IBM **Application Program** H20-0225-0

1130 Scientific Subroutine Package

Application Description

The Scientific Subroutine Package (SSP) is a collection of FORTRAN subroutines divided, for the sake of presentation, into three groups: statistics, matrix manipulation, and other mathematics. This manual gives brief abstracts of the subroutines and some of their common characteristics.

CONTENTS

Copies of this and other IBM publications can be obtained through IBM branch offices. Address comments concerning the contents of this publication to IBM, Technical Publications Department, 112 East Post Road, White Plains, N.Y. 10601

INTRODUCTION

OVERVIEW

The Scientific Subroutine Package (SSP) is a collection of FORTRAN subroutines divided, for the sake of presentation, into three groups: statistics, matrix manipulation, and other mathematics. It is a collection of input/output-free computational building blocks that can be combined with a user's input, output, or computational routines to meet his needs. The package can be applied to the solution of many problems in industry, science, and engineering.

The 1130 Scientific Subroutine Package makes available a mathematical and statistical subroutine library that the user may supplement or modify to meet his needs. This library includes a wide variety of subroutines to perform the functions listed below, but is not intended to be exhaustive in terms of either functions performed or methods used.

CONTENT

Individual subroutines, or a combination of them, can be used to carry out the listed functions in the following areas:

Statistics

- Analysis of variance (factorial design)
- Correlation analysis
- Multiple linear regression
- Polynomial regression
- Canonical correlation
- Factor analysis (principal components, varimax)
- Discriminant analysis (many groups)
- Time series analysis
- Data screening and analysis
- Nonparametric tests
- Random number generation (uniform, normal)

Matrix Manipulation

- Inversion
- Eigenvalues and vectors (real symmetric case)
- Simultaneous linear algebraic equations
- Transpositions
- Matrix arithmetic (addition, product, etc.)
- Partitioning
- Tabulation and sorting of rows or columns
- Elementary operations on rows or columns

Other Mathematical Areas

- Integration of given or tabulated functions
- Integration of up to six first-order differential equations
- Fourier analysis of given or tabulated functions
- Bessel and modified Bessel function evaluation
- Gamma function evaluation
- Legendre function evaluation
- Elliptic, exponential, sine, cosine, Fresnel integrals
- Finding real roots of a given function
- Finding real and complex roots of a real polynomial
- Polynomial arithmetic (addition, division, etc.)
- Polynomial evaluation, integration, differentiation

CHARACTERISTICS

Some of the characteristics of the Scientific Subroutine Package are:

- All subroutines are free of input/output statements
- Subroutines do not contain fixed maximum dimensions for the data arrays named in their calling sequences
- All subroutines are written in FORTRAN
- Many matrix manipulation subroutines handle symmetric and diagonal matrices (stored in economical, compressed formats) as well as general matrices. This can result in considerable saving in data storage for large arrays.
- The use of the more complex subroutines (or groups of them) is illustrated in the program documentation by sample main programs with input/output
- All subroutines are documented uniformly

DESIGN PHILOSOPHY

Choice of Algorithms

The algorithms in SSP have been chosen after considering questions of storage, accuracy, and past experience with the algorithm. Conservation of storage has been the primary criterion except in those situations where other considerations outweighed that of storage. As a result, many compromises have been made both with respect to level of sophistication and execution time. One such compromise is the use of the Runge-Kutta integration technique rather than predictor-corrector methods. A departure from the primary criterion of storage is illustrated by the algorithm for matrix inversion. If only row pivoting had been used, the subroutine would not have required working storage and would have needed fewer FORTRAN statements for implementation. However, pivoting on both rows and columns was chosen because of the accuracy requirement for matrix inversion in statistical operations.

Programming

The subroutines in SSP have been programmed in 1130 FORTRAN. Many of the larger functions such as those in statistics have been programmed as a series or sequence of subroutines.

An example of the use of sequences of subroutines is the statistical function called factor analysis. Factor analysis is a method of analyzing the underlying relations within a set of variables. It determines whether the variance in the original set of variables can be accounted for adequately by a smaller number of basic categories; namely, factors. In the 1130 Scientific Subroutine Package, factor analysis is normally performed by calling the following five subroutines in sequence:

1. CORRE - to find means, standard deviations, and correlation matrix
2. EIGEN - to compute eigenvalues and associated eigenvectors of the correlation matrix
3. TRACE - to select the eigenvalues that are greater than or equal to the control value specified by the user
4. LOAD - to compute a factor matrix
5. VARMX - to perform varimax rotation of the factor matrix

The multiple use of subroutines is illustrated by the fact that subroutine CORRE is also utilized in the multiple linear regression and canonical correlation. Subroutine EIGEN is used in canonical correlation as a third level subroutine.

RELATION TO SSP/360

SSP/1130 is intended to be compatible with SSP/360 (360A-CM-03X). All of the calling sequences are identical. The subroutine CONVT (dealing with single and double precision in System/360 FORTRAN) has been omitted from SSP/1130. The name TTEST in SSP/360 has been changed to TTSTT because the name TTEST is an 1130 Monitor function name. The ''minimum'' constant used in the SSP/1130 Routines is - 1.E38 instead of - 1.E75 (used in SSP/360) and the ''maximum'' constant is 1.E38 rather than 1.E75. The 1130 FORTRAN integers are 16 bits as opposed to 32 bits in the System/360 FORTRAN, causing some differences in the subroutines. In RANDU for example, SSP/360 will produce 2^{29} terms before repeating, whereas in SSP/1130, terms repeat after 2^{13}. (Because GAUSS uses RANDU, GAUSS will also have a shorter cycle length.) There are minor coding changes within the subroutines, and there will be some differences in results because 1130 and S/360 FORTRAN are numerically different.

The sample problems have different I/O and FORMAT statements, and the dimensions have been reduced to fit into 8K words of core.

SUBROUTINES

GENERAL REMARKS

Below are listed the subroutines of SSP/1130, grouped into related functional areas. In the case of six statistical entries (Multiple Linear Regression to Factor Analysis) the abstract gives the sequence of several SSP subroutines needed to perform the statistical function.

A tabulation of the subroutines of SSP, with detailed characteristics, is given in the appendices.

STATISTICS

Data Screening

TALLY--totals, means, standard deviations, minimums, and maximums

BOUND--selection of observations within bounds

SUBST--subset selection from observation matrix

ABSNT--detection of missing data

TAB1--tabulation of data (1 variable)

TAB2--tabulation of data (2 variables)

SUBMX--build subset matrix

Elementary Statistics

MOMEN--first four moments

TTSTT--tests on population means

Correlation

CORRE--means, standard deviations, and correlations

Multiple Linear Regression

Abstract (CORRE, ORDER, MINV, MULTR in sequence)

ORDER--rearrangement of intercorrelations

MULTR--multiple regression and correlation

Polynomial Regression

Abstract (GDATA, ORDER, MINV, MULTR in sequence)

GDATA--data generation

Canonical Correlation

Abstract (CORRE, CANOR, MINV, NROOT, EIGEN in sequence)

CANOR--canonical correlation

NROOT--eigenvalues and eigenvectors of a special nonsymmetric matrix

Analysis of Variance

Abstract (AVDAT, AVCAL, MEANQ in sequence)

AVDAT--data storage allocation

AVCAL--Σ and Δ operation

MEANQ--mean square operation

Discriminant Analysis

Abstract (DMATX, MINV, DISCR in sequence)

DMATX--means and dispersion matrix

DISCR--discriminant functions

Factor Analysis

Abstract (CORRE, EIGEN, TRACE, LOAD, VARMX in sequence)

TRACE--cumulative percentage of eigenvalues

LOAD--factor loading

VARMX--varimax rotation

Time Series

AUTO--autocovariances

CROSS--crosscovariances

SMO--application of filter coefficients (weights)

EXSMO--triple exponential smoothing

Nonparametric Statistics

CHISQ--x^2 test for a contingency table

UTEST--Mann-Whitney U-test

TWOAV--Friedman two-way analysis of variance

QTEST--Cochran Q-test

SRANK--Spearman rank correlation

KRANK--Kendall rank correlation

WTEST--Kendall coefficient of concordance

RANK--rank observations

TIE--calculation of ties in ranked observations

Random Number Generators

RANDU--uniform random numbers

GAUSS--normal random numbers

MATRIX MANIPULATION

MINV--Matrix inversion

EIGEN--eigenvalues and eigenvectors of a real, symmetric matrix

SIMQ--solution of simultaneous linear, algebraic equations

GMADD--add two general matrices

GMSUB--subtract two general matrices

GMPRD--product of two general matrices

GMTRA--transpose of a general matrix

GTPRD--transpose product of two general matrices

MADD--add two matrices

MSUB--subtract two matrices

MPRD--matrix product (row into column)

MTRA--transpose a matrix

TPRD--transpose product

MATA--transpose product of matrix by itself

SADD--add scalar to matrix

SSUB--subtract scalar from a matrix

SMPY--matrix multiplied by a scalar

SDIV--matrix divided by a scalar

RADD--add row of one matrix to row of another matrix

CADD--add column of one matrix to column of another matrix

SRMA--scalar multiply row and add to another row

SCMA--scalar multiply column and add to another column

RINT--interchange two rows

CINT--interchange two columns

RSUM--sum the rows of a matrix

CSUM--sum the columns of a matrix

RTAB--tabulate the rows of a matrix

CTAB--tabulate the columns of a matrix

RSRT--sort matrix rows

CSRT--sort matrix columns

RCUT-partition row-wise

CCUT--partition column-wise

RTIE--adjoin two matrices row-wise

CTIE--adjoin two matrices column-wise

MCPY--matrix copy

XCPY--copy submatrix from given matrix

RCPY--copy row of matrix into vector

CCPY--copy column of matrix into vector

DCPY--copy diagonal of matrix into vector

SCLA--matrix clear and add scalar

DCLA--raplace diagonal with scalar

MSTR--storage conversion

MFUN--matrix transformation by a function

RECP--reciprocal function for MFUN

LOC--location in compressed-stored matrix

ARRAY--vector storage—double dimensioned storage conversion

OTHER MATHEMATICAL AREAS

Integration

QUADR--integral of tabulated function

SMPSN--integral of given function by Simpson's rule

RK1--integral of first-order differential equation by Runge-Kutta method

RK2--tabulated integral of first-order differential equation by Runge-Kutta method

RK3--tabulated integral of a system of six first-order differential equations by Runge-Kutta method

Fourier Analysis

FORIF--Fourier analysis of a given function

FORIT--Fourier analysis of a tabulated function

Special Operations and Mathematical Functions

GAMMA--gamma function

LEGEN--Legendre polynomial

BESJ--J Bessel function

BESY--Y Bessel function

BESI--I Bessel function

BESK--K Bessel function

CELI1--elliptic integral of the first kind

CELI2--elliptic integral of the second kind

EXPI--exponential integral

SICI--sine cosine integral

CS--Fresnel integrals

Roots of Nonlinear Equations

RTWIT--refine estimate of root by Wegstein's iteration

RTMIT--determine root within a range by Mueller's iteration

RTNIT--refine estimate of root by Newton's iteration

Roots of Polynomial

POLRT--real and complex roots of a real polynomial

Polynomial Operations

PADD--add two polynomials

PADDM--multiply polynomial by constant and add to another polynomial

PCLA--replace one polynomial by another

PSUB--subtract one polynomial from another

PMPY--multiply two polynomials

PDIV--divide one polynomial by another

PQSD--quadratic synthetic division of a polynomial

PVAL--value of a polynomial

PVSUB--substitute variable of polynomial by another polynomial

PCLD--complete linear synthetic division

PILD--evaluate polynomial and its first derivative

PDER--derivative of a polynomial

PINT--integral of a polynomial

PGCD--greatest common divisor of two polynomials

PNORM--normalize coefficient vector of polynomial

OVERALL RULES OF USAGE

SUBROUTINE USAGE

All subroutines in the Scientific Subroutine Package (SSP) are entered by means of the standard FORTRAN CALL statement. These subroutines are purely computational in nature and do not contain any references to input/output devices. The user must therefore furnish, as part of his program, whatever input/output and other operations are necessary for the total solution of his problem. In addition, the user must define by DIMENSION statements all matrices to be operated on by SSP subroutines as well as those matrices utilized in his program. The subroutines contained in SSP are no different from any user-supplied subroutine. All of the normal rules of FORTRAN concerning subroutines must, therefore, be adhered to with the exception that the dimensioned areas in the SSP subroutine are not required to be the same as those in the calling program.

MATRIX OPERATIONS

Special consideration must be given to the subroutines that perform matrix operations. These subroutines have two characteristics that affect the format of the data in storage — variable dimensioning and data storage compression.

Variable Dimensioning

Those subroutines that deal with matrices can operate on any size array limited, in most cases, only by the available core storage and numerical analysis considerations. The subroutines do not contain fixed maximum dimensions for data arrays named in their calling sequence. The variable dimension capability has been implemented in SSP by using a vector storage approach. Under this approach, each column of a matrix is immediately followed in storage by the next column. Vector storage and two-dimensional storage result in the same layout of data in core, so long as the number of rows and columns in the matrix are the same as those in the user's dimension statement. If, however, the matrix is smaller than the dimensioned area, the two forms of storage are not compatible. A subroutine called ARRAY is available in SSP to change from one form of storage to the other. In addition, a subroutine called LOC is available to assist in referencing elements in an array stored in the vector fashion.

Storage Compression

Many subroutines in SSP can operate on compressed forms of matrices, as well as the normal form. Using this capability, which is called "storage mode", considerable savings in data storage can be obtained for special forms of large arrays. The three modes of storage are termed general, symmetric, and diagonal. In this context, general mode is one in which all elements of the matrix are in storage. Symmetric mode is one in which only the upper triangular portion of the matrix is retained column-wise in sequential locations in storage. (The assumption is made that the corresponding elements in the lower triangle have the same value.) Diagonal mode is one in which only the diagonal elements of the matrix are retained in sequential locations in storage. (The off-diagonal elements are assumed to be zero.) This capability has been implemented using the vector storage approach.

A special set of matrix subroutines is included in SSP. These subroutines (GMADD, GMSUB, GMPRD, GMTRA, and GTPRD) execute faster than their counterparts (MADD, MSUB, MPRD, MTRA, and TPRD) because they do not have the storage mode capability.

SAMPLE PROGRAMS

Distributed with the subroutines of SSP are 13 sample main programs with input/output, control (parameter) cards, and sample data. These sample main programs serve two purposes. First, they demonstrate input/output and the use of sequences of subroutines to carry out higher level functions. Secondly, many of the sample programs are useful as they stand. The user need only substitute his own data (in similar format).

There are sample main programs to do each of the following operations (the code names of the main programs are enclosed in parentheses):

1. Data screening (DASCR)
2. Regression (REGRE)
3. Polynomial regression (POLRG)
4. Canonical correlation (MCANO)
5. Analysis of variance, factorial design (ANOVA)
6. Discriminant analysis, many groups (MDISC)
7. Factor analysis (FACTO)
8. Exponential smoothing, third order (EXPON)
9. Matrix addition (ADSAM)

10. Integration of a tabulated function (QDINT)
11. Runge-Kutta solution of dy/dx=f(x, y) (RKINT)
12. Polynomial roots (SMPRT)
13. Simultaneous linear equations (SOLN)

Below is the sample main program QDINT and subroutine QUADR for numerical integration of a tabulated function, together with listings of input and output.

```
SAMPLE PROGRAM FOR INTEGRATION OF A TABULATED FUNCTION BY
NUMERICAL QUADRATURE - QDINT

PURPOSE
   INTEGRATES A SET OF TABULATED VALUES FOR F(X) GIVEN THE
   NUMBER OF VALUES AND THEIR SPACING

REMARKS
   THE NUMBER OF VALUES MUST BE MORE THAN ONE AND THE SPACING
   GREATER THAN ZERO
   I/O LOGICAL UNITS DETERMINED BY MX AND MY.

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
   QUADR

METHOD
   READS CONTROL CARD CONTAINING THE CODE NUMBER, NUMBER OF
   VALUES, AND THE SPACING OF THE FUNCTION VALUES CONTAINED
   ON THE FOLLOWING DATA CARDS.  DATA CARDS ARE THEN READ AND
   INTEGRATION IS PERFORMED.  MORE THAN ONE CONTROL CARD AND
   CORRESPONDING DATA CAN BE INTEGRATED IN ONE RUN.  EXECUTION
   IS TERMINATED BY A BLANK CONTROL CARD.
```

```
C        SAMPLE PROGRAM FOR INTEGRATION OF A TABULATED FUNCTION BY
C        NUMERICAL QUADRATURE - QDINT
C        THE FOLLOWING DIMENSION MUST BE AS LARGE AS THE MAXIMUM NUMBER
C        OF TABULATED VALUES TO BE INTEGRATED
      DIMENSION Z(500)
   10 FORMAT (2I5,F10.0)
   20 FORMAT(////65H INTEGRATION OF TABULATED VALUES FOR DY/DX USING SUB
     1ROUTINE QUADR///11H FUNCTION    ,I5,3X,I5,17H TABULATED VALUES,
     25X,10HINTERVAL =,E15.8//)
   22 FORMAT(/18H ILLEGAL CONDITION/)
   23 FORMAT(/44H NUMBER OF TABULATED VALUES IS LESS THAN TWO)
   24 FORMAT(/27H SPECIFIED INTERVAL IS ZERO)
   30 FORMAT(/7X,31HRESULTANT VALUE OF INTEGRAL IS ,E15.8)
   31 FORMAT(2I2)
   32 FORMAT(7F10.0)
      READ(2,31)MX,MY
   35 READ(MY,10)ICOD,NUMBR,SPACE
      IF(ICOD+NUMBR)70,70,38
   70 STOP
   38 WRITE(MX,20)ICOD,NUMBR,SPACE
      IF(NUMBR-1)100,50,50
   50 READ(MY,32)(Z(I),I=1,NUMBR)
      CALL QUADR(Z,NUMBR,SPACE,ANS,IER)
      IF(IER-1)60,100,200
   60 WRITE(MX,30)ANS
      GO TO 35
  100 WRITE(MX,22)
      WRITE(MX,23)
      GO TO 35
  200 WRITE(MX,22)
      WRITE(MX,24)
      GO TO 35
      END
```

```
      SUBROUTINE QUADR(Z,N,H,S,IER)
      DIMENSION Z(1)
      S=.0
      IF(N-1)32,32,33
   32 IER=1
      RETURN
   33 IF(H)34,35,34
   35 IER=2
      RETURN
   34 IF(N-5)40,38,38
C        CALCULATE USING 5 POINT FORMULA
   38 DO 39 I=5,N,4
   39 S=S+7.*Z(I-4)+32.*Z(I-3)+12.*Z(I-2)+32.*Z(I-1)+7.*Z(I)
      S=S*2./45.
   40 J=N-(N/4)*4+1
      GO TO (45,50,47,48),J
C        CALCULATE USING 4 POINT FORMULA
   45 S=S+.375*(Z(N-3)+3.*Z(N-2)+3.*Z(N-1)+Z(N))
      GO TO 50
C        CALCULATE USING 2 POINT FORMULA
   47 S=S+(Z(N-1)+Z(N))/2.
      GO TO 50
C        CALCULATE USING 3 POINT FORMULA
   48 S=S+(Z(N-2)+4.*Z(N-1)+Z(N))/3.
   50 S=S*H
      IER=0
      RETURN
      END
```

Input for Sample Problem QDINT

```
 1 2
12345   20       1.0
        2.0      2.0      2.0      2.0      2.0      2.0      2.0
        2.0      2.0      2.0      2.0      2.0      2.0      2.0
        2.0      2.0      2.0      2.0      2.0      2.0      2.0
  543   10       1.0
        1.0      2.0      3.0      4.0      5.0      6.0      7.0
        8.0      9.0     10.0     11.0
```

Output of Sample Problem QDINT

```
INTEGRATION OF TABULATED VALUES FOR DY/DX USING SUBROUTINE QUADR

FUNCTION  12345      20 TABULATED VALUES      INTERVAL = 0.10000002E 01

      RESULTANT VALUE OF INTEGRAL IS 0.38000007E 02


INTEGRATION OF TABULATED VALUES FOR DY/DX USING SUBROUTINE QUADR

FUNCTION    543      10 TABULATED VALUES      INTERVAL = 0.10000002E 01

      RESULTANT VALUE OF INTEGRAL IS 0.49500007E 02
```

REQUIRED SYSTEMS

PROGRAMMING SYSTEMS

The subroutines will compile and execute with both the IBM 1130 FORTRAN Compiler and the IBM 1130 Disk Monitor Programming System FORTRAN Compiler.

MACHINE CONFIGURATION

The machine configuration necessary to run SSP/1130 is dependent upon the use that is to be made of the package. All of the subroutines are I/O free, compile to less than 1500 words of core, and are, therefore, configuration independent. However, many of the routines are intended to be used in conjunction with other subroutines or to solve problems using large arrays of data. For this reason, many of the subroutines are not useful with less than 8K words of core.

The following items should be taken into consideration when deciding upon the applicability of the package to a particular machine configuration:

1. The size of problem that may be executed on a given 1130 depends upon the number of subroutines used, the size of the compiled subroutines, the size of the compiled main program, the size of the control program, and the data storage requirements.
2. SSP/1130 programs will be distributed in card form only.
3. Several of the sample problems require 8K words of core and the use of the Disk Monitor, and the remaining sample problems require 8K words of core.

It is possible to estimate program sizes by using the data processing techniques manual entitled Core Requirements for 1130 FORTRAN (C20-1641) in conjunction with the program listings found in the programmer's manual entitled System/360 Scientific Subroutine Package (H20-0205).

PRECISION

The accuracy of the computations in many of the SSP subroutines is highly dependent upon the number of significant digits available for arithmetic operations. Matrix inversion, integration, and many of the statistical subroutines fall into this category. By inserting one FORTRAN control card (*EXTENDED PRECISION) into a subroutine, the mantissa length is increased from 24 to 31 bits.

APPENDIX A: STORAGE REQUIREMENTS

The following table lists the number of characters of storage required by each of the subroutines in the Scientific Subroutine Package. The figures given were obtained by using the IBM 1130 Disk Monitor Programming System FORTRAN Compiler (the figures are preliminary and may be revised). The subroutines or function subprograms called or used by a given subroutine are also noted.

Name	Function	Storage Required (Words)
STATISTICS		
Data Screening		
TALLY	totals, means standard deviations, minimums, maximums	352
BOUND	selection of observations within bounds	202
SUBST	subset selection from observation matrix (needs user's Boolean subroutine)	224
ABSNT	detection of missing data	92
TAB1	tabulation of data (1 variable)	550
TAB2	tabulation of data (2 variables)	1030
SUBMX	build subset matrix	132
Elementary Statistics		
MOMEN	first four moments	396
TTSTT	tests on population means	532
Correlation		
CORRE	means, standard deviations, and correlations (needs user's subroutine to get data)	1098
Multiple Linear Regression		
(Usually requires CORRE, ORDER, MINV, MULTR in sequence)		
ORDER	rearrangement of intercorrelations	204
MULTR	multiple regression and correlation	512

Name	Function	Storage Required (Words)
Polynomial Regression		
(Usually requires GDATA, ORDER, MINV, MULTRA in sequence)		
GDATA	data generation	646
Canonical Correlation		
(Usually requires CORRE, CANOR, MINV, NROOT, EIGEN in sequence)		
CANOR	canonical correlation (CANOR calls MINV and NROOT)	1110
NROOT	eigenvalues and eigenvectors of a special nonsymmetric matrix (NROOT calls EIGEN)	734
Analysis of Variance		
(Usually requires AVDAT, AVCAL, MEANQ in sequence)		
AVDAT	data storage allocation	318
AVCAL	Σ and Δ operation	264
MEANQ	mean square operation	548
Discriminant Analysis		
(Usually requires DMATX, MINV, DISCR in sequence)		
DMATX	means and dispersion matrix	412
DISCR	discriminant functions	948
Factor Analysis		
(Usually requires CORRE, EIGEN, TRACE, LOAD, VARMX in sequence)		
TRACE	cumulative percentage of eigenvalues	158
LOAD	factor loading	96
VARMX	varimax rotation	1170

Name	Function	Storage Required (Words)
Time Series		
AUTO	autocovariances	162
CROSS	crosscovariances	218
SMO	application of filter coefficients (weights)	162
EXSMO	triple exponential smoothing	246
Nonparametric Statistics		
CHISQ	x^2 test for a contingency table	520
UTEST	Mann-Whitney U-test (UTEST calls RANK and TIE)	240
TWOAV	Friedman two-way analysis of variance (TWOAV calls RANK)	218
QTEST	Cochran Q-test	228
SRANK	Spearman rank correlation (SRANK calls RANK and TIE)	376
KRANK	Kendall rank correlation (KRANK calls RANK and TIE)	510
WTEST	Kendall coefficient of concordance (WTEST calls RANK and TIE)	490
RANK	rank observations	220
TIE	calculation of ties in ranked observations	194
Random Number Generators		
RANDU	uniform random numbers	52
GAUSS	normal random numbers GAUSS calls RANDU	68
MATRIX MANIPULATIONS		
MINV	matrix inversion	768
EIGEN	eigenvalues and eigenvectors of a real, symmetric matrix	1050
SIMQ	solution of simultaneous linear, algebraic equations	532
GMADD	add two general matrices	50
GMSUB	subtract two general matrices	50
GMPRD	product of two general matrices	154
GMTRA	transpose of a general matrix	86
GTPRD	transpose product of two general matrices	150
MADD	add two matrices (calls LOC)	224
MSUB	subtract two matrices (calls LOC)	224
MPRD	matrix product (row into column) (calls LOC)	226
MTRA	transpose a matrix (calls MCPY)	106
TPRD	transpose product (calls LOC)	226
MATA	transpose product of matrix by itself (calls LOC)	192
SADD	add scalar to matrix (calls LOC)	56
SSUB	subtract scalar from a matrix (calls LOC)	56
SMPY	matrix multiplied by a scalar (calls LOC)	56
SDIV	matrix divided by a scalar (calls LOC)	66
RADD	add row of one matrix to row of another matrix (calls LOC)	90
CADD	add column of one matrix to column of another matrix (call LOC)	92
SRMA	scalar multiply row and add to another row	106
SCMA	scalar multiply column and add to another column	110
RINT	interchange two rows	92
CINT	interchange two columns	96
RSUM	sum the rows of a matrix (calls LOC)	82

Name	Function	Storage Required (Words)
CSUM	sum the columns of a matrix (calls LOC)	96
RTAB	tabulate the rows of a matrix (calls LOC, RADD)	194
CTAB	tabulate the columns of a matrix (calls LOC, CADD)	194
RSRT	sort matrix rows (calls LOC)	294
CSRT	sort matrix columns (calls LOC and CCPY)	294
RCUT	partition row-wise (calls LOC)	160
CCUT	partition column-wise (calls LOC)	160
RTIE	adjoin two matrices row-wise (calls LOC)	176
CTIE	adjoin two matrices column-wise (calls LOC)	162
MCPY	matrix copy (calls LOC)	54
XCPY	copy submatrix from given matrix (calls LOC)	126
RCPY	copy row of matrix into vector (calls LOC)	78
CCPY	copy column of matrix into vector (calls LOC)	78
DCPY	copy diagonal of matrix into vector (calls LOC)	56
SCLA	matrix clear and add scalar (calls LOC)	52
DCLA	replace diagonal with scalar (calls LOC)	52
MSTR	storage conversion (calls LOC)	114
MFUN	matrix transformation by a function	70
RECP	reciprocal function for MFUN	44
LOC	location in compressed-stored matrix	108
ARRAY	vector storage--double dimensioned conversion	194

Name	Function	Storage Required (Words)
OTHER MATHEMATICS		
Integration		
QUADR	integral of tabulated function	152
SMPSN	integral of given function by Simpson's rule (needs user's function subprogram)	342
RK1	integral of first-order differential equation by Runge-Kutta method (needs user's function subprogram)	432
RK2	tabulated integral of first-order differential equation by Runge-Kutta method (needs user's function subprogram)	208
RK3	tabulated integral of a system of six first-order differential equations by Runge-Kutta method (needs six user's function subprograms)	990
Fourier Analysis		
FORIF	Fourier analysis of a given function (needs user's function subprogram)	292
FORIT	Fourier analysis of a tabulated function	284
Special Operations and Mathematical Functions		
GAMMA	gamma function	232
LEGEN	Legendre polynomial	268
BESJ	J Bessel function	438
BESY	Y Bessel function	824
BESI	I Bessel function	376
BESK	K Bessel function	804
CELI1	elliptic integral of the first kind	138
CELI2	elliptic integral of the second kind	212
EXPI	exponential integral	262

Name	Function	Storage Required (Words)
SICI	sine cosine integral	366
CS	Fresnel integrals	310
Roots of Nonlinear Functions		
RTWIT	refine estimate of root of Wegstein's iteration (needs user's function subprogram)	264
RTMIT	determine root within a range by Mueller's iteration (needs user's function subprogram)	454
RTNIT	refine estimate of root by Newton's iteration (needs user's function subprogram)	222
Roots of Polynomial		
POLRT	real and complex roots of polynomial	792
Polynomial Operations		
PADD	add two polynomials	110
PADDM	multiply polynomial by constant and add to another polynomial	116
PCLA	replace one polynomial by another	48
PSUB	subtract one polynomial from another	110
PMPY	multiply two polynomials	140
PDIV	divide one polynomial by another (calls PNORM)	196
PQSD	quadratic synthetic division of a polynomial	78
PVAL	value of a polynomial	54
PVSUB	substitute variable of polynomial by another polynomial (calls PMPY, PADDM, PCLA)	132
PCLD	complete linear division	74
PILD	evaluate polynomial and its first derivative (calls PQSD)	56
PDER	derivative of a polynomial	86
PINT	integral of a polynomial	86
PGCD	greatest common divisor of two polynomials (calls PDIV and PNORM)	106
PNORM	normalize coefficient vector of polynomial	46

APPENDIX B: TIMING

Two examples of timing are given.

1. Sample program SOLN was chosen to illustrate the overall timing of a problem. In all cases, the 1442 Card Read Punch, Model 7, is used for input and all necessary subroutines are already on disk.
   a. Compile time, using a LIST ALL card (gives a program listing of its 56 cards, and a memory map which includes variable allocations, statement allocations, features supported, called subprograms, integer constants, and core requirements), requires 1 minute 32 seconds on the 1132 Printer. (Compile time, minus the LIST ALL card, requires 36 seconds.)
   b. It takes ten seconds to store the program on disk.
   c. After the XEQ control card is read, the computer uses 17 seconds to locate the necessary main program and subprograms and load them in core.
   d. It takes ten seconds to read the data cards.
   e. The solution time is four seconds.
   f. Output printing time is 53 seconds on an 1132 Printer. (Output printing time is 3 minutes 32 seconds on the console typewriter.)
2. To illustrate the computational time used by an IBM 1130 Computing System, the following program was selected:

```
  DIMENSION A(1600),L(40),M(40)
  IX=3
2 PAUSE 1
  DO 1 I=1,1600
  CALL RANDU (IX,IY,Y)
  IX=IY
1 A(I)=Y
  PAUSE 2
  CALL MINV(A,10,D,L,M)
  PAUSE 3
  CALL MINV(A,15,D,L,M)
  PAUSE 4
  CALL MINV(A,20,D,L,M)
  PAUSE 5
  CALL MINV(A,30,D,L,M)
  PAUSE 6
  CALL MINV(A,40,D,L,M)
  PAUSE 7
  GO TO 2
  END
```

   a. RANDU - random number generator subroutine.
      To generate 1600 numbers, using subroutine RANDU, execution time is 5 seconds.
   b. MINV - matrix inversion subroutine.
      Matrix inversion, using subroutine MINV, is performed on five different-size matrices, with the following results in execution time:
      1. 10 x 10 matrix uses 4 seconds
      2. 15 x 15 matrix uses 12 seconds
      3. 20 x 20 matrix uses 27 seconds
      4. 30 x 30 matrix uses 1 minute 28 seconds
      5. 40 x 40 matrix uses 3 minutes 27 seconds

REFERENCE MATERIAL

IBM 1130 FORTRAN Language (C26-5933).

IBM 1130 Monitor System, Reference Manual (C26-3750).

IBM 1130 Card/Paper Tape Programming System, Operator's Guide (C26-3629).

Core Requirements for 1130 FORTRAN (C20-1641).

Printed in U.S.A. H20-0225-0

IBM®

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]