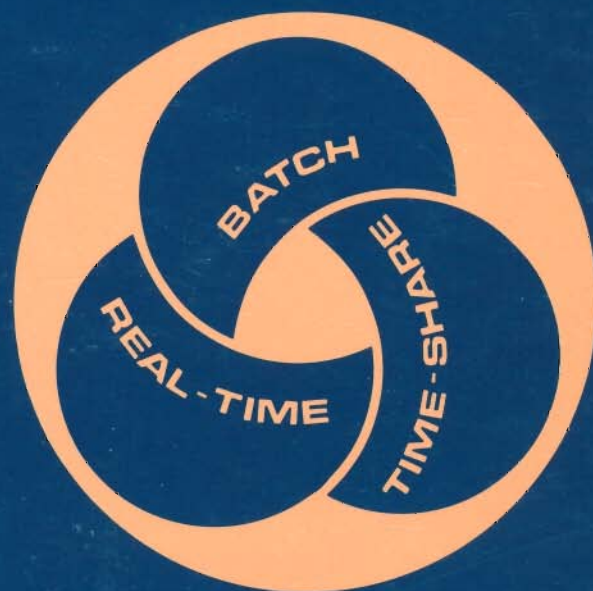


HP 3000

SCIENTIFIC LIBRARY



HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

HP 3000 SCIENTIFIC LIBRARY



03000-90010A

November 1972

List of Effective Pages

Pages	Effective Date
Title	Nov. 1972
Copyright	Nov. 1972
iii	Nov. 1972
v	Nov. 1972
vii to xiii	Nov. 1972
1-1 to 1-38	Nov. 1972
2-1 to 2-18	Nov. 1972
3-1	Nov. 1972

Printing History

Part No.	Date	Update Package	Date
03000-90010A	Nov. 1972		

PREFACE

HP 3000 Scientific Library (HP 03000-90010) is the programmer's reference to mathematical function, statistical, and related utility procedures available to users of Hewlett-Packard HP 3000 software.

The procedures in this library are provided for scientific mathematics users. The user must have a working knowledge of the programming language(s) used and the MPE/3000 Operating System or have access to the appropriate reference manuals listed below:

HP 3000 FORTRAN (HP 03000-90007)

HP 3000 Multiprogramming Executive Operating System (HP 03000-90005)

HP 3000 Systems Programming Language (HP 03000-90002)

HP 3000 supporting software, including the procedures in the Scientific Library, are written in Hewlett-Packard's System Programming Language (SPL/3000).

The Introduction to this manual defines the purpose of the Scientific Library and its relationship to the MPE/3000 Operating System and outlines the structural elements of this book. Section I contains mathematical function procedures, Section II describes statistical procedures, and Section III explains a utility procedure.

CONTENTS

PREFACE	iii
INTRODUCTION	vii
ORGANIZATION OF THIS BOOK	vii
Format for Procedures	vii
Text Conventions	viii
FUNCTION DIRECTORY	xi
ERROR FUNCTIONS	xi
GAMMA FUNCTIONS	xi
SPECIAL INTEGRALS	xii
SINGLE-PRECISION ELLIPTIC INTEGRALS	xii
EXTENDED-PRECISION ELLIPTIC INTEGRALS	xii
JACOBIAN ELLIPTIC FUNCTIONS	xii
BESSEL FUNCTIONS	xii
STATISTICS	xiii
UTILITY PROCEDURE	xiii
 SECTION I MATHEMATICAL FUNCTION PROCEDURES	 1-1
 SECTION II STATISTICAL PROCEDURES	 2-1
USER-WRITTEN DATA PROCEDURE	2-2
 SECTION III UTILITY PROCEDURE	 3-1

INTRODUCTION

The HP 3000 Scientific Library routines perform mathematical function, statistical and related utility computations for user programs. The HP 3000 Multiprogramming Executive Operating System (MPE/3000) links each user program to the Scientific Library routines needed.

Each routine is an SPL/3000 procedure (written in the HP 3000 Systems Programming Language, SPL/3000) and is an intrinsic (a system-provided procedure). Thus, an SPL/3000 user can include the name of each procedure to be used for a program in an intrinsic declaration (see *HP 3000 Systems Programming Language, 03000-90002*).

All of the Scientific Library procedures communicate only with the calling program. Thus, an error detected by any of these procedures is reported only by an error flag in the parameter list.

ORGANIZATION OF THIS BOOK

This book contains a function directory and three sections:

Section I: Mathematical Function Procedures

Section II: Statistical Procedures

Section III: Utility Procedure

Format for Procedures

Most of the procedures are described in a standard format. The following items are included in that format, when applicable:

NAME The procedure identifier.
(at top of page)

FUNCTION: Purpose of the procedure.

Declaration: The parts of the procedure declaration that define the requirements for actual parameters (arguments) included in a procedure call or calling sequence. Procedure declarations are defined in *HP 3000 Systems Programming Language (HP 03000-90002)*.

Definition: A comment on the algorithm for the procedure.

Accuracy: A statement of the procedure accuracy.

ATTRIBUTES:

Parameters:	The type(s) and ranges ¹ of value(s) allowed by the procedure. The SPL/3000 type LONG (real) is identical to the FORTRAN type DOUBLE PRECISION (real).
Result(s):	The type(s) and range ¹ of value(s).
Error(s):	A brief description of the error conditions.
COMMENTS:	When needed, special comments about programming considerations.
DISCUSSION:	When needed, elaborations of scientific characteristics.
EXAMPLES:	When needed, illustrations to support the descriptions.

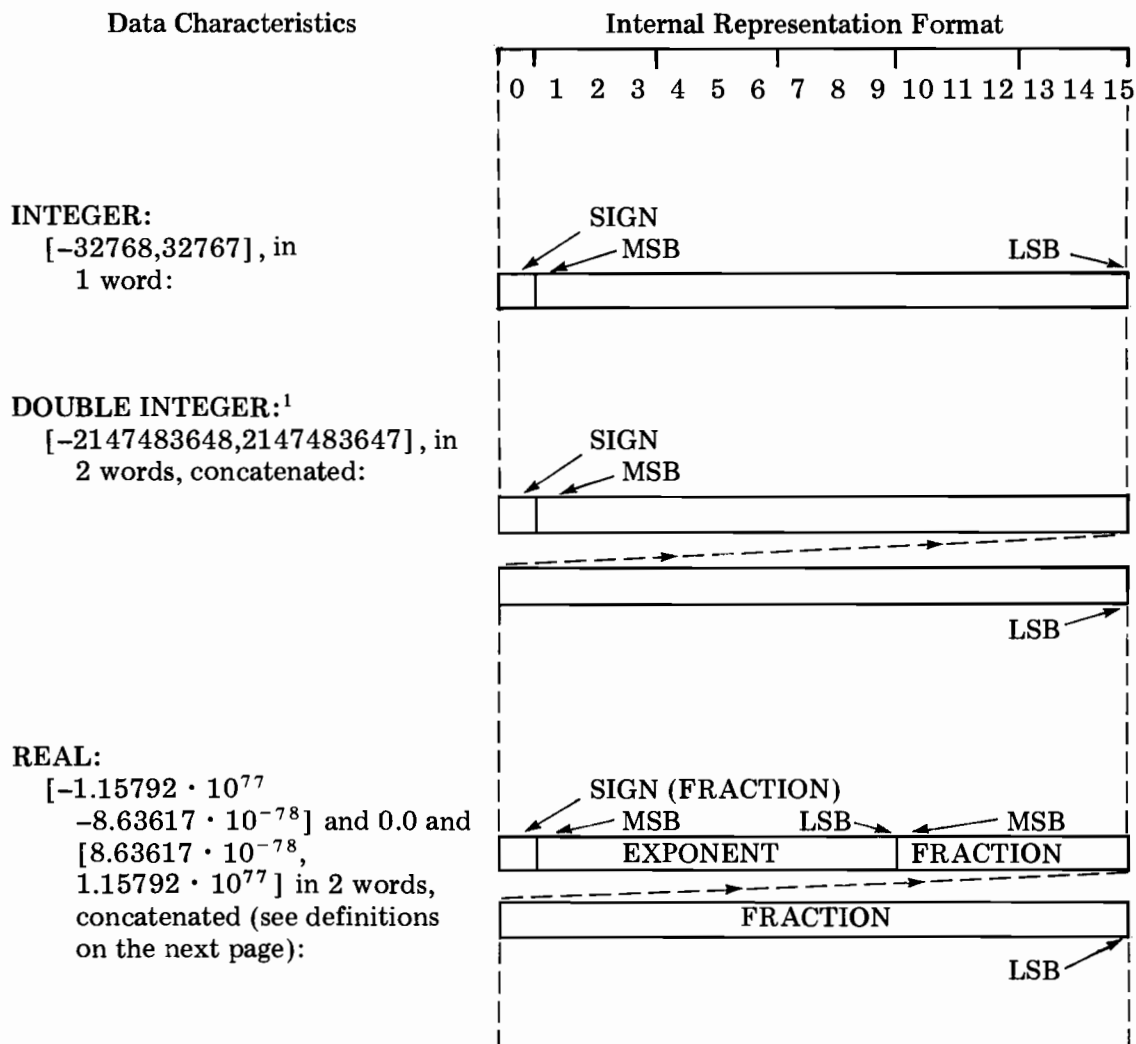
Text Conventions

The following conventions are used throughout the manual:

1. All numbers are decimal unless otherwise noted.
2. In all examples a blank space is represented by a delta Δ .
3. All appearances of the initials TOS refer to the top-of-stack, as defined in *HP 3000 Systems Programming Language (HP 03000-90002)*.
4. The notation $:=$ means "is replaced by."
5. Mathematical notation in the text include the following definitions:
 - A value x in the range (a, b) means $a < x < b$
 - A value x in the range $[a, b]$ means $a \leq x \leq b$
 - A value x in the range $(a, b]$ means $a < x \leq b$
 - A value x in the range $[a, b)$ means $a \leq x < b$

¹The range and form of internal representations are discussed in the text that follows.

6. All references to internal representations are based on the following (unless otherwise noted):



where MSB is the most significant bit and LSB is the least significant bit.

¹Not recognized in FORTRAN.

Data Characteristics

LONG:¹

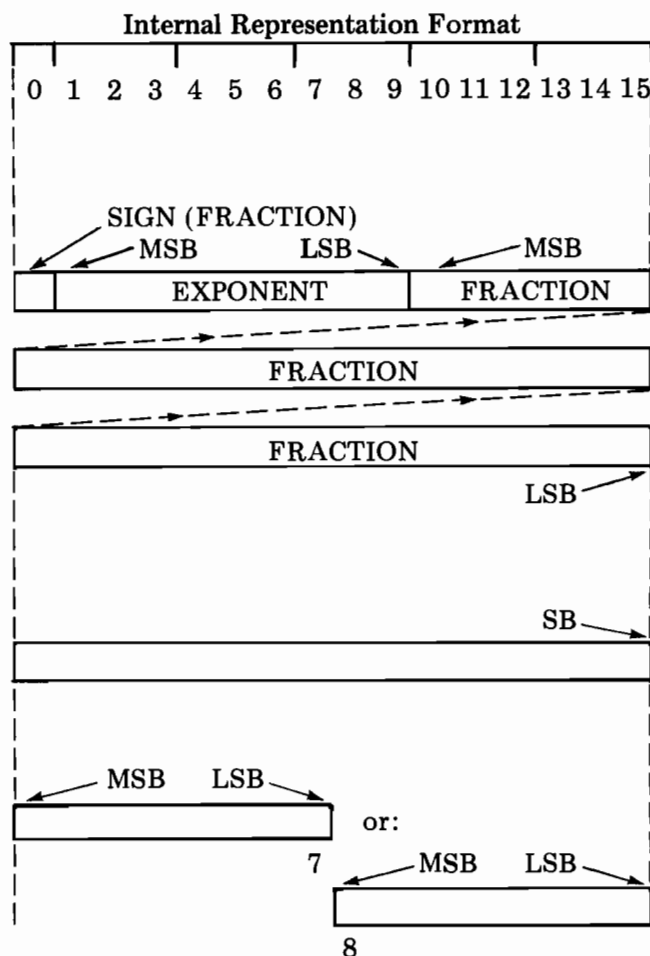
$[-1.1579208924 \cdot 10^{77}$,
 $-8.6361685551 \cdot 10^{-78}]$ and 0.0
 and $[8.6361685551 \cdot 10^{-78}$,
 $1.1579208924 \cdot 10^{77}]$ in 3 words,
 concatenated (see definitions
 below):

LOGICAL (Boolean):

[True (odd), False (even)], in
 1 word:

BYTE (ASCII character code):

[any alphameric], in
 $\frac{1}{2}$ word:



Definitions:

MSB = most significant bit

LSB = least significant bit

SB = significant bit (all others may be 0 or 1 for other uses)

SIGN = one bit for the sign of FRACTION, 0 for positive, 1 for negative

EXPONENT = $E = [0, 777_8] = [0, 511_{10}]$

FRACTION = $F = [0, 2^{22}-1]$ or $[0, 2^{38}-1]$

Real and double precision numbers are stored in normalized form with an implied "1." to the left of the FRACTION MSB. Thus, DECIMAL VALUE =

$$(-1)^S * 2^{E-256} * (1. + F * 2^{-22}) \quad \text{REAL}$$

$$(-1)^S * 2^{E-256} * (1. + F * 2^{-38}) \quad \text{LONG}^1$$

The exception occurs when $S = E = F = 0$; the decimal value is 0.0.

NOTE: When $E = 511_{10}$, $2^{E-256} = 2^{255}$.

¹In FORTRAN, type DOUBLE PRECISION (real).

FUNCTION DIRECTORY



The following list identifies each function provided in the HP 3000 Scientific Library and points to the descriptions in this manual. To simplify use of this list, the functions have been grouped by general categories. The grouping does not reflect the organization of this manual.

Function	Identifier	Page
Error Functions		
Calculate the error function of a single-precision ¹ number.	ERF	1-1
Calculate the complimentary error function of a single-precision ¹ number.	ERFC	1-2
Calculate the error function of an extended-precision ² number.	DERF	1-3
Calculate the complimentary error function of an extended-precision ² number.	DERFC	1-5
Gamma Functions		
Calculate the gamma function of a single-precision ¹ number.	GAMMA	1-6
Calculate the natural logarithm of the gamma function of a single-precision ¹ number.	LGAMMA	1-8
Calculate the gamma function of an extended-precision ² number.	DGAMMA	1-10
Calculate the natural logarithm of the gamma function of an extended-precision ² number.	DLGAMMA	1-12

¹In SPL/3000 and FORTRAN/3000, type REAL.

²In SPL/3000 type LONG; in FORTRAN/3000 type DOUBLE PRECISION.

Function	Identifier	Page
Special Integrals		
Calculate the exponential integral of a single-precision ¹ number.	EXPI	1-14
Calculate the sine integral and the cosine integral of a single-precision ¹ number.	SICI	1-15
Calculate the fresnel integrals of a single-precision ¹ number.	FRES	1-16
Single-precision¹ Elliptic Integrals		
Calculate the elliptic integral of the first kind.	EL1	1-17
Calculate the elliptic integral of the second kind.	EL2	1-18
Calculate the complete elliptic integral of the first kind.	CEL1	1-20
Calculate the complete elliptic integral of the second kind.	CEL2	1-21
Extended-precision² Elliptic Integrals		
Calculate the elliptic integral of the first kind.	DEL1	1-23
Calculate the elliptic integral of the second kind.	DEL2	1-25
Calculate the complete elliptic integral of the first kind.	DCEL1	1-27
Calculate the complete elliptic integral of the second kind.	DCEL2	1-29
Jacobian Elliptic Functions		
Calculate the jacobian elliptic functions sn, cn and dn using single-precision ¹ parameters.	JELF	1-31
Calculate the jacobian elliptic functions sn, cn and dn using extended-precision ¹ parameters.	DJELF	1-33
Bessel Functions		
Calculate the bessel function of the first kind of a single-precision ¹ number.	BESJ	1-35
Calculate the bessel function of the second kind of a single-precision ¹ number.	BESY	1-37

¹In SPL/3000 and FORTRAN/3000, type REAL.

²In SPL/3000 type LONG; in FORTRAN/3000 type DOUBLE PRECISION.

Function	Identifier	Page
Statistics		
Compute the statistical analyses: mean, standard deviation, standard error of the mean, variance, kurtosis, skewness, minimum, maximum and range.	STAT	2-5
Calculate product-moment correlation coefficients, means and standard deviations.	CORRELATE	2-8
Extract a subset of correlation coefficients and a vector of a dependent variable.	ARRANGE	2-12
Compute multiple linear regression coefficients and related analyses.	MULTLINR	2-14
Utility Procedure		
Invert a symmetric matrix stored in triangular form.	SMETINV	3-1

SECTION I
Mathematical Function Procedures

ERF

FUNCTION: Calculate the **error function** (also called the **gauss error function**) of a REAL number.

NOTE: For the complimentary error function, use the procedure *ERFC*.

Declaration: `PROCEDURE ERF(X, RESLT);
VALUE X;
REAL X, RESLT;
OPTION EXTERNAL;`

·
·
·

Computer
Museum

Definition: $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$

Accuracy: Depends on accuracy of x and EXP (see *HP 3000 Compiler Library (HP 03000-90009)*). When $|x| \geq 0.35$, absolute error in $\text{erf}(x)$ is less than absolute error in x . When $|x| < 0.35$, absolute error in x is amplified slightly.

ATTRIBUTES:

Parameters: X = a REAL number
 $RESLT$ = a REAL identifier

Result: $RESLT$, a REAL number.

Error: None.

COMMENTS:

1. To compute the gaussian probability integral $P(x)$, use the following transformation:

$$P(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-t^2/2} dt = \frac{1}{2} \left[2 + \text{erf} \left(\frac{x}{\sqrt{2}} \right) \right]$$

2. To compute the gaussian probability integral $A(x)$, use the following transformation:

$$A(x) = \frac{1}{\sqrt{2\pi}} \int_{-x}^x e^{-t^2/2} dt = \text{erf} \left(\frac{x}{\sqrt{2}} \right)$$

ERFC

FUNCTION: Calculate the **complimentary error function** of a REAL number.

Declaration: `PROCEDURE ERFC(X, RESLT);
VALUE X;
REAL X, RESLT;
OPTION EXTERNAL;
.
.
.`

Definition: $\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt$

Accuracy: See the error function procedure ERF.

ATTRIBUTES:

Parameters: X = a REAL number in the range [-10, 10].
RESLT = a REAL identifier

Result: RESLT, a REAL number (set to 0 if X is outside the range stated).

Error: None.

COMMENT: See Comments 1 and 2 for ERF.

DERF

FUNCTION: Calculate the **error function** (also called the **gauss error function**) of a **LONG**¹ (real) number, using extended precision arithmetic.

NOTE: For the complimentary error function, use the procedure *DERFC*.

Declaration: `PROCEDURE DERF(X, DRSLT);
VALUE X;
LONG X, DRSLT;
OPTION EXTERNAL;`

·
·
·

Definition: $\text{derf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$

Accuracy: Depends on the accuracy of x and DEXP (see *HP 3000 Compiler Library (HP 03000-90009)*). When $|x| \geq 0.35$, absolute error in $\text{erf}(x)$ is less than absolute error in x . When $|x| < 0.35$, absolute error in x is amplified slightly.

ATTRIBUTES:

Parameters: X = a LONG number
DRSLT = a LONG identifier

Result: DRSLT, a LONG number.

Error: None.

COMMENTS: 1. To compute the gaussian probability integral $P(x)$, use the following transformation:

$$P(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-t^2/2} dt = \frac{1}{2} \left[2 + \text{erf} \left(\frac{x}{\sqrt{2}} \right) \right]$$

¹In FORTRAN/3000, type DOUBLE PRECISION (real).

DERF (cont.)

2. To compute the gaussian probability integral $A(x)$, use the following transformation:

$$A(x) = \frac{1}{\sqrt{2\pi}} \int_{-x}^x e^{-t^2/2} dt = \operatorname{erf} \left(\frac{x}{\sqrt{2}} \right)$$

DERFC

FUNCTION: Calculate the complimentary error function of a LONG¹ (real) number using extended precision arithmetic.

Declaration: PROCEDURE DERFC(X, DRSLT);
VALUE X;
LONG X, DRSLT;
OPTION EXTERNAL;

.
.
.

Definition: $\text{derfc}(x) = 1 - \text{derf}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt$

Accuracy: See the LONG error function DERF.

ATTRIBUTES:

Parameters: X = a LONG number in the range [-10, 10].

DRSLT = a LONG identifier

Result: DRSLT, a LONG number (set to 0 if X is outside the range stated).

Error: None

COMMENT: See Comments 1 and 2 for DERF.

¹In FORTRAN/3000, type DOUBLE PRECISION (real).

GAMMA

FUNCTION: Calculate the **gamma function** (also called Euler's second integral) of a **REAL** number.

Declaration: **PROCEDURE GAMMA(X, GAM, ERR);**
 VALUE X;
 REAL X, GAM;
 INTEGER ERR;
 OPTION EXTERNAL;
 .
 .
 .

Definition: Depends on the value of x :

x value	Definition
Any positive noninteger	$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$
Any positive integer	$\Gamma(x) = (x - 1)!$
Zero or any negative integer	None. Such values are asymptotes of the gamma function. See "Results," below.
Any negative noninteger	(Same as first definition above.)

Accuracy: Depends on accuracy of **EXP** (see *HP 3000 Compiler Library (HP 03000-90009)*) and relative error in x . The relative error in x is amplified by $x \ln x$ in the result. However, restrictions on the size of the argument limit loss of significance to about four digits.

ATTRIBUTES:

Parameters: **X** = a **REAL** number
 GAM = a **REAL** identifier
 ERR = an **INTEGER** identifier

Results: **GAM**, a **REAL** number
 ERR, an **INTEGER** error flag set to
 0 for no error.
 1 for $X = 0$; gamma function is undefined. No computation occurs, and the procedure terminates.
 2 for X = a negative integer; gamma function is undefined. No computation occurs, and the procedure terminates.

GAMMA (cont.)

Error: If $X > \approx 58$, arithmetic trap FLPT OFLW (floating point overflow) occurs (if traps are enabled). For other errors, see "Results," above.

LGAMMA

FUNCTION: Calculate the **natural logarithm of the gamma function** of a REAL number.

Declaration: `PROCEDURE LGAMMA(X, LGAM, ERR);`
`VALUE X;`
`REAL X, LGAM;`
`INTEGER ERR;`
`OPTION EXTERNAL;`
`.`
`.`
`.`

Definition: Depends on the value of x :

x value	Definition
Any positive noninteger	$\ln(\Gamma(x))$, where $\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$
Any positive integer	$\ln(\Gamma(x))$, where $\Gamma(x) = (x-1)!$
Zero or any negative value	None. See "Results," below.

Accuracy: Depends on the accuracy of ALOG (see *HP 3000 Compiler Library (HP 03000-90009)*), the gamma function GAMMA, and the relative error in x . The relative error in the result is less than the relative error in x but approaches the relative error in x as x becomes larger.

ATTRIBUTES:

Parameters: X = a positive REAL number
 LGAM = a REAL identifier
 ERR = an INTEGER identifier

Results: LGAM, a REAL number
 ERR, an INTEGER error flag set to

- 0 for no error.
- 1 for $X = 0$ which is an asymptote of the gamma function. No computation occurs, and the procedure terminates.
- 2 for X = a negative integer, any negative integer is an asymptote of the gamma function. No computation occurs, and the procedure terminates.
- 3 for X = a negative noninteger value in the range $(m, m+1)$ where m is any odd negative integer; the gamma function would return a negative value which has no natural logarithm. No computation occurs, and the procedure terminates.

LGAMMA (cont.)

4 for $X =$ a negative noninteger value in the range $(n, n + 1)$ where n is any even negative integer; this procedure rejects such a value. No computation occurs, and the procedure terminates.

Error: If X is too large, the result will overflow; arithmetic trap FLPT OFLW (floating point overflow) occurs (if traps are enabled). For other errors, see “Results,” above.

DGAMMA

FUNCTION: Calculate the **gamma function** (also called Euler's second integral) of a LONG¹ (real) number, using extended precision arithmetic.

Declaration: PROCEDURE DGAMMA (X, DGAM, ERR);
 VALUE X;
 LONG X, DGAM;
 INTEGER ERR;
 OPTION EXTERNAL;
 .
 .
 .

Definition: Depends on the value of x :

x value	Definition
Any positive noninteger	$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$
Any positive integer	$\Gamma(x) = (x - 1)!$
Zero or any negative integer	None. Such values are asymptotes of the gamma function. See "Results," below.
Any negative noninteger	(Same as first definition above.)

Accuracy: Depends on accuracy of DEXP (see *HP 3000 Compiler Library (HP 03000-90009)*) and relative error in x . The relative error in x is amplified by $x \ln x$ in the result. However, restrictions on the size of the argument limit loss of significance to about four digits.

ATTRIBUTES:

Parameters: X = a LONG number
 DGAM = a LONG identifier
 ERR = an INTEGER identifier

Results: DGAM, a LONG number
 ERR, an INTEGER error flag set to
 0 for no error.
 1 for $X = 0$; gamma function is undefined. No computation occurs, and the procedure terminates.

¹In FORTRAN/3000, type DOUBLE PRECISION (real).

DGAMMA (cont.)

2 for $X =$ a negative integer; gamma function is undefined. No computation occurs, and the procedure terminates.

Error: If $X > \approx 58$, arithmetic trap FLPT OFLW (floating point overflow) occurs (if traps are enabled). For other errors, see “Results,” above.

DLGAMMA

FUNCTION: Calculate the **natural logarithm of the gamma function of a LONG¹** (real) number, using extended precision arithmetic.

Declaration: `PROCEDURE DLGAMMA(X, DLGAM, ERR);`
`VALUE X;`
`LONG X, DLGAM;`
`INTEGER ERR;`
`OPTION EXTERNAL;`
`.`
`.`
`.`

Definition: Depends on the value of x :

x value	Definition
Any positive noninteger	$\ln(\Gamma(x))$, where $\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$
Any positive integer	$\ln(\Gamma(x))$, where $\Gamma(x) = (x-1)!$
Zero or any negative value	None. See "Results," below.

Accuracy: Depends on the accuracy of DLOG (see *HP 3000 Compiler Library (HP 03000-90009)*), the LONG gamma function DGAMMA, and the relative error in x . The relative error in the result, less than the relative error in x , approaches the relative error in x as x becomes larger.

ATTRIBUTES:

Parameters: X = a positive LONG number
DLGAM = a LONG identifier
ERR = an INTEGER identifier

Results: DLGAM, a LONG number
ERR, an INTEGER error flag set to
0 for no error.
1 for $X = 0$ which is an asymptote of the gamma function. No computation occurs, and the procedure terminates.
2 for X = a negative integer, any negative integer is an asymptote of the gamma function. No computation occurs, and the procedure terminates.

¹In FORTRAN/3000, type DOUBLE PRECISION (real).

DLGAMMA (cont.)

3 for $X =$ a negative noninteger value in the range $(m, m + 1)$ where m is any odd negative integer; the gamma function would return a negative value which has no natural logarithm. No computation occurs, and the procedure terminates.

4 for $X =$ a negative noninteger value in the range $(n, n + 1)$ where n is any even negative integer; this procedure rejects such a value. No computation occurs, and the procedure terminates.

Errors: If X is too large, the result will overflow; arithmetic trap FLPT OFLW (floating point overflow) occurs (if traps are enabled). For other errors, see "Results," above.

EXPI

FUNCTION: Calculate the **exponential integral** of a REAL number. See "Definition," below.

Declaration: `PROCEDURE EXPI(X, RESLT, ERR);`
`VALUE X;`
`REAL X, RESLT;`
`INTEGER ERR;`
`OPTION EXTERNAL;`
`.`
`.`
`.`

Definition: Depends on the value of x :

x value	Definition
$x > 0$	$E(x) = \int_x^{\infty} \frac{e^{-t}}{t} dt$
$-4 \leq x < 0$	$E(x) = - \int_x^{\infty} \frac{e^{-t}}{t} dt$
$x = 0$	None. See "Results," below.

Accuracy: Depends on accuracy of EXP and ALOG (see *HP 3000 Compiler Library (HP 03000-90009)*) and the truncation error. When $x \geq 4$, the truncation error is less than $10^{-8}/xe^x$; when $|x| < 4$, the truncation error is less than $3 \cdot 10^{-8}$.

ATTRIBUTES:

Parameters: $X \geq -4$, a REAL number
 RESLT = a REAL identifier
 ERR = an INTEGER identifier

Results: RESLT, a REAL number
 ERR, an INTEGER error flag set to
 0 for no error.
 1 for $X = 0$; this value is the asymptote of the exponential integral. No computation occurs, and the procedure terminates.
 2 for $X < -4$; this procedure rejects such a value. No computation occurs, and the procedure terminates.

Error: See "Results," above.

SICI

FUNCTION: Calculate the **sine integral** and the **cosine integral** of a **REAL** number.

Declaration: `PROCEDURE SICI(X, SI, CI, ERR);
VALUE X;
REAL X, SI, CI;
INTEGER ERR;
OPTION EXTERNAL;`

·
·
·



Definition: Depends on the value of x :

x value

Definition

$$x > 0 \quad \text{si}(x) = \int_0^x \frac{\sin t}{t} dt$$

$$\text{ci}(x) = \int_x^\infty \frac{\cos t}{t} dt$$

$$x = 0 \quad \text{si}(x) = \text{Same as for } x > 0$$

$$\text{ci}(x) = \text{None. See "Results," below.}$$

$$x < 0 \quad \text{si}(x) = -\pi - \text{si}(-x)$$

$$\text{ci}(x) = \text{ci}(-x)$$

Accuracy: Depends on accuracy of SIN and COS and ALOG (see *HP 3000 Compiler Library (HP 03000-90009)*) and the truncation error. When $|x| > 4$, the truncation error is less than $2.3 \cdot 10^{-8}/x$; when $|x| \leq 4$, the truncation error is less than or equal to $5.6 \cdot 10^{-9} \cdot x^2$.

ATTRIBUTES:

Parameters: X = a **REAL** number

SI and CI = **REAL** identifiers

ERR = an **INTEGER** identifier

Results: SI and CI, **REAL** numbers

ERR, an **INTEGER** error flag set to

0 for no error.

1 for X = 0; the cosine integral is undefined.

Error: See "Results," above.

FRES

FUNCTION: Calculate the Fresnel integrals $C(x)$ and $S(x)$ of a REAL number.

Declarations: PROCEDURE FRES(X, C, S);
VALUE X;
REAL X, C, S;
OPTION EXTERNAL;
.
.
.

Definitions:
$$C(x) = \frac{1}{\sqrt{2\pi}} \int_0^{|x|} \frac{\cos t}{\sqrt{t}} dt$$

$$S(x) = \frac{1}{\sqrt{2\pi}} \int_0^{|x|} \frac{\sin t}{\sqrt{t}} dt$$

Accuracy: Depends on accuracy of SIN and COS and SQRT (see *HP 3000 Compiler Library (HP 03000-90009)*) and the truncation error. When $|x| > 4$, the truncation error for both $C(x)$ and $S(x)$ is less than $2.5 \cdot 10^{-8} \sqrt{x}$. When $|x| \leq 4.0$, the truncation error for $C(x)$ is less than or equal to $\sqrt{x} \cdot 2.6 \cdot 10^{-8}$; the truncation error for $S(x)$ is less than or equal to $x \sqrt{x} \cdot 3.5 \cdot 10^{-8}$.

ATTRIBUTES:

Parameters: X = a REAL number
C and S = REAL identifiers

Result: C and S, REAL numbers.

Error: None.

EL1

FUNCTION: Calculate the elliptic integral of the first kind, using REAL parameters.

Declaration: PROCEDURE EL1(KC, X, EL);
 VALUE KC, X;
 REAL KC, X, EL;
 OPTION EXTERNAL;
 .
 .
 .

Definition: Depends on the value of k_c :

k_c value	Definition
$k_c^2 > 0$	$\text{el1}(x, k_c) = \int_0^x \frac{dt}{\sqrt{(1+t^2)(1+k_c^2 t^2)}}$
$k_c = 0$	$\text{el1}(x, k_c) = \text{sign}(x) \ln \left(x + \sqrt{1+x^2} \right)$

Accuracy: Accurate to about six decimal digits.

ATTRIBUTES:

Parameters: KC = a REAL number
 X = a REAL number
 EL = a REAL identifier

Result: EL, a REAL number.

Error: None.

COMMENTS: 1. KC is the complementary modulus of the elliptic integral. An alternate, and perhaps more common, definition of the elliptic integral of the first kind is

$$\text{el1}(x, k) = \int_0^{\arctan x} \frac{dt}{\sqrt{1-k^2 \sin^2 t}} \quad \text{for } 0 \leq k \leq 1$$

where $k^2 + k_c^2 = 1$ and k_c^2 is any value in the range $(0, \infty)$. The parameter k_c enables k to be real or purely imaginary (when k_c^2 is greater than 1, k^2 is negative, so k is purely imaginary).

2. X is the upper bound of the integral.

EL2

FUNCTION: Calculate the **elliptic integral of the second kind**, using REAL parameters.

Declaration: PROCEDURE EL2(KC, X, A, B, EL);
 VALUE KC, X, A, B;
 REAL KC, X, A, B, EL;
 OPTION EXTERNAL;
 .
 .
 .

Definition: Depends on the value of k_c :

k_c value

Definition

$$k_c^2 > 0 \quad \text{el2}(x, k_c, a, b) = \int_0^x \frac{a + bt^2}{(1+t^2)\sqrt{(1+t^2)(1+k_c^2 t^2)}} dt$$

$$k_c = 0 \quad \text{el2}(x, k_c, a, b) = b \cdot \ln \left(|x| + \sqrt{1+x^2} \right) + \text{sign}(x)(a-b) \frac{x}{\sqrt{1+x^2}}$$

Accuracy: Accurate to about six decimal digits except close to the zeros of the function.

ATTRIBUTES:

Parameters: KC = a REAL number
 X = a REAL number
 A = a REAL number
 B = a REAL number
 EL = a REAL identifier

Result: EL, a REAL number.

Error: None.

COMMENTS: 1. KC is the complementary modulus of the elliptic integral. The standard definition for the elliptic integral of the second kind is

$$\text{el2}(x, k, a, b) = \int_0^{\arctan x} \frac{a + (b-a) \sin^2 t}{\sqrt{(1-k^2 \sin^2 t)}} dt \quad \text{for} \quad 0 \leq k \leq 1$$

EL2 (cont.)

where $k^2 + k_c^2 = 1$ and k_c^2 is any value in the range $(0, \infty)$. The parameter k_c enables k to be real or purely imaginary (when k_c^2 is greater than 1, k^2 is negative, so k is purely imaginary).

2. X is the upper bound of the integral.

CEL1

FUNCTION: Calculate the **complete elliptic integral of the first kind**, using a **REAL** parameter.

Declarations: **PROCEDURE** CEL1(KC, CEL, ERR);
 VALUE KC;
 REAL KC, CEL;
 INTEGER ERR;
 OPTION **EXTERNAL**;
 :
 :

Definition: Depends on the value of k_c :

k_c value

Definition

$$k_c^2 > 0 \qquad \text{cell}(k_c) = \int_0^{\infty} \frac{dt}{\sqrt{(1+t^2)(1+k_c^2 t^2)}}$$

$k_c = 0$ None, the complete elliptic integral is undefined.

Accuracy: Accurate to about six decimal digits.

ATTRIBUTES:

Parameters: KC = a **REAL** number
 CEL = a **REAL** identifier
 ERR = an **INTEGER** identifier

Results: CEL, a **REAL** number
 ERR, an **INTEGER** error flag set to
 0 for no error.
 1 for KC = 0; the complete elliptic integral is undefined. No computation occurs, and the procedure terminates.

Error: See "Results," above.

COMMENT: KC is the complementary modulus of the elliptic integral. An alternate definition of the complete elliptic integral of the first kind is

$$\text{cell}(k) = \int_0^{\pi/2} \frac{dt}{\sqrt{1 - k^2 \sin^2 t}} \quad \text{for} \quad 0 \leq k \leq 1$$

where $k^2 + k_c^2 = 1$ and k_c^2 is any value in the range $(0, \infty)$. The parameter k_c enables k to be real or purely imaginary (when k_c^2 is greater than 1, k^2 is negative, so k is purely imaginary).

CEL2

FUNCTION: Calculate the **complete elliptic integral of the second kind**, using **REAL** parameters.

Declaration: `PROCEDURE CEL2(KC, A, B, CEL, ERR);`
`VALUE KC, A, B;`
`REAL KC, A, B, CEL;`
`INTEGER ERR;`
`OPTION EXTERNAL;`
`.`
`.`
`.`

Definition: Depends on the value of k_c :

k_c value	Definition
$k_c^2 > 0$	$\text{cel2}(x, k_c, a, b) = \int_0^x \frac{a + bt^2}{(1 + t^2) \sqrt{(1 + t^2)(1 + k_c^2 t^2)}} dt$
$k_c = 0$	None, the complete elliptic integral is undefined.

Accuracy: Accurate to about six decimal digits except close to the zero of the function.

ATTRIBUTES:

Parameters: **KC** = a **REAL** number
A = a **REAL** number
B = a **REAL** number
CEL = a **REAL** identifier
ERR = a **REAL** identifier

Results: **CEL**, a **REAL** number
ERR, an **INTEGER** error flag set to
0 for no error.
1 for **KC** = 0; the complete elliptic integral is undefined. No computation occurs, and the procedure terminates.

Error: See "Results," above.

CEL2 (cont.)

COMMENT: KC is the complementary modulus of the elliptic integral. An alternate definition for the complete elliptic integral of the second kind is

$$\text{el2}(x, k, a, b) = \int_0^{\pi/2} \frac{a + (b - a) \sin^2 t}{\sqrt{(1 - k^2 \sin^2 t)}} dt \quad \text{for } 0 \leq k \leq 1$$

where $k^2 + k_c^2 = 1$ and k_c^2 is any value in the range $(0, \infty)$. The parameter k_c enables k to be real or purely imaginary (when k_c^2 is greater than 1, k^2 is negative, so k is purely imaginary).

DEL1

FUNCTION: Calculate the **elliptic integral of the first kind**, using LONG¹ parameters and extended precision arithmetic.

Declaration: PROCEDURE DEL1 (KC, X, EL);
 VALUE KC, X;
 LONG KC, X, EL;
 OPTION EXTERNAL;
 .
 .
 .

Definition: Depends on the value of k_c :

k_c value

Definition

$$k_c^2 > 0 \qquad \text{el1} (x, k_c) = \int_0^x \frac{dt}{\sqrt{(1+t^2)(1+k_c^2 t^2)}}$$

$$k_c = 0 \qquad \text{el1} (x, k_c) = \text{sign} (x) \ln \left(|x| + \sqrt{1+x^2} \right)$$

Accuracy: Accurate to about 11 decimal digits; depends on accuracy of DSQRT and DATAN and DLOG (see *HP 3000 Compiler Library (HP 03000-90009)*) and the iterative scheme of this procedure. When $|k_c|$ is very large (near overflow), as many as 11 iterations are used to evaluate the result.

ATTRIBUTES:

Parameters: KC = a LONG number
 X = a LONG number
 EL = a LONG identifier

Result: EL, a LONG number.

Error: None.

COMMENTS: 1. KC is the complementary modulus of the elliptic integral. An alternate, and perhaps more common, definition of the elliptic integral of the first kind is

$$\text{el1} (x, k) = \int_0^{\arctan x} \frac{dt}{\sqrt{1-k^2 \sin^2 t}} \qquad \text{for} \qquad 0 \leq k \leq 1$$

¹In FORTRAN/3000, type DOUBLE PRECISION (real).

DEL1 (cont.)

where $k^2 + k_c^2 = 1$ and k_c^2 is any value in the range $(0, \infty)$. The parameter k_c enables k to be real or purely imaginary (when k_c^2 is greater than 1, k^2 is negative, so k is purely imaginary).

2. X is the upper bound of the integral.

DEL2

FUNCTION: Calculate the elliptic integral of the second kind, using LONG¹ parameters and extended precision arithmetic.

Declaration: PROCEDURE DEL2 (KC, X, A, B, EL);
 VALUE KC, X, A, B;
 LONG KC, X, A, B, EL;
 OPTION EXTERNAL;
 :
 :

Definition: Depends on the value of k_c :

k_c value	Definition
$k_c^2 > 0$	$\text{el2}(x, k_c, a, b) = \int_0^x \frac{a + bt^2}{(1+t^2)\sqrt{(1+t^2)(1+k_c^2 t^2)}} dt$
$k_c = 0$	$\text{el2}(x, k_c, a, b) = b \cdot \ln \left(x + \sqrt{1+x^2} \right) + \text{sign}(x)(a-b) \frac{x}{\sqrt{1+x^2}}$

Accuracy: Accurate to about 11 decimal digits except close to the zeros of the function.

ATTRIBUTES:

Parameters: KC = a LONG number
 X = a LONG number
 A = a LONG number
 B = a LONG number
 EL = a LONG identifier

Result: EL, a LONG number.

Error: None.

COMMENTS: 1. KC is the complementary modulus of the elliptic integral. The standard definition for the elliptic integral of the second kind is

$$\text{el2}(x, k, a, b) = \int_0^{\arctan x} \frac{a + (b-a) \sin^2 t}{\sqrt{(1-k^2 \sin^2 t)}} dt \quad \text{for} \quad 0 \leq k \leq 1$$

¹In FORTRAN/3000, type DOUBLE PRECISION (real).

DEL2 (cont.)

where $k^2 + k_c^2 = 1$ and k_c^2 is any value in the range $(0, \infty)$. The parameter k_c enables k to be real or purely imaginary (when k_c^2 is greater than 1, k^2 is negative, so k is purely imaginary).

2. X is the upper bound of the integral.

DCEL1

FUNCTION: Calculate the **complete elliptic integral of the first kind**, using a LONG¹ parameter and extended precision arithmetic.

Declaration: PROCEDURE DCEL1 (KC, CEL, ERR);
 VALUE KC;
 LONG KC, CEL;
 INTEGER ERR;
 OPTION EXTERNAL;

.
.
.



Definition: Depends on the value of k_c :

k_c value

Definition

$$k_c^2 > 0 \qquad \text{cell} (k_c) = \int_0^{\infty} \frac{dt}{\sqrt{(1+t^2)(1+k_c^2 t^2)}}$$

$k_c = 0$ None, the complete elliptic integral is undefined.

Accuracy: Accurate to about 11 decimal digits.

ATTRIBUTES:

Parameters: KC = a LONG number
 CEL = a LONG identifier
 ERR = an INTEGER identifier

Results: CEL, a LONG number
 ERR, an INTEGER error flag set to
 0 for no error.
 1 for KC = 0; the complete elliptic integral is undefined. No computation occurs, and the procedure terminates.

Error: See "Results," above.

¹In FORTRAN/3000, type DOUBLE PRECISION (real).

DCEL1 (cont.)

COMMENT: KC is the complementary modulus of the elliptic integral. An alternate definition of the complete elliptic integral of the first kind is

$$\text{cel1}(k) = \int_0^{\pi/2} \frac{dt}{\sqrt{1 - k^2 \sin^2 t}} \quad \text{for} \quad 0 \leq k \leq 1$$

where $k^2 + k_c^2 = 1$ and k_c^2 is any value in the range $(0, \infty)$. The parameter k_c enables k to be real or purely imaginary (when k_c^2 is greater than 1, k^2 is negative, so k is purely imaginary).

DCEL2

FUNCTION: Calculate the **complete elliptic integral of the second kind**, using LONG¹ parameters and extended precision arithmetic.

Declaration: PROCEDURE DCEL2 (KC, A, B, CEL, ERR);
 VALUE KC, A, B;
 LONG KC, A, B, CEL;
 INTEGER ERR;
 OPTION EXTERNAL
 :
 :
 :

Definition: Depends on the value of k_c :

k_c value	Definition
$k_c^2 > 0$	$\text{cel2}(s, k_c, a, b) = \int_0^s \frac{a + bt^2}{(1+t^2)\sqrt{(1+t^2)(1+k_c^2 t^2)}} dt$
$k_c = 0$	None, the complete elliptic integral is undefined.

Accuracy: Accurate to about 11 decimal digits except close to the zeros of the function.

ATTRIBUTES:

Parameters: KC = a LONG number
 A = a LONG number
 B = a LONG number
 CEL = a LONG identifier
 ERR = an INTEGER identifier

Results: CEL, a LONG number
 ERR, an INTEGER error flag set to
 0 for no error.
 1 for KC = 0; the complete elliptic integral is undefined. No computation occurs, and the procedure terminates.

Error: See "Results," above.

¹In FORTRAN/3000, type DOUBLE PRECISION (real).

DCEL2 (cont.)

COMMENT: KC is the complementary modulus of the elliptic integral. An alternate definition for the complete elliptic integral of the second kind is

$$\text{cel2}(x, k, a, b) = \int_0^{\pi/2} \frac{a + (b-a) \sin^2 t}{\sqrt{1 - k^2 \sin^2 t}} dt \quad \text{for } 0 \leq k \leq 1$$

where $k^2 + k_c^2 = 1$ and k_c^2 is any value in the range $(0, \infty)$. The parameter k_c enables k to be real or purely imaginary (when k_c^2 is greater than 1, k^2 is negative, so k is purely imaginary).

JELF

FUNCTION: Calculate the jacobian elliptic functions sn, cn, and dn, using REAL parameters.

Declaration: PROCEDURE JELF(MC, X, SN, CN, DN);
 VALUE MC, X;
 REAL MC, X, SN, CN, DN;
 OPTION EXTERNAL;
 .
 .
 .

Definition: Depends on the value of mc :

mc value	Definition
$mc \neq 0$	$x = \int_0^{\text{sn}} \frac{dt}{\sqrt{(1-t^2)(1-k^2 t^2)}}$ $\text{sn}^2 + \text{cn}^2 = 1$ $k^2 \text{sn}^2 + \text{dn}^2 = 1$ $mc = k_c^2$ $k^2 = 1 - k_c^2$
$mc = 0$	$\text{sn}(x, mc) = \tanh(x)$ $\text{cn}(x, mc) = \text{dn}(x, mc) = \frac{1}{\cosh(x)}$

Accuracy: JELF depends on accuracy of SIN, and COS (see *HP 3000 Compiler Library (HP 03000-90009)*). SN depends on accuracy of SIN, CN depends on accuracy of COS, and DN depends on accuracy of both SIN and COS.

ATTRIBUTES:

Parameters: MC = a REAL number
 X = a REAL number
 SN = a REAL identifier
 CN = a REAL identifier
 DN = a REAL identifier

Result: SN and CN and DN, REAL numbers.

Error: None.

JELF (cont.)

COMMENT:

MC is the square of the complementary modulus of the integral.

X is the value for which the inverse integral is computed.

An alternate definition of the jacobian elliptic functions is the inverse integral

$$x = \int_0^{\alpha} \frac{dt}{\sqrt{1 - k^2 \sin^2 t}}$$

$$\operatorname{sn}(x, k) = \sin \alpha$$

$$\operatorname{cn}(x, k) = \cos \alpha$$

$$\operatorname{dn}(x, k) = \sqrt{1 - k^2 \sin^2 \alpha}$$

where k is defined under "Definition."

DJELF

FUNCTION: Calculate the jacobian elliptic functions sn, cn, and dn, using LONG¹ parameters and extended precision arithmetic.

Declaration: PROCEDURE DJELF (MC, X, SN, CN, DN);
 VALUE MC, X;
 LONG MC, X, SN, CN, DN;
 OPTION EXTERNAL;
 .
 .
 .

Definition: Depends on the value of mc :

mc value

Definition

$mc = 0$

$$x = \int_0^{\text{sn}} \frac{dt}{\sqrt{(1-t^2)(1-k^2 t^2)}}$$

$$\text{sn}^2 + \text{cn}^2 = 1$$

$$k^2 \text{sn}^2 + \text{dn}^2 = 1$$

$$mc = k_c^2$$

$$k^2 = 1 - k_c^2$$

$mc \neq 0$

$$\text{sn}(x, mc) = \tanh(x)$$

$$\text{cn}(x, mc) = \text{dn}(x, mc) = \frac{1}{\cosh(x)}$$

Accuracy: DJELF depends on accuracy of DSIN and DCOS (see *HP 3000 Compiler Library (HP 03000-90009)*). SN depends on accuracy of DSIN, CN depends on accuracy of DCOS, and DN depends on accuracy of both DSIN and DCOS.

ATTRIBUTES:

Parameters: MC = a LONG number
 X = a LONG number
 SN = a LONG identifier
 CN = a LONG identifier
 DN = a LONG identifier

Result: SN and CN and DN, LONG numbers

Error: None.

¹In FORTRAN/3000, type DOUBLE PRECISION (real).

DJELF (cont.)

COMMENT: MC is the square of the complementary modulus of the integral.
X is the value for which the inverse integral is computed.
An alternate definition of the jacobian elliptic functions is the inverse integral

$$x = \int_0^{\alpha} \frac{dt}{\sqrt{1 - k^2 \sin^2 t}}$$

$$\operatorname{sn}(x, k) = \sin \alpha$$

$$\operatorname{cn}(x, k) = \cos \alpha$$

$$\operatorname{dn}(x, k) = \sqrt{1 - k^2 \sin^2 \alpha}$$

where k is defined under “Definition.”

BESJ

FUNCTION: Calculate the **bessel function of the first kind** $J_n(x)$ of integer order n of a REAL number x . See "Definition," below.

NOTE: For the bessel function of the second kind, $Y_n(x)$, use the procedure BESY.

Declaration: PROCEDURE BESJ(X, N, SEQ, RESLT, ERR);
 VALUE X, N, SEQ;
 REAL X;
 INTEGER N, SEQ, ERR;
 REAL ARRAY RESLT;
 OPTION EXTERNAL;

⋮



Definition: The bessel functions of integer order n , $J_n(x)$ and $Y_n(x)$, are linearly independent pairs of solutions to the differential equation

$$x^2 \frac{d^2 w}{dx^2} + x \frac{dw}{dx} + (x^2 - n^2) w = 0$$

This procedure BESJ computes $J_n(x)$.

Accuracy: Depends on ALOG and SQRT and SIN and COS (see *HP 3000 Compiler Library (HP 03000-90009)*) and factors described in Comment 2.

ATTRIBUTES:

Parameters: X = a REAL number
 N = an INTEGER number
 SEQ = an INTEGER number
 RESLT = a REAL array identifier

Restriction:

RESLT size $\geq$ SEQ

ERR = an INTEGER identifier; not used by this procedure BESJ, but reserved for use by procedure BESY

Results: RESLT, one or more (as specified by SEQ) REAL numbers.

Error: None.

BESJ (cont.)

COMMENTS:

1. Procedure BESJ can be used to compute one integer order N or a sequence of integer orders starting at integer order N , of the bessel function of the first kind. To evaluate only one integer order, call the procedure with a statement such as

BESJ(X, N, 1, RESLT, ERR);

where RESLT is a REAL identifier for a single variable (or a one-element array) and the other parameters are as defined above. If $SEQ < 1$, it is treated as 1.

To evaluate a sequence of several integer orders starting at integer order N , call the procedure with a statement such as

BESJ(X,N,SEQ,RESLT,ERR);

where SEQ is the number of elements in the desired sequence.

2. Further, $J_n(x)$ has infinitely many zeros on the real axis; as a result it is virtually impossible to give outstanding results close to those zeros. In fact, the relative error is likely to be unbounded near those zeros. All the zeros of $J_n(x) > n$. The absolute error in $J_0(x)$ is less than the error in x .

BESY

FUNCTION: Calculate the **bessel function of the second kind** $Y_n(x)$ of integer order n of a REAL number x . See "Definition," below.

NOTE: For the bessel function of the first kind, $J_n(x)$, use the procedure BESJ.

Declaration: PROCEDURE BESY (X, N, SEQ, RESLT, ERR);
 VALUE X, N, SEQ;
 REAL X;
 INTEGER N, SEQ, ERR;
 REAL ARRAY RESLT;
 OPTION EXTERNAL;
 :
 :
 :

Definition: The bessel functions of integer order n , $J_n(x)$ and $Y_n(x)$, are linearly independent pairs of solutions to the differential equation:

$$x^2 \frac{d^2 w}{dx^2} + x \frac{dw}{dx} + (x^2 - n^2) w = 0$$

This procedure BESY computes $Y_n(x)$.

Accuracy: Depends on ALOG and SQRT and SIN and COS (see *HP 3000 Compiler Library (HP 03000-90009)*) and factors described in Comment 2.

ATTRIBUTES:

Parameters: X = a REAL number
 N = an INTEGER number
 SEQ = an INTEGER number
 RESLT = a REAL array identifier

Restriction:

RESULT size $\geq$ SEQ

ERR = an INTEGER identifier

Results: RESLT, one or more (as specified by SEQ) REAL numbers.
 ERR, an INTEGER error flag set to

0 for no error.

1 for $X \leq 0$; the bessel function of the second kind is undefined. No computation occurs, and the procedure terminates.

Error: See "Results," above.

BESY (cont.)

COMMENTS:

1. This procedure BESY is a second entry point to procedure BESJ; thus, it can be used to compute one integer order N or a sequence of integer orders starting at integer order N , of the bessel function of the second kind. To evaluate only one integer order, call the procedure with a statement such as

BESY(X, N, 1, RESLT, ERR);

where RESLT is a REAL identifier for a single variable (or a one-element array) and the other parameters are as defined above. If $SEQ < 1$, it is treated as 1.

To evaluate a sequence of several integer orders starting at integer order N , call the procedure with a statement such as

BESY(X,N,SEQ,RESLT,ERR);

where SEQ is the number of elements in the desired sequence.

2. The bessel function $Y_n(x)$ grows rapidly with increasing n ; thus the error at each step of the recurrence relation does not appear in the next step. However, there are infinitely many zeros of $Y_n(x)$ on the real axis, and severe loss of accuracy may occur near those zeros.

SECTION II

Statistical Procedures

Of the several statistical procedures described in this section, all but one, DATA, are provided in the HP 3000 Scientific Library. DATA is a user-defined procedure that must meet the requirements explained under "USER-WRITTEN DATA PROCEDURE" in this section.

Each of the statistical procedures provided in the library can be called by users' programs in the same manner as other procedures in the library. However, these statistical procedures are primarily intended for use through the HP 3000 Statistical Analysis Routines (STAR/3000) system described in the manual *HP 3000 Statistical Analysis Routines (HP 3000-90011)*.

The descriptions in this section include the following notation definitions:

Notation	Definition
$\bar{V}_j$	The MEAN, the arithmetic average of unweighted data.
X_{ij}	i th observation of variable j .
N (or n)	The total number of observations.
σ_j	The STANDARD DEVIATION, a measure of distribution about the mean.
s	The STANDARD ERROR OF THE MEAN, another measure of distribution about the mean.
σ_j^2	The VARIANCE of variable j , the square of the STANDARD DEVIATION.
kur_j	The KURTOSIS of variable j . Kurtosis is a measure of the general "peakness" of the distribution curve. Positive values indicate more "peakness" than the normal distribution; negative values indicate more "flatness" than the normal distribution.
skew_j	The SKEWNESS of variable j . A distribution is skewed when the number of extreme cases on one side of the normal distribution curve is considerably larger than on the other side. Thus, if $\text{skew}_j > 0$, the distribution is skewed right; if $\text{skew}_j < 0$, the distribution is skewed left.

Notation	Definition
MIN_j	The MINIMUM (the smallest of a series of observations) of variable j .
MAX_j	The MAXIMUM (the largest of a series of observations) of variable j .
Rnge_j	The RANGE ($\text{MAX}_j - \text{MIN}_j$) of observations of variable j .
δ_j	$\sigma_j \cdot \frac{N-1}{N}$

USER-WRITTEN DATA PROCEDURE

Two statistical procedures in the HP 3000 Scientific Library, STAT and CORRELATE, require the name of a user-written DATA procedure as a parameter. That DATA procedure is called by STAT or CORRELATE, to read cases into a data matrix. A case is a set of observation values consisting of one observation value for each variable to be processed.

Procedure DATA must be able to handle either of two situations:

1. The data matrix is large enough to contain all the data. Procedure DATA reads all cases into the data matrix, then returns execution to the calling procedure, STAT or CORRELATE.
2. The data matrix is not large enough to contain all the data. Procedure DATA reads as many complete cases as possible into the data matrix, returns execution to the calling procedure, then repeats; this cycle continues until all data has been read.

Five parameters are used by the DATA procedure. (In this discussion, the labels of the parameters and the procedure name DATA are all examples only. Any appropriate labels and name(s) can be used.) Four of the parameters must be declared as part of the procedure:

```

PROCEDURE DATA(BUFUR,SIZE,NUMOBS,NUM);
  REAL ARRAY BUFUR;
  INTEGER SIZE,NUMOBS,NUM;
  OPTION EXTERNAL;
  .
  .
  .

```

where

BUFUR = a REAL array identifier, for the data matrix.

SIZE = the INTEGER size of BUFUR, in two-word REAL numbers;
(upper bound – lower bound) + 1.

NUMOBS = the INTEGER number of cases (sets of observation values).

NUM = an INTEGER identifier, to accumulate the total number of cases read by DATA; NUM is initialized to 0 by STAT or CORRELATE before DATA is called. When NUM = NUMOBS, DATA execution is terminated.

The fifth parameter, NUMVAR, is an INTEGER, either a constant or a value passed through common. It specifies the number of variables in the data to be read.

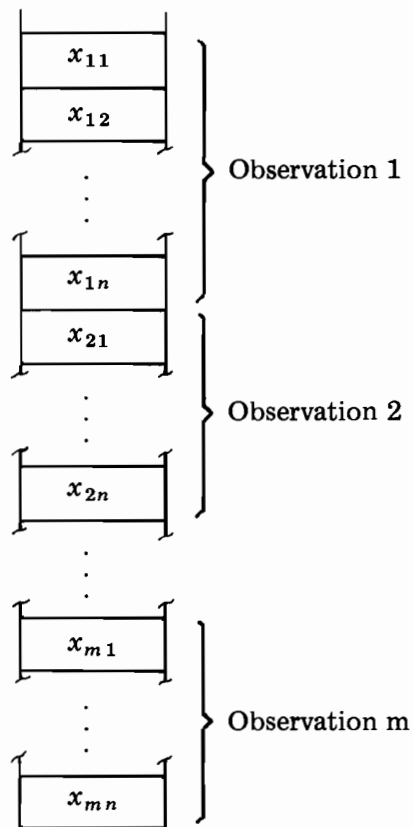
Restriction:

$$\text{NUMVAR} \leq \text{SIZE}$$

Here is an example of a DATA procedure written in FORTRAN, where 3 is the NUMVAR value:

```
SUBROUTINE DATA(BUFUR, SIZE, NUMOBS, NUM)
  INTEGER SIZE
  DIMENSION BUFUR(SIZE)
  C  DETERMINE HOW MANY COMPLETE CASES FIT IN BUFUR
  IFIT = SIZE/3
  C  3 = NUMVAR, THE NUMBER OF VARIABLES; REPLACE WITH AN
  C  APPROPRIATE INTEGER CONSTANT OR PASS AN INTEGER THROUGH
  C  COMMON
  C  DECIDE IF ALL REMAINING OBSERVATIONS FIT IN BUFUR
  IFIT = MIN0(IFIT, NUMOBS - NUM)
  C  INCREMENT TOTAL OBSERVATIONS ACCUMULATOR
  NUM = NUM + IFIT
  C  READ DATA INTO BUFUR
  DO 10 I = 1,IFIT
  DO 10 J = 1,3
10  BUFUR(3 * (I - 1) + J) = observation I of variable J
  RETURN
END
```

The data must be read into the data matrix as follows:



where

x_{ij} = i th observation of variable j .

m = observation (row) number

n = variable (column) number

STAT

FUNCTION: Compute, from unweighted data, the elementary statistical analyses:

MEAN
STANDARD DEVIATION
STANDARD ERROR OF THE MEAN
VARIANCE
KURTOSIS
SKEWNESS
MINIMUM
MAXIMUM
RANGE

see "Definitions," below.

Declaration: PROCEDURE STAT(BUFUR, SIZE, NUMVAR, RESULT, NUMOBS, DATA);
VALUE SIZE, NUMVAR, NUMOBS;
REAL ARRAY BUFUR, RESULT;
INTEGER SIZE, NUMVAR, NUMOBS;
PROCEDURE DATA;
OPTION EXTERNAL;
:
:



Definitions: MEAN:

$$\bar{V}_j = \frac{\sum_{i=1}^N X_{ij}}{N}$$

STANDARD DEVIATION:

$$\sigma_j = \left(\frac{\sum_{i=1}^N (X_{ij} - \bar{V}_j)^2}{N - 1} \right)^{1/2}$$

STANDARD ERROR OF THE MEAN:

$$s = \frac{\sigma_j}{\sqrt{N}}$$

VARIANCE (the second moment about the MEAN):

$$\sigma_j^2 = \frac{\sum_{i=1}^N (X_{ij} - \bar{V}_j)^2}{N - 1}$$

STAT (cont.)

KURTOSIS (the fourth moment about the MEAN):

$$\text{kur}_j = \frac{\sum_{i=1}^N \left(\frac{X_{ij} - \bar{V}_j}{\sigma_j} \right)^4}{N} - 3$$

SKEWNESS (the third moment about the MEAN):

$$\text{skew}_j = \frac{\sum_{i=1}^N \left(\frac{X_{ij} - \bar{V}_j}{\sigma_j} \right)^3}{N}$$

MINIMUM:

$$\text{MIN}_j = \text{MIN}(X_{1j}, X_{2j}, \dots, X_{nj})$$

MAXIMUM:

$$\text{MAX}_j = \text{MAX}(X_{1j}, X_{2j}, \dots, X_{nj})$$

RANGE:

$$\text{Rnge}_j = \text{MAX}_j - \text{MIN}_j$$

ATTRIBUTES:

Parameters: (Input)	BUFUR = a REAL array identifier, the data matrix formed by the DATA procedure ¹ which reads user-furnished data into BUFUR to form a data matrix.
	SIZE = an INTEGER number, the number of REAL elements that can be written into BUFUR. This parameter is passed to the DATA procedure.
	NUMVAR = an INTEGER number, the number of variables in the data.
	NUMOBS = an INTEGER number, the number of observations in the data. This parameter is passed to the DATA procedure.
	DATA = the DATA procedure identifier.
Parameters (Output)	RESULT = a REAL array identifier for the nine elementary statistics analyses produced by this STAT procedure for each variable. Restriction: $\text{RESULT size} \geq 9 \cdot \text{NUMVAR}$

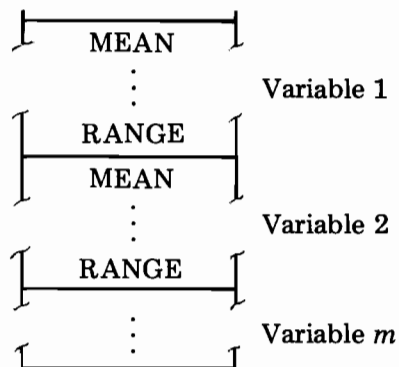
¹See "User-Written DATA Procedure," earlier in this section.

STAT (cont.)

The nine analyses appear in **RESULT** as follows:

MEAN
STD. DEV.
STD. ERR.
VARIANCE
KURTOSIS
SKEWNESS
MINIMUM
MAXIMUM
RANGE

If **NUMVAR** > 1, more than one variable is analyzed. The array **RESULT** contains the nine results for each variable in consecutive sets of nine locations:



NOTE: This *STAT* procedure detects no errors, but overflow or underflow could occur.

CORRELATE

FUNCTION: Calculate the **product-moment correlation coefficients, means, and standard deviations.**

Correlation produces an estimate of the linear association between two variables (see "Definition," below), for each variable in the data.

Declaration: `PROCEDURE CORRELATE(NUMOBS, NUMVAR, BUFUR, SIZE, CR,
MEAN, SIGMA, ERR, DATA);
VALUE NUMOBS, NUMVAR, SIZE;
INTEGER NUMOBS, NUMVAR, SIZE;
REAL ARRAY BUFUR, CR, MEAN, SIGMA;
LOGICAL ERR;
PROCEDURE DATA;
OPTION EXTERNAL;
:
:`

Definition:
$$r_{ij} = \frac{\text{COV}(x_i, x_j)}{\sigma_{x_i} \sigma_{x_j}}$$

where $\text{COV}(x_i, x_j)$ is the sample covariance between variables x_i and x_j and σ_{x_i} and σ_{x_j} are the sample standard deviations.

The value r satisfies the inequalities

$$-1 \leq r \leq 1$$

with $|r| = 1$ if and only if all of the sample points lie on a straight line.

The further $|r|$ is from unity, the further the observations are from the linear relationship. When $r_{ij} = 0$, there is no linear relationship between the variables x_i and x_j in the data.

ATTRIBUTES:

Parameters
(Input)

NUMOBS = an **INTEGER** number, the number of observations in the data. This parameter is passed to the **DATA** procedure¹ which reads user-furnished data into the **REAL** array **BUFUR** to form a data matrix.

NUMVAR = an **INTEGER** number, the number of variables in the data.

BUFUR = a **REAL** array identifier, the data matrix formed by the **DATA** procedure.

SIZE = an **INTEGER** number, the number of **REAL** elements that can be written into **BUFUR**. This parameter is passed to the **DATA** procedure.

DATA = the **DATA** procedure identifier.

¹ See "User-Written **DATA** Procedure," earlier in this section.

CORRELATE (cont.)

Parameters (Output)

CR = a REAL array identifier for the matrix of correlation coefficients, stored in triangular form (see "Discussion").

Restriction:

$$\text{CR size} \geq \frac{\text{NUVAR} (\text{NUMVAR} + 1)}{2}$$

MEAN = a REAL array identifier for the means, one for each variable.
MEAN (0), subscript 0, contains the mean of the first variable,
MEAN (1) contains the mean of the second variable, etc., etc.

Restriction:

$$\text{MEAN size} \geq \text{NUMVAR}$$

SIGMA = a REAL array identifier for the standard deviations, one for each variable. **SIGMA** (0) contains the standard deviation of the first variable, **SIGMA** (1) contains the standard deviation of the second variable, etc., etc.

Restriction:

$$\text{SIGMA size} \geq \text{NUMVAR}$$

ERR = a LOGICAL error flag set to

FALSE for no error

TRUE if NUMOBS < 2 or NUMVAR < 1. No computation occurs, and the procedure terminates.

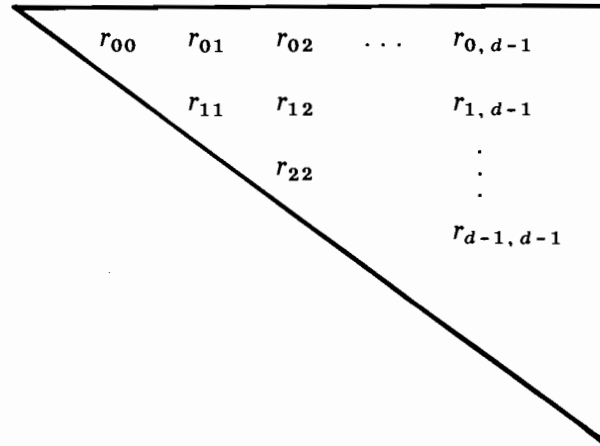
DISCUSSION:

1. *Triangular matrix format.* The triangular form of the correlation coefficients symmetric array (for this example, labeled CR) of dimension d is represented as follows:

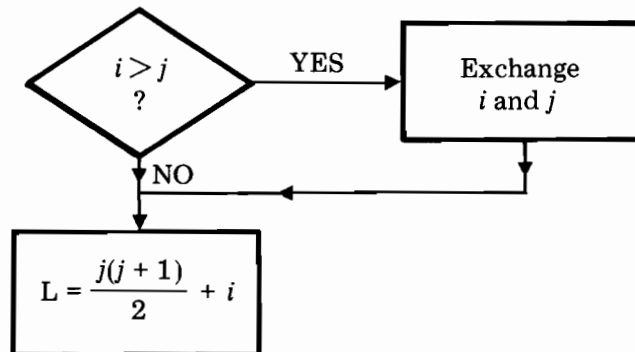
Conventional notation	Actual storage														
$\begin{bmatrix} r_{00} & r_{01} & \cdots & r_{0,d-1} \\ r_{10} & & & \\ \cdot & & & \\ \cdot & & & \\ r_{d-1,0} & r_{d-1,1} & \cdots & r_{d-1,d-1} \end{bmatrix}$	<table> <tr> <th>Reference</th><th>Contents</th></tr> <tr> <td>CR(0)</td><td>r_{00}</td></tr> <tr> <td>CR(1)</td><td>r_{01}</td></tr> <tr> <td>CR(2)</td><td>r_{11}</td></tr> <tr> <td>CR(3)</td><td>r_{02}</td></tr> <tr> <td>$\vdots$</td><td>$\vdots$</td></tr> <tr> <td>$\text{CR} \left(\frac{d(d+1)}{2} - 1 \right)$</td><td>$r_{d-1,d-1}$</td></tr> </table>	Reference	Contents	CR(0)	r_{00}	CR(1)	r_{01}	CR(2)	r_{11}	CR(3)	r_{02}	$\vdots$	$\vdots$	$\text{CR} \left(\frac{d(d+1)}{2} - 1 \right)$	$r_{d-1,d-1}$
Reference	Contents														
CR(0)	r_{00}														
CR(1)	r_{01}														
CR(2)	r_{11}														
CR(3)	r_{02}														
$\vdots$	$\vdots$														
$\text{CR} \left(\frac{d(d+1)}{2} - 1 \right)$	$r_{d-1,d-1}$														

CORRELATE (cont.)

Thus, the actual matrix is



Subscripting into the array uses the following algorithm to obtain $r_{x_i x_j}$:



Thus, $CR(L)$ contains $r_{x_i x_j}$.

2. **Mathematics.** Computation of cross products uses either of two methods: In the first method, only the first M observations are stored in the data matrix identified for BUFUR:

$$P_{ij} = \sum_{l=0}^{N-1} (x_{li} - T_i) (x_{lj} - T_j) - N (\mu_i - T_i) (\mu_j - T_j)$$

where

$$T_i = \frac{\sum_{l=0}^{M-1} x_{li}}{M} \quad \text{for } M < N$$

CORRELATE (cont.)

and

$$\mu_i = \frac{\sum_{l=0}^{N-1} x_{li}}{N}$$

In the second method, all data is in the data matrix:

$$P_{ij} = \sum_{l=0}^{N-1} (x_{li} - \mu_{li}) (x_{lj} - \mu_{lj})$$

Computation of correlation coefficients uses these methods:

$$r_{ij} = \frac{P_{ij}}{\sqrt{P_{ii}P_{jj}}} \quad \text{where} \quad \begin{array}{l} j = 1, \text{ NUMVAR} - 1 \\ i = 0, j - 1 \end{array}$$

and

$$r_{ii} = 1 \quad \text{where} \quad i = 0, \text{ NUMVAR} - 1$$

NOTE: The CORRELATE procedure uses the square root procedure SQRT (see HP 3000 Compiler Library (HP 03000-90009)).

ARRANGE

FUNCTION: Extract a **subset of correlation coefficients** and a **vector of a dependent variable** from the triangular matrix produced by the CORRELATE procedure.

Declaration: PROCEDURE ARRANGE (CR, NUMVAR, S, P, DEP, SUBCR, V, ERR);
VALUE NUMVAR, P, DEP;
REAL ARRAY CR, SUBCR, V;
INTEGER NUMVAR, P;
INTEGER ARRAY S;
LOGICAL DEP, ERR;
OPTION EXTERNAL;
.
.
.

ATTRIBUTES:

Parameters (Input):

CR = a REAL array identifier (procedure CORRELATE output parameter CR), the triangular matrix of correlation coefficients.

NUMVAR = the INTEGER order of CR (the number of variables in the data).

S = an INTEGER array identifier, a vector of numbers to specify which variables represented in CR are to be extracted.

$$S \text{ size} = P + 1$$

EXAMPLE:

$S(0) \leftarrow 3; S(1) \leftarrow 2; S(2) \leftarrow 4; S(3) \leftarrow 1;$

specify the third, second, fourth and first variables are to be extracted.

P = an INTEGER number, the number of *independent* variables correlation coefficients to be extracted. This parameter specifies the subscripts of vector S to be used: Subscripts S(0) through S(P-1) are the *independent* variables extracted for the output matrix SUBCR; subscript S(P) is the *dependent* variable extracted for output vector V.

Restrictions:

$$P \leq \text{NUMVAR}$$

$$P \geq 0$$

DEP = a LOGICAL identifier, set to FALSE to direct this ARRANGE procedure to extract both the *independent* variables matrix SUBCR and the *dependent* variable vector V; or set to TRUE to direct this procedure to extract only V without changing SUBCR.

ARRANGE (cont.)

Parameters
(Output):

SUBCR = a REAL array identifier for the independent variables correlation coefficients, stored in triangular form (see the example below).

$$\text{SUBCR size} = \frac{P(P+1)}{2}$$

EXAMPLE:

If array S is set as shown in the example for that input parameter, ARRANGE can be called by a statement such as:

```
ARRANGE(CR,4,S,3,FALSE,SUBCR,V,ERR);
```

This extracts from the triangular matrix CR:

r_{00}	r_{01}	r_{02}	r_{03}
	r_{11}	r_{12}	r_{13}
		r_{22}	r_{23}
			r_{33}



the independent variables 2, 1, and 3 and the dependent variable 0:

SUBCR			V
r_{22}	r_{12}	r_{23}	r_{02}
	r_{11}	r_{13}	r_{01}
		r_{33}	r_{03}

V = a REAL array identifier for the vector of the dependent variable.

ERR = a LOGICAL error flag set to

FALSE for no error

TRUE if $P > \text{NUMVAR}$ or a nonexistent variable is specified in array S. No extraction occurs and the procedure terminates.

MULTLINR

FUNCTION: Compute multiple linear regression coefficients and related analyses from the outputs of procedures CORRELATE, ARRANGE and SMETINV (see Section III of this manual). Further details are given under "Procedure Relationships," on the following page.

Declaration: PROCEDURE MULTLINR(SUBCR, V, SIGMA, MEAN, S, NUMOBS, P, BETA, B, INTERCEPT, STAT, SB, ERR);
VALUE NUMOBS, P;
REAL ARRAY SUBCR, V, SIGMA, MEAN, BETA, B, STAT, SB;
INTEGER ARRAY S;
INTEGER NUMOBS, P;
REAL INTERCEPT;
LOGICAL ERR;
OPTION EXTERNAL;
:
:

ATTRIBUTES:

Parameters (Input):

SUBCR = a REAL array identifier (procedure ARRANGE output parameter SUBCR *after inversion* by procedure SMETINV), the triangular matrix of independent variables correlation coefficients.

V = a REAL array identifier (procedure ARRANGE output parameter V), the vector of the dependent variable.

SIGMA = a REAL array identifier (procedure CORRELATE output parameter SIGMA), the standard deviations, one for each variable.

MEAN = a REAL array identifier (procedure CORRELATE output parameter MEAN), the means, one for each variable.

S = an INTEGER array identifier (procedure ARRANGE input parameter S), a vector of numbers to specify which variables represented in CR were extracted.

NUMOBS = an INTEGER number (procedure CORRELATE input parameter NUMOBS), the number of observations in the data.

P = an INTEGER number (procedure ARRANGE input parameter P), the number of independent variables correlation coefficients extracted.

MULTLINR (cont.)

Parameters
(Output):

BETA = a REAL array identifier for the normalized regression coefficients vector. This vector contains coefficients based on the values of the variables stated in units of standard deviation from the mean. For example,

$$x_i^* = \frac{x_i - \bar{x}_i}{\sigma_i}$$

These vector values are used to compare the relative influence of the independent variables on the dependent variable. See "Discussion" for further details.

Restriction:

BETA size = P

B = a REAL array identifier for the true (un-normalized) regression coefficients vector. This vector is calculated by

$$b_i = \beta_i \frac{\sigma_y}{\sigma_i}$$

and

$$b_0 = \bar{y} - \sum b_i \bar{x}_k \quad \text{where } b_0 \text{ is INTERCEPT}$$

B size = P

INTERCEPT = a REAL identifier for the constant of the regression equation.

STAT = a five-element REAL array identifier for the statistics vector.
This vector contains analyses as follows:

Subscript	Contents	Calculations
STAT(0)	R = the multiple regression coefficient, which is equal to $\frac{\text{regression sum of squares}}{\text{total sum of squares}}$	$R = \sqrt{\sum \beta_i r_{iy}}$
STAT(1)	SSR = the regression sum of squares	$\text{SSR} = R^2 \text{SST}$ where $\text{SST} = \sigma_y^2 (n - 1)$
STAT(2)	SSE = the residual sum of squares	$\text{SSE} = \text{SST} - \text{SSR}$
STAT(3)	S = the standard error of the estimate	$S = \sqrt{\frac{\text{SSE}}{n - p - 1}}$
STAT(4)	F = distribution with P and NUMOBS - P - 1 degrees of freedom	$F = \frac{\text{SSR}/p}{\text{SSE}/(n - p - 1)}$

MULTLINR (cont.)

SB = a REAL array identifier for the vector of standard errors for the sample regression coefficients. This vector is calculated by

$$S_i = S\sqrt{c_{ii}} \quad \text{where} \quad c_{ii} = \frac{c^{*ii}}{(n-1)\sigma_i^2}$$

$$\text{and} \quad C^* = (X^{*T} X)^{-1}$$

SB size = P

ERR = a LOGICAL error flag set to

FALSE for no error

TRUE if NUMOBS $\leq$ P + 1 or P < 0. No computation occurs, and the procedure terminates.

PROCEDURE RELATIONSHIPS

Regression analysis is separated into four procedures in the Scientific Library. The separation allows analysis of several subsets of independent variables in terms of one or more dependent variables without recalculation of the correlation coefficients, and their inverse, for each subset.

First, procedure CORRELATE produces CR, a matrix of correlation coefficients, MEAN, an array of means, and SIGMA, an array of standard deviations. Next, procedure ARRANGE extracts SUBCR, a matrix of correlation coefficients for a relevant subset of independent variables and V, a vector of a relevant dependent variable. Procedure SMETINV (described in Section III) follows, to invert SUBCR in place. These various results of procedures CORRELATE, ARRANGE and SMETINV are used by this procedure MULTLINR to analyze the relevant independent variables in terms of the relevant dependent variable V.

After a given MULTLINR analysis, the inverted matrix SUBCR for the subset of relevant independent variables can be retained while procedure ARRANGE is used to extract only a vector of another dependent variable. Then procedure MULTLINR is used again.

To extract a different subset of relevant independent variables, procedures ARRANGE, SMETINV and MULTLINR are used as described above. Procedure CORRELATE is not used again until another set of variables is to be made available.

EXAMPLES:

1. Simple regression

```
CORRELATE (100, 5, A, 100, R, MEAN, SIGMA, F, DATA);  
ARRANGE (R, 5, S, 4, FALSE, COV, COR, F);  
SMETINV (COV, 4, F);  
MULTLINR (COV, COR, SIGMA, MEAN, S, 100, 4, BETA, B, INTERCEPT,  
STAT, SB, F);
```

Assume

```
S(0) = 2  
S(1) = 3  
S(2) = 4  
S(3) = 5  
S(4) = 1
```

MULTLINR (cont.)

This example performs regression on a set of five variables, with the first variable in the set as the dependent variable.

2. Selective use of variables.

```
CORRELATE (100, 20, A, 100, R, MEAN, SIGMA, F, DATA);  
S(0) = 1; S(1) = 2; S(2) = 11;  
ARRANGE (R, 20, S, 2, FALSE, COV, COR, F);  
SMETINV (COV, 2, F);  
MULTLINR (COV, COR, SIGMA, MEAN, S, 100, 2, BETA, B, INTERCEPT,  
STAT, SB, F);  
:  
:  
S(0) = 2; S(1) = 5; S(2) = 4; S(3) = 13;  
ARRANGE (R, 20, S, 3, FALSE, COV, COR, F);  
SMETINV (COV, 3, F);  
MULTLINR (COV, COR, SIGMA, MEAN, S, 100, 3, BETA, B, INTERCEPT,  
STAT, SB, F);
```

Correlation is performed on a set of 20 variables. Variables 1 and 2 are independent, variable 11 is the dependent variable, and regression is performed. Then, a new set of variables is chosen (2, 5, and 4 are independent, 13 is dependent), and an additional regression analysis is performed. Note that correlation was performed just once for the entire set.

3. Change only the dependent variable.

```
CORRELATE (200, 10, A, 100, R, MEAN, SIGMA, F, DATA);  
S(0) = 4; S(1) = 5; S(2) = 6; S(3) = 9;  
ARRANGE (R, 10, S, 3, FALSE, COV, COR, F);  
SMETINV (COV, 3, F);  
MULTLINR (COV, COR, SIGMA, MEAN, S, 100, 3, BETA, B, INTERCEPT,  
STAT, SB, F);  
:  
:  
S(3) = 10;  
ARRANGE (R, 10, 5, 3, TRUE, COV, COR, F);  
MULTLINR (COV, COR, SIGMA, MEAN, S, 100, 3, BETA, B, INTERCEPT,  
STAT, SB, F);
```

Regression is performed first with variable 9 the dependent variable, then with variable 10. Since the independent variables did not change, it was not necessary to re-invert the correlation matrix.

MULTLINR (cont.)

DISCUSSION:

Assume that y is the dependent variable and $x_1 x_2 \dots x_p$ are the P independent variables. The objective; to fit to the equation

$$y = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_p x_p$$

where b_0 is INTERCEPT.

1. The alternate set of normal equations are solved:

$$\beta = (X^{*T} X^*)^{-1} X^{*T} Y^*$$

where $(X^{*T} X^*)^{-1}$ is the inverse of the matrix of correlation coefficients (SUBCR)

$$(X^{*T} X^*)^{-1} = \begin{bmatrix} 1 & r_{12} & \cdot & \cdot & \cdot & \cdot & r_{1p} \\ r_{21} & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ r_{p1} & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \end{bmatrix}^{-1}$$

and $X^{*T} Y^*$ is the vector of the correlations with y (V):

$$X^{*T} Y^* = \begin{bmatrix} r_{y1} \\ r_{y2} \\ \cdot \\ \cdot \\ \cdot \\ r_{yp} \end{bmatrix}$$

β , the vector of beta weights, represents the normalized regression coefficients, BETA, and provides the solution to the following equation:

$$y^* = \beta_1 x_1^* + \beta_2 x_2^* + \dots + \beta_p x_p^*$$

where y^* and the x_i^* are the normalized variables:

$$x_i^* = \frac{x_i - \bar{x}_i}{\sigma_i}$$

SECTION III
Utility Procedure

SMETINV

FUNCTION: Invert a symmetric matrix stored in triangular form (see procedure CORRELATE in Section II). The resulting inverse is stored over the input matrix.

Declaration: PROCEDURE SMETINV(SUBCR,MATDIM,ERR);
VALUE MATDIM;
REAL ARRAY SUBCR;
INTEGER MATDIM;
LOGICAL ERR;
OPTION EXTERNAL;
.
.
.



ATTRIBUTES:

Parameters: SUBCR = a REAL array identifier¹

MATDIM = an INTEGER number, the order of SUBCR

Restriction:

MATDIM $\geq$ 1

ERR = a LOGICAL error flag set to

FALSE for no error.

TRUE if the matrix to be inverted is singular. No inversion occurs, and the procedure terminates.

Results: SUBCR, the inverse of the original matrix (in place of the original).

Error: Floating point traps may occur, or see ERR above.

COMMENTS: This procedure can invert any triangular matrix. However, it is expected to be used most often for the statistical procedure MULTLINR described in Section II of this manual, where:

SUBCR is the procedure ARRANGE output parameter SUBCR.

MATDIM is the size of array S minus 1, where S is an input parameter to procedure ARRANGE.

¹ For example, see procedure ARRANGE output parameter SUBCR, in Section II.