List of Contents

Multics Programmers' Manual
Installation-Maintained Library
7/11/74

Commands

8/21/72	bmd
4/18/72	convert_360_fortran
7/11/74	list_help
1/27/72	online_consultant
2/5/73	trouble_report

Subroutines

10/30/72	forssp
9/20/71	plssp

bmd

Command
Installation-Maintained Library
8/21/72

Name: bmd

Twelve of the BMD (BioMeDical) programs are available on Multics. They are statistical routines developed by the UCLA Health Sciences Computing Facility and written in Fortran IV. Complete instructions for usage of the programs are available in "Biomedical Computer Programs," edited by W. J. Dixon, University of California Publications in Automatic Computation, No.2. Berkeley: University of California Press, 1971. This memo contains a list of the routines available and a brief description of each one.

The programs are in ">libraries>bmd," and this must be in the user's search path (i.e., enter the command "ssd >libraries>bmd").

Usage:

bmdxxx

where xxx is the particular routine desired.

Input for the routine, both control cards and data as described in the BMD manual, must be in a file called "file07". The control cards must be in the fixed format the manual specifies. Keywords in control cards, such as "PROBLM," must be in all lower case letters, rather than upper case as the manual describes. The user may wish to keep the control cards in a separate file from the data. In this case, merge the data into the proper place in the control file using the merge subcommand of edm before execution. Errors in input of this file by the BMD programs may result if the file is not closed, so the user must be sure the file is properly closed before execution of any BMD program.

Several of the routines create scratch files as they run. The user should be careful not to have files with these names. After the program is finished these scratch files should be deleted. The routines with scratch files and the names of the files are:

```
bmd02d    file01
bmd03m    file03
bmd02r    file01,file02
bmd06s    file01,file02,file03
```

The following table lists the routines available:

bmd01d Simple Data Description--computes simple averages and measures of dispersion of variables.

bmd02d Correlation with Transgeneration--computes simple correlation coefficients, averages, and measures of dispersion on entering or transgenerated variables from selected cases.

bmd03d Correlation with Item Deletion--computes correlation matrix, ommitting specified values of variables. The punch card option is not available.

bmd07d Description of Strata with Histograms--groups data and prints histograms for each variable; means, standard deviations, and correlation coefficients are computed for each group.

bmd08d Cross-Tabulation with Variable Stacking--computes two way frequency tables of integer data.

bmd03m General Factor Analysis--performs a principal component solution and an orthogonal rotation of the factor matrix.

bmd05m Discriminant Analysis for Several Groups--computes a set of linear functions to classify cases into groups.

bmd02r Stepwise Regression--computes a sequence of multiple linear regression equations in a stepwise manner.

bmd06s This program performs Guttman Scale analysis. bmd07s and bmd08s have been made subroutines of bmd06s. The punch card option is not available. The first line in the data file should be as follows:

Col. 1 7 if the program should continue with bmd07s, otherwise 0 to stop after bmd06s.

Col. 2 8 if the program should continue with bmd08s, otherwise 0 to stop after bmd07s.

The next line of data should be the PROBLM card as described in the BMD manual.

bmd02t Autocovariance and Power Spectral Analysis--computes autocovariance, power spectrum, cross-covariance, cross-spectrum, transfer function, and coherence function of time series.

Command
Installation-Maintained Library
4/26/72

Name: convert_360_fortran, c360f

This command converts an IBM FORTRAN IV program as read in by the Multics card reader to a form acceptable to the Multics FORTRAN compiler.

Usage:

convert_360_fortran path1 path2

Path1 is the path or entry name of the segment to be converted. This segment name must end in the suffix ".FORTRAN", but the suffix need not be typed in the command. path2 is optional and, if supplied, is the name of the resulting converted segment. This name must have the suffix ".fortran", but, again, the suffix need not be typed. If path2 is omitted, the output segment will have the same name as the input, except that the suffix will be changed from ".FORTRAN" to ".fortran".

Operation:

convert_360_fortran makes the following changes to the input segment to produce the output segment:

1) All upper case letters appearing outside of literals of commentary are changed to lower case.

2) All single quotes (') appearing outside of literals are changed to double quotes ("). Pairs of single quotes appearing in literals delimited by single quotes are changed to one single quote; double quotes appearing in such literals are replicated. Single quotes appearing in literals delimited by nH are not changed.

3) Trailing blanks and the contents of what were card columns 73-80 are eliminated. Trailing blanks which are a part of a literal are preserved.

4) The card continuation convention is changed to the Multics continuation convention.

5) All declaration statements are moved to a position in the segment immediately following the program header, if any. As a result, only comments, SUBROUTINE, FUNCTION and BLOCK DATA statements will precede them. The statements moved are: INTEGER, REAL, DOUBLE PRECISION, COMPLEX, LOGICAL, DIMENSION, and COMMON.

   The effect of this is to ensure that DATA and EQUIVALENCE statements appear in the position required by MULTICS FORTRAN, that is, following the above statements. For purposes or rearrangement, commentary is assumed to be associated with the next following non-comment line.

6) Length specifications in declarations are removed. REAL*8 is changed to "double precision."

Restrictions:

convert_360_fortran makes no claim to translate all the incompatibilities between IBM FORTRAN IV and Multics FORTRAN; it does only those translations specifically listed above. Suggestions for additional translations will be considered.

The continuatiion convention translation is effected in most cases simply be deleting the continuation character on the input card image. When a literal crossing card boundaries is encountered, the two card images are combined in a single line so that the literal is acceptable to Multics FORTRAN. This is the only case where this is done, therefore programs which have variables or keywords or operators broken across card boundaries will not be translated effectively.

It should be noted that a program which contains statements unnecessarily continued on additional cards may be translated incorrectly. Specifically, a statement must be classifiable on its first card (e.g., the equals sign of an assignment statement must occur on the first card).

Also note that the conversion from REAL*8 is not rigorous. REAL*8 XX, YY becomes "double precision xx, yy"; REAL FUNCTION ZZ*8 becomes "double precision function zz"; but REAL XX, YY*8, ZZ*4 becomes "real xx, yy, zz."

Command
Installation-Maintained Library
07/11/74

Names: list_help,lh

The list_help command may be used to find help (info) files on specified topics.

Usage: list_help argl ... argn

All arguments which are not control arguments are taken as "topics" to be searched for. All files whose names contain one of the topics will be listed.

Control arguments:

-all	list the names of all help files rather than looking for topics
-brief	list primary names only
-pathname xxx	search specified directory instead of the directories used by the help command

Examples:

To print the names of files on fortran: list_help fortran

To look for lisp and macsyma: lh lisp macsyma

To list all info files in >udd>A>B: lh -pn >udd>A>B -all

Notes:

1) The default directories to be searched are:

>documentation>info_segments
>documentation>iml_info_segments

2) The names of help (info) files always end with ".info", but list_help does not print this suffix, nor is it specified when printing the contents of a file using the help command.

Command
Installation-Maintained Library
1/27/72

Name: online_consultant, olc

This experimental command allows a user to type a message at his terminal which will be transmitted to a logged-in IPC staff member immediately. The staff member may then reply to the user via the "ipc_message_facility."

An IPC staff member is scheduled to be logged in every weekday between the hours of 2 p.m. and 5 p.m. This schedule will be modified if user demand and personnel availability dictate. Outside of the scheduled hours, staff members may sign on as consultant (they are encouraged to do so whenever convenient), but the Center cannot promise an on-line consultant outside scheduled hours.

Usage:

There are two ways to use this command:

online_consultant argument(s)

or

online_consultant

If arguments are supplied, a single message consisting of the arguments, concatenated and separated by blanks, will be transmitted to the current consultant.

If no argument is supplied, online_consultant will respond

Input

and read lines from the console. Each line will be sent to the current consultant as a separate message as soon as it is completed. The consultant's replies may, of course, be interspersed between message input lines. To exit from this message-input mode, the user types a line consisting of only a period (.).

Notes:

The on-line consultant will be attempting to complete his normal Multics work while signed on. Sometimes, this may mean that the messages sent to him are delayed until his program completes execution, so please be patient if your question is not answered right away.

If no consultant is signed on, the command will determine whether a consultant is scheduled or not. If not, a short message describing other means of bug reporting will be typed. If a consultant should be signed on, the command will attempt to locate a backup consultant or will type a message giving a telephone extension to call.

The online_consultant command calls the "ipc_message_facility" command "accept_messages" as it is exiting, to allow the consultant to reply. See the AML writeup for "ipc_message_facility" for details.

Command
Installation-Maintained Library
2/5/73

Names: trouble_report, tr

The command trouble_report is used to report usage problems to the programming consultants at the Information Processing Center. Invoking the command provides a series of prompting messages designed to assure that all the necessary information is reported. The brief option, with only one prompting message, is for experienced users of trouble_report.

Usage: trouble_report or tr

This will cause pertinent questions concerning the user and the problem to be typed out on the console. The user should respond with the requested information or answer yes or no and then hit the carriage return once. If the user fails to give the requested information, he will receive additional prompting.

After trouble_report asks the user to describe his programming problem, it accepts lines from the console until only a period (".") is typed at the beginning of a line. The user is requested to send any source and assembly listings or debugging attempts to Consultant, Room 39-454. The path names of the problem segments should be given and the ACL should be set to re *.*.*

Usage: tr -brief or tr -bf

This types only "Input." Tr then accepts lines from the console until a period (".") is typed at the beginning of a line.

The temporary segment containing all the positive responses to the questions is sent to Consultant IPC by using the mail command. The segment is deleted.

If deemed expedient, the reply to the trouble report will be sent to the user's mailbox. In order to permit this, the user should see that he has a segment in his login directory named "mailbox" and check that the ACL is set to rewa *.*.* If a user wants to restrict the access to his mailbox, the following projects should have at least "rwa" access to the mailbox and at least "e" access to the "home" directory: SysMaint, Multics, PDO, SysLib, and Consult.

When a user wants to restrict the access to the path names of segments that can be used in debugging the problem, he should provide "re" access to all the project groups mentioned in the previous paragraph on the ACL of the directory and on each relevant segment. As an alternative, he could set the CACL to "re" if many segments are involved.

Subroutine
Installation-Maintained Library
10/30/72

Name: forssp

Sixteen of the FORTRAN Scientific Subroutine Package (SSP) groups of subroutines are available on Multics. They are Data Screening, Correlation and Regression, Design Analysis, Discriminant Analysis, Factor Analysis, Time Series, Nonparametric Statistics, Generation of Random Variates--Distribution Functions, Elementary Statistics and Miscellany, Matrices: Storage, Matrices: Operations, Matrices: Inversion, Systems of Linear Equations and Related Topics, Matrices: Eigenanalysis and Related Topics, Polynomials: Operations, Polynomials: Roots, and Polynomials: Special Types. Instructions for usage of the programs are available in the IBM System/360 Scientific Subroutine Package Programmer's Manual (GH20-0205). This memo provides a listing of all the subroutines (in the same order as in the SSP manual) with a brief description of the function of each one.

Double precision versions have been created for most of the routines which did not already have them. These routines are starred in this memo and are not found in the SSP manual. They work exactly like their single precision counterparts except for the typing of their arguments. All real arguments of these routines must be double precision except as indicated in the following descriptions.

The subroutines are in ">libraries>forssp", and this must be in the user's search path (i.e., enter the command "ssd >libraries>forssp").

Usage:

call subroutine_name (arg)

The subroutines available are:

Data Screening

tally	*dtally	bound	*dbound	subst	*dsubst
absnt	*dabsnt	tab1	*dtab1	tab2	*dtab2
submx	*dsubmx				

Correlation and Regression

corre	*dcorre	misr	*dmisr	order	*dorder
multr	*dmultr	gdata	*dgdata	stprg	*dstprg
probt	*dprobt	canor	*dcanor		

Design Analysis

avdat	*davdat	avcal	*davcal	meanq	*dmeanq

Discriminant Analysis

dmatx	*ddmatx	disc	*ddisc

Factor Analysis

trace	*dtrace	load	*dload	varmx	*dvarmx

Time Series

auto	*dauto	cross	*dcross	smo	*dsmo
exsmo	*dexsmo				

Non-Parametric Statistics

kolmo	kolm2	smirn	*dsmirn	chisq	*dchisq
krank	*dkrank	mpair	*dmpair	qtest	*dqtest
rank	*drank	signt	*dsignt	srank	*dsrank
tie	*dtie	twoav	*dtwoav	utest	*dutest
wtest	*dwtest				

General Random Variates--Distribution Functions

gauss	ndtr	*dndtr	bdtr	cdtr	ndtri
*dndtri					

Elementary Statistics and Miscellany

momen	*dmomen	ttest	*dttest	biser	*dbiser
phi	*dphi	point	*dpoint	tetra	*dtetra
srate	*dsrate				

Matrices: Storage

mcpy	*dmcpy	rcpy	*drcpy	ccpy	*dccpy
dcpy	*ddcpy	xcpy	*dxcpy	mstr	*dmstr
loc	convt	array	*darray		

Matrices: Operations

gmadd	*dgmadd	gmsub	*dgmsub	gmprd	*dgmprd
gmtra	*dgmtra	gtprd	*dgtprd	madd	*dmadd
msub	*dmsub	mprd	*dmprd	mtra	*dmtra
tprd	*dtprd	mata	*dmata	sadd	*dsadd
ssub	*dssub	smpy	*dsmpy	sdiv	*dsdiv
scla	*dscla	dcla	*ddcla	radd	*dradd
cadd	*dcadd	srma	*dsrma	scma	*dscma
rint	*drint	cint	*dcint	rsum	*drsum
csum	*dcsum	rtab	*drtab	ctab	*dctab
rsrt	*drsrt	csrt	*dcsrt	rcut	*drcut
ccut	*dccut	rtie	*drtie	ctie	*dctie
mprc	*dmprc	mfun	*dmfun	recp	*drecp

Matrices: Inversion, Systems of Linear Equations and Related Topics

minv	*dminv	sinv	dsinv	simq	*dsimq
gelg	dgelg	rslmc	*drslmc	factr	*dfactr
mfgr	dmfgr	gels	dgels	gelb	dgelb
mtds	dmtds	mlss	dmlss	mchb	dmchb
mfss	dmfss	mfsd	dmfsd	llsq	dllsq

Matrices: Eigenanalysis and Related Topics

eigen	*deigen	nroot	*dnroot	ateig	*dateig
hsbg	*dhsbg				

Polynomials: Operations

padd	*dpadd	psub	*dpsub	pmpy	*dpmpy
pdiv	*dpdiv	pcla	*dpcla	paddm	*dpaddm
pval	*dpval	pvsub	*dpvsub	pild	*dpild
pder	*dpder	pint	*dpint	pqsd	*dpqsd
pcld	*dpcld	pgcd	*dpgcd	pnorm	*dpnorm
pecn	dpecn	pecs	dpecs		

Polynomials: Roots

polrt	*dpolrt	prqd	dprqd	prbm	dprbm
pqfb	dpqfb				

Polynomials: Special Types

cnp	dcnp	cnps	dcnps	tcnp	dtcnp
csp	dcsp	csps	dcsps	tcsp	dtcsp
hep	dhep	heps	dheps	thep	dthep
lap	dlap	laps	dlaps	tlap	dtlap
lep	dlep	leps	dleps	tlep	dtlep

Miscellaneous

dlgam	hist	plot

Data Screening

tally	to calculate total, mean, standard deviation, minimum, and maximum for each variable.
dtally *	double precision version of tally. Only inputs a, total, aver, and sd are double precision.
bound	to select the number of observations above, below, and between two bounds for each variable.
dbound *	double precision version of bound.
subst	to select a subset from an observation matrix.
dsubst *	double precision version of subst.
absnt	to tabulate missing or zero-values in an observation matrix.
dabsnt *	double precision version of absnt.
tab1	to compute frequency tabulation for one variable.
dtab1 *	double precision version of tab1.
tab2	to compute frequency tabulations for two variables.
dtab2 *	double precision version of tab2.
submx	to select a submatrix on the basis of results from absnt or subst.
dsubmx *	double precision version of submx.

Correlation and Regression

corre	to calculate means, standard deviations, and correlation coefficients.
dcorre *	double precision version of corre.
misr	to calculate means, standard deviations, third and fourth moments, correlations, regression coefficients, and regression errors. Considers that data may be missing.
dmisr *	double precision version of misr.
order	to rearrange intercorrelations.
dorder *	double precision version of order.
multr	to perform multiple linear regression.
dmultr *	double precision version of multr.
gdata	to generate data matrix for polynomial regression.
dgdata *	double precision version of gdata.
stprg	to perform stepwise multiple linear regression.
dstprg *	double precision version of stprg.
probt	to perform probit analysis.
dprobt *	double precision version of probt.
canor	to perform canonical correlation.
dcanor *	double precision version of canor.

Design Analysis

avdat	to prepare data storage for analysis of variance.
davdat *	double precision version of avdat.
avcal	to perform the calculus for the general k-factorial experiment: operators Σ and Δ .
davcal *	double precision version of avcal.
meanq	to perform mean square operation.
dmeanq *	double precision version of meanq.

Discriminant Analysis

dmatx	to calculate means and dispersion matrix.
ddmatx *	double precision version of dmatx.
disc	to calculate discriminant functions.
ddisc *	double precision version of disc.

Factor Analysis

trace	to compute cumulative percentage of eigenvalues.
dtrace *	double precision version of trace.
load	to compute factor matrix from eigenvalues and associated eigenvectors.
dload *	double precision version of load.
varmx	to perform an orthogonal rotation of a factor matrix.
dvarmx *	double precision version of varmx.

Time Series

auto	to compute autocovariances.
dauto *	double precision version of auto.
cross	to compute cross covariances.
dcross *	double precision version of cross.
smo	to calculate smoothed or filtered series.
dsmo *	double precision version of smo.
exsmo	to calculate triple exponential smoothed series.
dexsmo *	double precision version of exsmo.

Non-Parametric Statistics

kolmo	to perform Kolmogorov-Smirnov one sample test. See program comments for double precision usage.
kolm2	to perform Kolmogorov-Smirnov two sample test. See program comments for double precision usage.
smirn	to compute values of the Kolmogorov-Smirnov limiting distribution. This program was modified to prevent exponent underflows.
dsmirn *	double precision version of smirn.
chisq	to compute chi-square for a contingency table.
dchisq *	double precision version of chisq.
krank	to compute the Kendall rank correlation coefficient.
dkrank *	double precision version of krank.
mpair	to perform Wilcoxin's signed ranks test.
dmpair *	double precision version of mpair.
qtest	to calculate Cochran's Q.
dqtest *	double precision version of qtest.
rank	to rank a vector of values.
drank *	double precision version of rank.
signt	to perform non-parametric sign test.
dsignt *	double precision version of signt.
srank	to calculate Spearman rank correlation coefficients.
dsrank *	double precision version of srank.
tie	to calculate correction factor due to ties during ranking.

dtie *	double precision version of tie.
twoav	to calculate Friedman two-way analysis of variance statistics.
dtwoav *	double precision version of twoav.
utest	to perform the Mann-Whitney U-test.
dutest *	double precision version of utest.
wtest *	to compute the Kendall coefficient of concordance.
dwtest *	double precision version of wtest.

General Random Variates--Distribution Functions

gauss	to calculate normally distributed random numbers.
ndtr	to compute normal distribution function.
dndtr *	double precision version of ndtr.
bdtr	to compute beta distribution function.
cdtr	to compute chi-square distribution function.
ndtri	to compute inverse of normal distribution function.
dndtri *	double precision version of ndtri.

Elementary Statistics and Miscellany

momen	to calculate first four moments.
dmomen *	double precision version of momen.
ttest	to test population means.
dttest *	double precision version of ttest.
biser	to compute biserial correlation coefficients.
dbiser *	double precision version of biser.
phi	to calculate phi coefficients.
dphi *	double precision version of phi.
point	to compute point-biserial correlation coefficients.
dpoint *	double precision version of point.
tetra	to calculate tetrachoric correlation coefficients.
dtetra *	double precision version of tetra.
srate	to compute survival rates.
dsrate *	double precision version of srate.

Matrices: Storage

mcpy	to copy an entire matrix.
dmcpy *	double precision version of mcpy.
rcpy	to copy a specified row of a matrix into a vector.
drcpy *	double precision version of rcpy.
ccpy	to copy a specified column of a matrix into a vector.
dccpy *	double precision version of ccpy.
dcpy	to copy diagonal elements of a matrix into a vector.
ddcpy *	double precision version of dcpy.
xcpy	to copy a portion of a matrix.
dxcpy *	double precision version of xcpy.
mstr	to change the storage mode of a matrix.
dmstr *	double precision version of mstr.
loc	to compute a vector subscript for an element in a matrix of specified storage mode.
convt	to convert numbers from single precision to double precision or vice versa.
array	to convert a data array from single to double dimension or vice versa. This subroutine is used to link the user program, which has double dimension arrays, and the SSP subroutines, which operate on arrays of data in a vector fashon.
darray *	double precision version of array.

Matrices: Operations

gmadd	to add two general matrices to form a resultant general matrix.
dgmadd *	double precision version of gmadd.
gmsub	to subtract one general matrix from another to form a resultant matrix.
dgmsub *	double precision version of gmsub.
gmprd	to multiply two general matrices to form a resultant general matrix.
dgmprd *	double precision version of gmprd.
gmtra	to transpose a general matrix.
dgmtra *	double precision version of gmtra.
gtprd	to premultiply a general matrix by the transpose of another general matrix.
dgtprd *	double precision version of gtprd.
madd	to add two matrices element by element to form a resultant matrix.
dmadd *	double precision version of dmadd.
msub	to subtract two matrices element by element to form a resultant matrix.
dmsub *	double precision version of msub.
mprd	to multiply two matrices to form a resultant matrix.
dmprd *	double precision version of mprd.
mtra	to transpose a matrix.

dmtra *	double precision version of mtra.
tprd	to transpose a matrix and postmultiply by another to form a resultant matrix.
dtprd *	double precision version of tprd.
mata	to premultiply a matrix by its transpose to form a symmetric matrix.
dmata *	double precision version of mata.
sadd	to add a scalar to each element of a matrix to form a resultant matrix.
dsadd *	double precision version of sadd.
ssub	to subtract a scalar from each element of a matrix to form a resultant matrix.
dssub *	double precision version of ssub.
smpy	to multiply each element of a matrix by a scalar to form a resultant matrix.
dsmpy *	double precision version of smpy.
sdiv	to divide each element of a matrix by a scalar to form a resultant matrix.
dsdiv *	double precision version of sdiv.
scla	to set each element of a matrix equal to a given scalar.
dscla *	double precision version of scla.
dcla	to set each diagonal element of a matrix equal to a scalar.
ddcla *	double precision version of dcla.
radd	to add a row of one matrix to a row of another matrix.
dradd *	double precision version of radd.

cadd	to add a column of one matrix to a column of another matrix.
dcadd *	double precision version of cadd.
srma	to multiply a row of a matrix by a scalar and to add to another row of the same matrix.
dsrma *	double precision version of srma.
scma	to multiply a column of a matrix by a scalar and to add to another column of the same matrix.
dscma *	double precision version of scma.
rint	to interchange two rows of a matrix.
drint *	double precision version of rint.
cint	to interchange two columns of a matrix.
dcint *	double precision version of cint.
rsum	to sum the elements of each row of a matrix to form a column vector.
drsum *	double precision version of rsum.
csum	to sum the elements of each column of a matrix to form a row vector.
dcsum *	double precision version of csum.
rtab	to tabulate rows of a matrix to form a summary matrix.
drtab *	double precision version of rtab.
ctab	to tabulate columns of a matrix to form a summary matrix.
dctab *	double precision version of ctab.
rsrt	to sort rows of a matrix.
drsrt *	double precision version of rsrt.

csrt	to sort columns of a matrix.
dcsrt *	double precision version of csrt.
rcut	to partition a matrix between specified rows to form two resultant matrices.
drcut *	double precision version of rcut.
ccut	to partition a matrix between specified columns to form two resultant matrices.
dccut *	double precision version of ccut.
rtie	to adjoin two matrices with the same column dimension to form one resultant matrix.
drtie *	double precision verison of rtie.
ctie	to adjoin two matrices with the same row dimension to form one resultant matrix.
dctie *	double precision version of ctie.
mprc,dmprc	to permute the rows or columns of a given matrix according to a given transposition vector or its inverse.
mfun	to apply a function to each element of a matrix to form a resultant matrix.
dmfun *	double precision version of mfun.
recp	to calculate the reciprocal of an element. This is a FORTRAN function subprogram that may be used as an argument by the subroutine MFUN.
drecp *	double precision version of recp.

Matrices: Inversion, Systems of Linear Equations and Related Topics

minv	to invert a square matrix.
dminv *	double precision version of minv.
sinv,dsinv	to invert a symmetric positive definite matrix.
simq	to solve simultaneous linear algebraic equations.
dsimq *	double precision version of simq.
gelg,dgelg	to solve a system of general simultaneous linear equations by Gauss elimination.
rslmc	to solve simultaneous linear equations with iterative refinement.
drslmc *	double precision version of rslmc. Inputs a, af, b, x, v, and per should be double precision. Input epsi should be single precision.
factr	to compute a triangular factorization of a nonsingular matrix.
dfactr *	double precision version of factr.
mfgr,dmfgr	to factor a matrix and determine its rank.
gels,dgels	to solve a system of general simultaneous linear equations with symmetric coefficients.
gelb,dgelb	to solve a system of general simultaneous linear equations with band-structured coefficients.
mtds,dmtds	to divide a matrix by a triangular matrix.
mlss,dmlss	to solve simultaneous linear equations with symmetric positive semidefinite matrix.
mchb,dmchb	to compute a triangular factorization of a symmetric positive definite band matrix.

mfss,dmfss	to compute a triangular factorization and determine the rank of a symmetric positive semidefinite matrix.
mfsd,dmfsd	to compute a triangular factorization of a symmetric positive definite matrix.
llsq,dllsq	to solve linear least squares problem.

Matrices: Eigenanalysis and Related Topics

eigen	to compute eigenvalues and eigenvectors for a real symmetric matrix.
deigen *	double precision version of eigen.
nroot	to calculate eigenvalues and eigenvectors for a real nonsymmetric matrix of the form B-inverse times A.
dnroot *	double precision version of nroot.
hsbg	to reduce a real matrix into upper almost triangular form (hessenberg form).
dhsbg *	double precision version of hsbg.
ateig	to compute eigenvalues for a real almost triangular matrix.
dateig *	double precision version of ateig.

Polynomials: Operations

padd	to add two polynomials.
dpadd *	double precision version of padd.
psub	to subtract two polynomials.
dpsub *	double precision version of psub.
pmpy	to multiply two polynomials.
dpmpy *	double precision version of pmpy.
pdiv	to divide one polynomial by another.
dpdiv *	double precision version of pdiv. Inputs p, x, and y should be double precision. Input tol should be single precision.
pcla	to replace one polynomial by another.
dpcla *	double precision version of pcla.
paddm	to multiply one polynomial by a constant and add it to another polynomial.
dpaddm *	double precision version of paddm.
pval	to evaluate a polynomial.
dpval *	double precision verision of pval.
pvsub	to substitute the variable of one polynomial by another polynomial.
dpvsub *	double precision version of pvsub.
pild	to evaluate a polynomial and its first derivative.
dpild *	double precision version of pild.

pder	to take the derivative of a polynomial.
dpder *	double precision version of pder.
pint	to integrate a polynomial.
dpint *	double precision version of pint.
pqsd	to perform quadratic synthetic division of a polynomial.
dpqsd *	double precision version of pqsd.
pcld	to perform complete linear synthetic division of a polynomial.
dpcld *	double precision version of pcld.
pgcd	to find the greatest common divisor of two polynomials.
dpgcd *	double precision version of pgcd. Inputs x, y, and work should be double precision. Input eps should be single precision.
pnorm	to normalize the coefficient vector of a polynomial.
dpnorm *	double precision version of pnorm. Only input x should be double precision. Input eps should be single precision.
pecn,dpecn	to economize a polynomial for a symmetric range.
pecs,dpecs	to economize a polynomial for an unsymmetric range.

Polynomials: Roots

polrt	to find real and complex roots of a real polynomial.
dpolrt *	double precision version of polrt.
prqd,dprqd	to find roots of a real polynomial by QD algorithm with displacement.
prbm,dprbm	to find roots of a real polynomial by Bairstow's algorithm.
pqfb,dpqfb	to determine a quadratic factor of a real polynomial.

Polynomials: Special Types

cnp,dcnp	to compute the values of the Chebyshev polynomials for a given argument x and orders zero up to n.
cnps,dcnps	to compute the value of a series expansion in Chebyshev polynomials.
tcnp,dtcnp	to transform a given series expansion in Chebyshev polynomials to a polynomial.
csp,dcsp	to compute the values of the shifted Chebyshev polynomials for a given argument x and orders zero up to n.
csps,dcsps	to compute the value of a series expansion in shifted Chebyshev polynomials.
tcsp,dtcsp	to transform a given series expansion in shifted Chebyshev polynomials to a polynomial.
hep,dhep	to compute the values of the Hermite polynomials for a given argument x and orders zero up to n.
heps,dheps	to compute the value of a series expansion in Hermite polynomials.
thep,dthep	to transform a given series expansion in Hermite polynomials to a polynomial.
lap,dlap	to compute the values of the normalized Laguerre polynomials for a given argument x and orders zero up to n.
laps,dlaps	to compute the value of a series expansion in normalized Laguerre polynomials.
tlap,dtlap	to transform a given series expansion in normalized Laguerre polynomials to a polynomial.
lep,dlep	to compute the values of the Legendre polynomials for a given argument x and orders zero up to n.

leps,dleps to compute the value of a series expansion in Legendre polynomials.

tlep,dtlep to transform a given series expansion in Legendre polynomials to a polynomial.

Miscellaneous

dlgam to compute the double precision natural logarithm of the gamma function of a given double precision argument. This routine is from the Special Functions section and is used by routines bdtr and cdtr from the Random Variates section.

hist to print a histogram of frequencies versus intervals. This routine is from the Special Sample Subroutines section.

plot to plot several cross-variables versus a base variable. This routine is from the Special Sample Subroutines section.

Subroutine
Installation-Maintained Library
9/20/71

Name: plssp

All but 15 members* of the PL/1 Scientific Subroutine Package (SSP) are available on Multics. Instructions for using the subroutines are available in the IBM System/360 Scientific Subroutine Package (PL/1) Program Description and Manual (GH20-0586-0).

This memo contains directions for using the subroutines on Multics and a list of all the subroutines available with a brief description of the function of each one.

The subroutines are in ">libraries>plssp", and this must be in the user's search path (i.e., type "ssd >libraries>plssp").

Usage:

call subroutine_name (arg)

The subroutines available and descriptions of the functions of these subroutines are on the following pages.

---

*ahie,ahim,bdtr,cdtr,fft,fftm,fmfp,momn,mvat,mveb,mvsu,mvub,pec, prtc and ptc will not be available until the installation of the new Multics PL/1 compiler.

1. Mathematics

acfe interpolates the function value yval for a given argument xval using xst, the starting value of the arguments, dx, the increment of the argument values and vector y of function values.

acfm interpolates the function value yval for a given argument xval using a given table (x,y) of arguments and function values.

alie interpolates the function value yval for a given argument value xval using xst, the starting value of the arguments, dx, the increment of the argument values and the vector y of function values.

alim interpolates the function value yval for a given argument value xval using a given table (x,y) of argument and function values.

apc1 sets up the normal equations for a weighted polynomial least squares fit to a given discrete function, using Chebyshev polynomials as fundamental functions.

apc2 sets up the normal equations for a polynomial least squares fit to a given discrete function, using Chebyshev polynomials as fundamental functions.

apl1 sets up normal equations for a polynomial least squares fit to a given discrete function.

asn computes the solution of normal equations set up by procedures apc1, apc2 and apl1.

cel1 computes the complete elliptic integral of the first kind:

$$\int_0^{\pi/2} dt/\sqrt{1-k^2\sin^2 t} \qquad 0\leq k<1$$

cel2 computes the generalized elliptic integral of the second kind:

$$\int_0^{\pi/2} \frac{[a+(b-a)\sin^2 t]dt}{\sqrt{1-k^2-\sin^2 t}} \qquad 0\leq k<1$$

dere performs one integration step for a system of first order ordinary differential equations y'=f(x,y) with given initial values y. The stepsize h is adjusted for accuracy requirements and speed considerations.

det3 computes a vector $z = (z_1, z_2, \ldots, z_{dim})$ of derivative values, given a vector $y = (y_1, y_2, \ldots, y_{dim})$ of function values whose components y_i correspond to dim equidistantly spaced argument values x_i with $x_i - x_{i-1} = h$ for $i = 2, \ldots,$ dim.

det5 computes a vector $z = (z_1, z_2, \ldots, z_{dim})$ of derivative values $y = (y_1, y_2, \ldots, y_{dim})$ of function values whose components correspond to dim equidistantly spaced argument values x_i, with $x_i - x_{i-1} = h$.

dfeo given argument x and function fct(x), defined in the one-sided interval [x,x+h], dfeo computes an approximation z to the derivative.

dfec given the argument x and the function fct(x), defined in the closed interval [x-|h|, x+|h|], dfec computes an approximation z to the derivative of the function fct(x).

dgt3 computes a vector $z = (z_1, z_2, \ldots, z_{dim})$, of derivative values, when vectors $x = (x_1, x_2, \ldots, x_{dim})$ of argument values and $y = (y_1, y_2, \ldots, y_{dim})$ of corresponding function values are given.

eli1 computes the incomplete elliptic integral of first kind for given values of an argument x and complementary modulus ck.

eli2 computes the incomplete elliptic integral of second kind for given values of an argument x, complementary modulus ck and constants a and b.

exsm develops the triple exponential smoothed series s of the given series x.

jelf calculates the three Jacobian elliptic functions sn, cn and dn.

lgam computes the double-precision natural logarithm of the gamma function of a given double-precision argument.

post transforms a given series expansion in orthogonal polynomials to a polynomial. The independent variable of the given expansion is assumed to be $x_0 + x_1 \cdot x$; that is, a linear transforamtion of the range is built in. The coefficient vector $c = (c_1, c_2, \ldots, c_n)$ is given. Procedure post calculates $pol = (pol_1, pol_2, \ldots, pol_n)$ satisfying

$$\sum_{i=1}^{n} c_i f_{i-1}(x_0 + x_1 \cdot x) = \sum_{i=1}^{n} (pol_i)(x^{i-1}).$$

For the specified set of orthogonal polynomials (f_k) the user has the choice of:

Chebyshev polynomials (t_0, t_1, ..., t_{n-1}) with opt = "t"

Legendre polynomials (p_0, p_1, ..., p_{n-1}) with opt = "p"

Laguerre polynomials (l_0, l_1, ..., l_{n-1}) with opt = "l"

Hermite polynomials (h_0, h_1, ..., h_{n-1}) with opt = "h".

posv computes the value of the sum

$$\sum_{k=1}^{n} c_k f_{k-1}(x)$$

for a given vector c = (c_1, c_2, ..., c_n) and a specified set of orthogonal polynomials (f_k).

The user has a choice of:

Chebyshev polynomials (t_0, t_1, ..., t_{n-1}) with opt = "t"

Legendre polynomials (p_0, p_1, ..., p_{n-1}) with opt = "p"

Laguerre polynomials (l_0, l_1, ..., l_{n-1}) with opt = "l"

Hermite polymonials (h_0, h_1, ..., h_{n-1}) with opt = "h".

pov computes the values of the first n orthogonal polynomials. The user has the choice of:

Chebyshev polynomials (t_0, t_1, ..., t_{n-1}) with opt = "t"

Legendre polynomials (p_0, p_1, ..., p_{n-1}) with opt = "p"

Laguerre polynomials (l_0, l_1, ..., l_{n-1}) with opt = "l"

Hermite polynomials (h_0, h_1, ..., h_{n-1}) with opt = "h".

se13 computes a vector $z = (z_1, z_2, \ldots, z_{dim})$ of smoothed function values. It requires a vector $y = (y_1, y_2, \ldots, y_{dim})$ where the y components correspond to equidistantly spaced argument values x_i, assuming $x_i - x_{i-1} = h$.

se15 computes a vector $z = (z_1, z_2, \ldots, z_{dim})$ of smoothed function values, given a vector $y = (y_1, y_2, \ldots, y_{dim})$ of function values whose components y_i correspond to dim equidistantly spaced argument values x_i with $x_i - x_{i-1} = h$ for $i = 2, \ldots, dim$.

se35 computes a vector $z = (z_1, z_2, \ldots, z_{dim})$ of smoothed function values, given a vector $y = (y_1, y_2, \ldots, y_{dim})$ of function values whose components y_i correspond to dim equidistantly spaced argument values x_i with $x_i - x_{i-1} = h$ for $i = 2, \ldots, dim$.

sg13 computes a vector $z = (z_1, z_2, \ldots, z_{dim})$ of smoothed function values. It requires a vector $y = (y_1, y_2, \ldots, y_{dim})$ and a vector $x = (x_1, x_2, \ldots, x_{dim})$ of argument values, where y_i corresponds to x_i.

2. Matrix Operations

canc computes the canonical correlations between two sets of variables. It is normally preceded by a call to procedure corr.

mags computes c = a + b if opt = "1"
c = a - b if opt = "2"
c = b - a if opt = "3"
for given matrices a and b which are general and symmetric respectively.

mate reduces a given real matrix to upper almost triangular (Hessenberg) form by means of a sequence of similarities.

matu reduces a given real matrix to almost triangular (Hessenberg) form by means of a sequence of orthogonal transformations.

mdlg for a system of equations $a \cdot x = r$, where $a = l \cdot u$ is a general nonsingular matrix, mdlg performs the following calculations, depending on the character of an input parameter opt:

opt = "1" r is replaced by $l^{-1} \cdot r$
opt = "2" r is replaced by $u^{-1} \cdot r$
otherwise r is replaced by $(l \cdot u)^{-1} \cdot r$.

mdls for a system of equations $a \cdot x = r$ with symmetric positive definite matrix $a = t \cdot t^t$, mdls performs the following calculations depending on the character of the input parameter opt:

opt = "1" r is replaced by $t^{-1} \cdot r$
opt = "2" r is replaced by $(t^{-1})^t \cdot r$
otherwise r is replaced by $(t \cdot t^t)^{-1} \cdot r$.

mdrs for a system of equations $x \cdot a = r$ with symmetric positive definite matrix $a = t \cdot t^t$, mdrs performs the following calculations, depending on the character of an input parameter opt:

opt = "1" r is replaced by $r \cdot (t^{-1})^t$
opt = "2" r is replaced by $r \cdot t^{-1}$
otherwise r is replaced by $r \cdot (t \cdot t^t)^{-1}$.

mdsb depending on the character of the input parameter opt, mdsb performs the following operations on a system of equations $a \cdot x = r$ with symmetric positive definite band matrix $a = t^t \cdot t$:

opt = "1" r is replaced by $(t^{-1})^t \cdot r$
opt = "2" r is replaced by $t^{-1} \cdot r$
otherwise r is replaced by $(t^t \cdot t)^{-1} \cdot r$.

meat computes the eigenvalues of a real upper almost triangular matrix (Hessenberg form, see subroutines mate and matu) using double QR iteration.

mebs computes a lower and an upper bound for the eigenvalues of a real symmetric matrix.

mest computes the eigenvalues of a real symmetric tridiagonal matrix (see subroutine mstu).

mfg factorizes a general nonsingular matrix a into a product of a lower triangular matrix l and an upper triangular matrix u overwritten on a, omitting the unit diagonal of u.

mfgr performs the following for a given rectangular matrix:

determines rank and linearly independent rows and columns

factorizes a submatrix of maximal rank

expresses nonbasic rows in terms of basic rows

expresses basic variables in terms of free variables.

mfs computes a triangular factorization of a symmetric positive definite matrix using the square root method of Cholesky.

mfsb computes a triangular factorization of a symmetric positive definite band matrix using the square root method of Cholesky.

mgbl performs the following operations on an equation system $a \cdot x = r$ with general band matrix $a = l \cdot u$, depending on the character of an input parameter opt:

opt = "l" u replaces a and $l^{-1} \cdot r$ replaces r

opt = "u" u replaces a and $u^{-1} \cdot r$ replaces r

opt = "b" $u^{-1} \cdot r$ replaces r for a given u on storage locations of a

otherwise u replaces a and the solution $x = a^{-1} \cdot r$ replaces r.

mgb2 performs the following operations on an equation system $a \cdot x = r$ with general band matrix $a = l \cdot u$, depending on the character of an input parameter opt:

opt = "l" a is replaced by upper band factor u, r is replaced by $l^{-1} \cdot r$ and lower band factor l is stored in a one-dimensional array l omitting the unit diagonal

opt = "f" a is replaced by the upper band factor u and the lower band factor l is stored in the array l; the right-hand side r remains unchanged

opt = "a" r is replaced by $l^{-1} \cdot r$ for the given upper factor u in array a and the lower factor l in vector l

opt = "c" r is replaced by the solution $x = a^{-1} \cdot r$ for given u and l

otherwise a is replaced by the upper factor u; the lower factor l is calculated and stored in l, and r is replaced by the solution $x = a^{-1} \cdot r$.

mgdu computes eigenvalues and eigenvectors of a real matrix of the form b-inverse times a, where a is symmetric and b is positive definite.

mig inverts a general nonsingular matrix a, which is given in the factored form $a = l \cdot u$, where the upper triangular matrix u contains the unit diagonal which is not stored.

minv inverts a general square matrix.

mis inverts a symmetric positive definite matrix a, which is given in factored form (Cholesky):

$a = t \cdot \text{transpose}(t)$.

mlsq calculates x satisfying $a \cdot x = b$, that is, the solution of a system of linear equations using Householder transformations. The least squares solution is obtained in case of an overdetermined system of equations.

mmgg computes the standard matrix product $c = a \cdot b$.

mmgs calculates $g \cdot s$ if opt = "1" or
$s \cdot g$ if opt = "2"
where g is a general and s a symmetric matrix.

mmgt calculates $a \cdot a^t$ if opt = "1" or
$a^t \cdot a$ if opt = "2"
where a is a general matrix.

mmss computes the standard product $p = a \cdot b$ of two symmetric matrices.

mpit calculates the permutation vector corresponding to the inverse of a given permutation if opt = "i" and the transposition vectors of the given permutation and of its inverse if opt = "t".

mprm permutes rows (if opt = "r") or columns (if opt = "c") of a given matrix a according to the permutation p (if inv = "0") or its inverse p^{-1} (if inv = "1"). The permutation p is given in the form of its transposition vector t.

mscg expands the compressed one-dimensional storage allocation of a symmetric matrix to general two-dimensional form.

mscs compresses the storage allocation of a symmetric two-dimensional matrix to a one-dimensional array.

msdu computes eigenvalues and eigenvectors of a real symmetric matrix.

mstu reduces a given real symmetric matrix to tridiagonal form by means of a sequence of orthogonal transformation.

mtpi calculates the permutation vector if inv = "0" and the inverse permutation vector if inv = "1" from a given transposition vector.

mvst provides the eigenvector corresponding to a given eigenvalue of a symmetric tridiagonal matrix.

3. Statistics

abst tests for missing or zero elements in observation matrix a.

avar performs an analysis of variance for a complete factorial design.

boun selects from a set (or a subset) of observations the number of observations under, between and over two given bounds for each variable.

chsq computes chi-square from a contingency table.

corr computes means, standard deviations, sums of cross-products of deviations and correlation coefficients.

dmtx computes means of variables in each group and a pooled dispersion matrix for all the groups. This subroutine is used in the performance of discriminant analysis.

dscr performs a discriminant analysis by calculating a set of linear functions that serves as indices for classifying an individual into one of k groups.

htes calculates the Kruskal-Wallis H-statistic from the ranks of observations obtained from three or more independent samples.

klmo tests the difference between empirical and theoretical distributions using the Kolmogorov-Smirnov test.

klm2 tests the difference between two sample distributions using the Kolmogorov-Smirnov test.

krnk measures the correlation between two variables by means of the Kendall rank correlation coefficient.

load computes a factor matrix (loading) from eigenvalues and associated eigenvectors.

mltr performs a multiple linear regression analysis for a dependent variable and a set of independent variables.

ndti computes $x = p^{-1}(y)$ such that $y = p(x)$, the probability that the random variable x, distributed normally (0,1), is less than or equal to x; f(x), the ordinate of the normal density at x, is also computed.

ndtr computes y = p(x), the probability that the random available x, distributed normally (0,1), is less than or equal to x; f(x), the ordinate of the normal density at x, is also computed.

ordr is used to choose a dependent variable and a set of independent variables from a matrix of correlation coefficients and to form a submatrix of correlation coefficients to be used in performing a multiple linear regression analysis.

qtst uses the Cochran Q-test to determine whether three or more matched groups of dichotomious data differ sygnificantly.

rank ranks a vector of data.

sbst derives a subset vector indicating which observations in a set have satisfied certain conditions on the variables.

smir computes values of the limiting distribution function for the Kolmogorov-Smirnov statistic.

srnk tests the correlation between two variables by means of the Spearman rank correlation coefficient.

strg performs a setpwise multiple linear regression analysis for a dependent variable and a set of independent variables.

subm copies a submatrix from an observation matrix. The elements of this submatrix satisfy conditions specified by an input vector. This subroutine is used in preparing data for input to a statistical analysis such as multiple regression.

tab1 tabulates, for one variable in an observation matrix (or a matrix subset), the frequency and percent frequency over given class intervals. In addition, t calculates for the same variable the total, mean, standard deviation, minimum and maximum.

tab2 performs a two-way classification, for two variables in an observation matrix (or a matrix subset), of the frequency, percent frequency and other statistics over given class intervals.

taly calculates total, mean, standard deviation, minimum and maximum for each variable in a set (or a subset) of observations.

tie calculates correction factor due to ties.

trac computes cumulative percentage of eigenvalues greater than or equal to a constant specified by the user.

ttst calculates certain t-statistics on the means of populations.

twav tests whether a number of samples are from the same population by the Friedman two-way analysis of variance test.

utst tests whether two independent groups are from the same population by means of the Mann-Whitney U-test.

vrmx performs an orthogonal rotation of a factor matrix.

wtst measures the degree of association among a number of variables by the Kendall coefficient of concordance.