618034      0.000000
```

(例 2.) ADI 法

〔問題〕

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \rho$$

$$A_{jl} = C_{jl} = D_{jl} = F_{jl} = -1.0$$
$$B_{jl} = E_{jl} = 2.0$$

$$\text{alpha} = 2\left[1 - \cos\left(\frac{\pi}{\text{jmax}}\right)\right]$$
$$\text{beta} = 2\left[1 - \cos\frac{(\text{jmax}-1)\pi}{\text{jmax}}\right]$$

```
PROGRAM d17r2(input,output);
(* driver for routine ADI *)
LABEL 1;
CONST
   jmax=11;
   pi=3.1415926;
TYPE
   double = real;
   gljmax = ARRAY [1..jmax,1..jmax] OF double;
VAR
   alim,alpha,beta,eps : double;
   i,j,k,mid,twotok : integer;
   a,b,c,d,e,f,g,u : gljmax;

(*$I ¥pas¥ADI.PAS *)

BEGIN
   FOR i := 1 to jmax DO BEGIN
      FOR j := 1 to jmax DO BEGIN
         a[i,j] := -1.0;
         b[i,j] := 2.0;
         c[i,j] := -1.0;
         d[i,j] := -1.0;
         e[i,j] := 2.0;
         f[i,j] := -1.0;
         g[i,j] := 0.0;
         u[i,j] := 0.0
      END
   END;
   mid := (jmax DIV 2)+1;
   g[mid,mid] := 2.0;
   alpha := 2.0*(1.0-cos(pi/jmax));
   beta := 2.0*(1.0-cos((jmax-1)*pi/jmax));
   alim := ln(4.0*jmax/pi);
   k := 0;
   twotok := 1;
   REPEAT
      k := k+1;
      twotok := 2*twotok;
   UNTIL twotok >= alim;
   eps := 1.0e-4;
   adi(a,b,c,d,e,f,g,u,jmax,k,alpha,beta,eps);
   writeln('ADI Solution:');
   FOR i := 1 to jmax DO BEGIN
      FOR j := 1 to jmax DO write(u[i,j]:7:2);
      writeln
   END;
   writeln;
   writeln('Test that solution satisfies difference eqns:');
   FOR i := 2 to (jmax-1) DO BEGIN
      FOR j := 2 to (jmax-1) DO BEGIN
         g[i,j] := -4.0*u[i,j]+u[i+1,j]
            +u[i-1,j]+u[i,j-1]+u[i,j+1]
      END;
      write(' ':7);
      FOR j := 2 to (jmax-1) DO write(g[i,j]:7:2);
      writeln
   END
END.
```

〔実行結果〕

```
ADI Solution:
   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
   0.00  -0.02  -0.04  -0.06  -0.08  -0.09  -0.08  -0.06  -0.04  -0.02   0.00
   0.00  -0.04  -0.09  -0.13  -0.17  -0.19  -0.17  -0.13  -0.09  -0.04   0.00
   0.00  -0.06  -0.13  -0.20  -0.28  -0.32  -0.28  -0.20  -0.13  -0.06   0.00
   0.00  -0.08  -0.17  -0.28  -0.41  -0.55  -0.41  -0.28  -0.17  -0.08   0.00
   0.00  -0.09  -0.19  -0.32  -0.55  -1.05  -0.55  -0.32  -0.19  -0.09   0.00
   0.00  -0.08  -0.17  -0.28  -0.41  -0.55  -0.41  -0.28  -0.17  -0.08   0.00
   0.00  -0.06  -0.13  -0.20  -0.28  -0.32  -0.28  -0.20  -0.13  -0.06   0.00
   0.00  -0.04  -0.09  -0.13  -0.17  -0.19  -0.17  -0.13  -0.09  -0.04   0.00
   0.00  -0.02  -0.04  -0.06  -0.08  -0.09  -0.08  -0.06  -0.04  -0.02   0.00
   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00

Test that solution satisfies difference eqns:
         -0.00   0.00   0.00   0.00  -0.00   0.00   0.00   0.00  -0.00
          0.00  -0.00  -0.00  -0.00   0.00  -0.00  -0.00  -0.00   0.00
          0.00  -0.00  -0.00  -0.00   0.00  -0.00  -0.00  -0.00   0.00
          0.00  -0.00  -0.00  -0.00   0.00  -0.00  -0.00  -0.00   0.00
         -0.00   0.00   0.00   0.00   2.00   0.00   0.00   0.00  -0.00
          0.00  -0.00  -0.00  -0.00   0.00  -0.00  -0.00  -0.00   0.00
          0.00  -0.00  -0.00  -0.00   0.00  -0.00  -0.00  -0.00   0.00
          0.00  -0.00  -0.00  -0.00   0.00  -0.00  -0.00  -0.00   0.00
         -0.00   0.00   0.00   0.00  -0.00   0.00   0.00   0.00  -0.00
```

5. おわりに

Numerical Recipesとそのソフトウエアについて述べてきたが，内容的に見てみると解法のアルゴリズムの点で，多少，偏りはあるものの，その低価格性を見ると，かなり良くできたソフトウエアと言える。ただし，筆者がNEC PC-9800 上のTurbo-Pascalで全てのコードを試したところ全部O.K.というわけでなく，数個のコードはそのままでは動かず，多少の手直しを必要とした。

参考文献

[1] W.T.Vettering et al., Numerical Recipes, Cambrige University Press, (1986).

[2] W.T.Vettering et al., Numerical Recipes Example Book(Pascal), Cambrige Cambrige University Press, (1986).

[3] Numerical Recipes Software, Numerical Recipes Pascal Diskette V1.0, Cambrige University Press, (1986).

[4] Numerical Recipes Software, Numerical Recipes Example Diskette V1.0, Cambrige University Press, (1986).

[5] Turbo Pascal User's Guide and Refference Manual, Borland, 1986.

FLMOON	phases of the moon, calculated by date
JULDAY	Julian day number, calculated by date
BADLUK	Friday the 13th when the moon is full
CALDAT	calendar date, calculated from Julian day number
GAUSSJ	matrix inversion and linear equation solution, Gauss-Jordan
LUDCMP	linear equation solution, *LU* decomposition
LUBKSB	linear equation solution, backsubstitution
TRIDAG	linear equation solution, tridiagonal equations
MPROVE	linear equation solution, iterative improvement
VANDER	linear equation solution, Vandermonde matrices
TOEPLZ	linear equation solution, Toeplitz matrices
SVDCMP	singular value decomposition of a matrix
SVBKSB	singular value backsubstitution
SPARSE	linear equation solution, sparse matrix, conjugate-gradient method
POLINT	interpolation, polynomial
RATINT	interpolation, rational function
SPLINE	interpolation, construct a cubic spline
SPLINT	interpolation, evaluate a cubic spline
LOCATE	search an ordered table, bisection
HUNT	search an ordered table, correlated calls
POLCOE	polynomial coefficients from a table of values
POLCOF	polynomial coefficients from a table of values
POLIN2	interpolation, two-dimensional polynomial
BCUCOF	interpolation, two-dimensional, construct bicubic
BCUINT	interpolation, two-dimensional, evaluate bicubic
SPLIE2	interpolation, two-dimensional, construct two-dimensional spline
SPLIN2	interpolation, two-dimensional, evaluate two-dimensional spline
TRAPZD	integrate a function by trapezoidal rule
QTRAP	integrate a function to desired accuracy, trapezoidal rule
QSIMP	integrate a function to desired accuracy, Simpson's rule
QROMB	integrate a function to desired accuracy, Romberg adaptive method
MIDPNT	integrate a function by extended midpoint rule
QROMO	integrate a function to desired accuracy, open Romberg
MIDINF	integrate a function on a semi-infinite interval
MIDSQL	integrate a function with a square-root singularity
MIDSQU	integrate a function with an inverse square-root singularity
MIDEXP	integrate a function which decreases exponentially
QGAUS	integrate a function by Gaussian quadratures
GAULEG	Gauss-Legendre weights and abscissas, compute
QUAD3D	integrate a function over a three-dimensional space
EULSUM	sum a series, Euler-van Wijngaarden algorithm
DDPOLY	polynomial, fast evaluation of specified derivatives
POLDIV	polynomials, divide one by another
CHEBFT	fit a Chebyshev polynomial to a function
CHEBEV	Chebyshev polynomial evaluation
CHINT	integrate a function already Chebyshev fitted
CHDER	derivative of a function already Chebyshev fitted
CHEBPC	polynomial coefficients from a Chebyshev fit
PCSHFT	polynomial coefficients of a shifted polynomial
GAMMLN	logarithm of gamma function
FACTRL	factorial function
BICO	binomial coefficients function
FACTLN	factorial function, logarithm
BETA	beta function
GAMMP	gamma function, incomplete
GAMMQ	gamma function, incomplete, complementary
GSER	gamma function, incomplete, series evaluation
GCF	gamma function, incomplete, continued fraction evaluation
ERF	error function
ERFC	error function, complementary
ERFCC	error function, complementary, concise routine
BETAI	beta function, incomplete
BETACF	beta function, incomplete, continued fraction evaluation
BESSJ0	Bessel function J_0
BESSY0	Bessel function Y_0
BESSJ1	Bessel function J_1
BESSY1	Bessel function Y_1
BESSJ	Bessel function J of integer order
BESSY	Bessel function Y of integer order
BESSI0	Modified Bessel function I_0
BESSK0	Modified Bessel function K_0
BESSI1	Modified Bessel function I_1
BESSK1	Modified Bessel function K_1
BESSI	Modified Bessel function I of integer order
BESSK	Modified Bessel function K of integer order
PLGNDR	Legendre polynomials, associated (spherical harmonics)
EL2	Elliptic integral of the first and second kinds
CEL	Elliptic integrals, complete, all three kinds
SNCNDN	Jacobian elliptic functions
RAN0	random deviates, improve an existing generator
RAN1	random deviates, uniform
RAN2	random deviates, uniform
RAN3	random deviates, uniform, subtractive method
EXPDEV	random deviates, exponential
GASDEV	random deviates, normally distributed (Box-Muller)
GAMDEV	random deviates, gamma-law distribution
POIDEV	random deviates, Poisson distributed
BNLDEV	random deviates, binomial distributed
IRBIT1	random bit sequence, generate
IRBIT2	random bit sequence, generate
RAN4	random deviates, uniform, using Data Encryption Standard
DES	encryption, using the Data Encryption Standard
KS	encryption, key schedule for Data Encryption Standard
CYFUN	encryption, cipher function for Data Encryption Standard
PIKSRT	sort an array by straight insertion
PIKSR2	sort two arrays by straight insertion
SHELL	sort an array by Shell's method
SORT	sort an array by heapsort method
SORT2	sort two arrays by heapsort method
INDEXX	sort, construct an index for an array
SORT3	sort, use an index to sort 3 or more arrays
RANK	sort, construct a rank table for an array
QCKSRT	sort an array by quicksort method
ECLASS	determine equivalence classes
ECLAZZ	determine equivalence classes
SCRSHO	graph a function to search for roots
ZBRAC	roots of a function, search for brackets on
ZBRAK	roots of a function, search for brackets on
RTBIS	root of a function, find by bisection
RTFLSP	root of a function, find by false-position
RTSEC	root of a function, find by secant method
ZBRENT	root of a function, find by Brent's method
RTNEWT	root of a function, find by Newton-Raphson
RTSAFE	root of a function, find by Newton-Raphson and bisection
LAGUER	root of a polynomial, Laguerre's method
ZROOTS	roots of a polynomial, Laguerre's method with deflation
QROOT	root of a polynomial, complex or double, Bairstow
MNEWT	nonlinear systems of equations, Newton-Raphson

MNBRAK	minimum of a function, bracket
GOLDEN	minimum of a function, find by golden section search
BRENT	minimum of a function, find by Brent's method
DBRENT	minimum of a function, find using derivative information
AMOEBA	minimum of a function, multidimensions, downhill-simplex
POWELL	minimum of a function, multidimensions, Powell's method
LINMIN	minimum of a function, along a ray in multidimensions
F1DIM	minimum of a function, used by LINMIN
FRPRMN	minimum of a function, multidimensions, conjugate-gradient
DF1DIM	minimum of a function, used by LINMIN
DFPMIN	minimum of a function, multidimensions, variable metric
SIMPLX	linear programming maximization of a linear function
SIMP1	linear programming, used by SIMPLX
SIMP2	linear programming, used by SIMPLX
SIMP3	linear programming, used by SIMPLX
ANNEAL	minimize by simulated annealing (traveling salesman problem)

JACOBI	eigenvalues and eigenvectors of a symmetric matrix
EIGSRT	eigenvectors, sorts into order by eigenvalue
TRED2	Householder reduction of a real, symmetric matrix
TQLI	eigenvalues and eigenvectors of a symmetric tridiagonal matrix
BALANC	balance a nonsymmetric matrix
ELMHES	Hessenberg form, reduce a general matrix to
HQR	eigenvalues of a Hessenberg matrix

FOUR1	Fourier transform (FFT) in one dimension
TWOFFT	Fourier transform of two real functions
REALFT	Fourier transform of a single real function
SINFT	sine transform using FFT
COSFT	cosine transform using FFT
CONVLV	convolution or deconvolution of data using FFT
CORREL	correlation or autocorrelation of data using FFT
SPCTRM	power spectrum estimation using FFT

MEMCOF	power spectrum estimation, evaluate maximum entropy coefficients
EVLMEM	power spectrum estimation using maximum entropy coefficients
FIXRTS	roots of a polynomial, reflects inside unit circle
PREDIC	linear prediction using MEM coefficients
FOURN	Fourier transform (FFT) in multidimensions

MOMENT	moments of a data set, calculate
MDIAN1	median of a data set, calculate by sorting
MDIAN2	median of a data set, calculate iteratively
TTEST	Student's t-test for difference of means
AVEVAR	mean and variance of a data set, calculate
TUTEST	Student's t-test for means, with unequal variances
TPTEST	Student's t-test for means, with paired data
FTEST	F-test for difference of variances
CHSONE	chi-square test for difference between data and model
CHSTWO	chi-square test for difference between two data sets
KSONE	Kolmogorov-Smirnov test of data against model
KSTWO	Kolmogorov-Smirnov test between two data sets
PROBKS	Kolmogorov-Smirnov probability function
CNTAB1	contingency table analysis using chi-square
CNTAB2	contingency table analysis using entropy measure
PEARSN	correlation between two data sets, Pearson's
SPEAR	correlation between two data sets, Spearman's rank
CRANK	rank, replaces array elements by their
KENDL1	correlation between two data sets, Kendall's tau
KENDL2	contingency table analysis using Kendall's tau
SMOOFT	smooth data using FFT

FIT	fit data to a straight line, least squares
LFIT	linear least squares fit, general, normal equations
COVSRT	covariance matrix, sort, used by LFIT
SVDFIT	linear least squares fit, general, singular value decomposition
SVDVAR	variances from singular value decomposition
FPOLY	fit a polynomial, using LFIT or SVDFIT
FLEG	fit a Legendre polynomial, using LFIT or SVDFIT
MRQMIN	nonlinear least squares fit, Marquardt's method
MRQCOF	nonlinear least squares fit, used by MRQMIN
FGAUSS	fit a sum of Gaussians, using MRQMIN
MEDFIT	fit data to a straight line robustly, least absolute deviation
ROFUNC	fit data robustly, used by MEDFIT

RK4	integrate one step of ODEs, 4th order Runge-Kutta
RKDUMB	integrate ODEs by 4th order Runge-Kutta
RKQC	integrate one step of ODEs with accuracy monitoring
ODEINT	integrate ODEs with accuracy monitoring
MMID	integrate ODEs by modified midpoint method
BSSTEP	integrate ODEs, Bulirsch-Stoer step
RZEXTR	rational function extrapolation, used by BSSTEP
PZEXTR	polynomial extrapolation, used by BSSTEP

SHOOT	two-point boundary value problem, solve by shooting
SHOOTF	two-point boundary value problem, shooting to a fitting point
SOLVDE	two-point boundary value problem, solve by relaxation
BKSUB	backsubstitution, used by SOLVDE
PINVS	diagonalize a sub-block, used by SOLVDE
RED	reduce columns of a matrix, used by SOLVDE
SFROID	spheroidal functions, obtain using SOLVDE
DIFEQ	spheroidal matrix coefficients, used by SFROID

SOR	elliptic PDE solved by simultaneous overrelaxation method
ADI	elliptic PDE solved by alternating direction implicit method