Draft:Not for distribution
version: June 20, 1994

TOPAZ FORTLIB

short writeup

TOPAZ Offline Group

Abstract

TOPAZ Fortlib is a collection of subroutines developped by TOPA Group members for the use of TOPAZ data analysis. These routines are written using FORTAN77 and Assmebler language of FACOM MSP system. Porting to UNIX system is in progress and test version is available from *miyamoto@kekux1.kek.jp*. This manual contains the breif summary of each subroutines. Since each source codes includes the writeups of how to use, please get consults from there. Source codes are in T#PB.FORTLIB.FORT.

Contents

1 Bevington Libraries.

```
AREA     Integrate the area beneath a set of data points.
CURFIT   Make a least-squres fit to a non-linear function with a
        linearization of the fitting function using the algorithm of
        Marquardt which combines a gradient search with an analytical
        solution developped from linearization of the fitting function.
DETERM   Calculate the determinant of a square matrix.
FCHISQ   Evaluate reduced chi-square for fit to data.
GQDRIV   Evaluate derivative of function for least-square search with
         form of a gaussian peak plus quadratic polynomial.
GQFUNC   Evaluate terms of function for non-linear least-square search
         with form of a gaussian peak plus quadratic polynomial

         GQFUNC(X,I,A) = A(1)*EXP(-Z**2/2) + A(4) + A(5)*X + A(6)*X**2

         where
            X = X(i)
            Z = (X - A(2))/A(3)

INTEG    Integrate the area beneath two data points
INTERP   Interpolate between data points to evaluate a function.
         Make an (n-1)th-degree polynomial fit to pairs of data points
       (Xi,Yi) ordered for monotonically increasing bu not necessarily
       equally spaced Xi, and to interpolate ( or extrapolate ) to find
       the value of Y corresponding to a given value of X.
LINFIT   Make a least-squares fit to data with a straight line
             Y = A + B*x
MATINV   Invert a symmetric matrix.
POLFIT  Make a least-square fit to data with a polynomial curve.

           Y=A(1)+A(2)*X+A(3)*X**2+A(3)*X**3+ . . .

        This program is modified from orginal version.  Use MATRIX
        inversion technique to output error of fitted parameter.
POLFT2  Make a least-square fit to data with a polynomial curve.
           Y=A(1)+A(2)*X+A(3)*X**2+A(3)*X**3+ . . .
```

2 System utilities

```
GETUID   Return the name of the user name executting the
NARG     Return # of arguments in the calling routine.
NUADDR   Get address of the input variable
RAN      VAX compatible random number generator.
```

RANN   Function to generate normal random number with
  m   : mean value = 0.0
  sigma : s.d.   = 1.0.
TERMID   Return the name of the terminal, where the program is executed.
TIMEL   CERNLIB compatible routine to tell remaining CPU time of
the job.
UERASE   Clear screen of fullscreen terminal.
USUBMT   Submit JCL.
WAIT   Suspend the execution of the job by a given amount of time.
WAITS   Suspend the execution of the job by a given amount of time.
Wait time is specified by machine cycle unit.

3 Bit manupilation

IIBITS   Extracts a contiguous part of bits from input I*2 variables.
Same algorithm as UTAKBT. Same calling sequence as VAX FORTRAN IBITS.
JIBITS   Extracts a contiguous part of bits from input I*4 variables.
Same algorithm as UTAKBT. Same calling sequence as VAX FORTRAN IBITS.
UPUTBT   Puts IPRT into IX at the bit location given by ILO and NBT
and returns the results as IOUT
USETBT   Inserts IPRT as a contiguous NBT bits into IX at ILO.
UTAKBT   Extracts a contiguous part of bits from input I*4 variables.

4 Character string manupulation.(Assembler routine.

NUCFND   Find pointer of the next character CHAR after IAD1.
NUCLEN   Get the effective character length of STRING.
NUCNXT   Find pointer of the next character after IAD1.
NUSFND   Find pointer of the next character CHAR after IAD1.
UCASC   Translate ASCII character to displayable EBCDIC character.
UCASC2 Translate ASCII code to EBCDIC.
UCCHTI Convert NWORD of integer array INTGR into character
type array CHOUT.
UCEBC   Translate EBCDIC code to ASC-ii code.
UCITCH Convert NWORD of integer array INTGR into character
type array CHOUT.
UCUPER   Translate lower case character to upper case.
USMALL   Convert character to lower case.

5 System utilities

NUCRMV Remove heading and trailing blanks in the input STRING.
UCMACH Seraches the string matched with the input Tststr.
UCHANG Change character string CFROM to CTO in the Character variable STRING. Corresponds to the CHANGE command of PFDE editor.
UCPICK Pick up the characters between SEP1 and SEP2 in the input sstring.
UCSTRP Remove the Heading, in between blanks from the Input string and output together with the valid character Length.
UCTOKN Devide INPUT string by SEP to NTOKs strings.

6 Libraries for TOPAZ data analysis

PUELOC Tells when error happened in production job.
RUELOC Tells when error happened in reduction job.
SUELOC Tells when error happened in simulator.
UERINI Initializes TOPAZ error handling facility.
UERLOC Tells when error happened.
UERROR Handles error occured in Simulator, Reduction, and Production jobs.
UERTRM Prints out error statistics.

7 File manupulation

UALCDA Allocate direct access file.
UALCDS Allocate VBS format data set.
UALCPS Allocate PS data set of FB, VB, VBA, FBA, or VBS format.
UCMTAL Dynammically allocates Raw data or DST tapes.
UCTLAL Dynammically allocates CTL files.
UCTLFR Dynammically free CTL files.
UCTLSZ Accumulate output data size on disk and CTL.

8 Data copy/Move/clear

UBCPYH To move the NW half-words data of between addresses IAD0 and IAD0+NW-1 to the same array of addresses between IAD1 and IAD1+NW-1.
UBSWAP Swap bytes of the I*2 array data IDATA
UDUMPH Dumps INTEGER*2 array.
UI2SWP Swap even I*2 word and odd I*2 word
Argument INPUT, OUTPUT must be 4 byte integer.
UVCOPY Copy array.

```
UVCPYH   Copy array ( 2 byte unit.)
UVFILL   Fill value into ARRAY.
UVGET    Copy the contents of memory from IADD by NBYTES
         to the ARRAY.
UVSET    Set memory of address IADD equal to the value VAL
UVSETC   Copy character in STRING to the address start from IADD
UVZERO   Zero clear array.    .
UWSWAP   Do word swap
```

9 Sorting

```
USORTC   Sort character array.
USORTI   Quick sorter of an interger array.
USORTR   Quick sorter of an real*4 array.
UXSORT   Sorts out X(*) in ascending order by giving a set of pointers
         without touching the array itself.
```

10 Date and time manipulation

```
UCDATE  Converts TOPAZ Date format into Character*8 format.
            yymmdd ---> yy-mm-dd
UCTIME   Converts TOPAZ time format into Character*8 format.
            msec   ---> hh:mm:ss
UDAYDF   Calculate day difference between IDATE and KDATE.
UIDATE    Get current date by TOPAZ standard date format.
UTIME    Returns a time of a day by character format as follows.
            JIKAN = hh:mm:ss
         Using the subroutine TIME in Facom fortran library.
```

11 Others

```
UBINIT  Decides which bin X falls on when the number of bins and
        bin boundary are given by NBIN and BNDRY, respectively.
UCONST   Block data for Numerical constants.
```

12 Helix parameter manipulation

```
UCDXDA  Calculate @X/@A at given deflection angle, where X is positon
        vector and A is helix parameter vector.
UCPSER   Calculate position error matrix at specified deflection angle.
UCRCLX   Calculates intersections of 2 circles.
UPLNX    Calculates intersection of a line and a plane.
UCRFT2   Do circle fit by least square fitting.
```

UCYLX   Calculates intersection of a line and a cylinder.
UFCLSP   Finds a point on a helix which is closest to a given point
and calculates tangent vector and deflection angle therat.
UFGEOC   Do geometrical constraint fit to the Track parameter.
UFGEOI*   Called from PGFTRK to store initial parameter for the
geometrical constraint fit to the buffer.
UFGEOK*   Calculate chi-squre, first derivative of Chi-square and
2nd-derivative of the matrix for geometrical constraint fit.
UFGEOT*   Called from PGFTRK to unpack fit results.
UFLTRK   Fit straight line track in R-Phi space.
UFPAIR   Get approximate intersection of 2 helices by 1-st calculating
intersections of 2 circles and then checking their Z coords.
UFTVTX   Calculates vertex point consists of NTRK tracks.
The #tracks should not exceed MX$PAR-3. At presetn MX$PAR is
set at 53 corresponding to 50 tracks. User who needs more
can extend MX$PAR.
UFVRTX   Calculates vertex point consists of NTRK tracks.
UPHELX   Calculates position and momentum of a helix track at given
deflection angle.
UTRKAV   Convolutes multi-track segments into a single helx.
UTRKIV   Change charge sign of the track parameter.
UTRKMV   Transforms helix parameters and their error matrix to those
of a new pivot XP.

13 Statistics

UCHIIN   Computes the inverse of the distribution function
of the chi-squre distribution win N degrees of freedom
UCHISQ(NTERMS,PARAM)
UCONFL   Gives confidence level.
UCONF8   Give confidence level.(Double precission.)
UGAUIN   Calculates the inverse of the distribution function of the
standard nominal distribution.
UPRBIN   To estimate the probability exist in a region of
Gaussian distribution.

14 Kinematics calculation

UABSV3 Calculate the absolute value of   3-dim. vector.
UADD3 Adds two 3-dim. vector.
UADD4 Adds two 4-vector.
UBASV3   Sets the standard basis vector
UBSTBK routine to transform PB(4) in PR(4)-rest frame to PA(4)
in PR(4)-moving frame.(Boost 4 vector)

```
UBSTFD  routine to transform PB(4) to PR(4)-rest frame ,PA(4).
UBTRAN    Rotates PB to PA so that PB in the frame E is PA
          in the standard frame.
UCOLLN   routine to calculate collinearity.
UCONJ4   Take Lorentz conjugate.
UCOPV3   Copies 3-dim. vector.
UCOPV4   Copies 4 vector.
UCROSD   Calculate vector product of 3-dim vector (real*8)
UCROSS   Calculate vector product of 3-dim vector (real*4)
UCTRAN   Coordinate transformation utility routine.
UDCAY2   Simulates isotropic 2-body decay.
UDCMPV   routine to decompose a vector as:
           X=XTV+XLV
         with XR as a reference.
UDELTA   calculates invariant triangle function.
UDIST    routine to find distance between X(3) and Y(3).
UDOT2     Calculates scalar-product of X and Y.( 2-dim. vector)
UDOT3     Calculates scalar-product of X and Y.( 3-dim. vector)
UDOT3D    Calculates scalar-product of X and Y.( 3-dim. vector, Real*8)
UDOT4     routine to calculate invariant 4-scalar product.
UHANGL    calculates decay angles of particle DK in RES rest frame.
          ZED is ref. Z direction  (Beam for t channel, -MM for
          s channel)
          If ZED = +/-RES,  (RES helicity frame) decay plane is
          undefined & PHI will be zero.
          DK, RES, and ZED are all 4-vectors.
UMCSTH   Gives COS(theta) and invariant mass for the hypothesis
UNCROS   calculates 3-vector normed-cross product.
               C=AXB, Then C = C/|C|
UNDOT3   calculates 3-vector normed-dot product. After normalizing A and
         B to unity, makes dot-product of the two.
UNRMV3   Normalizes the given vector,X.( 3-dim.)
UNULV3   Zeros X. ( 3-dim. vector.)
UNULV4   Zeros a 4-vector, P(4).
UOPANG   Calculates opening angle between X(3) and Y(3).
UPHIAN   Calculates phi-angle of P measured from the axis normal to
         the Z-PR plane with PR as a polar axis.
UPLNG    calculates longitudinal momentum.
UPLV     Calculates longitudinal vector PLV with PR as a reference.
UPSTAR   Calculates the absolute value of daughter momenta in the parent
         rest frame for 2-body decay.
UPTPL    routine to calculate PT and PL of P with PR as a reference.
UPTPLV   routine to calculate transverse vector PTV and longitudinal
         vector PLV as PR as a reference. PT and PL are their absolute
         values.
```

```
UPTV     Calculates longitudinal vector PLV with PR as a reference.
UROT2    Rotate 2-dim. vector by given angle.
USCLM3   Multiplies the 3-dim vector by a factor A.
USETRF   Setups a reference frame with E(*,3) along P and E(*,1)
         perpendicular to P-E3(*) plane.
         (i.e. E(*,1) in E1-E2 plane)
USETV4   Sets 4-momentum P2 using 3-momentum P1 and mass AM.
USUB3    Subtacts Y from X and returns the result as Z.( 3-dim.)
USUB4    Carries out 4-vector subtraction.
UTEE     Calculates TEE variables.
             TEE = (P-PBEAM)*(P-PBEAM)
UXFYN    routine to calculate Feynman X of P in PR-rest frame.
```

15 Fitting

```
UHIFIT    Make a minimum search of a non-linear function with a
          parabolic expansion.
UINIMZ   Main routine to minimize a function in a multi-parameter
         space.
UIRCX2   Do circle fit to the data points.
UMATIG   Invert a real matrix. At present MX$ORD is set at 103 which
         enables inversion of a matrix with dimension up to 103x103.
         User can change MX$ORD, if necessary.
         (Caution)
         For inversion of positive definite real symmetric matrices,
         use UMATIN which is faster and has no restriction to array
         size.
UMATIN   Invert a positive definite real symmetric matrix.
URADLS   Make a gradient search of a minimum for a specified function
         which is not linear in coefficient.
UURFIT   Make a minimum search of a non-linear function with a
         linearization of the fitting function.
```