Appendix D

Compute Function Examples

Appendix D - Compute Function Examples

Index of Functions

		Page
ABS	Absolute function	3
ACC	Running accumulation	5
AMREG	Apply multiple linear regression equation	6
CONIC	Conic interpolation from elevation/area table	8
CORR	Compute correlation coefficients	10
COS	Cosine trigonometric function	12
COUNT	Count the number of valid and missing data values	13
CMA	Centered moving average smoothing	14
DDT	Differences per unit time	16
DECPAR	Decaying basin wetness parameter	18
DIFF	Successive differences	20
ESTLIN	Estimate values for missing data	21
ESTPPT	Estimate values for missing precipitation data	22
FMA	Forward moving average	23
GENTSR	Generate a regular interval time series	24
INT	Truncate to whole numbers	25
LAST	Last valid value in a time series	25
LOG	Natural log base "e"	25
LOG10	Log base 10	25
MATE	Generate data pairs from two time-series	26
MAX	Maximum value in a time series	28
MEAN	Mean value in a time series	28
MIN	Minimum value in a time series	28
MREG	Multiple linear regression function	29
MRG	Merge two time series	31
MRGP	Merge two paired data series	32
MUSK	Muskingum hydrologic routing	33
NINT	Round to nearest whole number	35
OLY	Olympic smoothing	36
PERCON	Period constants	39
POLY	Polynomial transformation	39
POLY2	Polynomial transformation with integral	42
PULS	Modified Puls or Working R&D routing function	45
QAC	Flow accumulator gage processor	47

RND	*Round off*	*47*
RTABLE	*Rating table interpolation*	*49*
RTABLR	*Reverse rating table interpolation*	*49*
RTABL2	*Two-variable rating table interpolation*	*52*
SCRN1	*Screen for possible erroneous values based on maximum/minimum range*	*57*
SCRN2	*Screen for possible erroneous values*	*57*
SDEV	*Compute the standard deviation of one independent variable.*	*59*
SELECT	*Extract time series data at unique time specification.*	*60*
SHIFT	*Shift adjustment*	*65*
SIN	*Sine trigonometric function*	*12*
SKEW	*Compute the skew coefficient of one independent variable*	*59*
SQRT	*Square root function*	*67*
SS	*Straddle Stagger routing function*	*67*
SSW	*Wilmington District Straddle Stagger routing function*	*67*
TAN	*Tangent trigonometric function*	*12*
TS1	*Interpolated data at regular intervals*	*69*
TS2	*Period averages at regular intervals*	*69*
TS3	*Period mins or maxs at regular intervals*	*69*
TS4	*Interpolated data at irregular intervals*	*71*
TSCYCL	*Time series cyclic analysis*	*72*
TSHIFT	*Shift time series in time*	*76*
TSNAP	*Snap Irregular times to nearest Regular period*	*77*
TTSR	*Transform time series to regular*	*78*
TTSI	*Transform time series to irregular*	*83*

Introduction

Appendix D is designed to help the user understand the DSSMATH functions by the use of examples. The examples are primarily designed to demonstrate the use of the function and may or may not be good examples in real life applications, but where possible, valid real life examples were used. Each of the functions has, at a minimum, one example of its use. Where appropriate, graphics and tabular outputs have been used to make it more clear as to the functions use. The examples are shown as a set of commands associated with a PREAD macro. The example macros consist of more than one function in many of the examples. Some of the functions have multiple options and require more than one example to demonstrate their use.

Name: *ABS Absolute function*
Use: *CO TY=ABS(TX)*

Compute the absolute value of TX and store in TY. TX and TY can be the same variable. TX can be scalar, time series or paired data. If a value of TX is undefined, resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged.

Example: The following example uses the ABS function to convert a set of difference values between observed flow and computed to positive values. This is done to allow the computation of the absolute maximum difference and the average difference.

Macro:

```
MACRO TABS
TIME 13JUL1975 0800 16JUL1975 1200
GET OBS=WS6N.DSS:/RAHWAY/SPRINGFIELD/FLOW/01JUL1975/1HOUR/OBS/
GET CAL=WS6N.DSS:/RAHWAY/SPRINGFIELD/FLOW/01JUL1975/1HOUR/OPT/
COMP DIFF=OBS-CAL
COMP DABS=ABS(DIFF)
PUT.A DABS=/RAHWAY/SPRINGFIELD/DIFF-FLOW/01JUL1975/1HOUR/CAL/
COMP MAXVAL=MAX(DABS)
COMP AVGDIF=MEAN(DABS)
SHOW MAXVAL AVGDIF
ENDMACRO
```

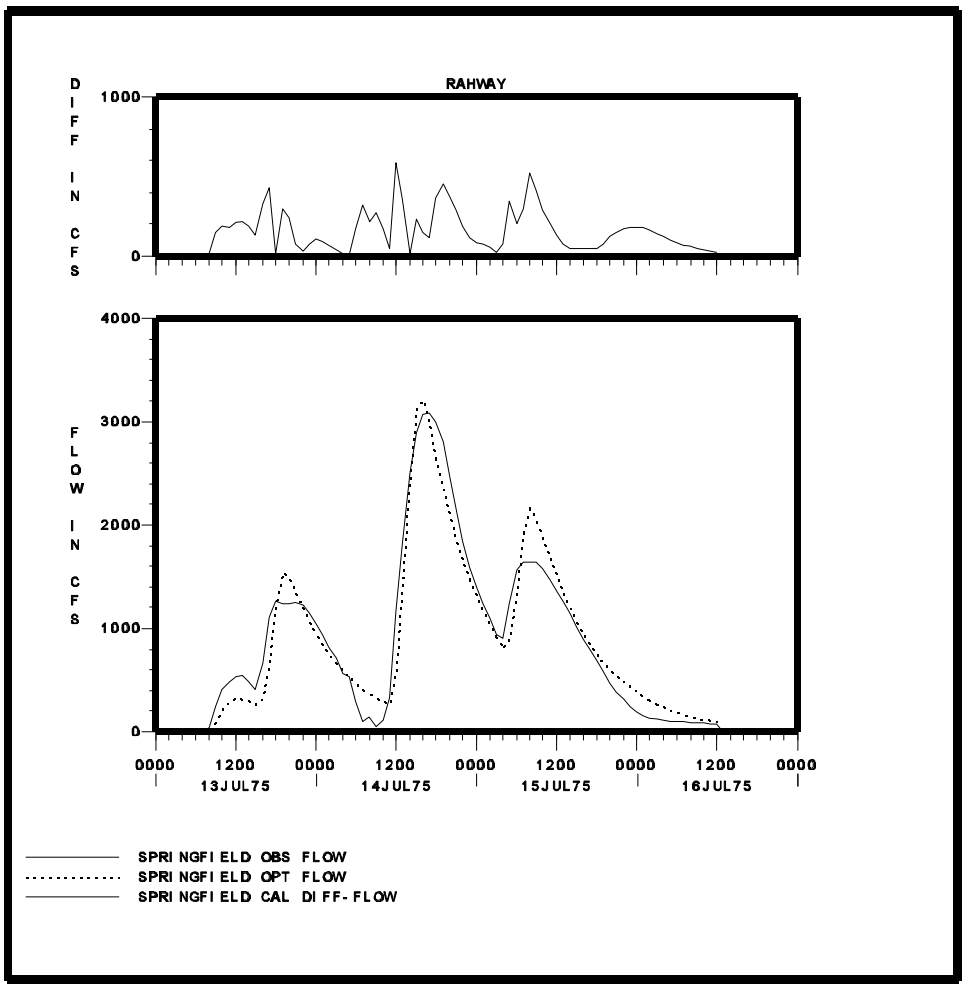
Execution:

```
I>!R TABS
   -----DSS---ZOPEN:  Existing File Opened,  File: WS6N.DSS
                  Unit:  71;  DSS Version: 6-GX
 -----DSS--- ZREAD Unit  71; Vers.    1:  /RAHWAY/SPRINGFIELD/FLOW/01JUL1975/1HOUR/OBS/
 -----DSS--- ZREAD Unit  71; Vers.    1:  /RAHWAY/SPRINGFIELD/FLOW/01JUL1975/1HOUR/OPT/
 -----DSS---ZWRITE Unit  71; Vers.    3:  /RAHWAY/SPRINGFIELD/DIFF-FLOW/01JUL1975/1HOUR/CAL/

 MAX=    591.306  MIN=    11.733  MEAN=    166.694  LAST VALUE=    24.544


 VARIABLE=MAXVAL
 MAXVAL  =  591.306200

 VARIABLE=AVGDIF
 AVGDIF  =  166.694000
```



Name: *ACC* *Running accumulation*
Use: *CO TY=ACC(TX)*

Compute a running accumulation of the values in TX and store the result in TY. If a TX value is undefined or a concurrent IF condition is not satisfied, the value of TX is not added to the accumulation and the corresponding TY remains the same as the previous TY. Units of TY are the same as those of TX. If TX is typed as PER-AVER or PER-CUM, TY will be typed INST-VAL or INST-CUM, respectively.

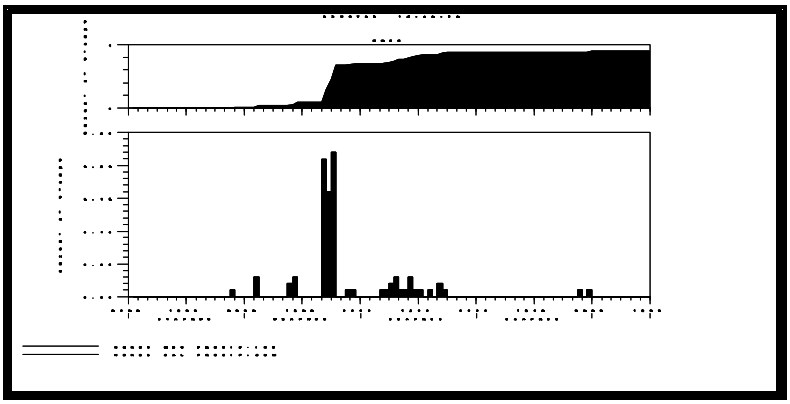
Example: The following example uses the accumulation function to take incremental precipitation and convert into a cumulative value.

Macro:

```
MACRO TACC
** COMPUTE A RUNNING ACCUMULATION OF THE VALUES IN PPT
** AND STORE IN PACC
CL ALL
TIME 01APR83 0100 05APR83 2400
GET PPT=testdb.dss:/GOES/NBKD3/PRECIP-INC/01ARP1983/1HOUR/OBS/
COMP PACC=ACC(PPT)
PUT.A PACC=C=PRECIP-CUM F=CAL
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
I>!R TACC
  ALL VARIABLES HAVE BEEN CLEARED
    -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                    Unit:  71;  DSS Version: 6-IA
 -----DSS--- ZREAD Unit  71; Vers.  145:  /GOES/NBKD3/PRECIP-INC/01APR1983/1HOUR/OBS/
  FINAL ACCUMULATED VALUE =   8.999997E-01
 -----DSS---ZWRITE Unit  71; Vers.    2:  /GOES/NBKD3/PRECIP-CUM/01APR1983/1HOUR/CAL/
```



Name: *AMREG* *Apply multiple linear regression equation.*
Use: *CO TY=MREG(PF,TX1,TX2,...,TXn,min,max)*

*This function is used to calculate a regular time series (TY) based on a linear regression equation of the general form (Y = B0 + B1*X1 + B2*X2 + Bn*Xn). Where Y is the dependent variable and the X1, X2, and Xn are independent variables. The AMREG function requires that the function parameters TY, TX1, TX2, and TXn be regular time series variables and that they all have the same time interval. It is further required that the time variables be retrieved in the same sequential order as specified in the function parameter list. In other words TX1 is retrieved from DSS first followed by TX2 and so on. The number of independent time series variables that can be specified is limited to one less the maximum number of time series variables that the program allows. On the DOS PC this value will be 4 independent variables. The linear regression coefficients (B0 - Bn) are stored in a paired function variable (PF) which is normally retrieved from DSS. Two optional function parameters "min max" are available to specify a range of values that control which values in the dependent variable are to be accepted as valid computed values. These two parameters must be specified as the last two parameters and if specified, both must be specified as real numbers. Units of TY are the same as those of TX1. An IF condition has no effect.*

Example: The following example uses the AMREG function to take previously determined linear regression coefficients determined by MREG function and apply them to flow values at two upstream gages and determine flow values at a downstream gage.

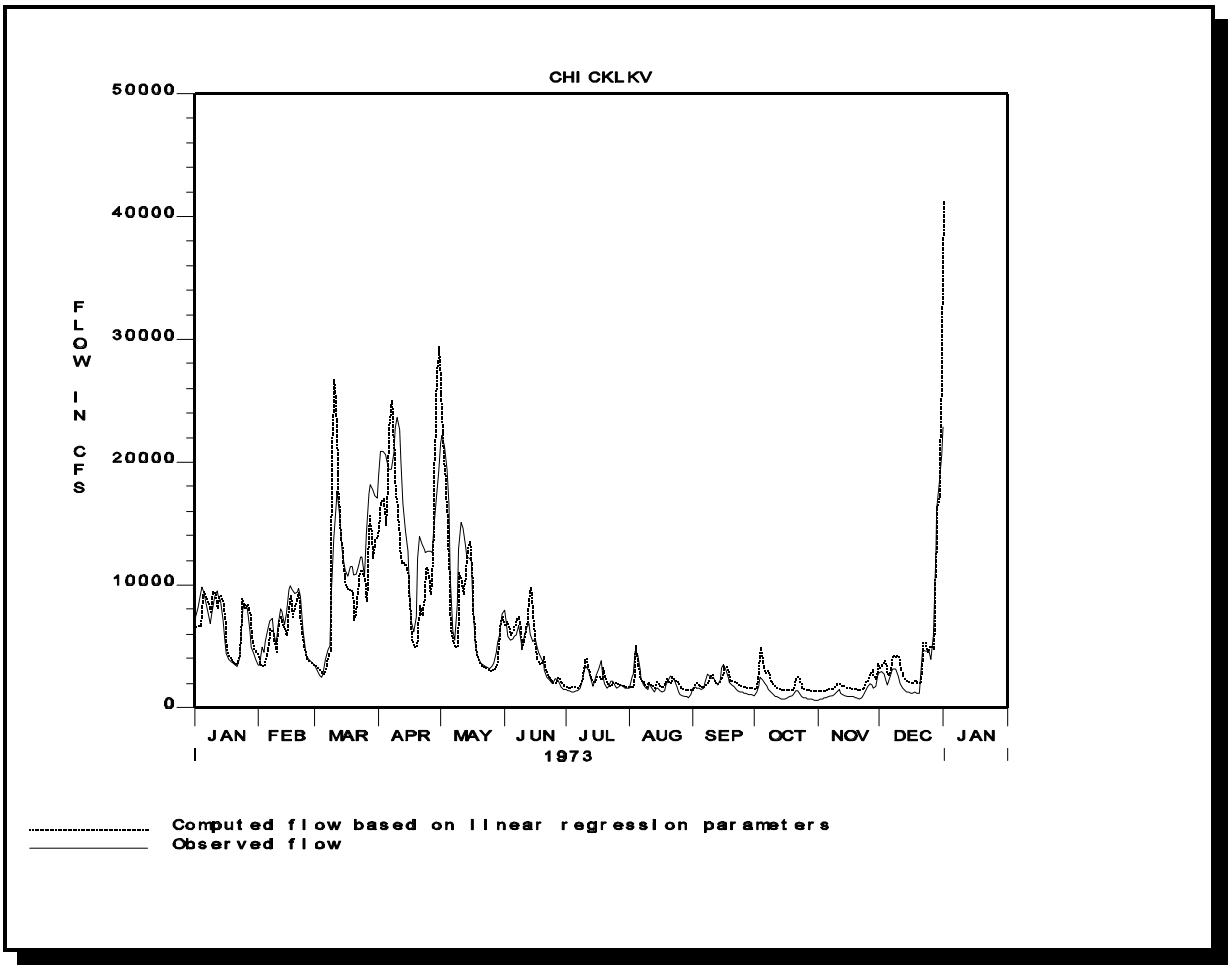
Macro:

```
 MACRO AMREG2
!=R AMREG2
CLEAR ALL
TIME 2400 01OCT1972 2400 30SEP1977
GET PF=MREG.DSS:/LINEAR REGRESSION/COEFFICIENTS/5YEARS////
GET TX2=MREG.DSS:/CHICKENT/TX2/FLOW/01JAN1972/1DAY/OBS/
GET TX3=MREG.DSS:/BUCKNER/TX3/FLOW/01JAN1972/1DAY/OBS/
COMP TX2=TSHIFT(TX2,4D)
COMP TX3=TSHIFT(TX3,1D)
COMP TX4=AMREG(PF,TX2,TX3)
PUT.A TX4=A=CHICKLKV F=COMPUTED5
ENDMACRO
```

Execution:

```
ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: MREG.DSS
                 Unit:  71;  DSS Version: 6-HD
```

```
-----DSS--- ZREAD Unit  71; Vers.   11:  /LINEAR REGRESSION/COEFFICIENTS/5YEARS////
-----DSS--- ZREAD Unit  71; Vers.    3:  /CHICKENT/TX2/FLOW/01JAN1972/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    1:  /CHICKENT/TX2/FLOW/01JAN1973/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    1:  /CHICKENT/TX2/FLOW/01JAN1974/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    1:  /CHICKENT/TX2/FLOW/01JAN1975/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    1:  /CHICKENT/TX2/FLOW/01JAN1976/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    1:  /CHICKENT/TX2/FLOW/01JAN1977/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    5:  /BUCKNER/TX3/FLOW/01JAN1972/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    3:  /BUCKNER/TX3/FLOW/01JAN1973/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    3:  /BUCKNER/TX3/FLOW/01JAN1974/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    3:  /BUCKNER/TX3/FLOW/01JAN1975/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    3:  /BUCKNER/TX3/FLOW/01JAN1976/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    3:  /BUCKNER/TX3/FLOW/01JAN1977/1DAY/OBS/
WARNING -- EXISTING VARIABLE RE-DEFINED
WARNING -- EXISTING VARIABLE RE-DEFINED
-----DSS---ZWRITE Unit  71; Vers.    4:  /CHICKLKV/TX3/FLOW/01JAN1972/1DAY/COMPUTED5/
-----DSS---ZWRITE Unit  71; Vers.    4:  /CHICKLKV/TX3/FLOW/01JAN1973/1DAY/COMPUTED5/
-----DSS---ZWRITE Unit  71; Vers.    4:  /CHICKLKV/TX3/FLOW/01JAN1974/1DAY/COMPUTED5/
-----DSS---ZWRITE Unit  71; Vers.    4:  /CHICKLKV/TX3/FLOW/01JAN1975/1DAY/COMPUTED5/
-----DSS---ZWRITE Unit  71; Vers.    4:  /CHICKLKV/TX3/FLOW/01JAN1976/1DAY/COMPUTED5/
-----DSS---ZWRITE Unit  71; Vers.    4:  /CHICKLKV/TX3/FLOW/01JAN1977/1DAY/COMPUTED5/
  -----DSS---ZCLOSE Unit:  71,   File: MREG.DSS
```



Name: *CONIC* *Conic interpolation from elevation/area table*
Use: *CO TY=CONIC(TX,TB,input,output,[scale])*

Interpolate values for TX using conic interpolation table TB and store the result in TY. TX and TY variables must be regular or irregular time series data. TB is a paired data variable that can be created using program DSSPD. The first paired data values contain the initial conic depth in feet or meters "x(1)" and storage in acre-feet or cubic meters below the first elevation "y(1)". The rest of the paired values "x(2--),y(2--)" contain the elevation in ft-msl or m-msl and area in acres or sq. meters. If the initial conic depth is undefined, the function will calculate one. The third and fourth parameters are used to specify the type of input (TX) and output (TY) data desired. The input type has to be either STORAGE or ELEVATION. The output type has to be either STORAGE, ELEVATION, or AREA. An optional fifth parameter can be used to specify a scale value to use with the input and output storage values. The default scale value is 1.0.

Example: The following example uses the CONIC function to convert elevation values to storage and surface area based on the conic interpolation table stored in variable PF.

Macro:

```
MACRO TCONIC
CLEAR ALL
TIME 01JAN1993 2400 10JAN1993 2400
GET TX=TESTDB.DSS:/CONIC FUNCTION/EAHGH/STAGE/01JAN1993/1DAY/TEST/
GET PF=TESTDB.DSS:/CONIC DATA/EAHGH/ELEV-AREA////
COMP TY=CONIC(TX,PF,ELEV,STOR)
COMP TA=CONIC(TX,PF,ELEV,AREA)
SD TY UNITS=AC-FT TYPE=INST
SD TA UNITS=ACRES TYPE=INST
PUT.A TY=TESTDB.DSS:/CONIC FUNCTION/EAHGH/STORAGE/01JAN1993/1DAY/CAL/
PUT.A TY=TESTDB.DSS:/CONIC FUNCTION/EAHGH/AREA/01JAN1993/1DAY/CAL/
TAB TX TY TA
TAB PF
ENDMACRO
```

Execution:

```
I>!R TCONIC
  ALL VARIABLES HAVE BEEN CLEARED
    -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                    Unit:  71;  DSS Version: 6-IA
 -----DSS--- ZREAD Unit  71; Vers.    1:  /CONIC FUNCTION/EAHGH/STAGE/01JAN1993/1DAY/TEST/
 -----DSS--- ZREAD Unit  71; Vers.    1:  /CONIC DATA/EAHGH/ELEV-AREA////
 -----DSS---ZWRITE Unit  71; Vers.    4:  /CONIC FUNCTION/EAHGH/STORAGE/01JAN1993/1DAY/CAL/
```

-----DSS---ZWRITE Unit 71; Vers. 1: /CONIC FUNCTION/EAHGH/AREA/01JAN1993/1DAY/CAL/

```
Variables ----          TX          TY          TA
Units     ----         FEET       AC-FT       ACRES
Time   Date
2400 01JAN1993    3336.000   486002.400    5777.025
2400 02JAN1993    3338.000   497635.600    5856.243
2400 03JAN1993    3340.000   509427.800    5936.000
2400 04JAN1993    3360.000   639317.800    7050.000
2400 05JAN1993    3380.000   793573.900    8310.000
2400 06JAN1993    3400.000   973569.400    9593.000
2400 07JAN1993    3450.000  1538441.000   13160.000
2400 08JAN1993    3500.000  2330557.000   18613.000
2400 09JAN1993    3525.000  2832292.000   21421.140
2400 10JAN1993    3565.000  3774320.000   25736.000

VARIABLE=PF
 UNITS = FEET      ACRES
 TYPES = UNT       UNT
 LABEL = SURFACE
 SHIFT =          OFFSET =          DATUM =
   X-AXIS        SURFACE
     292.7000  486000.0000    Initial conic depth in feet "x(1)" and storage in acre-feet below the first elevation
                              "y(1)"
    3336.0000    5777.0000
    3340.0000    5936.0000
    3350.0000    6500.0000
    3360.0000    7050.0000
    3370.0000    7750.0000
    3380.0000    8310.0000
    3390.0000    9052.0000
    3400.0000    9593.0000
    3410.0000   10220.0000
    3420.0000   10920.0000
    3430.0000   11600.0000
    3440.0000   12380.0000
    3450.0000   13160.0000
    3460.0000   14110.0000
    3470.0000   15570.0000
    3480.0000   16480.0000
    3490.0000   17182.0000
    3500.0000   18613.0000
    3510.0000   19880.0000
    3520.0000   20850.0000
    3530.0000   22000.0000
    3540.0000   22970.0000
    3550.0000   24080.0000
    3560.0000   25160.0000
    3565.0000   25736.0000
```

Name: *CORR* *Compute correlation coefficients*
Use: *CO SY=CORR(TX1,TX2,INDEX)*

Compute the correlation coefficients, determination coefficients, and standard errors of regression between the two variables TX1 and TX2. TX1 and TX2 must be of the same type (either uniform time series or irregular time series) and contain the same number of rows or ordinates. Variable INDEX is used to set the scalar variable SY as follows:

Index	*Description*
1	*Number of valid pairs for correlation.*
2	*Regression constant.*
3	*Regression coefficient.*
4	*Determination coefficient.*
5	*Standard error of regression.*
6	*Determination coefficient adjusted for degrees of freedom.*
7	*Standard error adjusted for degrees of freedom.*

Example: The following example uses the statistical function CORR to determine a linear regression equation for a downstream flow gage dependent on the observed flow values on one upstream flow gage. Note the use of the time shift function and the TIME.N command to account for the routing time of flows from the upstream to the downstream gage.

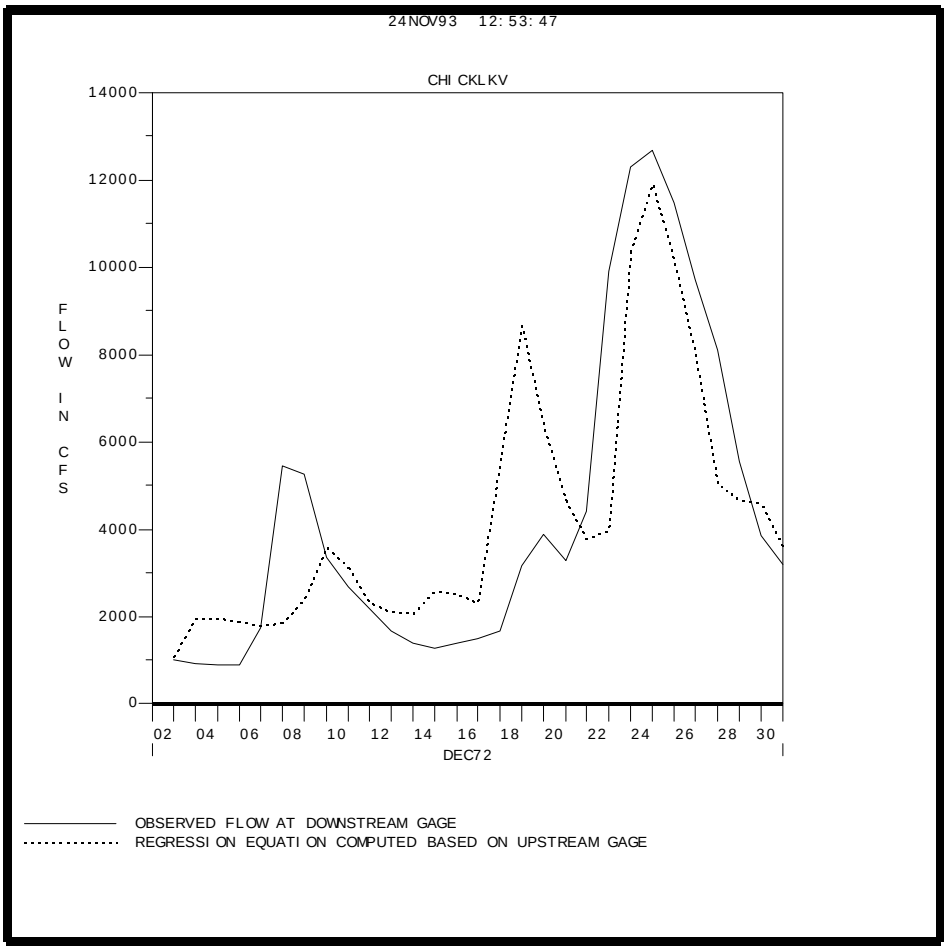
Macro:

```
MACRO TCORR
CL ALL
TIME 0100 03DEC1972 2400 30DEC1972
GET TY=MREG.DSS:/CHICKLKV/TX1/FLOW/01JAN1972/1DAY/OBS/
TIME.N 0100 01DEC1972 2400 28DEC1972
GET TX=MREG.DSS:/CHICKENT/TX2/FLOW/01JAN1972/1DAY/OBS/
COMP TX=TSHIFT(TX,+2D)
COMP RCON=CORR(TX,TY,2)
COMP RCOE=CORR(TX,TY,3)
COMP TYCOM=RCOE*TX
COMP TYCOM=TYCOM+RCON
PUT.A TYCOM=MREG.DSS:/CHICKLKV/TX1/FLOW/01JAN1972/1DAY/REG-EQ/
ENDMACRO
```

Execution:

```
I>!R TCORR

 ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: MREG.DSS
                    Unit:  71;  DSS Version: 6-HD
-----DSS--- ZREAD Unit  71; Vers.    4:  /CHICKLKV/TX1/FLOW/01JAN1972/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    3:  /CHICKENT/TX2/FLOW/01JAN1972/1DAY/OBS/
WARNING -- EXISTING VARIABLE RE-DEFINED
 NUMBER OF VALID PAIRS FOR CORRELATION      28.000000
 REGRESSION CONSTANT    1465.112000
 REGRESSION COEFFICIENT       2.002728
 DETERMINATION COEFFICIENT   6.243303E-01
 STANDARD ERROR OF REGRESSION    2257.026000
 DETERMINATION COEFFICIENT ADJ DEGREES OF FREEDOM   6.098815E-01
 STANDARD ERROR ADJ FOR DEGREES OF FREEDOM    2300.020000
 WARNING -- EXISTING VARIABLE RE-DEFINED
-----DSS---ZWRITE Unit  71; Vers.    5:  /CHICKLKV/TX1/FLOW/01JAN1972/1DAY/REG-EQ/
```



Name: *COS, SIN, TAN Trigonometric functions*
Use: *CO TY=COS(TX), TY=SIN(TX), TY=TAN(TX)*

Compute the trigonometric function of TX and store in TY. TX and TY can be the same variable. TX can be scalar, time series or paired data. If a value of TX is undefined resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged.

Example: The following examples are only meant to show the proper syntax of its application. Note that the parameter in the trigonometric function must be expressed in radian and not in degrees.

Macro:

```
MACRO TCOS
!* Change 45 degrees to radians
COMP RAD=45/180
COMP RAD=RAD*3.141593
COMP SCOS=COS(RAD)
COMP SSIN=SIN(RAD)
COMP STAN=TAN(RAD)
SHOW SCOS,SSIN,STAN
$CONTINUE
ENDMACRO
```

Execution:

```
I>!R TCOS

VARIABLE=SCOS
 SCOS    =   7.071067E-01

VARIABLE=SSIN
 SSIN    =   7.071067E-01

VARIABLE=STAN
 STAN    =   1.000
```

Name: *COUNT* *Count valid or missing data values*
Use: *CO SY=COUNT(TX,TYPE)*

This function is used to count the valid and missing values in time series variable TX and stores it in scalar variable SY. Possible values for TYPE are: "VALID" and "MISSING".

Example: The following example of the COUNT function is only meant to show the proper syntax of its application.

Macro:

```
MACRO TCOUNT
CL ALL
TIME 0100 08APR83 1200 10APR83
GET TX=testdb.dss:/SCIOTO/MISSING   RIVER/FLOW/01APR1983/1HOUR/OBS/
COMP NVALID=COUNT(TX,VALID)
  COMP NMISS=COUNT(TX,MISSING)
  SHOW NVALID,NMISS
ENDMACRO
```

Execution:

```
I>!R TCOUNT
  ALL VARIABLES HAVE BEEN CLEARED
    -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                    Unit:  71;  DSS Version: 6-IA
 -----DSS--- ZREAD Unit  71; Vers.    1:  /SCIOTO/MISSING RIVER/FLOW/01APR1983/1HOUR/OBS/

  NUMBER VALID=      55  NUMBER MISSING=       5
  NUMBER VALID=      55  NUMBER MISSING=       5
 VARIABLE=NVALID
 NVALID  =      55.000000
 VARIABLE=NMISS
 NMISS   =       5.000000

Variables ----     TX
Units     ----     CFS
Time   Date
0030 08APR1983    11995.000
0130 08APR1983     5628.000
0230 08APR1983     4832.000
0330 08APR1983     3241.000
0430 08APR1983     4837.000
0530 08APR1983     3243.000
0630 08APR1983     4042.000
0730 08APR1983     5641.000
0830 08APR1983     6445.000
0930 08APR1983     4058.000
1030 08APR1983     4065.000
1130 08APR1983     5670.000
1230 08APR1983     4878.000
1330 08APR1983     3283.000
1430 08APR1983     4084.000
1530 08APR1983     4084.000
1630 08APR1983        M
1730 08APR1983     3286.000
1830 08APR1983     3284.000
1930 08APR1983     3283.000
2030 08APR1983     3281.000
2130 08APR1983     3279.000
2230 08APR1983     2476.000
2330 08APR1983     3275.000
0030 09APR1983     2473.000
0130 09APR1983     2448.000
0230 09APR1983     2448.000
0330 09APR1983     3275.000
0430 09APR1983     5682.000
0530 09APR1983     5682.000
0630 09APR1983     6486.000
0730 09APR1983     4888.000
0830 09APR1983        M
0930 09APR1983        M
1030 09APR1983        M
1130 09APR1983        M
1230 09APR1983     9761.000
1330 09APR1983    11380.000
1430 09APR1983     8169.000
1530 09APR1983    11398.000
      .            .
      .            .
```

Name: *CMA* *Centered moving average smoothing*
Use: *CO TY=CMA(TX,NP[,CLEVEL])*

Compute a centered, moving average of NP values in TX and store in TY. NP must be odd and greater than 2. The default CLEVEL is "LEVEL3" and NP/2 values at the beginning and end of TY will be undefined. If a value in TX is undefined or a concurrent IF condition is not satisfied, then the resulting TY values are the averages of one less value. TX must be a regular-interval time-series and TY and TX cannot be the same. Useful for filtering instantaneous data containing high-frequency variations. Units and type are unchanged. The following levels are available:

LEVEL1 *Only valid values will be averaged. Valid values that do not have all valid values within the averaging period number will be missing. Values at the start and end of the data that do not have enough valid values to average will be assigned missing values.*

LEVEL2 *Only valid values will be averaged. Valid values that do not have all valid values within the averaging period number will be missing. Valid values at the start and end of the data that do not have enough valid values to average will be averaged over a reduced number of values.*

LEVEL3 *(Default setting) All values will be averaged based on valid values within the averaging period number. If all values are missing for the period number, the value will be missing. Values at the start and end of the data that do not have enough valid values to average will be assigned missing values.*

LEVEL4 *All values will be averaged based on valid values within the averaging period number. If all values are missing for the period number, the value will be missing. Values at the start and end of the data that do not have valid values to average will be averaged based on a reduced number of values.*

Example: The following example uses the CMA function to take flow values and smooth them using a moving average of 3.

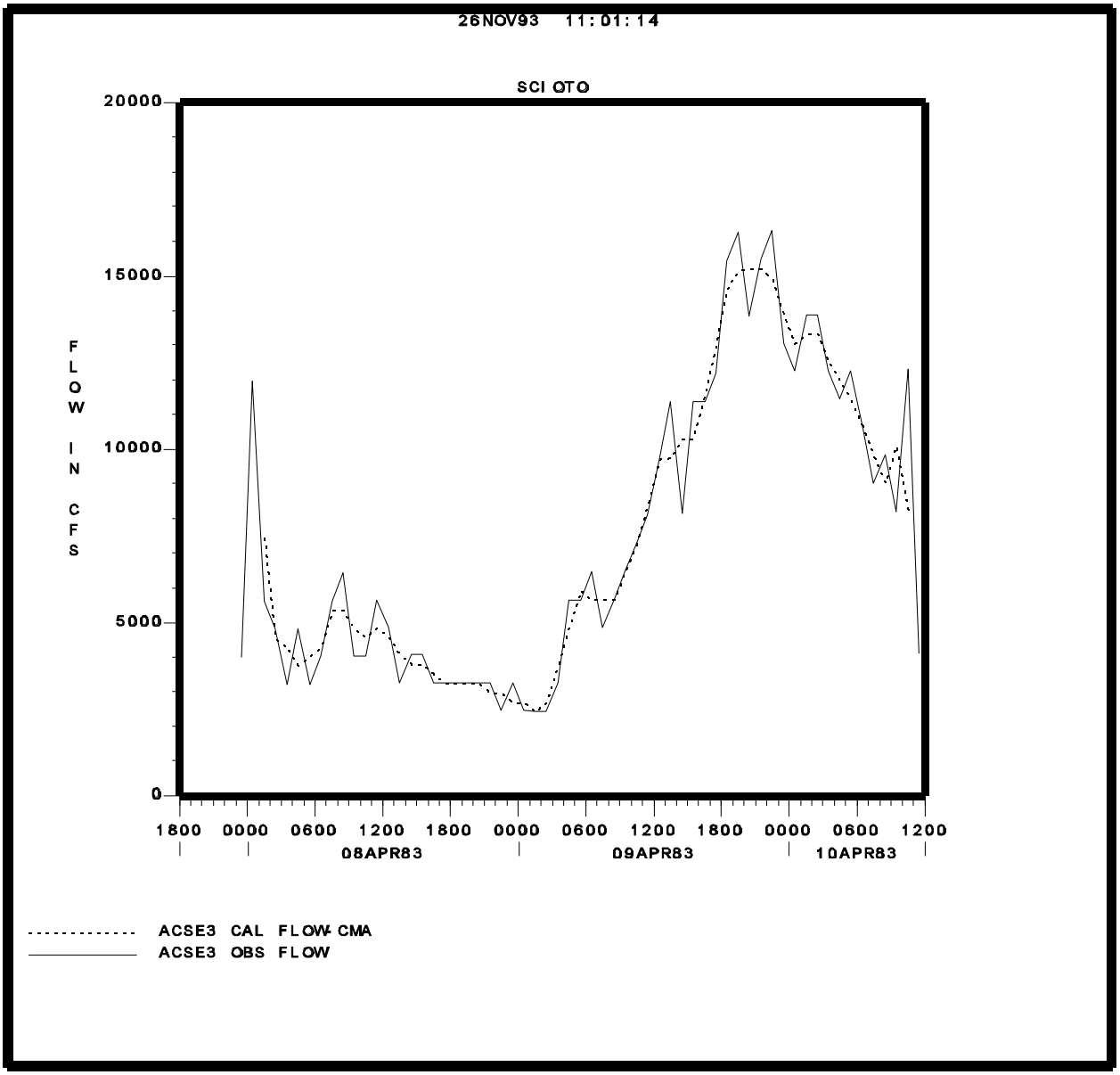
Macro:

```
MACRO TCMA
** COMPUTE A CENTERED ,MOVING AVERAGE OF NUMPOINT VALUES IN FLOW AND PUT.A

CL ALL
TIME 08APR1983 0100 10APR1983 1200
GET FLOW=testdb.dss:/SCIOTO/ACSE3/FLOW/01APR1983/1HOUR/OBS/
COMP AVE=CMA(FLOW,3,LEVEL3)
PUT.A AVE=C=FLOW-CMA F=CAL
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
I>!R TCMA
 ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                    Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.    3:  /SCIOTO/ACSE3/FLOW/01APR1983/1HOUR/OBS/
-----DSS---ZWRITE Unit  71; Vers.    1:  /SCIOTO/ACSE3/FLOW-CMA/01APR1983/1HOUR/CAL/
```



Name: *DDT* *Differences per unit time*
Use: *CO TY=DDT(TX)*

Computes the successive differences in TX per time period:

TY(t)=(TX(t)-TX(t-1))/DT

where DT is the time difference in days between t and t-1. If a value of TX is undefined, TY is undefined. If a concurrent IF condition is not satisfied, TY is unchanged. TY and TX cannot be the same variable. TY is assigned the type 'PER-AVER.' The units of TY are undefined. An example of the use of DDT is the computation of reservoir inflow from outflow and the change in storage, where the change in storage is transformed to average flow volume per day by the function. The conversion factor for storage to flow is 0.50416 when the storage change is in ac-ft day.

Example: The following example uses the DDT function to convert storage into change in storage per day. The macro in total converts pool elevation to storage, tailwater stage to outflow, and computed inflow to a reservoir.

Macro:

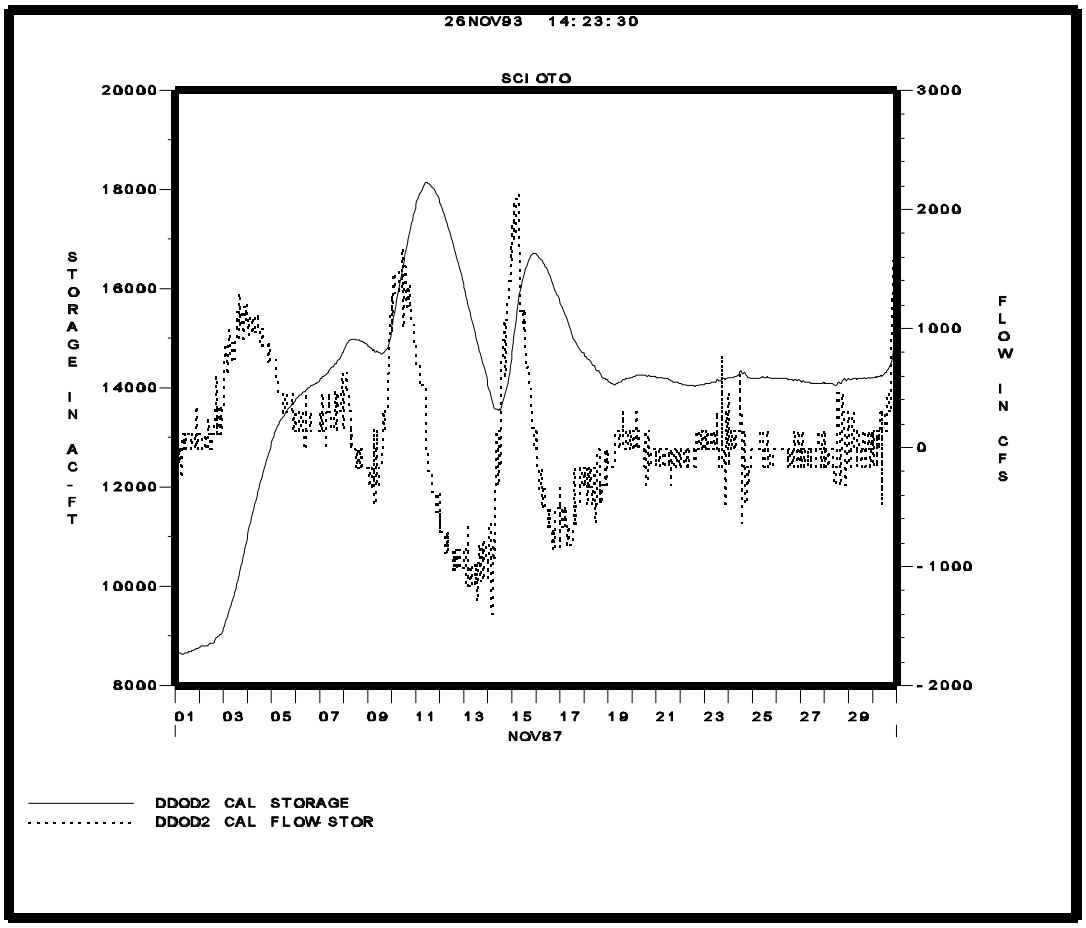
```
MACRO TDDT

CL ALL
TI 01NOV87 0100 30NOV87 2400
GET PELEV=MASTDB:/SCIOTO/DDOD2/ELEV/01APR1983/1HOUR/OBS/
GET PELST=/SCIOTO/DDOD2/ELEV-STOR////
COM STOR=RTABLE(PELEV,PELST)
SD STOR UNITS=AC-FT TYPE=INST-VAL
PUT.A STOR=/SCIOTO/DDOD2/STORAGE/01APR1983/1HOUR/CAL/
COM DSTIN=DDT(STOR)
COM DSTOR=DSTIN*.50416
SD DSTOR UNITS=CFS TYPE=PER-AVE
PUT.A DSTOR=/SCIOTO/DDOD2/FLOW-STOR/01APR1983/1HOUR/CAL/
CLEAR PELEV PELST
GET TELEV=/SCIOTO/DDOOF/ELEV/01APR1983/1HOUR/OBS/
GET OSTFL=/SCIOTO/DDOOF/ELEV-FLOW///USGS TABLE NO.10/
COM QOINST=RTABLE(TELEV,OSTFL)
CLEAR TELEV DSTIN
COM QOUT=TS2(QOINST,1H,0H)
COM QIN=DSTOR+QOUT
SD QIN U=CFS TY=PER-AVER
PUT.A QIN=MASTDB:/SCIOTO/DDOD2/FLOW-RES IN/01APR1983/1HOUR/CAL/
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Appendix D-18
Function Examples

Execution:

```
I>!R TDDT
 ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: MASTDB.DSS
                    Unit:  71;  DSS Version: 6-GU
-----DSS--- ZREAD Unit  71; Vers.  140:  /SCIOTO/DDOD2/ELEV/01NOV1987/1HOUR/OBS/
-----DSS--- ZREAD Unit  71; Vers.    1:  /SCIOTO/DDOD2/ELEV-STOR////
LOGLOG TRANSFORMATION
OFFSET =     890.00     SHIFT =       .00     DATUM =       .00
-----DSS---ZWRITE Unit  71; Vers.    1:  /SCIOTO/DDOD2/STORAGE/01NOV1987/1HOUR/CAL/
-----DSS---ZWRITE Unit  71; Vers.    1:  /SCIOTO/DDOD2/FLOW-STOR/01NOV1987/1HOUR/CAL/
 PELEV CLEARED
 PELST CLEARED
-----DSS--- ZREAD Unit  71; Vers.  140:  /SCIOTO/DDOOF/ELEV/01NOV1987/1HOUR/OBS/
-----DSS--- ZREAD Unit  71; Vers.    1:  /SCIOTO/DDOOF/ELEV-FLOW///USGS TABLE NO.10/
LOGLOG TRANSFORMATION
OFFSET =      79.00     SHIFT =       .00     DATUM =     800.00
 TELEV CLEARED
 DSTIN CLEARED
-----DSS---ZWRITE Unit  71; Vers.    1:  /SCIOTO/DDOD2/FLOW-RES IN/01NOV1987/1HOUR/CAL/
```



Name: *DECPAR* *Decaying basin wetness parameter*
Use: *CO TY=DECPAR(TY,TZ,R)*

Compute a time-series of parameters TY as a function of TZ and R:

*TY(t)=R*TY(t-1)+TZ(t)*

where R is the decay rate and TZ is precipitation. R is less than 1. The function extends TY: the first value in the series TY is used as the starting value, and any other TY are computed in the sequence. If the first value in TY is undefined, it is assumed to be zero. The first TZ value is ignored. TY and TZ must be regular-interval time series with identical times. R should be appropriate for the interval. The type and units of TY are unchanged. An IF condition has no effect.

Example: The following example uses the DECPAR function to create a set of decaying basin wetness parameters.

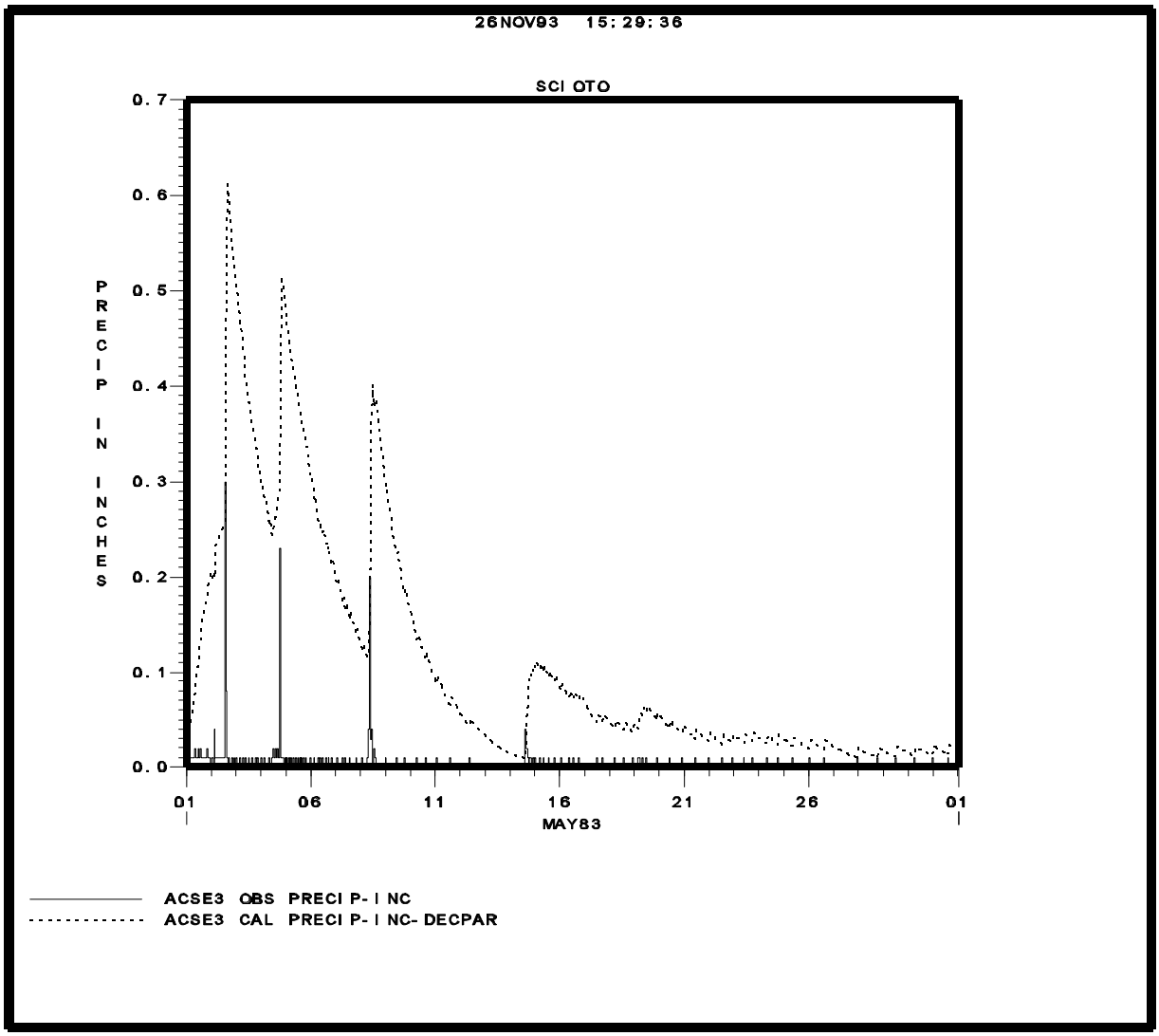
Macro:

```
MACRO TDECPAR

** PAR(t)=RATE*PAR(t-1) + PPT(t)   DECAYING WETNESS PARAMETER
CL ALL
TI 01MAY83 0100 31MAY83 2400
GET PPT=testdb.dss:/SCIOTO/ACSE3/PRECIP-INC/01MAY1983/1HOUR/OBS/
GET PAR=testdb.dss:/SCIOTO/ACSE3/PRECIP-INC/01MAY1983/1HOUR/OBS/
COM PAR=DECPAR(PAR,PPT,.97)
PUT.A PAR=C=PRECIP-INC-DECPAR,F=CAL
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
I>!R TDECPAR
 ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                      Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.   22:  /SCIOTO/ACSE3/PRECIP-INC/01MAY1983/1HOUR/OBS/
-----DSS--- ZREAD Unit  71; Vers.   22:  /SCIOTO/ACSE3/PRECIP-INC/01MAY1983/1HOUR/OBS/
TYPE FOR PARAM NO. 3 IS =          4
WARNING -- EXISTING VARIABLE RE-DEFINED
R =    9.700000E-01
-----DSS---ZWRITE Unit  71; Vers.    1:  /SCIOTO/ACSE3/PRECIP-INC-DECPAR/01MAY1983/1HOUR/CAL/
```



Name: *DIFF* *Successive differences*
Use: *CO TY=DIFF(TX)*

Compute the successive differences of TX and store in TY. TX and TY cannot be the same variable. TX must be of type INST-VAL or INST-CUM. If a value of TX is undefined, resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged. The type and units of TY are undefined.

Example: The following example uses the DIFF function to take cumulative precipitation and convert it into incremental precipitation.

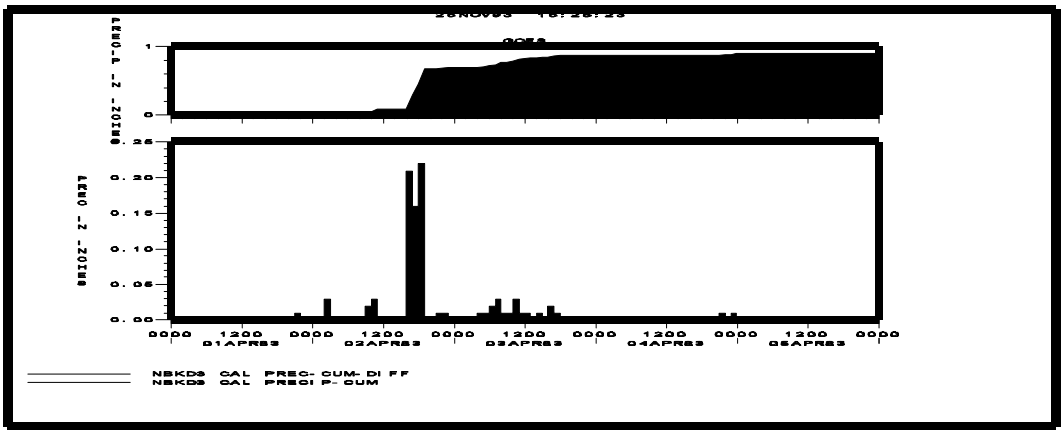
Macro:

```
MACRO TDIFF

** SUCCESSIVE DIFFERENCE
CL ALL
TIME 01APR83 0100 05APR83 2400
GET PPT=testdb.dss:/GOES/NBKD3/PRECIP-CUM/01ARP1983/1HOUR/CAL/
COM PDIFF=DIFF(PPT)
SD PDIFF UNITS=INCHES TYPE=PER-CUM
PUT.A PDIFF=C=PREC-CUM-DIFF,F=CAL
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
I>!R TDIFF
  ALL VARIABLES HAVE BEEN CLEARED
    -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                    Unit:  71;  DSS Version: 6-IA
 -----DSS--- ZREAD Unit  71; Vers.    2:  /GOES/NBKD3/PRECIP-CUM/01APR1983/1HOUR/CAL/
 -----DSS---ZWRITE Unit  71; Vers.    1:  /GOES/NBKD3/PREC-CUM-DIFF/01APR1983/1HOUR/CAL/
```



Name: *ESTLIN* *Estimate values for missing data*
Use: *CO TY=ESTLIN(TX,NMAX,QFLAG)*

Linearly interpolate estimates for values in TX with flags equal or lesser in quality to QFLAG and place the results in TY. Do not estimate more that NMAX continuous missing values. Possible flags, in order of decreasing quality, are: Q = questionable and M = missing. An IF condition has no effect. TX must be a time-series, and TX and TY may be the same variable. Type and units of TY are the same as TX.

Example: The following example uses the ESTLIN function to linearly interpolate values for missing values. The maximum number of missing consecutive values it will interpolate values for is specified as four.

Macro:

```
MACRO TESTLIN
** Estimate values for missing flow data. Do not attempt to estimate
** more than 4 values and only consider missing, not questionable values.

CL ALL
TI 24MAY83 1600 25MAY83 1600
GET FLOW=testdb.dss:/SCIOTO/BMRI2/FLOW/01MAY1983/1HOUR/OBS/
COM EFLOW=ESTLIN(FLOW,4,M)
PUT.A EFLOW=B=TESTLIN,F=CAL
TAB.F FLOW EFLOW
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
 ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                    Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.   24:  /SCIOTO/BMRI2/FLOW/01MAY1983/1HOUR/OBS/

 ** WARNING: FLAGGED DATA AT BEGINNING OF RECORD
 ** WARNING: FLAGGED DATA AT END OF RECORD
 ** WARNING: TOO MANY MISSING POINTS BETWEEN NON MISSING VALUES

-----DSS---ZWRITE Unit  71; Vers.    6:  /SCIOTO/TESTLIN/FLOW/01MAY1983/1HOUR/CAL/

Variables ----      FLOW        EFLOW
Units     ----      CFS         CFS
Time   Date
1600 24MAY1983      M           M                                   0400 25MAY1983      M           M
1700 24MAY1983      93.934      93.934                              0500 25MAY1983      M           M
1800 24MAY1983      92.643      92.643                              0600 25MAY1983      82.696      82.696
1900 24MAY1983      91.676      91.676                              0700 25MAY1983      82.376      82.376
2000 24MAY1983      90.710      90.710                              0800 25MAY1983      81.419      81.419
2100 24MAY1983      90.388      90.388                              0900 25MAY1983      81.101      81.101
2200 24MAY1983      88.781      88.781                              1000 25MAY1983      M           80.464
2300 24MAY1983      87.817      87.817                              1100 25MAY1983      79.827      79.827
2400 24MAY1983      87.817      87.817                              1200 25MAY1983      79.508      79.508
0100 25MAY1983      M           M                                   1300 25MAY1983      79.190      79.190
0200 25MAY1983      M           M                                   1400 25MAY1983      78.554      78.554
0300 25MAY1983      M           M                                   1500 25MAY1983      77.602      77.602
```

```
1600 25MAY1983      M         M
```

Name: *ESTPPT* *Estimate values for missing precipitation data*

Use: *CO TY=ESTPPT(TX,NMAX,QFLAG)*

Linearly interpolate estimates for cumulative precipitation values in TX with quality flags equal to or lesser in quality to QFLAG and place the results in TY. Possible flags, in order of decreasing quality, are: Q = questionable and M = missing. If the values bracketing the missing period are increasing with time, do not estimate more that NMAX continuous missing values. If the values bracketing the missing period are equal, then estimate any number of missing values. If the values bracketing the missing period are decreasing with time, do not estimate any missing values. TX must be a data type of INST-CUM. An IF condition has no effect.

Example: The following example uses the ESTPPT function to take cumulative precipitation and linearly interpolate values for missing values. The maximum number of missing consecutive values it will interpolate values for is specified as five.

Macro:

```
MACRO TESTPPT
CL ALL
TI 01APR83 0100 03APR83 1000
GET PPT=testdb.dss:/SCIOTO/ACSE3/PRECIP/01APR1983/1HOUR/OBS/
COM EPPT=ESTPPT(PPT,5,M)
PUT.A EPPT=B=TESTPPT,F=CAL
TAB.F PPT EPPT
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
 ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                    Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.  142:  /SCIOTO/ACSE3/PRECIP/01APR1983/1HOUR/OBS/
 ** WARNING: SOME POINTS WERE NOT ESTIMATED
-----DSS---ZWRITE Unit  71; Vers.    3:  /SCIOTO/TESTPPT/PRECIP/01APR1983/1HOUR/CAL/

Variables ----      PPT        EPPT
Units      ----   INCHES      INCHES
Time   Date
0100 01APR1983     13.630     13.630          1300 01APR1983     13.630     13.630          0100 02APR1983     13.630     13.630
0200 01APR1983     13.630     13.630          1400 01APR1983     13.630     13.630          0200 02APR1983     13.630     13.630
0300 01APR1983     13.630     13.630          1500 01APR1983     13.630     13.630          0300 02APR1983     13.630     13.630
0400 01APR1983      M         13.630          1600 01APR1983     13.630     13.630          0400 02APR1983     13.630     13.630
0500 01APR1983      M         13.630          1700 01APR1983     13.630     13.630          0500 02APR1983     13.630     13.630
0600 01APR1983      M         13.630          1800 01APR1983     13.630     13.630          0600 02APR1983     13.630     13.630
0700 01APR1983      M         13.630          1900 01APR1983     13.630     13.630          0700 02APR1983     13.630     13.630
0800 01APR1983      M         13.630          2000 01APR1983     13.630     13.630          0800 02APR1983     13.630     13.630
0900 01APR1983      M         13.630          2100 01APR1983     13.630     13.630          0900 02APR1983     13.630     13.630
1000 01APR1983     13.630     13.630          2200 01APR1983     13.630     13.630          1000 02APR1983     13.650     13.650
1100 01APR1983     13.630     13.630          2300 01APR1983     13.630     13.630          1100 02APR1983      M          M
1200 01APR1983     13.630     13.630          2400 01APR1983     13.630     13.630          1200 02APR1983      M          M
```

```
1300 02APR1983      M         M
1400 02APR1983      M         M
1500 02APR1983      M         M
1600 02APR1983      M         M
1700 02APR1983     14.410     14.410
1800 02APR1983     14.450     14.450
1900 02APR1983     14.490     14.490
2000 02APR1983     14.540     14.540
2100 02APR1983     14.580     14.580
2200 02APR1983     14.620     14.620
2300 02APR1983     14.650     14.650
2400 02APR1983     14.690     14.690
0100 03APR1983     14.720     14.720
0200 03APR1983     14.750     14.750
0300 03APR1983      M        14.780
0400 03APR1983      M        14.810
0500 03APR1983      M        14.840
0600 03APR1983     14.870     14.870
0700 03APR1983     14.900     14.900
0800 03APR1983     14.920     14.920
0900 03APR1983     14.950     14.950
1000 03APR1983     14.970     14.970
```

Name: *FMA* *Forward moving average*
Use: *CO TY=FMA(TX,NP)*

Compute a moving average of the last NP values in TX and store in TY. NP must greater than 2. NP values at the beginning of TY will be missing. If a value in TX is missing, the value is not used for computing TY, and the average is over one less value. At least 2 values of TX must be defined, else TY is missing. Useful for computing flow durations. Also, may be used to determine parameters for use in the SCRN2 screening function. Units and type are unchanged.

Example: The following example uses the forward moving average function to take flow values and smooth them using a forward moving average of 3 and 5.

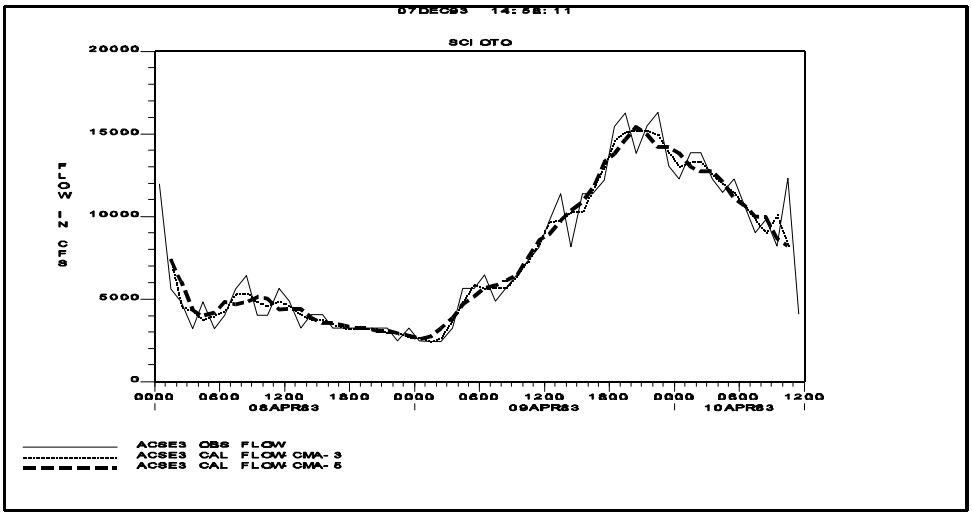
Macro:

```
MACRO TFMA
CL ALL
TIME 08APR1983 0100 10APR1983 1200
GET FLOW=testdb.dss:/SCIOTO/ACSE3/FLOW/01APR1983/1HOUR/OBS/
COMP AVE=FMA(FLOW,3)
PUT.A AVE=C=FLOW-FMA-3 F=CAL
COMP AVE=FMA(FLOW,5)
PUT.A AVE=C=FLOW-FMA-5 F=CAL
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
 ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                      Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.    3:  /SCIOTO/ACSE3/FLOW/01APR1983/1HOUR/OBS/
-----DSS---ZWRITE Unit  71; Vers.    1:  /SCIOTO/ACSE3/FLOW-FMA-3/01APR1983/1HOUR/CAL/
```

WARNING -- EXISTING VARIABLE RE-DEFINED
-----DSS---ZWRITE Unit 71; Vers. 1: /SCIOTO/ACSE3/FLOW-FMA-5/01APR1983/1HOUR/CAL/



Name: *GENTSR* *Generate a regular interval time series*
Use: *CO TY=GENTSR(DT,TOFF,Y0,QFLAG)*

Generate a new, regular-interval time series TY with a time interval of DT, a time interval offset of TOFF, a constant value Y0, and all values internally flagged with QFLAG. TOFF is time from the beginning of the standard interval to the actual time of the data. DT and TOFF are expressed as units of time as nnT, where T is M for minutes, H for hours, and D for days. Possible flags are: N = none, M = missing, E = estimated, and Q = questionable. If Y0 is -901., then the flag is automatically M. Units and type must be set independently (use SDD). An IF condition has no effect.

Example: The following example uses the GENTSR function to generate a regular time series that has all its values set to missing. The generated time series is then merged with an irregular time series that reports at hourly intervals, but not always every hour. The following macro inserts missing values into the irregular time series and then uses the TS1 function to convert the irregular time series to regular.

Macro:

```
MACRO TGENTSR
CL ALL
TIME 01MAY1983 0100 02MAY1983 0100
COM FLOWG=GENTSR(1H,0H,-901.,M)
GET FLOW=testdb.dss:/SCIOTO/BMRI2/FLOW/01MAY1983/IR-MONTH/OBS-I/
COM FLOWM=MRG(FLOW,FLOWG,N)
COM FLOWR=TS1(FLOWM,1H,0H)
TAB.F FLOW FLOWG FLOWR
PUT.A FLOWR=testdb.dss:/SCIOTO/BMRI2/FLOW/01MAY1983/1HOUR/CAL-I/
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
 ALL VARIABLES HAVE BEEN CLEARED
*** WARNING : QUALITY FLAG RESET TO NONE
   -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                    Unit:  71;  DSS Version: 6-IA

Variables ----      FLOW        FLOWG       FLOWR
Units     ----      CFS                     CFS
Time   Date
0100 01MAY1983     100.735      M        100.735
0200 01MAY1983     100.085      M        100.085
0300 01MAY1983      99.436      M         99.436
0400 01MAY1983       -          M         M
0500 01MAY1983      98.140      M         98.140
0600 01MAY1983       -          M         M
0700 01MAY1983      96.844      M         96.844
0800 01MAY1983      96.196      M         96.196
0900 01MAY1983      95.873      M         95.873
1000 01MAY1983       -          M         M
1100 01MAY1983      97.491      M         97.491
1200 01MAY1983     102.035      M        102.035
1300 01MAY1983     106.597      M        106.597
1400 01MAY1983       -          M         M
1500 01MAY1983     109.211      M        109.211
1600 01MAY1983     111.504      M        111.504
1700 01MAY1983     112.815      M        112.815
1800 01MAY1983       -          M         M
1900 01MAY1983     115.443      M        115.443
2000 01MAY1983     115.443      M        115.443
2100 01MAY1983     114.457      M        114.457
2200 01MAY1983       -          M         M
```

```
2300 01MAY1983    112.815    M    112.815
2400 01MAY1983    112.487    M    112.487
0100 02MAY1983    111.504    M    111.504
```

Name: *INT* *Truncate to whole numbers*

Use: *CO TY=INT(TX)*

Truncate to a whole number TX and store in TY. TX and TY can be the same variable. TX can be scalar, time series or paired data. If a value of TX is undefined resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged.

Name: *LAST* *Last valid value in a time series*

Use: *CO SY=LAST(TX)*

Find the last valid value in time-series TX and place it in scalar SY. Ignores missing values or values concurrent with an unsatisfied IF condition.

Name: *LOG* *Natural log base "e"*

Use: *CO TY=LOG(TX)*

Compute the natural log base e of TX and store in TY. TX and TY can be the same variable. TX can be scalar, time series or paired data. If a value of TX is undefined, resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged. If a value in TX is 0 or negative, the value in TY will be set to missing.

Name: *LOG10* *Log base 10*

Use: *CO TY=LOG10(TX)*

Compute the log base 10 of TX and store in TY. TX and TY can be the same variable. TX can be scalar, time series or paired data. If a value of TX is undefined resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged. If a value in TX is 0 or negative, the value in TY will be set to missing.

Example: *The following demonstrate the use of the LAST, INT, LOG, and LOG10 functions.*

Macro:

```
 MACRO TLAST
CL ALL
TIME 01APR83 0100 05APR83 2400
GET PPT=testdb.dss:/GOES/NBKD3/PRECIP-INC/01ARP1983/1HOUR/OBS/
COMP PACC=ACC(PPT)
COMP SUM=LAST(PACC)
COMP SUMI=INT(SUM)
COMP SLOG=LOG(SUM)
COMP SLOG10=LOG10(SUM)
SHOW SUM SUMI SLOG SLOG10
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
 ALL VARIABLES HAVE BEEN CLEARED
  -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                     Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.  145:  /GOES/NBKD3/PRECIP-INC/01APR1983/1HOUR/OBS/
```

```
 FINAL ACCUMULATED VALUE =   8.999997E-01
MAX=       .900  MIN=       .000  MEAN=       .566  LAST VALUE=      .900
VARIABLE=SUM
SUM     =   8.999997E-01
VARIABLE=SUMI
SUMI    =   0.000000E+00
VARIABLE=SLOG
SLOG    =  -1.053608E-01
VARIABLE=SLOG10
SLOG10  =  -4.575762E-02
```

Name: *MATE* *Generate data pairs from two time-series*
Use: *CO PF=MATE(TX,TY,SORT/NOSORT)*

Derive a paired-function PF by pairing values in the time series TX and TY. TX and TY values must have identical times. (Functions TS1 and TS4 may be useful in conjunction with MATE). The paired data are sorted into ascending order of TX if SORT is specified. The units of PF are respectively the same as those of TX and TY. The types of PF are undefined. An IF condition has no effect.

Example: The following example uses the accumulation function to take incremental precipitation and convert into a cumulative value.

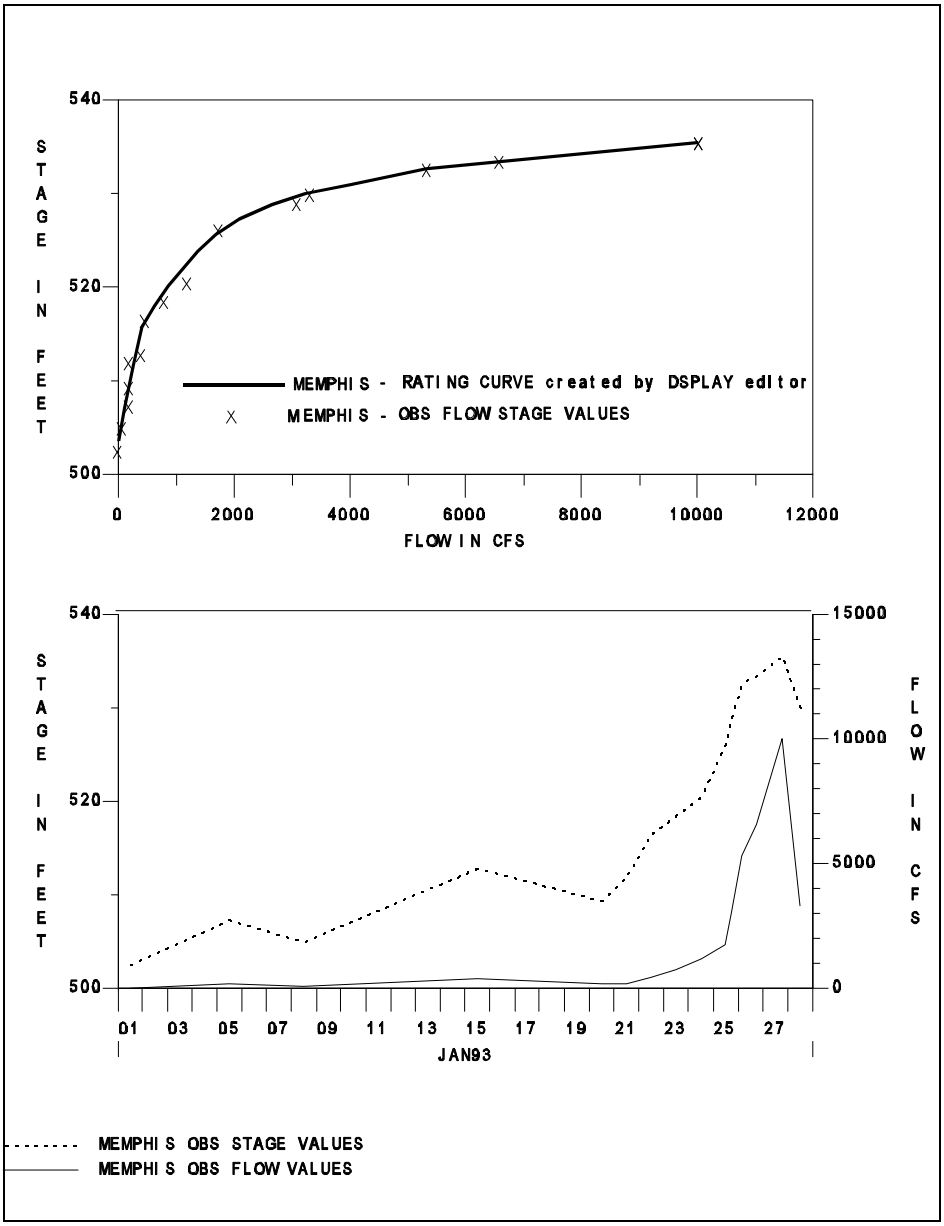
Macro:

```
MACRO TMATE
** COMPUTE DATA PAIRS FROM TWO TIME-SERIES
CL ALL
TIME 01JAN93 0100 30JAN93 2400
GET STAGE=testdb.dss:/DAVIS/MEMPHIS/STAGE/01JAN1993/IR-MONTH/OBS/
GET FLOW=testdb.dss:/DAVIS/MEMPHIS/FLOW/01JAN1993/IR-MONTH/OBS/
COMP PAIRS=MATE(FLOW,STAGE,SORT)
COMP PAIRN=MATE(FLOW,STAGE,NOSORT)
SD PAIRS LABEL=RATING-S
PUT.A PAIRS=testdb.dss:/DAVIS/MEMPHIS/FLOW-STAGE/01JAN1993/SORTED/CAL/
SD PAIRN LABEL=RATING-N
PUT.A PAIRN=testdb.dss:/DAVIS/MEMPHIS/FLOW-STAGE/01JAN1993/NOSORTED/CAL/
TAB.F FLOW STAGE
TAB.F PAIRS PAIRN
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
 ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                      Unit:  71;  DSS Version: 6-IA
-----DSS---ZWRITE Unit  71; Vers.    1:  /DAVIS/MEMPHIS/FLOW-STAGE/01JAN1993/SORTED/CAL/
-----DSS---ZWRITE Unit  71; Vers.    1:  /DAVIS/MEMPHIS/FLOW-STAGE/01JAN1993/NOSORTED/CAL/
```

Variables ----	FLOW	STAGE
Units ----	CFS	FEET
Time Date		
1200 01JAN1993	10.000	502.500
1200 05JAN1993	200.000	507.400
1200 08JAN1993	85.000	505.100
1200 15JAN1993	400.000	512.900
1200 20JAN1993	205.000	509.400
1200 21JAN1993	200.000	512.000
1200 22JAN1993	475.000	516.500
1200 23JAN1993	800.000	518.500
1200 24JAN1993	1200.000	520.500
1200 25JAN1993	1750.000	526.300
1800 25JAN1993	3100.000	529.000
0300 26JAN1993	5350.000	532.800
1800 26JAN1993	6600.000	533.600
1800 27JAN1993	10035.000	535.500
1200 28JAN1993	3330.000	530.000



VARIABLE=PAIRS
UNITS = CFS FEET

TYPES = UNT UNT
LABEL = RATING-S
SHIFT = OFFSET = DATUM =

X-AXIS	RATING-S
10.0000	502.5000
85.0000	505.1000
200.0000	507.4000
200.0000	512.0000
205.0000	509.4000
400.0000	512.9000
475.0000	516.5000
800.0000	518.5000
1200.0000	520.5000
1750.0000	526.3000
3100.0000	529.0000
3330.0000	530.0000
5350.0000	532.8000
6600.0000	533.6000
10035.0000	535.5000

VARIABLE=PAIRN
UNITS = CFS FEET

TYPES = UNT UNT
LABEL = RATING-N
SHIFT = OFFSET = DATUM =

X-AXIS	RATING-N
10.0000	502.5000
200.0000	507.4000
85.0000	505.1000
400.0000	512.9000
205.0000	509.4000
200.0000	512.0000
475.0000	516.5000
800.0000	518.5000
1200.0000	520.5000
1750.0000	526.3000
3100.0000	529.0000
5350.0000	532.8000
6600.0000	533.6000
10035.0000	535.5000
3330.0000	530.0000

Name: *MAX* *Maximum value in a time series*
Use: *CO SY=MAX(TX)*

Find the maximum value in time-series TX and place it in scalar SY. Ignore missing values or values concurrent with an unsatisfied IF condition.

Name: *MEAN* *Mean value in a time series*
Use: *CO SY=MEAN(TX)*

Compute the mean value in time-series TX and place the result in scalar SY. Ignore missing values or values concurrent with an unsatisfied IF condition.

Name: *MIN* *Minimum value in a time series*
Use: *CO SY=MIN(TX)*

Find the minimum value in time-series TX and place the result in scalar SY. Ignore missing values or values concurrent with an unsatisfied IF condition.

Example: The following demonstrate the use of the MAX, MIN, and MEAN functions.

Macro:

```
I> MACRO TMEAN
CL ALL
TIME 09APR1983 1200 10APR1983 1200
GET FLOW=testdb.dss:/SCIOTO/ACSE3/FLOW/01APR1983/1HOUR/OBS/
COM FMIN=MIN(FLOW)
COM FMAX=MAX(FLOW)
COM FMEAN=MEAN(FLOW)
SHOW FMIN FMAX FMEAN
TAB.F FLOW
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
 ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                  Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.    3:  /SCIOTO/ACSE3/FLOW/01APR1983/1HOUR/OBS/

MAX=   16321.000  MIN=   4156.000  MEAN=   11737.840  LAST VALUE=   4156.000
MAX=   16321.000  MIN=   4156.000  MEAN=   11737.840  LAST VALUE=   4156.000
MAX=   16321.000  MIN=   4156.000  MEAN=   11737.840  LAST VALUE=   4156.000

VARIABLE=FMIN                          Units     ----     CFS                          1730 09APR1983   12217.000
FMIN  =  4156.000000                   Time   Date                                    1830 09APR1983   15466.000
VARIABLE=FMAX                          1130 09APR1983    8146.000                     1930 09APR1983   16282.000
FMAX  = 16321.000000                   1230 09APR1983    9761.000                     2030 09APR1983   13872.000
VARIABLE=FMEAN                         1330 09APR1983   11380.000                     2130 09APR1983   15501.000
FMEAN = 11737.840000                   1430 09APR1983    8169.000                     2230 09APR1983   16321.000
                                       1530 09APR1983   11398.000                     2330 09APR1983   13093.000
Variables ----    FLOW                 1630 09APR1983   11403.000                     0030 10APR1983   12278.000
```

```
0130 10APR1983   13902.000
0230 10APR1983   13908.000
0330 10APR1983   12287.000
0430 10APR1983   11479.000
0530 10APR1983   12294.000
0630 10APR1983   10669.000
0730 10APR1983    9043.000
0830 10APR1983    9863.000
0930 10APR1983    8234.000
1030 10APR1983   12324.000
1130 10APR1983    4156.000
```

Name: *MREG* *Multiple linear regression function.*
Use: *CO PF=MREG(TY,TX1,TX2,...,TXn,min,max)*

*This function is used to determine the coefficients (Bn) of a linear regression equation of the general form (Y = B0 + B1*X1 + B2*X2 + Bn*Xn). Where Y is the dependent variable and the X1, X2, and Xn are independent variables. The MREG function requires that the function parameters TY, TX1, TX2, and TXn be regular time series variables and that they all have the same time interval. It is further required that the time variables be retrieved in the same sequential order as specified in the function parameter list. In other words TY is retrieved from DSS first followed by TX1 and so on. The number of independent time series variables that can be specified is limited to one less the maximum number of time series variables that the program allows. On the DOS PC this value will be 4 independent variables. The calculated linear regression coefficients will be stored in a paired function variable (PF) specified by the user. The user should store this variable to DSS with the PUT command if it is to be used later with the AMREG function. Two optional function parameters "min, max" are available to specify a range of values that control which values in the dependent variable are to be used for calculating the regression coefficients. These two parameters must be specified as the last two parameters and if specified, both must be specified as real numbers. The statistics output is written to file "MREG.REP" in addition to being printed out in the regular output.*

Example: The following example uses the MREG function to determine the linear regression coefficients for a downstream flow gage in relation to two upstream flow gages. The coefficients are saved as paired data to a DSS file and may later be used by the AMREG function to predict flows in the future from the observed flow values at the two upstream gages.

Macro:

```
MACRO MREG2
 TIME 0100 01OCT1972 2400 30NOV1977
 GET TX1=MREG.DSS:/CHICKLKV/TX1/FLOW/01JAN1972/1DAY/OBS/
 GET TX2=MREG.DSS:/CHICKENT/TX2/FLOW/01JAN1972/1DAY/OBS/
 GET TX3=MREG.DSS:/BUCKNER/TX3/FLOW/01JAN1972/1DAY/OBS/
 COMP TX2=TSHIFT(TX2,4D)
 COMP TX3=TSHIFT(TX3,1D)
 COMP PF=MREG(TX1,TX2,TX3)
 PUT.A PF=MREG.DSS:/LINEAR REGRESSION/COEFFICIENTS/5YEARS////
 ENDMACRO
```

Execution:

```
-----DSS---ZOPEN:  Existing File Opened,  File: MREG.DSS
                   Unit:  71;  DSS Version: 6-HD
-----DSS--- ZREAD Unit  71; Vers.    4:  /CHICKLKV/TX1/FLOW/01JAN1972/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    3:  /CHICKLKV/TX1/FLOW/01JAN1973/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    3:  /CHICKLKV/TX1/FLOW/01JAN1974/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    3:  /CHICKLKV/TX1/FLOW/01JAN1975/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    3:  /CHICKLKV/TX1/FLOW/01JAN1976/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    3:  /CHICKLKV/TX1/FLOW/01JAN1977/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    3:  /CHICKENT/TX2/FLOW/01JAN1972/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    1:  /CHICKENT/TX2/FLOW/01JAN1973/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    1:  /CHICKENT/TX2/FLOW/01JAN1974/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    1:  /CHICKENT/TX2/FLOW/01JAN1975/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    1:  /CHICKENT/TX2/FLOW/01JAN1976/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    1:  /CHICKENT/TX2/FLOW/01JAN1977/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    5:  /BUCKNER/TX3/FLOW/01JAN1972/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    3:  /BUCKNER/TX3/FLOW/01JAN1973/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    3:  /BUCKNER/TX3/FLOW/01JAN1974/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    3:  /BUCKNER/TX3/FLOW/01JAN1975/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    3:  /BUCKNER/TX3/FLOW/01JAN1976/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    3:  /BUCKNER/TX3/FLOW/01JAN1977/1DAY/OBS/
WARNING -- EXISTING VARIABLE RE-DEFINED
WARNING -- EXISTING VARIABLE RE-DEFINED
 Multiple Regression Report Written to: "mreg.rep"
ERROR SUM OF SQUARES   1.032530E+10
REGRESSION SUM OF SQUARES   5.177439E+10
TOTAL SUM OF SQUARES   6.209969E+10
COEFFICIENT OF MULTIPLE DETERMINATION   8.337302E-01

REGRESSION COEFFICIENTS
  NUMBER      VALUE
     0    1108.491000
     1        .525309
     2       3.366069
 INDEX     1882.000000
 ---------------------------------------------------
Mean absolute error   =   .40 % (1882. d)
Mean negative error   =  -.17 % ( 576. d)
Mean positive error   =   .50 % (1306. d)
Total volume error    =  -.30 % (1882. d)
```

```
Percentage of errors <= 5 %    =      12.805530
Percentage of errors <= 10 %   =      23.910730
Percentage of errors <= 15 %   =      34.909670
Percentage of errors <= 20 %   =      43.942620
Percentage of errors <= 25 %   =      51.328370
Percentage of errors  > 25 %   =      48.671630
 ---------------------------------------------------
-----DSS---ZWRITE Unit  71; Vers.   11:  /LINEAR
REGRESSION/COEFFICIENTS/5YEARS////
   -----DSS---ZCLOSE Unit:  71,   File: MREG.DSS
```

Name: *MRG* *Merge two time series*
Use: *CO TY=MRG(TX,TZ,QFLAG)*

Merge TX with TZ. TY includes all values in TX and TZ, except where TX and TZ occur at the same time. In that case, the value for TX is used unless it is flagged with a quality equal to or less than QFLAG and TZ is flagged with a quality greater than QFLAG. Possible flags, in order of decreasing quality, are: N = no flag,
E = estimated, Q = questionable and M = missing. An IF condition has no effect. The type and units of TY are undefined.

Execution: The following example uses the GENTSR function to generate a regular time series that has all its values set to missing. The generated time series is then merged, using MRG, with an irregular time series that reports at hourly intervals, but not always every hour. The following macro inserts missing values into the irregular time series and then uses the TS1 function to convert the irregular time series to regular.

Macro:

```
MACRO TGENTSR
CL ALL
TIME 01MAY1983 0100 02MAY1983 0100
COM FLOWG=GENTSR(1H,0H,-901.,M)
GET FLOW=testdb.dss:/SCIOTO/BMRI2/FLOW/01MAY1983/IR-MONTH/OBS-I/
COM FLOWM=MRG(FLOW,FLOWG,N)
COM FLOWR=TS1(FLOWM,1H,0H)
TAB.F FLOW FLOWG FLOWR
PUT.A FLOWR=testdb.dss:/SCIOTO/BMRI2/FLOW/01MAY1983/1HOUR/CAL-I/
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
 ALL VARIABLES HAVE BEEN CLEARED
*** WARNING : QUALITY FLAG RESET TO NONE
  -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                Unit:  71;  DSS Version: 6-IA

Variables ----      FLOW        FLOWG       FLOWR
Units     ----      CFS                     CFS
Time   Date
0100 01MAY1983    100.735      M        100.735
0200 01MAY1983    100.085      M        100.085
0300 01MAY1983     99.436      M         99.436
0400 01MAY1983      -          M         M
0500 01MAY1983     98.140      M         98.140
0600 01MAY1983      -          M         M
0700 01MAY1983     96.844      M         96.844
0800 01MAY1983     96.196      M         96.196
0900 01MAY1983     95.873      M         95.873
1000 01MAY1983      -          M         M
1100 01MAY1983     97.491      M         97.491
1200 01MAY1983    102.035      M        102.035
1300 01MAY1983    106.597      M        106.597
1400 01MAY1983      -          M         M
1500 01MAY1983    109.211      M        109.211
1600 01MAY1983    111.504      M        111.504
1700 01MAY1983    112.815      M        112.815
1800 01MAY1983      -          M         M
1900 01MAY1983    115.443      M        115.443
2000 01MAY1983    115.443      M        115.443
2100 01MAY1983    114.457      M        114.457
2200 01MAY1983      -          M         M
2300 01MAY1983    112.815      M        112.815
2400 01MAY1983    112.487      M        112.487
0100 02MAY1983    111.504      M        111.504
```

Name: *MRGP* *Merge two paired data series*
Use: *CO PY=MRGP(PX,PZ,NCURVE)*

Merge PZ with PX. PY includes all values in PX and the curve specified by the index curve number NCURVE in PZ. It is required that PX and PZ have the same number of data values and that the X axis values for both PX and PZ are identical. NCURVE is used to specify the curve number data in PZ that is to be merged with PX. A value of 0 for NCURVE signifies that all curves in PZ are to be merged with PX.

Example: The following example uses the MRGP function to merge two paired data curves into a single paired data that contains both curves.

Macro:

```
MACRO TMRGP
 TIME T-3D T
 GET XX=testdb.dss:/PULS//FLOW-STORAGE////
 GET YY=F=TEST
 COMP ZZ=MRGP(XX,YY,1)
 PUT ZZ=B=TMRGP
 TAB.F XX YY ZZ
 ENDMACRO
```

Execution:

```
-----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                   Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.    2:  /PULS//FLOW-STORAGE////
-----DSS--- ZREAD Unit  71; Vers.    1:  /PULS//FLOW-STORAGE///TEST/
-----DSS---ZWRITE Unit  71; Vers.    6:  /PULS/TMRGP/FLOW-STORAGE///TEST/
-----DSS---ZCLOSE Unit:  71,   File: TESTDB.DSS
```

```
VARIABLE=XX
  UNITS = AC-FT      CFS
  TYPES = UNT        UNT
  LABEL = FLOW
  SHIFT =  OFFSET =  DATUM =

    X-AXIS        FLOW
       .0000       .0000
     10.0000      50.0000
    100.0000     500.0000
    120.0000    1000.0000
    320.0000    5000.0000
    550.0000   10000.0000
    800.0000   20000.0000
   1500.0000   40000.0000
   3000.0000   80000.0000
```

```
VARIABLE=YY
  UNITS = AC-FT      CFS
  TYPES = UNT        UNT
  LABEL = FLOW
  SHIFT =  OFFSET =  DATUM =

    X-AXIS        FLOW
       .0000       .0000
     10.0000      50.0000
    100.0000    3921.5690
    120.0000    4405.7790
    320.0000    9247.8850
    550.0000   15310.4600
    800.0000   24705.8800
   1500.0000   54453.4600
   3000.0000   95374.0000
```

```
VARIABLE=ZZ
  UNITS = AC-FT      AC-FT
  TYPES = UNT        UNT
  LABEL = FLOW
  LABEL = FLOW
  SHIFT =  OFFSET =  DATUM =

    X-AXIS        FLOW         FLOW
       .0000       .0000        .0000
     10.0000      50.0000      50.0000
    100.0000     500.0000    3921.5690
    120.0000    1000.0000    4405.7790
    320.0000    5000.0000    9247.8850
    550.0000   10000.0000   15310.4600
    800.0000   20000.0000   24705.8800
   1500.0000   40000.0000   54453.4600
   3000.0000   80000.0000   95374.0000
```

Name: *MUSK* *Muskingum hydrologic routing*
Use: *CO TY=MUSK(TX,NR,K,X)*

Route the uniform time series variable TX by the Muskingum hydrologic routing method and store it in variable TY. The "IF" compute options has no effect on this function. The "K" parameter is the Muskingum "k" in hours, "X" parameter is the Muskingum "x" (range between 0 and .5), and "NR" is the number of routing subreaches.

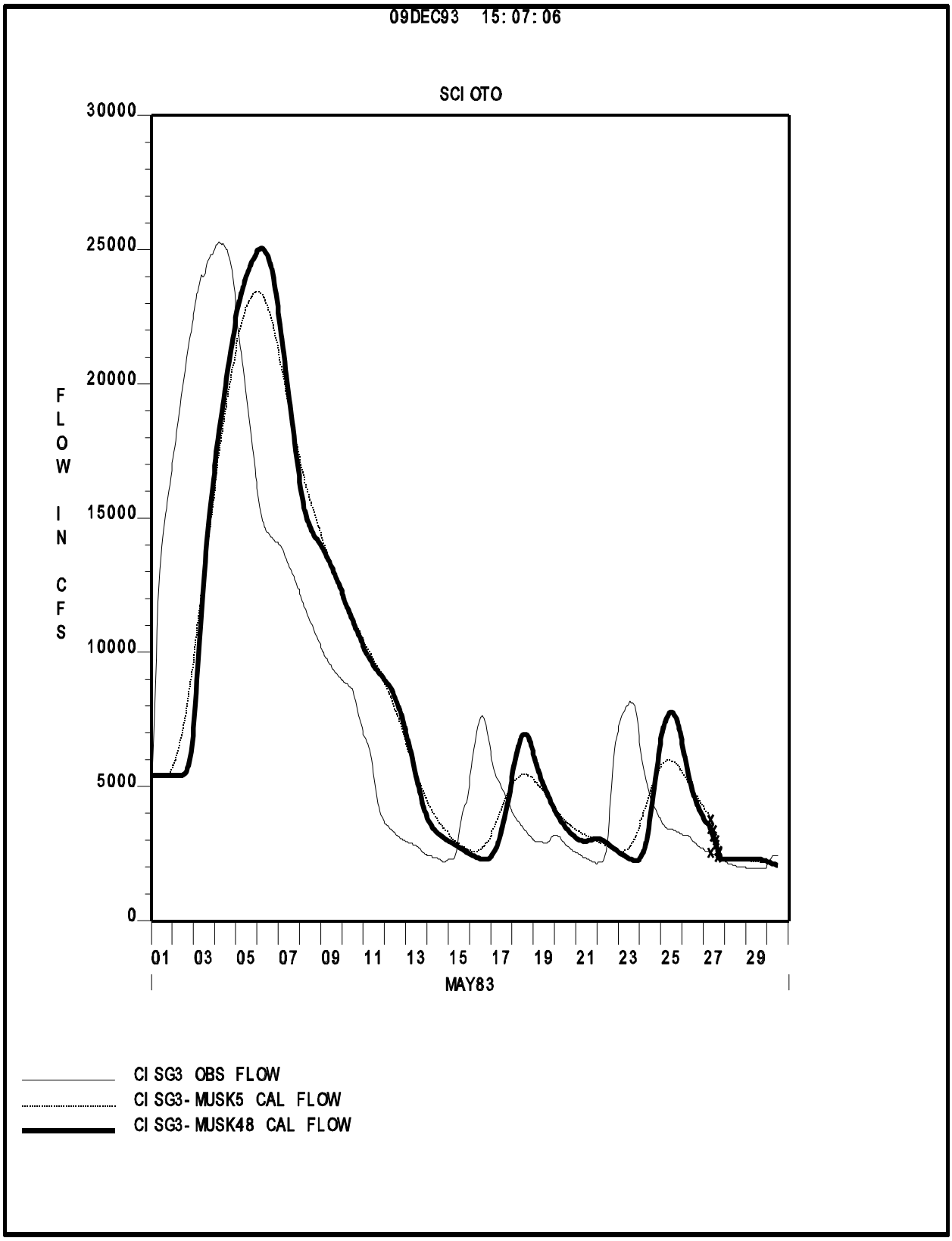
Example: The following example uses the MUSK function to perform a Muskingum routing using 5 and 48 routing steps.

Macro:

```
 MACRO TMUSK
** COMPUTE A MUSKINGUM ROUTING
CL ALL
TIME 01APR1983 0100 30MAY1983 1200
GET FLOW=testdb.dss:/SCIOTO/CISG3/FLOW/01MAY1983/1HOUR/OBS/
COMP RFLOW=MUSK(FLOW,5,48,.01)
PUT.A RFLOW=B=CISG3-MUSK5,F=CAL
COMP RFLOW=MUSK(FLOW,48,48,.01)
PUT.A RFLOW=B=CISG3-MUSK48,F=CAL
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
 ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                      Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.    1:  /SCIOTO/CISG3/FLOW/01APR1983/1HOUR/OBS/
-----DSS--- ZREAD Unit  71; Vers.   22:  /SCIOTO/CISG3/FLOW/01MAY1983/1HOUR/OBS/
 WARNING - NUMBER OF ROUTING STEPS SPECIFIED   5  NOT EQUAL TO COMPUTED   48
-----DSS---ZWRITE Unit  71; Vers.    1:  /SCIOTO/CISG3-MUSK5/FLOW/01APR1983/1HOUR/CAL/
-----DSS---ZWRITE Unit  71; Vers.    1:  /SCIOTO/CISG3-MUSK5/FLOW/01MAY1983/1HOUR/CAL/
WARNING -- EXISTING VARIABLE RE-DEFINED
-----DSS---ZWRITE Unit  71; Vers.    1:  /SCIOTO/CISG3-MUSK48/FLOW/01APR1983/1HOUR/CAL/
-----DSS---ZWRITE Unit  71; Vers.    1:  /SCIOTO/CISG3-MUSK48/FLOW/01MAY1983/1HOUR/CAL/
```


09DEC93 15:07:06
SCIOTO
FLOW IN CFS
30000
25000
20000
15000
10000
5000
0
01 03 05 07 09 11 13 15 17 19 21 23 25 27 29
MAY83
CISG3 OBS FLOW
CISG3-MUSK5 CAL FLOW
CISG3-MUSK48 CAL FLOW

Name: *NINT* *Round to nearest whole number*
Use: *CO TY=NINT(TX)*

Round to nearest whole number TX and store in TY. TX and TY can be the same variable. TX can be scalar, time series or paired data. If a value of TX is undefined resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged.

Example: The following example demonstrates how the NINT function will round up when a value is greater than .5 and round down when it is less than .5.

Macro:

```
MACRO TNINT
COMP VRUP=NINT(10.50)
COMP VRDN=NINT(10.49999)
SHOW VRUP VRDN
ENDMACRO
```

Execution:

```
VARIABLE=VRUP
VRUP    =     11.000000

VARIABLE=VRDN
VRDN    =     10.000000
```

Name: *OLY* *Olympic smoothing*
Use: *CO TY=OLY(TX,NP[,CLEVEL])*

Compute a smoothed time-series from TX using the Olympic smoothing scheme: same as centered, moving average except the minimum and maximum values in the span NP are ignored. Place the result in TY. Units and type are unchanged. The following levels are available:

LEVEL1 *Only valid values will be averaged. Valid values that do not have all valid values within the averaging period number will be missing. Values at the start and end of the data that do not have enough valid values to average will be assigned missing values.*

LEVEL2 *Only valid values will be averaged. Valid values that do not have all valid values within the averaging period number will be missing. Valid values at the start and end of the data that do not have enough valid values to average will be averaged over a reduced number of values.*

LEVEL3 *(Default setting) All values will be averaged based on valid values within the averaging period number. If all values are missing for the period number, the value will be missing. Values at the start and end of the data that do not have enough valid values to average will be assigned missing values.*

LEVEL4 *All values will be averaged based on valid values within the averaging period number. If all values are missing for the period number, the value will be missing. Values at the start and end of the data that do not have valid values to average will be averaged based on a reduced number of values.*

Note: *Questionable and estimated flagged values are used in the computations.*

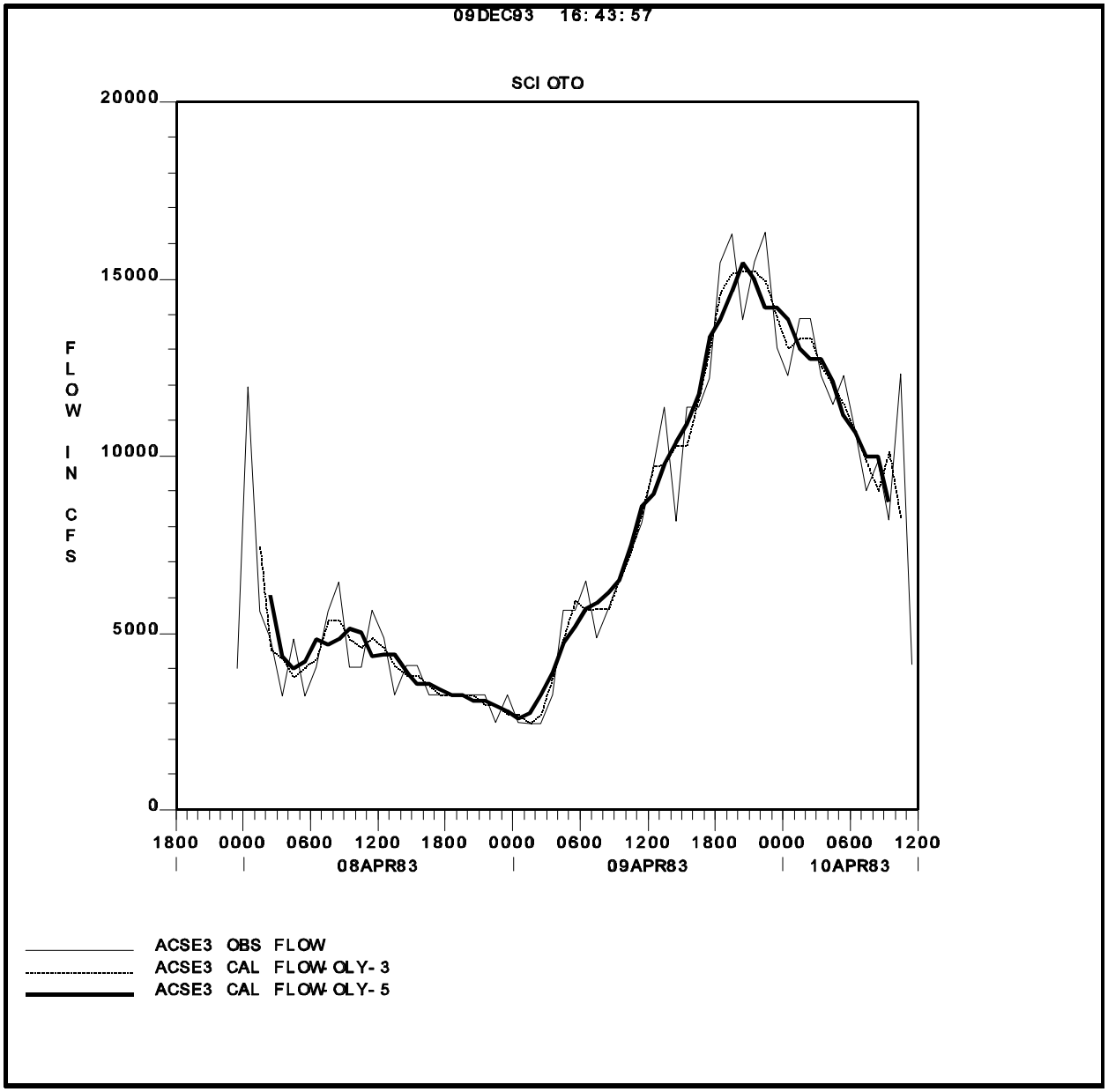
Example: The following example uses the OLY function to take flow values and smooth them using a moving average of 3 and 5.

Macro:

```
MACRO TOLY
CL ALL
TIME 08APR1983 0100 10APR1983 1200
GET FLOW=testdb.dss:/SCIOTO/ACSE3/FLOW/01APR1983/1HOUR/OBS/
COMP AVE3=OLY(FLOW,3,LEVEL3)
PUT.A AVE3=C=FLOW-OLY-3 F=CAL
COMP AVE5=OLY(FLOW,5,LEVEL3)
PUT.A AVE5=C=FLOW-OLY-5 F=CAL
TAB.F FLOW AVE3 AVE5
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
 ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                      Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.    3:  /SCIOTO/ACSE3/FLOW/01APR1983/1HOUR/OBS/
-----DSS---ZWRITE Unit  71; Vers.    1:  /SCIOTO/ACSE3/FLOW-OLY-3/01APR1983/1HOUR/CAL/
WARNING -- EXISTING VARIABLE RE-DEFINED
-----DSS---ZWRITE Unit  71; Vers.    1:  /SCIOTO/ACSE3/FLOW-OLY-5/01APR1983/1HOUR/CAL/
```


09DEC93 16:43:57
SCIOTO
20000
15000
10000
5000
0
FLOW IN CFS
1800 0000 0600 1200 1800 0000 0600 1200 1800 0000 0600 1200
08APR83
09APR83
10APR83
ACSE3 OBS FLOW
ACSE3 CAL FLOW-OLY-3
ACSE3 CAL FLOW-OLY-5

Variables ----		FLOW	AVE3	AVE5
Units	----	CFS	CFS	CFS
Time	Date			
0030	08APR1983	11995.000	M	M
0130	08APR1983	5628.000	7485.000	M
0230	08APR1983	4832.000	4567.000	6106.600
0330	08APR1983	3241.000	4303.333	4356.200
0430	08APR1983	4837.000	3773.667	4039.000
0530	08APR1983	3243.000	4040.667	4200.800
0630	08APR1983	4042.000	4308.667	4841.600
0730	08APR1983	5641.000	5376.000	4685.800
0830	08APR1983	6445.000	5381.333	4850.200
0930	08APR1983	4058.000	4856.000	5175.800
1030	08APR1983	4065.000	4597.667	5023.200
1130	08APR1983	5670.000	4871.000	4390.800
1230	08APR1983	4878.000	4610.333	4396.000
1330	08APR1983	3283.000	4081.667	4399.800
1430	08APR1983	4084.000	3817.000	3923.200
1530	08APR1983	4084.000	3818.333	3604.800
1630	08APR1983	3287.000	3552.333	3605.000
1730	08APR1983	3286.000	3285.667	3444.800
1830	08APR1983	3284.000	3284.333	3284.200
1930	08APR1983	3283.000	3282.667	3282.600
2030	08APR1983	3281.000	3281.000	3120.600
2130	08APR1983	3279.000	3012.000	3118.800
2230	08APR1983	2476.000	3010.000	2956.800
2330	08APR1983	3275.000	2741.333	2790.200
0030	09APR1983	2473.000	2732.000	2624.000
0130	09APR1983	2448.000	2456.333	2783.800
0230	09APR1983	2448.000	2723.667	3265.200
0330	09APR1983	3275.000	3801.667	3907.000
0430	09APR1983	5682.000	4879.667	4714.600
0530	09APR1983	5682.000	5950.000	5202.600
0630	09APR1983	6486.000	5685.333	5687.400
0730	09APR1983	4888.000	5691.000	5854.600
0830	09APR1983	5699.000	5701.667	6184.600
0930	09APR1983	6518.000	6516.333	6516.600
1030	09APR1983	7332.000	7332.000	7491.200
1130	09APR1983	8146.000	8413.000	8627.400
1230	09APR1983	9761.000	9762.333	8957.600
1330	09APR1983	11380.000	9770.000	9770.800
1430	09APR1983	8169.000	10315.670	10422.200
1530	09APR1983	11398.000	10323.330	10913.400
1630	09APR1983	11403.000	11672.670	11730.600
1730	09APR1983	12217.000	13028.670	13353.200
1830	09APR1983	15466.000	14655.000	13848.000
1930	09APR1983	16282.000	15206.670	14667.600
2030	09APR1983	13872.000	15218.330	15488.400
2130	09APR1983	15501.000	15231.330	15013.800
2230	09APR1983	16321.000	14971.670	14213.000
2330	09APR1983	13093.000	13897.330	14219.000
0030	10APR1983	12278.000	13091.000	13900.400
0130	10APR1983	13902.000	13362.670	13093.600
0230	10APR1983	13908.000	13365.670	12770.800
0330	10APR1983	12287.000	12558.000	12774.000
0430	10APR1983	11479.000	12020.000	12127.400

```
 0530 10APR1983   12294.000   11480.670   11154.400
 0630 10APR1983   10669.000   10668.670   10669.600
 0730 10APR1983    9043.000    9858.333   10020.600
 0830 10APR1983    9863.000    9046.667   10026.600
 0930 10APR1983    8234.000   10140.330    8724.000
 1030 10APR1983   12324.000    8238.000       M
 1130 10APR1983    4156.000       M          M
```

Name: *PERCON* *Period constants*
Use: *CO TY=PERCON(SY,TX)*

Generates a series TY at times concurrent with TX and with values equal to the previous chronological SY value. If no previous SY is present, then TY values are undefined. SY and TX may be irregular- or regular-interval time series. An IF condition has no effect.

Name: *POLY* *Polynomial transformation*
Use: *CO TY=POLY(TX,TP)*

Compute a polynomial transformation of TX using the polynomial coefficients TP. Store the result in TY. If a TX value is missing TY is undefined. If a concurrent IF condition is not satisfied, TY is unchanged. Units and type of TY are defined by TP. TP can be created with the utility DSSPD.

Example: The following example uses the PERCON and POLY functions to determine the surface area (AC) and evaporation (ACFT) for Black Butte Reservoir. Note the extensive use of PREAD function keys and variable parameters used with macros.

Macro:

```
MACRO TPERCON
!TEACH i TESTDB.DSS
!TEACH j TESTDB.DSS
!TEACH k TESTDB.DSS
!TEACH m .M
TIME 0100 01JAN93 1200 01DEC93
!R DEV ^i ^j ^k "STONY CR" "BLACK BUTTE"
ENDMACRO
```

```
MACRO DEV  %HPDB %EPDB %EVDB %APART %BPART
****|**********|****************************|**********************
**  Compute    Daily EVAP-RES AF
**                (Evap for Reservoir in ACFT)
**  INPUT:
**    %HPDB   Reservoir Elevation Database
**    %EPDB   Pan Evap Database
**    %EVDB   Lake Evap Database
**    %APART  "A" part of pathname
**    %BPART  "B" part of pathname
**  OUTPUT:
**    EV$     EVAP-RES AF (Reservoir Evaportion, acft)
**  VARIABLES:
**    EP$     Daily EVAP-PAN (Pan Evaporation at reservoir, inches)
**    HP$     Daily ELEV (Reservoir Elevation from Mean Sea Level).  Be sure
to
**            retrieve the previous value outside the time window.
**    POLY$   ELEV-AREA polynomial table for reservoir
**    COEF$   Evaporation coefficient time-series (period average values)
*****************************************************************************
*
CL ALL
GE EP$=%EPDB:/%APART/%BPART/EVAP-PAN//1DAY/OBS/
TI.N -30D
GE.P MCOEF$=TESTDB.DSS:/ALL CORPS RES//EVAP COEFS//1MON/AVG
MONTHLY/
TI.N +30D
CO COEF$=PERCON(MCOEF$,EP$)
TAB.F MCOEF$ EP$ COEF$
CO EV$=EP$*COEF$
CL MCOEF$
!RUN DLA  ^k  ^k "%APART" "%BPART"
GE LA$=^k:/%APART/%BPART/AREA-LAKE//1DAY//
CO EV$=LA$*EV$
CO EV$=RND(EV$,8,0)
SD EV$ UNITS=ACFT TYPE=PER-CUM
PU^m EV$=%EVDB:/%APART/%BPART/EVAP-LAKE AF//1DAY//
$CONTINUE
ENDMACRO
```

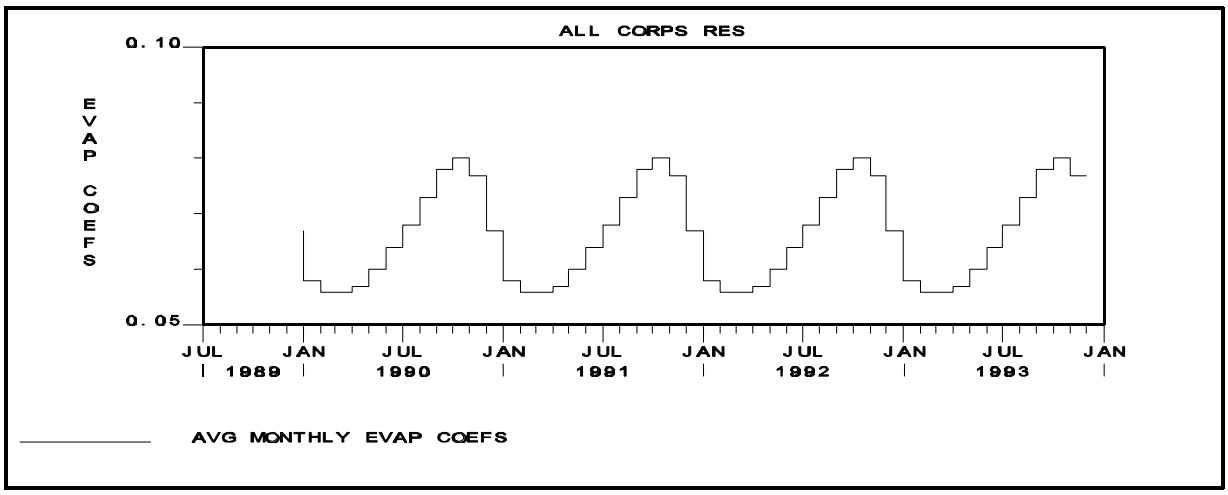
```
MACRO DLA %DHPDB %DLADB %APART %BPART
****|**********|****************************|**********************
**  Compute    Daily Midnight Lake Area
**  ELEV-AREA POLYNOMIALS
**  INPUT:
**    %DHPDB  Daily Reservoir Elevation Database
**    %DLADB  Daily Lake Area Database
**    %APART  "A" part of pathname
**    %BPART  "B" part of pathname
**  VARIABLES:
**    DHP$    ELEV (Reservoir Elevation from Mean Sea Level).
**    POLY$   ELEV-AREA polynomial table for reservoir
*****************************************************************************
***
GE DHP$=%DHPDB:/%APART/%BPART/ELEV//1DAY//
GE POLY$=TESTDB.DSS:/%APART/%BPART/ELEV-AREA///POLY/
STAT POLY$
TAB.F POLY$
CO DLA$=POLY(DHP$,POLY$)
CL POLY$
CO DLA$=RND(DLA$,8,0)
SD DLA$ UNITS=ACRES TYPE=INST-VAL
PU^m DLA$=%DLADB:/%APART/%BPART/AREA-LAKE//1DAY//
CL DLA$
$CONTINUE
ENDMACRO
```

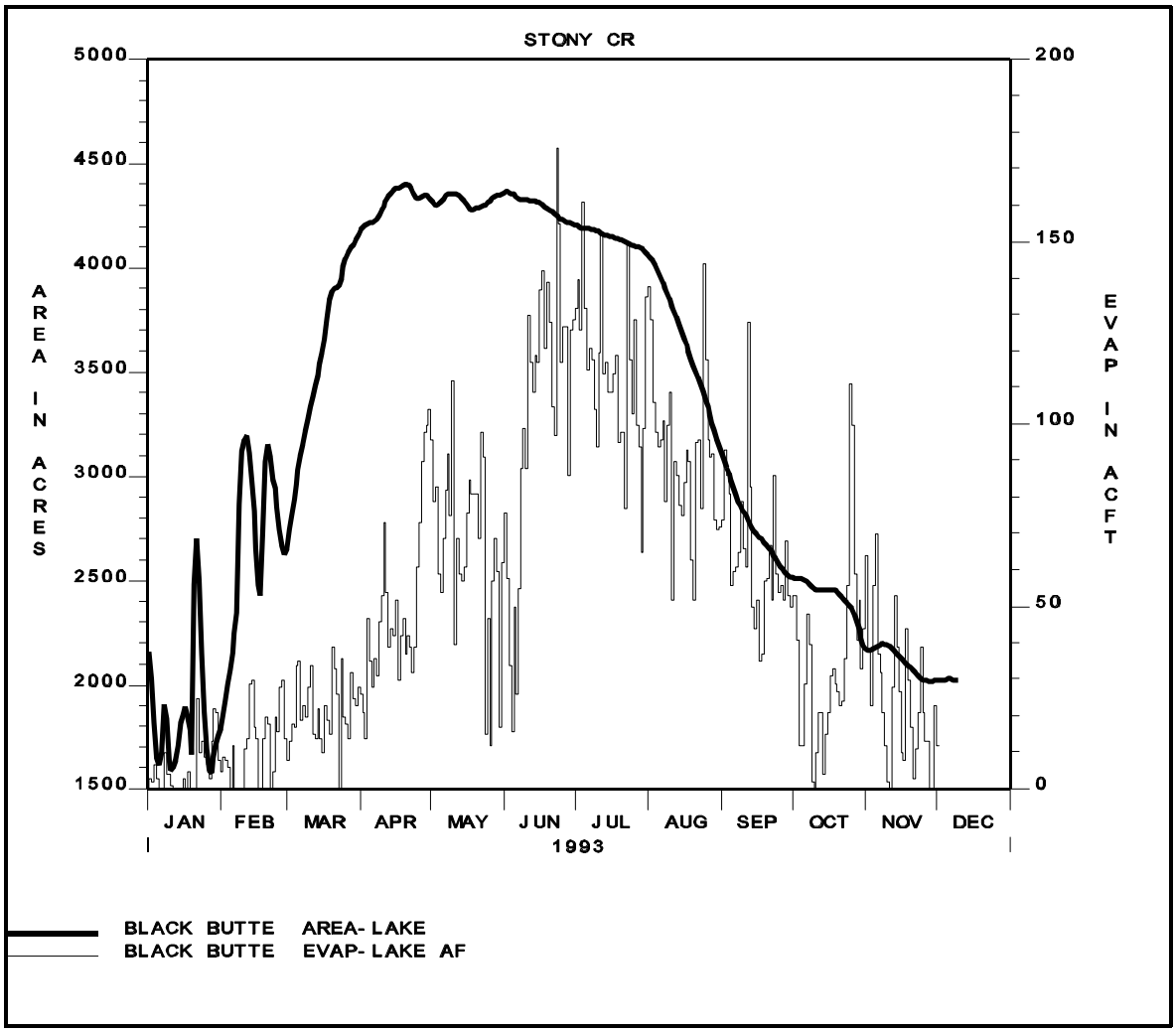
Execution:

```
 ALL VARIABLES HAVE BEEN CLEARED
  -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.    1:  /STONY CR/BLACK BUTTE/EVAP-PAN/01JAN1993/1DAY/OBS/
-----DSS--- ZREAD Unit  71; Vers.    3:  /ALL CORPS RES//EVAP COEFS/01JAN1990/1MON/AVG MONTHLY/
```

```
Variables ----   MCOEF$        EP$        COEF$
Units     ----               INCHES        UNK
Time   Date
0002 01DEC1992        .077       -          -
0002 01JAN1993        .067       -          -
2400 01JAN1993       -          .000       .067
2400 02JAN1993       -          .020       .067
2400 03JAN1993       -          .020       .067
2400 04JAN1993       -          .060       .067
2400 05JAN1993       -          .030       .067
2400 06JAN1993       -          .000       .067
2400 07JAN1993       -          .000       .067
2400 08JAN1993       -          .080       .067
2400 09JAN1993       -          .040       .067
2400 10JAN1993       -          .040       .067
2400 11JAN1993       -          .010       .067
2400 12JAN1993       -          .000       .067
2400 13JAN1993       -          .000       .067
2400 14JAN1993       -          .000       .067
2400 15JAN1993       -          .000       .067
2400 16JAN1993       -          .020       .067
2400 17JAN1993       -          .000       .067
2400 18JAN1993       -          .040       .067
2400 19JAN1993       -          .000       .067
2400 20JAN1993       -          .000       .067
2400 21JAN1993       -          .000       .067
2400 22JAN1993       -          .150       .067
2400 23JAN1993       -          .070       .067
2400 24JAN1993       -          .100       .067
2400 25JAN1993       -          .080       .067
2400 26JAN1993       -          .070       .067
2400 27JAN1993       -          .030       .067
2400 28JAN1993       -          .120       .067
2400 29JAN1993       -          .190       .067
2400 30JAN1993       -          .180       .067
2400 31JAN1993       -          .070       .067
0002 01FEB1993       .058       -          -
2400 01FEB1993       -          .050       .058
2400 02FEB1993       -          .080       .058
2400 03FEB1993       -          .070       .058
2400 04FEB1993       -          .050       .058
2400 05FEB1993       -          .000       .058
2400 06FEB1993       -          .090       .058
2400 07FEB1993       -          .000       .058
2400 08FEB1993       -          .000       .058
2400 09FEB1993       -          .000       .058
2400 10FEB1993       -          .000       .058
2400 11FEB1993       -          .060       .058
2400 12FEB1993       -          .080       .058
2400 13FEB1993       -          .170       .058
2400 14FEB1993       -          .180       .058
2400 15FEB1993       -          .110       .058
2400 16FEB1993       -          .100       .058
2400 17FEB1993       -          .000       .058
2400 18FEB1993       -          .000       .058
      -                         -          -
      .                         .          .
```

```
 MCOEF$ CLEARED
-----DSS--- ZREAD Unit  71; Vers.  400:  /STONY CR/BLACK BUTTE/ELEV/01JAN1993/1DAY//
-----DSS--- ZREAD Unit  71; Vers.    4:  /STONY CR/BLACK BUTTE/ELEV-AREA///POLY/
VARIABLE=POLY$
 UNITS = FEET-MSL   ACRES
 TYPES = UNT        UNT
 LABEL =
 SHIFT =      0.0   OFFSET =      0.0   DATUM =  375.0
   X-AXIS
     -1.09760          1.0000
      .49540          2.0000
     -.02000          3.0000
      .00080          4.0000
     -.11936E-04       5.0000
      .77074E-07       6.0000
     -.18529E-09       7.0000
 POLY$ CLEARED
WARNING -- EXISTING VARIABLE RE-DEFINED
-----DSS---ZWRITE Unit  71; Vers.  417:  /STONY CR/BLACK BUTTE/AREA-LAKE/01JAN1993/1DAY//
 DLA$ CLEARED
-----DSS--- ZREAD Unit  71; Vers.  417:  /STONY CR/BLACK BUTTE/AREA-LAKE/01JAN1993/1DAY//
WARNING -- EXISTING VARIABLE RE-DEFINED
-----DSS---ZWRITE Unit  71; Vers.   10:  /STONY CR/BLACK BUTTE/EVAP-LAKE AF/01JAN1993/1DAY//
```


ALL CORPS RES
0.10
0.05
EVAP COEFS
JUL
JAN
JUL
JAN
JUL
JAN
JUL
JAN
JUL
JAN
1989
1990
1991
1992
1993
AVG MONTHLY EVAP COEFS


STONY CR
5000
4500
4000
3500
3000
2500
2000
1500
AREA IN ACRES
200
150
100
50
0
EVAP IN ACFT
JAN
FEB
MAR
APR
MAY
JUN
JUL
AUG
SEP
OCT
NOV
DEC
1993
BLACK BUTTE AREA- LAKE
BLACK BUTTE EVAP- LAKE AF

Name: *POLY2* *Polynomial transformation with integral*
Use: *CO TY=POLY2(TX,TP)*

Compute a polynomial transformation of TX using the integral of the polynomial defined by coefficients TP. Store the result in TY. If a TX value is missing TY is undefined. If a concurrent IF condition is not satisfied, TY is unchanged. Units and type are defined by TP. TP can be created with the utility DSSPD.

Example: The following example uses the PERCON and POLY2 functions to determine the storage volume for Black Butte Reservoir. Note the extensive use of PREAD function keys and variable parameters used with macros.

Macro:

```
MACRO TPOLY2
!TEACH j TESTDB.DSS
!TEACH k TESTDB.DSS
!TEACH m .M
TIME 0100 01FEB93 2400 30APR93
!R DLS  ^j ^k "STONY CR" "BLACK BUTTE"
ENDMACRO

MACRO DLS %HPDB %LSDB %APART %BPART
****|**********|******************************|**********************
**  Compute    Daily STOR-RES EOP
**  INPUT:
**    %HPDB   Elevation Database
**    %LSDB   Storage Database
**    %APART  "A" Part of Pathname
**    %BPART  "B" Part of Pathname
*******************************************************************************
GE HP$=%HPDB:/%APART/%BPART/ELEV//1HOUR//
GE POLY=TESTDB.DSS:/%APART/%BPART/ELEV-AREA///POLY/
CO X$=GENTSR(1D,0D,-901,M)
CO HP$$=PERCON(HP$,X$)
CO LS$=POLY2(HP$$,POLY)
** Round storage to units
CO LSR$=RND(LS$,8,0)
SD LSR$ UNITS=ACFT  TYPE=INST-VAL
PU^m LSR$=%LSDB:/%APART/%BPART/STOR-RES EOP//1DAY/CAL/
$CONTINUE
ENDMACRO
```

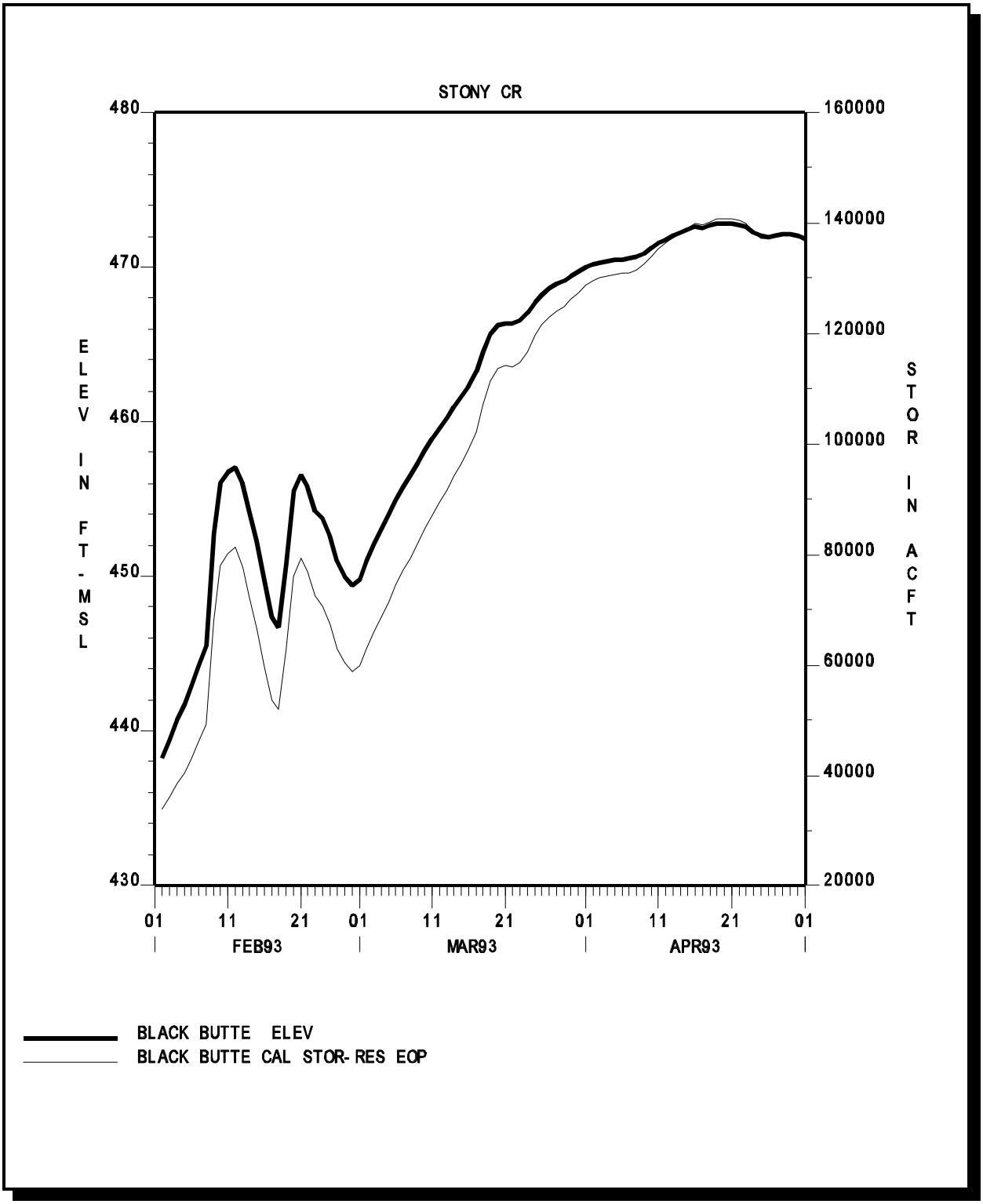
Execution:

```
   -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                      Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.  693:  /STONY CR/BLACK BUTTE/ELEV/01FEB1993/1HOUR//
-----DSS--- ZREAD Unit  71; Vers.  783:  /STONY CR/BLACK BUTTE/ELEV/01MAR1993/1HOUR//
-----DSS--- ZREAD Unit  71; Vers.  756:  /STONY CR/BLACK BUTTE/ELEV/01APR1993/1HOUR//
```

```
-----DSS--- ZREAD Unit  71; Vers.    6:  /STONY CR/BLACK BUTTE/ELEV-AREA///POLY/

*** WARNING : QUALITY FLAG RESET TO NONE
DATUM=     375.000000
-----DSS---ZWRITE Unit  71; Vers.    5:  /STONY CR/BLACK BUTTE/STOR-RES EOP/01JAN1993/1DAY/CAL/
```

Variables ----	X$	HP$	HP$$	LS$	LSR$
Units ----		FT-MSL	UNK	ACRES	ACFT
Time Date					
0100 01FEB1993	-	437.590	-	-	-
0200 01FEB1993	-	437.610	-	-	-
0300 01FEB1993	-	437.620	-	-	-
0400 01FEB1993	-	437.640	-	-	-
0500 01FEB1993	-	437.660	-	-	-
0600 01FEB1993	-	437.680	-	-	-
0700 01FEB1993	-	437.690	-	-	-
0800 01FEB1993	-	437.730	-	-	-
0900 01FEB1993	-	437.760	-	-	-
1000 01FEB1993	-	437.790	-	-	-
1100 01FEB1993	-	437.830	-	-	-
1200 01FEB1993	-	437.860	-	-	-
1300 01FEB1993	-	437.890	-	-	-
1400 01FEB1993	-	437.930	-	-	-
1500 01FEB1993	-	437.970	-	-	-
1600 01FEB1993	-	438.010	-	-	-
1700 01FEB1993	-	438.050	-	-	-
1800 01FEB1993	-	438.090	-	-	-
1900 01FEB1993	-	438.130	-	-	-
2000 01FEB1993	-	438.160	-	-	-
2100 01FEB1993	-	438.200	-	-	-
2200 01FEB1993	-	438.250	-	-	-
2300 01FEB1993	-	438.280	-	-	-
2400 01FEB1993	M	438.310	438.310	33958.550	33959.000
0100 02FEB1993	-	438.360	-	-	-
0200 02FEB1993	-	438.390	-	-	-
0300 02FEB1993	-	438.420	-	-	-
0400 02FEB1993	-	438.460	-	-	-
0500 02FEB1993	-	438.500	-	-	-
0600 02FEB1993	-	438.540	-	-	-
0700 02FEB1993	-	438.570	-	-	-
0800 02FEB1993	-	438.620	-	-	-
0900 02FEB1993	-	438.660	-	-	-
1000 02FEB1993	-	438.710	-	-	-
1100 02FEB1993	-	438.770	-	-	-
1200 02FEB1993	-	438.820	-	-	-
1300 02FEB1993	-	438.880	-	-	-
1400 02FEB1993	-	438.930	-	-	-
1500 02FEB1993	-	438.970	-	-	-
1600 02FEB1993	-	439.050	-	-	-
1700 02FEB1993	-	439.090	-	-	-
1800 02FEB1993	-	439.150	-	-	-
1900 02FEB1993	-	439.210	-	-	-
2000 02FEB1993	-	439.270	-	-	-
2100 02FEB1993	-	439.330	-	-	-
2200 02FEB1993	-	439.390	-	-	-
2300 02FEB1993	-	439.430	-	-	-
2400 02FEB1993	M	439.470	439.470	36143.720	36144.000
0100 03FEB1993	-	439.530	-	-	-
0200 03FEB1993	-	439.580	-	-	-
0300 03FEB1993	-	439.640	-	-	-
0400 03FEB1993	-	439.690	-	-	-
0500 03FEB1993	-	439.740	-	-	-
0600 03FEB1993	-	439.790	-	-	-
0700 03FEB1993	-	439.840	-	-	-
0800 03FEB1993	-	439.890	-	-	-
0900 03FEB1993	-	439.950	-	-	-
1000 03FEB1993	-	439.990	-	-	-
1100 03FEB1993	-	440.050	-	-	-
1200 03FEB1993	-	440.090	-	-	-


STONY CR
480
470
460
450
440
430
ELEV IN FT-MSL
160000
140000
120000
100000
80000
60000
40000
20000
STOR IN ACFT
01
11
21
01
11
21
01
11
21
01
FEB93
MAR93
APR93
BLACK BUTTE ELEV
BLACK BUTTE CAL STOR-RES EOP

Name: *PULS* *Modified Puls or Working R&D routing function*
Use: *CO TY=PULS(TX,PF,X,NR,STOR1,Q01)*

Route the uniform time series variable TX by the Modified Puls or Working R&D hydrologic routing method and store it in variable TY. The variable PF must have been previously defined in a GET command. It references a storage-discharge paired function relationship for the reach. Storage must be the first variable, discharge the second variable, and each variable must repeat only once. Variable X is the wedge coefficient (Muskingum X) for use in working R&D. Use 0.0 value to route by Modified Puls method. Variable NR is the number of routing reaches. STOR1 and Q01 are initial storage and flow respectively. Use a value of -1 for both STOR1 and Q01 in order to use the first flow value in TX and interpolate a corresponding storage value from PF. The IF compute command has no effect on this function.

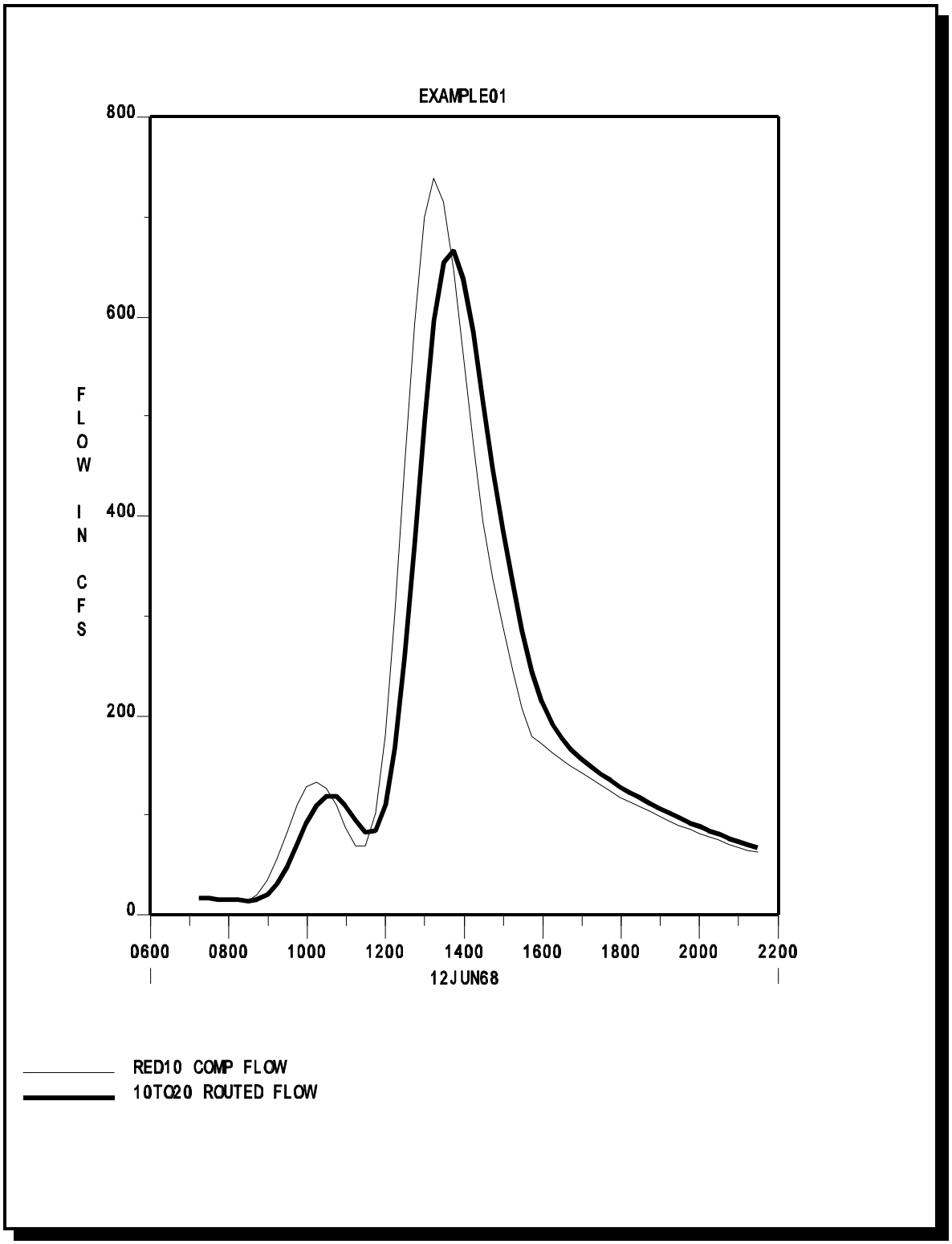
Example: The following example uses the PULS function to perform a PULS routing on a computed hydrograph.

Macro:

```
MACRO TPULS
** COMPUTE A PULS ROUTING
CL ALL
TIME 12JUN1968 0600 12JUN1968 2200
GET PD=HEC101:/EXAMPLE01/10TO20/STOR-FLOW////
GET FLOW=/EXAMPLE01/RED10/FLOW/12JUN1968/15MIN/COMP/
COMP RFLOW=PULS(FLOW,PD,0.0,1,-1,-1)
PUT.A RFLOW=B=TPULS,F=CAL
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
 ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: HEC101.DSS
                      Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.    1:  /EXAMPLE01/10TO20/STOR-FLOW////
-----DSS--- ZREAD Unit  71; Vers.    1:  /EXAMPLE01/RED10/FLOW/01JUN1968/15MIN/COMP/
-----DSS---ZWRITE Unit  71; Vers.    1:  /EXAMPLE01/TPULS/FLOW/01JUN1968/15MIN/CAL/
```


EXAMPLE01
800
600
400
200
0
FLOW IN CFS
0600
0800
1000
1200
1400
1600
1800
2000
2200
12JUN68
RED10 COMP FLOW
10TO20 ROUTED FLOW

Name: *QAC* *Flow accumulator gage processor*
Use: *CO TY=QAC(TX,TC)*

Compute period-average flows from a flow accumulator type gage:

TY(t) = (TX(t)-TX(t-1))/(TC(t)-TC(t-1))

TX and TC are respectively, time series of the accumulated flow and the count. TX and TC values must occur at the same times. If corresponding values for TX and TC decrease from the previous period, then the accumulation is assumed to have reset to zero at the beginning of the interval. An IF condition has no effect. TY is assigned the type 'PER-AVER.'

Name: *RND* *Round off*
Use: *CO TY=RND(TX,NDIG,IPLAC)*

Round off values to NDIG or IPLAC, whichever controls. NDIG is the number of digits to round to and can range from 1 to 8. IPLAC is a magnitude of 10 to which to round to: for example, -1 specifies rounding to one-tenth (0.1). The number of digits shown is never less than 1, however. The following example illustrates the effects on rounding if NDIG=3 and IPLACE=0 (ie., round to ones place):

	Rounds	to
1445.1	"	1450.
144.51	"	145.
14.451	"	14.
1.4451	"	1.0

If TX is undefined, TY is undefined. If a concurrent IF condition is unsatisfied, TY is unchanged.

Example: The following example uses the QAC function to take cumulative flow readings (15 minute) and their count and convert them into period average flow values. The RND function is used to round the flow values.
Macro:

```
MACRO TQAC1
TIME 01JAN93 0100 30JAN93 2400
!R HQR2 TESTDB.DSS TESTDB.DSS "STONY CR" "BLACK BUTTE" FLOW
ENDMACRO
MACRO HQR2 %INDB %QRDB %APART %BPART %CPART
****|**********|*****************************|**********************
**  Compute    Hourly FLOW or FLOW-RES OUT    for NO USGS RTable
**            Used when no USGS rating table is available
**  INPUT:
**    QV$      FLOW-CUM (Accumulated flow, cfs, HADA-PO).  Be sure to
retrieve
**            the previous value outside the time window.
**    NC$      N COUNT (N count, #, HADA-PO)
**  OUTPUT:
**    QR$      FLOW (OUTFLOW, STREAM FLOW, cfs) rounded to tenth if < 10
cfs
**            interpolated to 1 hour values
*******************************************************************************
*
GE.P QV$=%INDB:/%APART/%BPART/FLOW-CUM//1HOUR//
GE.P NC$=%INDB:/%APART/%BPART/N COUNT//1HOUR//
CO QR$=QAC(QV$,NC$)
CO QRR$=QR$
!RUN RNDQR  QRR$ -1
SD QR$ UNITS=CFS TYPE=PER-AVER
PU.M QR$=%QRDB:/%APART/%BPART/%CPART//1HOUR//
$CONTINUE
ENDMACRO
```

```
MACRO RNDQR %QR %IPLAC
****|**********|*****************************|**********************
**  Round      FLOW - HADA Gages              IF FLOW < 0 THEN set to 0
**  Round Flows -- HADA gaged stations
**
**  INPUT/OUTPUT:
**        %QR      Time series of flows to be rounded
**        %IPLAC   Magnitude of 10 to round to ( -1 rounds to one-tenth)
*******************************************************************************
***
** Round flow to nearest %IPLAC cfs for all flows
CO %QR=RND(%QR,8,%IPLAC)
** Round flow to nearest one cfs for flows > 10
CO IF(%QR GE 10) %QR=RND(%QR,8,0)
** account for negative flows caused by RTABLE doing negative interpolation
CO IF(%QR LT 0) %QR=0
ENDMACRO
```

Execution:

```
  -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                     Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.  749:  /STONY CR/BLACK BUTTE/FLOW-CUM/01JAN1993/1HOUR//
-----DSS--- ZREAD Unit  71; Vers.  749:  /STONY CR/BLACK BUTTE/N COUNT/01JAN1993/1HOUR//
WARNING -- EXISTING VARIABLE RE-DEFINED
WARNING -- EXISTING VARIABLE RE-DEFINED
-----DSS---ZWRITE Unit  71; Vers.    2:  /STONY CR/BLACK BUTTE/FLOW/01JAN1993/1HOUR//
Variables ----     QV$          NC$          QR$         QRR$
Units     ----     CFS                       CFS         CFS
Time   Date
0100 01JAN1993     2116.000       4.000       M           M
0200 01JAN1993     4232.000       8.000     529.000     529.000
0300 01JAN1993     6371.000      12.000     534.750     535.000
0400 01JAN1993     8510.000      16.000     534.750     535.000
0500 01JAN1993    10650.000      20.000     535.000     535.000
0600 01JAN1993    12789.000      24.000     534.750     535.000
0700 01JAN1993    14928.000      28.000     534.750     535.000
0800 01JAN1993    17067.000      32.000     534.750     535.000
0900 01JAN1993    19206.000      36.000     534.750     535.000
1000 01JAN1993    21346.000      40.000     535.000     535.000
1100 01JAN1993    23485.000      44.000     534.750     535.000
1200 01JAN1993    27120.000      48.000     908.750     909.000
1300 01JAN1993    32592.000      52.000    1368.000    1368.000
1400 01JAN1993    40215.000      56.000    1905.750    1906.000
1500 01JAN1993    49915.000      60.000    2425.000    2425.000
1600 01JAN1993    59930.000      64.000    2503.750    2504.000
1700 01JAN1993    69945.000      68.000    2503.750    2504.000
1800 01JAN1993    79960.000      72.000    2503.750    2504.000
1900 01JAN1993    89976.000      76.000    2504.000    2504.000
2000 01JAN1993    99991.000      80.000    2503.750    2504.000
2100 01JAN1993   110056.000      84.000    2516.250    2516.000
2200 01JAN1993   120104.000      88.000    2512.000    2512.000
2300 01JAN1993   130119.000      92.000    2503.750    2504.000
2400 01JAN1993   140135.000      96.000    2504.000    2504.000
0100 02JAN1993    10015.000       4.000    2503.750    2504.000
0200 02JAN1993    20030.000       8.000    2503.750    2504.000
0300 02JAN1993    30046.000      12.000    2504.000    2504.000
0400 02JAN1993    40061.000      16.000    2503.750    2504.000
```

0500	02JAN1993	50076.000	20.000	2503.750	2504.000
0600	02JAN1993	60091.000	24.000	2503.750	2504.000
0700	02JAN1993	70106.000	28.000	2503.750	2504.000
0800	02JAN1993	80122.000	32.000	2504.000	2504.000
0900	02JAN1993	89589.000	36.000	2366.750	2367.000
1000	02JAN1993	98741.000	40.000	2288.000	2288.000
1100	02JAN1993	110323.000	44.000	2895.500	2896.000
1200	02JAN1993	124563.000	48.000	3560.000	3560.000
1300	02JAN1993	140703.000	52.000	4035.000	4035.000
1400	02JAN1993	158913.000	56.000	4552.500	4553.000
1500	02JAN1993	181581.000	60.000	5667.000	5667.000
1600	02JAN1993	202367.000	64.000	5196.500	5197.000
1700	02JAN1993	222702.000	68.000	5083.750	5084.000
1800	02JAN1993	242813.000	72.000	5027.750	5028.000
1900	02JAN1993	262924.000	76.000	5027.750	5028.000
2000	02JAN1993	283035.000	80.000	5027.750	5028.000
2100	02JAN1993	303147.000	84.000	5028.000	5028.000
2200	02JAN1993	323258.000	88.000	5027.750	5028.000
2300	02JAN1993	343258.000	92.000	5000.000	5000.000
2400	02JAN1993	363258.000	96.000	5000.000	5000.000
0100	03JAN1993	20000.000	4.000	5000.000	5000.000
0200	03JAN1993	39884.000	8.000	4971.000	4971.000
0300	03JAN1993	59769.000	12.000	4971.250	4971.000
0400	03JAN1993	79653.000	16.000	4971.000	4971.000
0500	03JAN1993	99538.000	20.000	4971.250	4971.000
0600	03JAN1993	119422.000	24.000	4971.000	4971.000

Name: *RTABLE* *Rating table interpolation*
Use: *CO TY=RTABLE(TX,TB)*

Interpolate values for TX using table TB and store the result in TY. TB must be created using the program DSSPD (use /R option) with specific information in the header: type of interpolation, offset, shift, and datum. If the type of interpolation is LOGLOG, table x values are adjusted by subtracting the offset. The shift is added to and the datum subtracted from all incoming TX values. The header information in TB also defines the units of TY.

Name: *RTABLR* *Reverse rating table interpolation*
Use: *CO TY=RTABLR(TX,TB)*

Interpolate values for TX using the reverse of table TB and store the result in TY. TB must be created using the program DSSPD with specific information in the header: type of interpolation, offset, shift, and datum. If the type of interpolation is LOGLOG, table x values are adjusted by subtracting the offset. The shift is subtracted from and the datum added to all resulting TX values. The header information in TB also defines the units of TY.

Example: The following example uses the RTABLE and RTABLR function to compute flow values at Black Butte reservoir.

Macro:

```
MACRO TRTABL1
TIME 01JAN93 0100 30JAN93 2400
!R HQR1 TESTDB.DSS TESTDB.DSS "STONY CR" "BLACK BUTTE"
FLOW
ENDMACRO

MACRO RNDQR %QR %IPLAC
****|**********|****************************|**********************
**  Round      FLOW - HADA Gages              IF FLOW < 0 THEN set
to 0
**  Round Flows -- HADA gaged stations
**
**  INPUT/OUTPUT:
**        %QR      Time series of flows to be rounded
**        %IPLAC   Magnitude of 10 to round to ( -1 rounds to
one-tenth)
**********************************************************************
*********
** Round flow to nearest %IPLAC cfs for all flows
CO %QR=RND(%QR,8,%IPLAC)
** Round flow to nearest one cfs for flows > 10
CO IF(%QR GE 10) %QR=RND(%QR,8,0)
** account for negative flows caused by RTABLE doing negative
interpolation
CO IF(%QR LT 0) %QR=0
ENDMACRO
```

```
MACRO HQR1 %INDB %QRDB %APART %BPART %CPART
****|**********|****************************|**********************
**  Compute    Hourly FLOW or FLOW-RES OUT    w/ HADA & USGS
RTable
**  Compute FLOW (from accumulated flow and N count ,ESI)
Type 1 Short Period
**                used when a HADA PO and USGS rating table exists
**  INPUT:
**    %INDB    Database for ACCUMULATED FLOW and N-COUNT
**    %QRDB    Flow Database
**    %APART   "A" part of pathname
**    %BPART   "B" part of pathname
**    %CPART   Flow type being computed -- "C" part -- (FLOW-RES
OUT or FLOW)
**  OUTPUT:
**    QR$      FLOW (OUTFLOW, STREAM FLOW, cfs) rounded to
tenth if < 10 cfs
**
**********************************************************************
*******
GE.P QV$=%INDB:/%APART/%BPART/FLOW-CUM//1HOUR//
GE.P NC$=%INDB:/%APART/%BPART/N COUNT//1HOUR//
GE HADA$=TESTDB.DSS:/%APART/%BPART/STAGE-FLOW///HADA
PO/
GE USGS$=TESTDB.DSS:/%APART/%BPART/STAGE-FLOW///USGS/
CO TQR$=QAC(QV$,NC$)
CO AHR$=RTABLR(TQR$,HADA$)
CO QR$=RTABLE(AHR$,USGS$)
!RUN RNDQR  QR$ -1
SD QR$ UNITS=CFS TYPE=PER-AVER
PU.M QR$=%QRDB:/%APART/%BPART/%CPART//1HOUR//
$CONTINUE
ENDMACRO
```

Execution:

```
   -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                      Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.  749:  /STONY CR/BLACK BUTTE/FLOW-CUM/01JAN1993/1HOUR//
-----DSS--- ZREAD Unit  71; Vers.  749:  /STONY CR/BLACK BUTTE/N COUNT/01JAN1993/1HOUR//
-----DSS--- ZREAD Unit  71; Vers.    8:  /STONY CR/BLACK BUTTE/STAGE-FLOW///HADA PO/
-----DSS--- ZREAD Unit  71; Vers.   10:  /STONY CR/BLACK BUTTE/STAGE-FLOW///USGS/
LINLIN TRANSFORMATION
OFFSET =       .00     SHIFT =       .00     DATUM =       .00
LOGLOG TRANSFORMATION
OFFSET =       .90     SHIFT =       .00     DATUM =       .00
WARNING -- EXISTING VARIABLE RE-DEFINED
WARNING -- EXISTING VARIABLE RE-DEFINED
-----DSS---ZWRITE Unit  71; Vers.    7:  /STONY CR/BLACK BUTTE/FLOW/01JAN1993/1HOUR//
```

Variables ----	AHR$	TQR$	QR$
Units ----	FEET	CFS	CFS
Time Date			
0100 01JAN1993	M	M	M
0200 01JAN1993	4.550	529.000	525.000
0300 01JAN1993	4.560	534.750	530.000
0400 01JAN1993	4.560	534.750	530.000
0500 01JAN1993	4.560	535.000	530.000
0600 01JAN1993	4.560	534.750	530.000
0700 01JAN1993	4.560	534.750	530.000
0800 01JAN1993	4.560	534.750	530.000
0900 01JAN1993	4.560	534.750	530.000
1000 01JAN1993	4.560	535.000	530.000
1100 01JAN1993	4.560	534.750	530.000
1200 01JAN1993	5.145	908.750	896.000
1300 01JAN1993	5.642	1368.000	1351.000
1400 01JAN1993	6.070	1905.750	1892.000
1500 01JAN1993	6.383	2425.000	2403.000
1600 01JAN1993	6.430	2503.750	2489.000
1700 01JAN1993	6.430	2503.750	2489.000
1800 01JAN1993	6.430	2503.750	2489.000
1900 01JAN1993	6.430	2504.000	2489.000
2000 01JAN1993	6.430	2503.750	2489.000
2100 01JAN1993	6.438	2516.250	2503.000
2200 01JAN1993	6.435	2512.000	2498.000
2300 01JAN1993	6.430	2503.750	2489.000
2400 01JAN1993	6.430	2504.000	2489.000
0100 02JAN1993	6.430	2503.750	2489.000
0200 02JAN1993	6.430	2503.750	2489.000
0300 02JAN1993	6.430	2504.000	2489.000
0400 02JAN1993	6.430	2503.750	2489.000
0500 02JAN1993	6.430	2503.750	2489.000
0600 02JAN1993	6.430	2503.750	2489.000
0700 02JAN1993	6.430	2503.750	2489.000
0800 02JAN1993	6.430	2504.000	2489.000
0900 02JAN1993	6.347	2366.750	2341.000
1000 02JAN1993	6.300	2288.000	2259.000
1100 02JAN1993	6.628	2895.500	2869.000
1200 02JAN1993	6.935	3560.000	3544.000
1300 02JAN1993	7.129	4035.000	4008.000
1400 02JAN1993	7.328	4552.500	4520.000
1500 02JAN1993	7.691	5667.000	5548.000
1600 02JAN1993	7.556	5196.500	5157.000
1700 02JAN1993	7.524	5083.750	5067.000
1800 02JAN1993	7.508	5027.750	5022.000
1900 02JAN1993	7.508	5027.750	5022.000
2000 02JAN1993	7.508	5027.750	5022.000
2100 02JAN1993	7.508	5028.000	5022.000
2200 02JAN1993	7.508	5027.750	5022.000
2300 02JAN1993	7.500	5000.000	5000.000
2400 02JAN1993	7.500	5000.000	5000.000
0100 03JAN1993	7.500	5000.000	5000.000
0200 03JAN1993	7.489	4971.000	4968.000
0300 03JAN1993	7.489	4971.250	4968.000
0400 03JAN1993	7.489	4971.000	4968.000
0500 03JAN1993	7.489	4971.250	4968.000
0600 03JAN1993	7.489	4971.000	4968.000
0700 03JAN1993	7.478	4942.250	4936.000
0800 03JAN1993	7.478	4942.250	4936.000
0900 03JAN1993	7.287	4445.250	4410.000
1000 03JAN1993	6.653	2951.250	2922.000
1100 03JAN1993	6.653	2951.500	2922.000

Paired Data Variables

```
VARIABLE=HADA$
 UNITS = FEET       CFS
 TYPES = UNT        UNT
 LABEL = FLOW
 SHIFT =      0.0   OFFSET =      0.0   DATUM =
   0.0

  X-AXIS        FLOW
      .0000       .0000
      .9000       .0000
     1.0000       .1000
     1.1000       .3000
     1.2000       .7000
     1.3000      1.3000
     1.4000      2.3000
     1.5000      3.5000
     1.6000      4.9000
     1.7000      7.0000
     1.8000      9.8000
     2.0000     17.7000
     2.2000     28.3000
     2.4000     41.8000
     2.6000     57.3000
     2.7000     65.8000
     2.8000     76.0000
     3.0000    101.0000
     3.2500    139.9000
     3.5000    184.0000
     4.0000    314.0000
     4.5000    500.0000
     5.0000    790.0000
     5.5000   1200.0000
     6.0000   1790.0000
     6.5000   2620.0000
     7.0000   3700.0000
     7.5000   5000.0000
     9.0000  10250.0000
    11.5000  24000.0000
```

```
VARIABLE=USGS$
 UNITS = FEET       CFS
 TYPES = UNT        UNT
 LABEL = FLOW
 SHIFT =      0.0   OFFSET =      0.9   DATUM =
   0.0

  X-AXIS        FLOW
      .0000       .0000
      .9000       .0000
     1.0000       .1000
     1.1000       .3000
     1.2000       .7000
     1.3000      1.3000
     1.4000      2.3000
     1.5000      3.5000
     1.6000      4.9000
     1.7000      7.0000
     1.8000      9.8000
     2.0000     17.7000
     2.2000     28.3000
     2.4000     41.8000
     2.6000     57.3000
     2.7000     65.8000
     2.8000     76.0000
     3.0000    101.0000
     3.2500    139.9000
     3.5000    184.0000
     4.0000    314.0000
     4.5000    500.0000
     5.0000    790.0000
     5.5000   1200.0000
     6.0000   1790.0000
     6.5000   2620.0000
     7.0000   3700.0000
     7.5000   5000.0000
     8.0000   6530.0000
     9.0000  10250.0000
     9.5000  12500.0000
    11.5000  24000.0000
```

Name: *RTABL2* *Two-variable rating table interpolation*
Use: *CO TY=RTABL2(TX,TZ,TB)*

TY is a function of two independent variables TX and TZ. The functional relationship is specified by the table TB, which consists of sets of TX/TY pairs, each set corresponding to a TZ value. TB is created with the program DSSPD with values for TZ specified as labels for the sets of TX/TY pairs. RTABL2 interpolates linearly in table TB. No extrapolation is done: if the TX or TY value are outside the range bounded by TB, TY is set to missing. TX and TZ must be concurrent time series. The header information in TB also defines the units of TY.

Example: The following example uses the RTABL2 function to calculate the outflows from Lynn Lake from stage readings at a gage (LYNN) downstream from the outlet works. The gage rating is effected by backwater on the Cheat River. A family of rating curves has been established for LYNN gage that relates each rating curve to a stage reading at the PMAW gage.

Macro:

```
MACRO TRTABL2
CL ALL
TIME 28NOV1993 0100 02DEC1993 2400
GET LABL1=TESTDB.DSS:/MONONGAHELA/PMRP1/STAGE-L/01JUN1989/1DAY/OBS/
CO LABL2=ESTLIN(LABL1,24,M)
CO LABLE=TS1(LABL2,1H,0H)
TAB.F LABL1 LABL2 LABLE
PUT LABLE=TESTDB.DSS:/MONONGAHELA/PMRP/STAGE/01JUN1989/1HOUR/CMPTD/
TIME 01DEC1993 0100 02DEC1993 2400
GET TABLE=TESTDB.DSS:/MONONGAHELA/PMAW/STAGE-FLOW////
GET LABLE=TESTDB.DSS:/MONONGAHELA/PMRP/STAGE/01JUN1989/1HOUR/CMPTD/
GET STAGE=TESTDB.DSS:/MONONGAHELA/PMAW/STAGE/01JUN1989/1HOUR/OBS/
COMP FLOW=RTABL2(STAGE,LABLE,TABLE)
SD FLOW U=CFS TY=INST-VAL
PUT FLOW=B=LNNW,C=FLOW,F=CAL
ENDMACRO
```

Execution:

```
ALL VARIABLES HAVE BEEN CLEARED
  -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.    1:  /MONONGAHELA/PMRP1/STAGE-L/01JAN1993/1DAY/OBS/
-----DSS---ZWRITE Unit  71; Vers.    2:  /MONONGAHELA/PMRP/STAGE/01NOV1993/1HOUR/CMPTD/
-----DSS---ZWRITE Unit  71; Vers.    3:  /MONONGAHELA/PMRP/STAGE/01DEC1993/1HOUR/CMPTD/
WARNING - TIME SERIES VARIABLES WERE CLEARED
BECAUSE START DAY OF TIME WINDOW HAS CHANGED.
NOTE:  CLEAR CAN BE SUPPRESSED VIA .N OPTION.
-----DSS--- ZREAD Unit  71; Vers.    4:  /MONONGAHELA/PMAW/STAGE-FLOW////
-----DSS--- ZREAD Unit  71; Vers.    3:  /MONONGAHELA/PMRP/STAGE/01DEC1993/1HOUR/CMPTD/
```

-----DSS--- ZREAD Unit 71; Vers. 200: /MONONGAHELA/PMAW/STAGE/01DEC1993/1HOUR/OBS/
-----DSS---ZWRITE Unit 71; Vers. 2: /MONONGAHELA/LNNW/FLOW/01DEC1993/1HOUR/CAL/

Variables ----	LABL1	LABL2	LABLE
Units ----	FEET	FEET	FEET
Time Date			
0100 28NOV1993	-	-	M
0200 28NOV1993	-	-	M
0300 28NOV1993	-	-	M
0400 28NOV1993	-	-	M
0500 28NOV1993	-	-	M
0600 28NOV1993	-	-	M
0700 28NOV1993	13.700	13.700	13.700
0800 28NOV1993	-	-	13.675
0900 28NOV1993	-	-	13.650
1000 28NOV1993	-	-	13.625
1100 28NOV1993	-	-	13.600
1200 28NOV1993	-	-	13.575
1300 28NOV1993	-	-	13.550
1400 28NOV1993	-	-	13.525
1500 28NOV1993	-	-	13.500
1600 28NOV1993	-	-	13.475
1700 28NOV1993	-	-	13.450
1800 28NOV1993	-	-	13.425
1900 28NOV1993	-	-	13.400
2000 28NOV1993	-	-	13.375
2100 28NOV1993	-	-	13.350
2200 28NOV1993	-	-	13.325
2300 28NOV1993	-	-	13.300
2400 28NOV1993	-	-	13.275
0100 29NOV1993	-	-	13.250
0200 29NOV1993	-	-	13.225
0300 29NOV1993	-	-	13.200
0400 29NOV1993	-	-	13.175
0500 29NOV1993	-	-	13.150
0600 29NOV1993	-	-	13.125
0700 29NOV1993	13.100	13.100	13.100
0800 29NOV1993	-	-	13.104
0900 29NOV1993	-	-	13.108
1000 29NOV1993	-	-	13.113
1100 29NOV1993	-	-	13.117
1200 29NOV1993	-	-	13.121
1300 29NOV1993	-	-	13.125
1400 29NOV1993	-	-	13.129
1500 29NOV1993	-	-	13.133
1600 29NOV1993	-	-	13.137
1700 29NOV1993	-	-	13.142
1800 29NOV1993	-	-	13.146
1900 29NOV1993	-	-	13.150
2000 29NOV1993	-	-	13.154
2100 29NOV1993	-	-	13.158
2200 29NOV1993	-	-	13.163
2300 29NOV1993	-	-	13.167
2400 29NOV1993	-	-	13.171
0100 30NOV1993	-	-	13.175
0200 30NOV1993	-	-	13.179
0300 30NOV1993	-	-	13.183
0400 30NOV1993	-	-	13.188
0500 30NOV1993	-	-	13.192
0600 30NOV1993	-	-	13.196
0700 30NOV1993	13.200	13.200	13.200
0800 30NOV1993	-	-	13.167
0900 30NOV1993	-	-	13.133
1000 30NOV1993	-	-	13.100
1100 30NOV1993	-	-	13.067
1200 30NOV1993	-	-	13.033
1300 30NOV1993	-	-	13.000
1400 30NOV1993	-	-	12.967
1500 30NOV1993	-	-	12.933
1600 30NOV1993	-	-	12.900
1700 30NOV1993	-	-	12.867
1800 30NOV1993	-	-	12.833
1900 30NOV1993	-	-	12.800
2000 30NOV1993	-	-	12.767
2100 30NOV1993	-	-	12.733
2200 30NOV1993	-	-	12.700
2300 30NOV1993	-	-	12.667
2400 30NOV1993	-	-	12.633
0100 01DEC1993	-	-	12.600
0200 01DEC1993	-	-	12.567
0300 01DEC1993	-	-	12.533
0400 01DEC1993	-	-	12.500
0500 01DEC1993	-	-	12.467
0600 01DEC1993	-	-	12.433
0700 01DEC1993	12.400	12.400	12.400
0800 01DEC1993	-	-	12.383
0900 01DEC1993	-	-	12.367
1000 01DEC1993	-	-	12.350
1100 01DEC1993	-	-	12.333
1200 01DEC1993	-	-	12.317
1300 01DEC1993	-	-	12.300
1400 01DEC1993	-	-	12.283
1500 01DEC1993	-	-	12.267
1600 01DEC1993	-	-	12.250
1700 01DEC1993	-	-	12.233
1800 01DEC1993	-	-	12.217
1900 01DEC1993	-	-	12.200
2000 01DEC1993	-	-	12.183
2100 01DEC1993	-	-	12.167
2200 01DEC1993	-	-	12.150
2300 01DEC1993	-	-	12.133
2400 01DEC1993	-	-	12.117
0100 02DEC1993	-	-	12.100
0200 02DEC1993	-	-	12.083
0300 02DEC1993	-	-	12.067
0400 02DEC1993	-	-	12.050
0500 02DEC1993	-	-	12.033
0600 02DEC1993	-	-	12.017
0700 02DEC1993	12.000	12.000	12.000
0800 02DEC1993	-	-	M
0900 02DEC1993	-	-	M
1000 02DEC1993	-	-	M
1100 02DEC1993	-	-	M
1200 02DEC1993	-	-	M
1300 02DEC1993	-	-	M
1400 02DEC1993	-	-	M
1500 02DEC1993	-	-	M
1600 02DEC1993	-	-	M
1700 02DEC1993	-	-	M
1800 02DEC1993	-	-	M
1900 02DEC1993	-	-	M
2000 02DEC1993	-	-	M
2100 02DEC1993	-	-	M
2200 02DEC1993	-	-	M
2300 02DEC1993	-	-	M
2400 02DEC1993	-	-	M
0100 01DEC1993	12.600	10.700	6804.998
0200 01DEC1993	12.567	10.730	6896.498
0300 01DEC1993	12.533	10.730	6896.498
0400 01DEC1993	12.500	10.730	6896.498
0500 01DEC1993	12.467	10.700	6804.998
0600 01DEC1993	12.433	10.460	6173.333
0700 01DEC1993	12.400	10.520	6313.333
0800 01DEC1993	12.383	10.490	6243.332
0900 01DEC1993	12.367	10.460	6173.333
1000 01DEC1993	12.350	10.460	6173.333
1100 01DEC1993	12.333	10.460	6173.333
1200 01DEC1993	12.317	10.460	6173.333
1300 01DEC1993	12.300	10.460	6173.333
1400 01DEC1993	12.283	10.460	6173.333
1500 01DEC1993	12.267	10.490	6243.332
1600 01DEC1993	12.250	10.460	6173.333
1700 01DEC1993	12.233	10.490	6243.332
1800 01DEC1993	12.217	10.460	6173.333
1900 01DEC1993	12.200	10.460	6173.333
2000 01DEC1993	12.183	10.460	6173.333
2100 01DEC1993	12.167	10.460	6173.333

Paired Data Variable (TABLE)

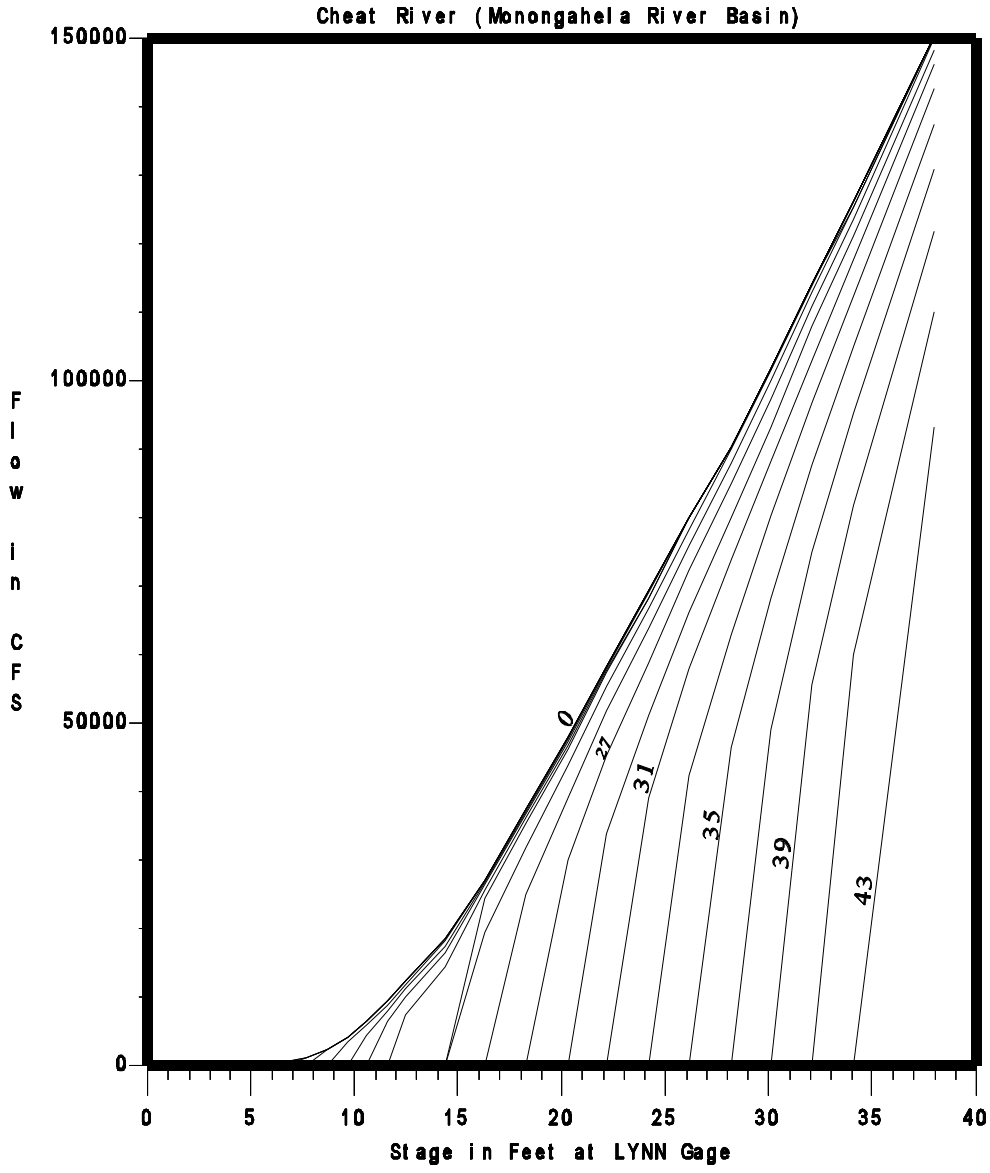
VARIABLE=TABLE
UNITS = FEET CFS
TYPES = UNT UNT
LABEL = 0
LABEL = 14
LABEL = 15
LABEL = 16
LABEL = 17
LABEL = 18
LABEL = 19
LABEL = 20
LABEL = 21
LABEL = 23
LABEL = 25
LABEL = 27
LABEL = 29
LABEL = 31
LABEL = 33
LABEL = 35
LABEL = 37
LABEL = 39
LABEL = 41
LABEL = 43
SHIFT= 0.0 OFFSET=0.0 DATUM= 0.00

X-AXIS	0	14	15	16	17
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6.7000	600.0000	600.0000	0.0000	0.0000	0.0000
7.7000	1400.0000	1400.0000	1350.0000	0.0000	0.0000
8.7000	2600.0000	2600.0000	2550.0000	2400.0000	0.0000
9.7000	4400.0000	4400.0000	4400.0000	4400.0000	3550.0000
10.6000	6500.0000	6500.0000	6500.0000	6500.0000	6050.0000
11.6000	9550.0000	9550.0000	9550.0000	9550.0000	9100.0000
12.5000	12400.0000	12400.0000	12400.0000	12400.0000	12000.0000
14.4000	18800.0000	18800.0000	18800.0000	18800.0000	18500.0000
16.3000	27200.0000	27200.0000	27200.0000	27200.0000	27200.0000
18.3000	37400.0000	37400.0000	37400.0000	37400.0000	37400.0000
20.3000	48000.0000	48000.0000	48000.0000	48000.0000	48000.0000
22.2000	58500.0000	58500.0000	58500.0000	58500.0000	58500.0000
24.2000	69500.0000	69500.0000	69500.0000	69500.0000	69500.0000
26.2000	80000.0000	80000.0000	80000.0000	80000.0000	80000.0000
28.2000	90500.0000	90500.0000	90500.0000	90500.0000	90500.0000
30.1000	102000.0000	102000.0000	102000.0000	102000.0000	102000.0000
32.1000	114400.0000	114400.0000	114400.0000	114400.0000	114400.0000
34.1000	126300.0000	126300.0000	126300.0000	125300.0000	126300.0000
38.0000	150400.0000	150400.0000	150400.0000	150400.0000	150400.0000

X-AXIS	18	19	20	21	23
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6.7000	0.0000	0.0000	0.0000	0.0000	0.0000
7.7000	0.0000	0.0000	0.0000	0.0000	0.0000
8.7000	0.0000	0.0000	0.0000	0.0000	0.0000
9.7000	0.0000	0.0000	0.0000	0.0000	0.0000
10.6000	4700.0000	0.0000	0.0000	0.0000	0.0000
11.6000	8200.0000	6500.0000	0.0000	0.0000	0.0000
12.5000	11400.0000	10000.0000	7600.0000	0.0000	0.0000
14.4000	17500.0000	16500.0000	14500.0000	0.0000	0.0000
16.3000	27000.0000	26500.0000	25700.0000	24500.0000	19500.0000
18.3000	37200.0000	37000.0000	36400.0000	35300.0000	32000.0000
20.3000	47900.0000	47500.0000	47000.0000	46400.0000	44000.0000
22.2000	58500.0000	58000.0000	57900.0000	57500.0000	55500.0000
24.2000	69500.0000	69400.0000	68500.0000	68500.0000	67000.0000
26.2000	80000.0000	80000.0000	80000.0000	80000.0000	78500.0000
28.2000	90500.0000	90500.0000	90500.0000	90500.0000	90000.0000
30.1000	102000.0000	102000.0000	102000.0000	102000.0000	101500.0000
32.1000	114400.0000	114400.0000	114400.0000	114400.0000	114000.0000
34.1000	126300.0000	126300.0000	126300.0000	126300.0000	126300.0000
38.0000	150400.0000	150400.0000	150400.0000	150400.0000	150400.0000

X-AXIS	25	27	29	31	33
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6.7000	0.0000	0.0000	0.0000	0.0000	0.0000
7.7000	0.0000	0.0000	0.0000	0.0000	0.0000
8.7000	0.0000	0.0000	0.0000	0.0000	0.0000
9.7000	0.0000	0.0000	0.0000	0.0000	0.0000
10.6000	0.0000	0.0000	0.0000	0.0000	0.0000
11.6000	0.0000	0.0000	0.0000	0.0000	0.0000
12.5000	0.0000	0.0000	0.0000	0.0000	0.0000
14.4000	0.0000	0.0000	0.0000	0.0000	0.0000
16.3000	0.0000	0.0000	0.0000	0.0000	0.0000
18.3000	25100.0000	0.0000	0.0000	0.0000	0.0000
20.3000	39400.0000	30000.0000	0.0000	0.0000	0.0000
22.2000	52000.0000	45500.0000	34000.0000	0.0000	0.0000
24.2000	64000.0000	59000.0000	51300.0000	39200.0000	0.0000
26.2000	76000.0000	72500.0000	66200.0000	58000.0000	42500.0000
28.2000	88000.0000	85000.0000	80000.0000	73900.0000	63000.0000
30.1000	100000.0000	97500.0000	93500.0000	88400.0000	80300.0000
32.1000	113000.0000	111000.0000	108000.0000	103000.0000	97000.0000
34.1000	125500.0000	123800.0000	121300.0000	117700.0000	112400.0000
38.0000	150400.0000	150000.0000	148400.0000	146300.0000	142800.0000

X-AXIS	35	37	39	41	43
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6.7000	0.0000	0.0000	0.0000	0.0000	0.0000
7.7000	0.0000	0.0000	0.0000	0.0000	0.0000
8.7000	0.0000	0.0000	0.0000	0.0000	0.0000
9.7000	0.0000	0.0000	0.0000	0.0000	0.0000
10.6000	0.0000	0.0000	0.0000	0.0000	0.0000
11.6000	0.0000	0.0000	0.0000	0.0000	0.0000
12.5000	0.0000	0.0000	0.0000	0.0000	0.0000
14.4000	0.0000	0.0000	0.0000	0.0000	0.0000
16.3000	0.0000	0.0000	0.0000	0.0000	0.0000
18.3000	0.0000	0.0000	0.0000	0.0000	0.0000
20.3000	0.0000	0.0000	0.0000	0.0000	0.0000
22.2000	0.0000	0.0000	0.0000	0.0000	0.0000
24.2000	0.0000	0.0000	0.0000	0.0000	0.0000
26.2000	0.0000	0.0000	0.0000	0.0000	0.0000
28.2000	46500.0000	0.0000	0.0000	0.0000	0.0000
30.1000	68400.0000	49300.0000	0.0000	0.0000	0.0000
32.1000	87800.0000	75000.0000	55800.0000	0.0000	0.0000
34.1000	105200.0000	95300.0000	82000.0000	60000.0000	0.0000
38.0000	137500.0000	131000.0000	122000.0000	110000.0000	93400.0000


Cheat River (Monongahela River Basin)
150000
100000
50000
0
Flow in CFS
0
27
31
35
39
43
0
5
10
15
20
25
30
35
40
Stage in Feet at LYNN Gage

Variables ----	LABLE	STAGE	FLOW
Units ----	FEET	FEET	CFS
Time Date			
0100 01DEC1993	12.600	10.700	6804.998
0200 01DEC1993	12.567	10.730	6896.498
0300 01DEC1993	12.533	10.730	6896.498
0400 01DEC1993	12.500	10.730	6896.498
0500 01DEC1993	12.467	10.700	6804.998
0600 01DEC1993	12.433	10.460	6173.333
0700 01DEC1993	12.400	10.520	6313.333
0800 01DEC1993	12.383	10.490	6243.332
0900 01DEC1993	12.367	10.460	6173.333
1000 01DEC1993	12.350	10.460	6173.333
1100 01DEC1993	12.333	10.460	6173.333
1200 01DEC1993	12.317	10.460	6173.333
1300 01DEC1993	12.300	10.460	6173.333
1400 01DEC1993	12.283	10.460	6173.333
1500 01DEC1993	12.267	10.490	6243.332
1600 01DEC1993	12.250	10.460	6173.333
1700 01DEC1993	12.233	10.490	6243.332
1800 01DEC1993	12.217	10.460	6173.333
1900 01DEC1993	12.200	10.460	6173.333
2000 01DEC1993	12.183	10.460	6173.333
2100 01DEC1993	12.167	10.460	6173.333
2200 01DEC1993	12.150	10.460	6173.333
2300 01DEC1993	12.133	10.460	6173.333
2400 01DEC1993	12.117	10.460	6173.333
0100 02DEC1993	12.100	9.320	3716.000
0200 02DEC1993	12.083	9.290	3662.000
0300 02DEC1993	12.067	9.290	3662.000
0400 02DEC1993	12.050	9.260	3608.001
0500 02DEC1993	12.033	9.260	3608.001
0600 02DEC1993	12.017	9.260	3608.001
0700 02DEC1993	12.000	9.260	3608.001
0800 02DEC1993	11.967	10.400	6033.332
0900 02DEC1993	11.933	10.400	6033.332
1000 02DEC1993	11.900	11.090	7994.500
1100 02DEC1993	11.867	10.280	5753.333
1200 02DEC1993	11.833	10.340	5893.333
1300 02DEC1993	11.800	10.190	5543.332
1400 02DEC1993	11.767	10.610	6530.498
1500 02DEC1993	11.733	10.430	6103.333
1600 02DEC1993	11.700	10.730	6896.498
1700 02DEC1993	11.667	10.400	6033.332
1800 02DEC1993	11.633	10.700	6804.998
1900 02DEC1993	11.600	10.460	6173.333
2000 02DEC1993	11.567	11.000	7719.999
2100 02DEC1993	11.533	9.920	4913.333
2200 02DEC1993	11.500	9.650	4310.000
2300 02DEC1993	11.467	6.800	680.000
2400 02DEC1993	11.433	6.320	465.882

Name: *SCRN1* *Screen for possible erroneous values based on maximum/minimum range*

Use: *CO TY=SCRN1(TX,XMIN,XMAX,MAXDEL,QFLAG)*

Flag any value in TX that falls outside the range XMIN - XMAX or exceeds the maximum change MAXDEL from the previous value. The maximum change comparison is done only when the consecutive values are not flagged. Possible values for QFLAG are: M = missing data or Q = questionable. The result is placed in TY.

Name: *SCRN2* *Screen for possible erroneous values based on forward moving average maximum*

Use: *CO TY=SCRN2(TX,NPTS,MAXDEL,QFLAG)*

Flag any value in TX that exceeds the maximum change MAXDEL from the forward moving average of NPTS ending at the previous value. Missing values in TX are not counted in the moving average and the divisor of the average is less one for every missing value. Values which fail the screen are not counted either. At least 2 values must be defined else the moving average is undefined and the screen passes the relevant TX value. Possible flags are: M = missing data or Q = questionable. The result is placed in TY. SCRN2 is useful for detecting and removing spikes. The FMA function may be useful for determining appropriate NPTS and MAXDEL parameters.

Example: The following example uses the SCRN1 and SCRN2 functions to demonstrate their use in flagging possible bad data.

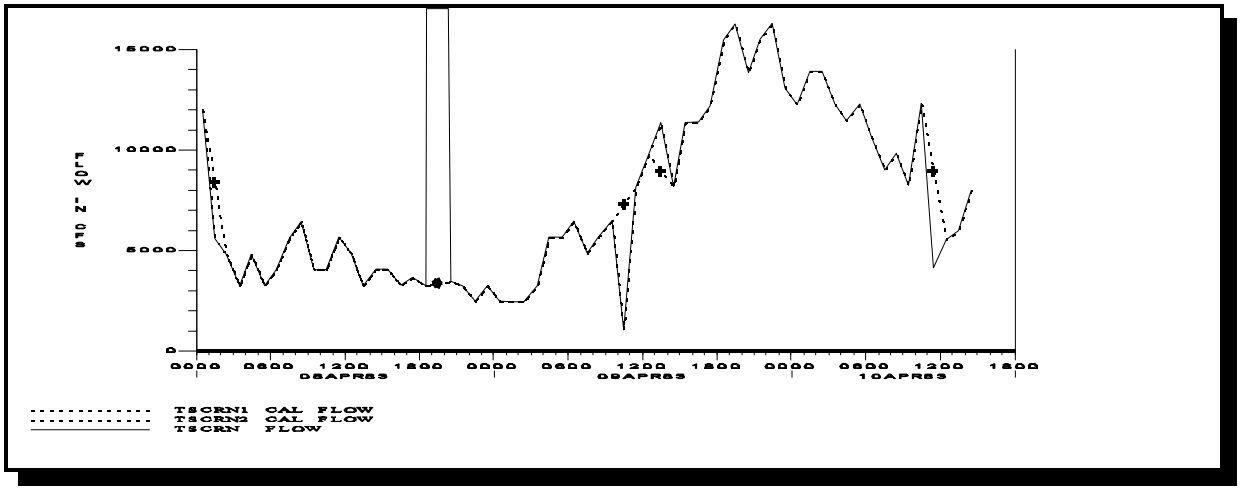
Macro:

```
MACRO TSCRN3
CL ALL
TIME 08APR1983 0100 10APR1983 1500
GET FLOW=testdb.dss://TSCRN/FLOW/01APR1983/1HOUR//
COMP SFLOW1=SCRN1(FLOW,0,50000,5000,M)
COMP SFLOW2=SCRN2(FLOW,4,5000,M)
PUT.A SFLOW1=B=TSCRN1,F=CAL
PUT.A SFLOW2=B=TSCRN2,F=CAL
TAB.F FLOW SFLOW1 SFLOW2
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
ALL VARIABLES HAVE BEEN CLEARED
  -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.    6:  //TSCRN/FLOW/01APR1983/1HOUR//
DMIN,DMAX,DEL =    0.000000E+00    50000.000000     5000.000000
-----DSS---ZWRITE Unit  71; Vers.    6:  //TSCRN1/FLOW/01APR1983/1HOUR/CAL/
-----DSS---ZWRITE Unit  71; Vers.    6:  //TSCRN2/FLOW/01APR1983/1HOUR/CAL/
```

Variables ----	FLOW	SFLOW1	SFLOW2
Units ----	CFS	CFS	CFS
Time Date			
0030 08APR1983	12005.000	12005.000	12005.000
0130 08APR1983	5638.000	M	5638.000
0230 08APR1983	4842.000	4842.000	4842.000
0330 08APR1983	3251.000	3251.000	3251.000
0430 08APR1983	4847.000	4847.000	4847.000
0530 08APR1983	3253.000	3253.000	3253.000
0630 08APR1983	4052.000	4052.000	4052.000
0730 08APR1983	5651.000	5651.000	5651.000
0830 08APR1983	6455.000	6455.000	6455.000
0930 08APR1983	4068.000	4068.000	4068.000
1030 08APR1983	4075.000	4075.000	4075.000
1130 08APR1983	5680.000	5680.000	5680.000
1230 08APR1983	4888.000	4888.000	4888.000
1330 08APR1983	3293.000	3293.000	3293.000
1430 08APR1983	4094.000	4094.000	4094.000
1530 08APR1983	4094.000	4094.000	4094.000
1630 08APR1983	3297.000	3297.000	3297.000
1730 08APR1983	3686.274	3686.274	3686.274
1830 08APR1983	3294.000	3294.000	3294.000
1930 08APR1983	99999.000	M	M
2030 08APR1983	3490.196	3490.196	3490.196
2130 08APR1983	3289.000	3289.000	3289.000
2230 08APR1983	2486.000	2486.000	2486.000
2330 08APR1983	3285.000	3285.000	3285.000
0030 09APR1983	2483.000	2483.000	2483.000
0130 09APR1983	2448.000	2448.000	2448.000
0230 09APR1983	2458.000	2458.000	2458.000
0330 09APR1983	3285.000	3285.000	3285.000
0430 09APR1983	5692.000	5692.000	5692.000
0530 09APR1983	5692.000	5692.000	5692.000
0630 09APR1983	6496.000	6496.000	6496.000
0730 09APR1983	4898.000	4898.000	4898.000
0830 09APR1983	5709.000	5709.000	5709.000
0930 09APR1983	6528.000	6528.000	6528.000
1030 09APR1983	1058.823	M	1058.823
1130 09APR1983	8156.000	8156.000	8156.000
1230 09APR1983	9771.000	9771.000	9771.000
1330 09APR1983	11390.000	11390.000	M
1430 09APR1983	8179.000	8179.000	8179.000
1530 09APR1983	11408.000	11408.000	11408.000
1630 09APR1983	11413.000	11413.000	11413.000
1730 09APR1983	12227.000	12227.000	12227.000
1830 09APR1983	15476.000	15476.000	15476.000
1930 09APR1983	16292.000	16292.000	16292.000
2030 09APR1983	13882.000	13882.000	13882.000
2130 09APR1983	15511.000	15511.000	15511.000
2230 09APR1983	16331.000	16331.000	16331.000
2330 09APR1983	13103.000	13103.000	13103.000
0030 10APR1983	12288.000	12288.000	12288.000
0130 10APR1983	13912.000	13912.000	13912.000
0230 10APR1983	13918.000	13918.000	13918.000
0330 10APR1983	12297.000	12297.000	12297.000
0430 10APR1983	11489.000	11489.000	11489.000
0530 10APR1983	12304.000	12304.000	12304.000
0630 10APR1983	10679.000	10679.000	10679.000
0730 10APR1983	9053.000	9053.000	9053.000
0830 10APR1983	9873.000	9873.000	9873.000
0930 10APR1983	8244.000	8244.000	8244.000
1030 10APR1983	12334.000	12334.000	12334.000
1130 10APR1983	4166.000	M	M
1230 10APR1983	5550.000	5550.000	5550.000
1330 10APR1983	6000.000	6000.000	6000.000
1430 10APR1983	8010.000	8010.000	8010.000



Name: *SDEV* *Compute the standard deviation of one independent variable.*

Use: *CO SY=SDEV(TX)*

This function is used to compute the standard deviation of one independent variable. TX can be either regular or irregular time series. SY is a scalar and is set to the computed standard deviation. This function requires a minimum of three valid values to compute.

Name: *SKEW* *Compute the skew coefficient of one independent variable*

Use: *CO SY=SKEW(TX)*

This function is used to compute the skew coefficient of one independent variable. TX can be either regular or irregular time series. SY is a scalar and is set to the computed skew coefficient. This function requires a minimum of three valid values to compute.

Example: The following example uses the SDEV and SKEW functions to determine the standard deviation and skew of a set of time series data. Note that the SDEV and SKEW function are the same, with the only difference being that each sets either the scalar variable to standard deviation or skew.

Macro:

```
MACRO SDSKEW
TIME 01MAY1983 0100 06MAY1983 2300
CL ALL
GE TX=TESTDB.DSS:/SCIOTO/BMRI2/FLOW/01MAY1983/1HOUR/OBS/
COMP SDEV=SDEV(TX)
COMP TSKEW=SKEW(TX)
SHOW SDEV TSKEW
ENDMACRO
```

Execution:

```
ALL VARIABLES HAVE BEEN CLEARED
  -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.   24:  /SCIOTO/BMRI2/FLOW/01MAY1983/1HOUR/OBS/
NUMBER OF VALID PAIRS        143
NUMBER OF MISSING VALUES          0
MEAN     143.799200
STANDARD DEVIATION      48.882410
SKEW COEFFICIENT   9.202288E-01
NUMBER OF VALID PAIRS        143
NUMBER OF MISSING VALUES          0
MEAN     143.799200
STANDARD DEVIATION      48.882410
SKEW COEFFICIENT   9.202288E-01
```

```
VARIABLE=SDEV
SDEV    =      48.882410
VARIABLE=TSKEW
TSKEW   =   9.202288E-01
```

Name: *SELECT* *Extract time series data at unique time specification.*

Use: *CO TS= SELECT(TX,LEVEL,RANGE,FLAG,TWINDOW,TFLAG)*

This function is used to extract data from any regular or irregular time series, based on any number of unique time specifications. The extraction process can take place on any of five different selection LEVELs which can be mutually inclusive or exclusive as defined by the FLAG and RANGE parameters. Each LEVEL has a RANGE parameter that is either a specific value or a range of values. The extracted data is defaulted to be irregular time series, but the user has the option to redefine it as regular by the TFLAG parameter. CAUTION: Defining the extracted data to be regular will cause it to be stored in a consecutive manner at regular time interval specified on the E pathname part without regard to its actual irregular date and time.

Parameter	*Description*
TX	*Input variable name of regular or irregular time series.*
LEVEL	*Level of time specific selection.*
RANGE	*Beginning and ending values of LEVEL. If only one value is specified, then the beginning and ending values are assumed to be the same. The following table shows all the valid LEVEL and RANGE values.*

LEVEL	*RANGE*	*Example*
YEAR	*Numeric: Four digit year value or a range of values. #### or ####-####*	*1938 or 1938-1945*
MONTH	*Alpha: First three characters of a month or a range of months. aaa or aaa-aaa*	*JAN or OCT-FEB*
DAYMON	*Numeric/Alpha: One or two digit day value or a range of values or the key word "LASTDAY" which specifies the last day of the month.*	*15 or 1-15 or LASTDAY or 15-LASTDAY*
DAYWEE	*Alpha: First three characters of a week day or a range of week days. aaa or aaa-aaa Note: Sunday is day 1 and Saturday is 7.*	*MON or SUN-SAT*
TIME	*Numeric: Four digit military style 24 hour clock consisting of a single time or a range of time. #### or ####-####*	*2300 or 0300-0600*

Parameter	*Description*
FLAG	*The valid entries for the processing FLAG are "INCLUDE" (default) or "EXCLUDE". This entry controls the inclusion or exclusion of all data specified by LEVEL and RANGE.*
TWINDOW	*This parameter is only used with the TIME component of the LEVEL parameter and is expressed in minutes before and after the time of day within which the data will be extracted. The window is assumed to be symmetric about the time specified in RANGE. At the time boundaries of 0000 and 2400, the time window will not cross into the previous or next day.*
TFLAG	*The valid entries for the TFLAG are "IRREGULAR" (default) or "REGULAR". This parameter allows the user to control how DSSMATH treats the data extracted as it relates to its internal definition of the type of time series data it is. This allows the user to override the protection in the program to write irregular data as regular.*

Examples : This first example is designed to demonstrate how the SELECT function works and is not a real working example. The second example given is a real working example acquired from NPD in which the SELECT function is used to select the drawdown elevations for the last day in April for the upper rule curve and for a reservoir simulation study.

Example 1:

Macro:

```
MACRO SELECT
 CL ALL
 TI 01JAN19 0100 31DEC24 2400
 GET TY=RMS.DSS:/ALLEGHENY/P109.159/PRECIP-INC/01APR1919/IR-MONTH/HARRIS/
 COMP TS=SELECT(TY,YEAR,1922-1922,INCLUDE,0,IRREGULAR)
 PUT.A TS=F=SELECT YEAR
 CLEAR TY
 COMP TX=SELECT(TS,MONT,SEP-OCT,INCLUDE,0,IRREGULAR)
 PUT.A TX=F=SELECT MONTH
 COMP TS1=SELECT(TX,DAYM,20-LASTDAY,INCLUDE,0,IRREGULAR)
 PUT.A TS1=F=SELECT DAYMONTH
 COMP TX1=SELECT(TS1,DAYW,MON-SAT,INCLUDE,0,IRREGULAR)
 PUT.A TX1=F=SELECT DAYWEEK
 COMP TS2=SELECT(TX1,TIME,2400,INCLUDE,10,IRREGULAR)
 PUT.A TS2=F=SELECT TIME
 $CONTINUE INCASE OF ERROR
 ENDMACRO
```

Execution:

```
ALL VARIABLES HAVE BEEN CLEARED
 -----DSS---ZOPEN:  Existing File Opened,  File: RMS.DSS
                    Unit:  71;  DSS Version: 6-GM
-----DSS---ZWRITE Unit  71; Vers.   11:  /ALLEGHENY/P109.159/PRECIP-INC/01MAY1922/IR-MONTH/SELECT YEAR/
 TY CLEARED
-----DSS---ZWRITE Unit  71; Vers.   11:  /ALLEGHENY/P109.159/PRECIP-INC/01SEP1922/IR-MONTH/SELECT MONTH/
```

Variables ----	TS	TX	TS1	TX1	TS2
Units ----	INCHES	INCHES	INCHES	INCHES	INCHES
Time Date					
0400 30JUL1922	.529	-	-	-	-
1600 30JUL1922	.529	-	-	-	-
2000 30JUL1922	.529	-	-	-	-
2400 30JUL1922	.529	-	-	-	-
0400 01AUG1922	.529	-	-	-	-
1200 01AUG1922	.529	-	-	-	-
0400 22AUG1922	.529	-	-	-	-
2000 25AUG1922	.529	-	-	-	-
2400 26AUG1922	.529	-	-	-	-
0400 28AUG1922	.529	-	-	-	-
1600 28AUG1922	.529	-	-	-	-
2000 28AUG1922	.529	-	-	-	-
2400 28AUG1922	.529	-	-	-	-
0400 29AUG1922	.529	-	-	-	-
1600 29AUG1922	.529	-	-	-	-
2000 29AUG1922	.529	-	-	-	-
2400 29AUG1922	.529	-	-	-	-
0400 31AUG1922	.529	-	-	-	-
1200 31AUG1922	.529	-	-	-	-
0400 21SEP1922	.483	.483	.483	.483	-
2000 24SEP1922	.455	.455	.455	-	-
2400 25SEP1922	.455	.455	.455	.455	.455
0400 27SEP1922	.455	.455	.455	.455	-
1600 27SEP1922	.455	.455	.455	.455	-
2000 27SEP1922	.455	.455	.455	.455	-
2400 27SEP1922	.455	.455	.455	.455	.455
0400 28SEP1922	.455	.455	.455	.455	-
1600 28SEP1922	.455	.455	.455	.455	-
2000 28SEP1922	.455	.455	.455	.455	-
2400 28SEP1922	.455	.455	.455	.455	.455
0400 30SEP1922	.455	.455	.455	.455	-
1200 30SEP1922	.455	.455	.455	.455	-
0400 26OCT1922	.356	.356	.356	.356	-
2000 29OCT1922	.356	.356	.356	-	-
2400 30OCT1922	.356	.356	.356	.356	.356
0400 01NOV1922	.356	-	-	-	-
1600 01NOV1922	.356	-	-	-	-
2000 01NOV1922	.356	-	-	-	-
2400 01NOV1922	.356	-	-	-	-
0400 02NOV1922	.356	-	-	-	-
1600 02NOV1922	.356	-	-	-	-
2000 02NOV1922	.356	-	-	-	-
2400 02NOV1922	.356	-	-	-	-
0400 04NOV1922	.356	-	-	-	-
1200 04NOV1922	.356	-	-	-	-
0400 05DEC1922	.275	-	-	-	-
2000 08DEC1922	.000	-	-	-	-
2400 09DEC1922	.152	-	-	-	-
0400 11DEC1922	.152	-	-	-	-

Example 2:

Macro:

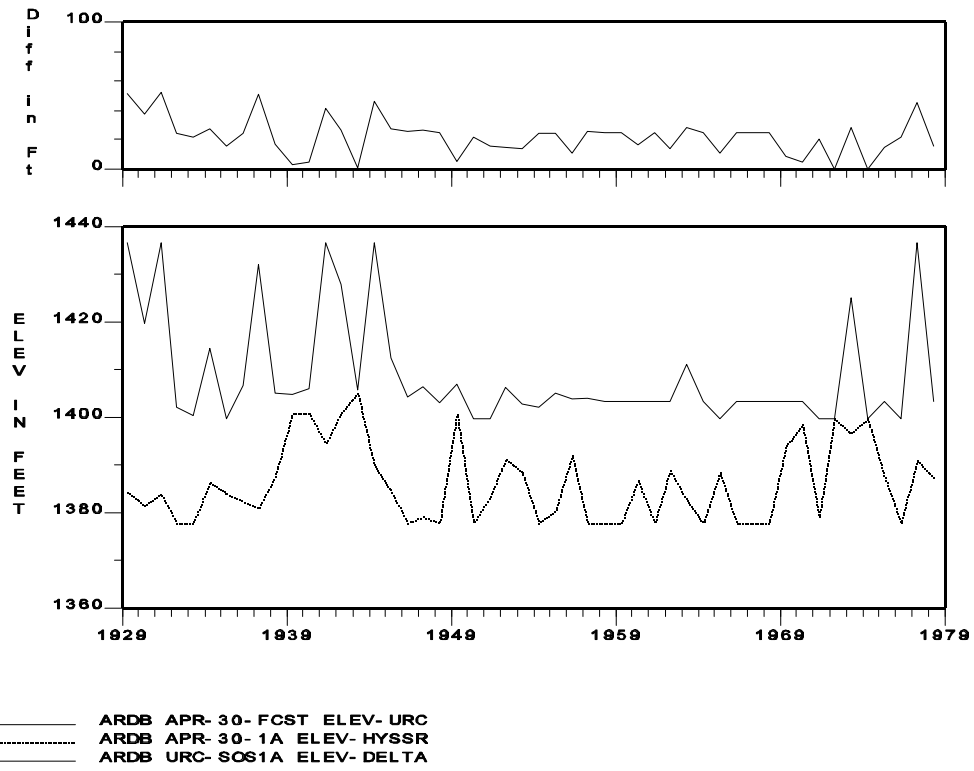
```
MACRO TSELURC
** Using SELECT to compare URC to SOR's 30 apr drawdown pts.
CL ALL
TI 01JAN1929 2400 30SEP1978 2400
GET URCF=FIXED.DSS://ARDB/ELEV-URC//1DAY/FCST/
CO URCFA=SELECT(URCF,MONTH,APR,INCLUDE,0)
CO URCFAE=SELECT(URCFA,DAYMONTH,LAST,INCLUDE,0)
CO URC=URCFAE
SD URC U=FEET T=INST-VAL
PUT.A URC=COMPSOR.DSS://ARDB/ELEV-URC//IR-DECADE/APR-30-FCST/
CL URCF URCFA URCFAE
GET S1AHSR=1A.DSS://ARDB/ELEV-HYSSR//IR-DECADE/1A/
CO S1AHS=SELECT(S1AHSR,MONTH,APR,INCLUDE,0)
CO S1AH=SELECT(S1AHS,DAYMONTH,LAST,INCLUDE,0)
CO S1A=S1AH
SD S1A U=FEET T=INST-VAL
PUT.A S1A=COMPSOR.DSS://ARDB/ELEV-HYSSR//IR-DECADE/APR-30-1A/
CL S1AHSR S1AHS S1AH
CO DIFF=URC-S1A
SD DIFF U=FEET T=INST-VAL
PUT.A DIFF=COMPSOR.DSS://ARDB/ELEV-DELTA//IR-DECADE/URC-SOS1A/
TAB.F URC S1A DIFF
$CONTINUE
ENDMACRO
```

Execution:

```
     -----DSS---ZOPEN:  Existing File Opened,  File: FIXED.DSS
                    Unit:  71;  DSS Version: 6-IC
-----DSS--- ZREAD Unit  71; Vers.    1:  //ARDB/ELEV-URC/01JAN1929/1DAY/FCST/
-----DSS---ZOPEN:  Existing File Opened,  File: COMPSOR.DSS
                    Unit:  72;  DSS Version: 6-IC
-----DSS---ZWRITE Unit  72; Vers.    4:  //ARDB/ELEV-URC/01JAN1920/IR-DECADE/APR-30-FCST/
URCF CLEARED
URCFA CLEARED
URCFAE CLEARED
 -----DSS---ZOPEN:  Existing File Opened,  File: 1A.DSS
                    Unit:  73;  DSS Version: 6-IC
-----DSS---ZWRITE Unit  72; Vers.    4:  //ARDB/ELEV-HYSSR/01JAN1920/IR-DECADE/APR-30-1A/

S1AHSR CLEARED
S1AHS CLEARED
S1AH CLEARED
-----DSS---ZWRITE Unit  72; Vers.    4:  //ARDB/ELEV-DELTA/01JAN1920/IR-DECADE/URC-SOS1A/
```

Variables ----	URC	S1A	DIFF
Units ----	FEET	FEET	FEET
Time Date			
2400 30APR1929	1436.700	1384.400	52.300
2400 30APR1930	1419.800	1381.900	37.900
2400 30APR1931	1436.700	1384.200	52.500
2400 30APR1932	1402.200	1377.900	24.300
2400 30APR1933	1400.500	1377.900	22.600
2400 30APR1934	1414.500	1386.700	27.800
2400 30APR1935	1399.900	1384.100	15.800
2400 30APR1936	1406.900	1382.600	24.300
2400 30APR1937	1432.100	1381.200	50.900
2400 30APR1938	1405.100	1387.900	17.200
2400 30APR1939	1404.900	1401.100	3.800
2400 30APR1940	1406.200	1401.100	5.100
2400 30APR1941	1436.700	1394.700	42.000
2400 30APR1942	1428.000	1401.100	26.900
2400 30APR1943	1406.000	1405.200	0.800
2400 30APR1944	1436.700	1389.900	46.800
2400 30APR1945	1412.600	1384.600	28.000
2400 30APR1946	1404.400	1377.900	26.500
2400 30APR1947	1406.700	1379.400	27.300
2400 30APR1949	1407.000	1401.100	5.900
2400 30APR1950	1399.900	1377.900	22.000
2400 30APR1951	1399.900	1383.500	16.400
2400 30APR1952	1406.400	1391.500	14.900
2400 30APR1953	1403.100	1388.900	14.200
2400 30APR1954	1402.400	1377.900	24.500
2400 30APR1955	1405.100	1380.500	24.600
2400 30APR1956	1403.900	1392.200	11.700
2400 30APR1957	1404.100	1377.900	26.200
2400 30APR1958	1403.400	1377.900	25.500
2400 30APR1959	1403.400	1377.900	25.500
2400 30APR1960	1403.400	1386.900	16.500
2400 30APR1961	1403.400	1377.900	25.500
2400 30APR1962	1403.400	1389.000	14.400
2400 30APR1963	1411.200	1382.800	28.400
2400 30APR1964	1403.400	1377.900	25.500
2400 30APR1965	1399.900	1388.500	11.400
2400 30APR1966	1403.400	1377.900	25.500
2400 30APR1967	1403.400	1377.900	25.500
2400 30APR1968	1403.400	1377.900	25.500
2400 30APR1969	1403.400	1394.100	9.300
2400 30APR1970	1403.400	1398.700	4.700
2400 30APR1971	1399.900	1379.500	20.400
2400 30APR1972	1399.900	1399.800	0.100
2400 30APR1973	1425.300	1396.700	28.600
2400 30APR1974	1399.900	1399.800	0.100
2400 30APR1975	1403.400	1387.800	15.600
2400 30APR1976	1399.900	1377.900	22.000
2400 30APR1977	1436.700	1391.200	45.500
2400 30APR1978	1403.400	1387.700	15.700
2400 30APR1948	1403.300	1377.900	25.400



Name: *SHIFT* *Shift adjustment*
Use: *CO TY=SHIFT(TS,TX)*

Generate a time series of shift adjustments TY with times at TX and actual shifts TS, possibly at other times. TX and TY cannot be the same variables. TS values are interpolated at TX times. The interpolation is linear between TS values, unless TX time is greater than last TS time. Then, the last value of TS is held constant for the remaining TX times. If no previous TS value brackets TY, then TY is set to zero. An IF condition has no effect.

Example: The following example uses the SHIFT and PERCON functions to demonstrate how they can be used in conjunction with an irregular time series variable of gage shift values to create a regular time series of shift adjustments. Note that the SHIFT function interpolates while the PERCON function holds them constant for the period.

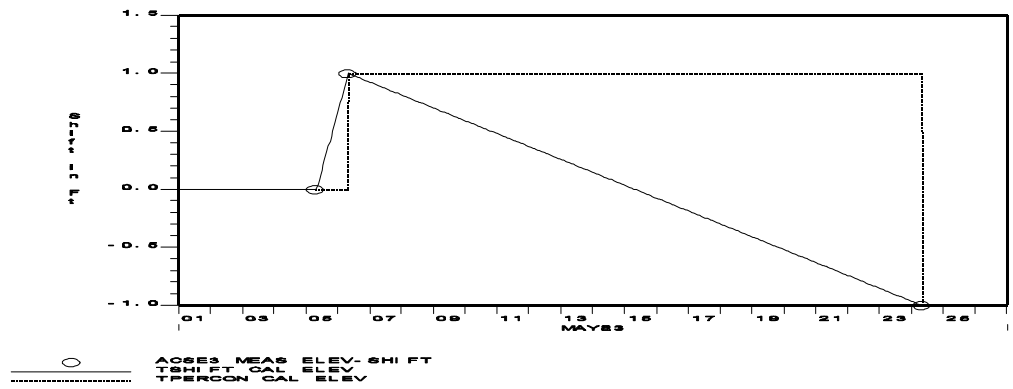
Macro:

```
MACRO TSHIFT
CL ALL
TI 0100 01MAY83 2400 26MAY83
GET ELEV=testdb.dss:/SCIOTO/ACSE3/ELEV/01MAY1983/1HOUR/OBS/
GET SHFT=testdb.dss:/SCIOTO/ACSE3/ELEV-SHIFT/01JAN1983/IR-YEAR/MEAS/
GET ELEV=testdb.dss:/SCIOTO/ACSE3/ELEV/01MAY1983/1HOUR/OBS/
COM TES=SHIFT(SHFT,ELEV)
COM TEP=PERCON(SHFT,ELEV)
PUT.A TES=B=TSHIFT,F=CAL
PUT.A TEP=B=TPERCON,F=CAL
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                      Unit:  71;  DSS Version: 6-IA
 -----DSS--- ZREAD Unit  71; Vers.   22:  /SCIOTO/ACSE3/ELEV/01MAY1983/1HOUR/OBS/
 ELEV   CLEARED
 -----DSS--- ZREAD Unit  71; Vers.   22:  /SCIOTO/ACSE3/ELEV/01MAY1983/1HOUR/OBS/
 -----DSS---ZWRITE Unit  71; Vers.    1:  /SCIOTO/TSHIFT/ELEV/01MAY1983/1HOUR/CAL/
 -----DSS---ZWRITE Unit  71; Vers.    1:  /SCIOTO/TPERCON/ELEV/01MAY1983/1HOUR/CAL/
```

Variables ----	SHFT	ELEV	TES	TEP
Units ----	FEET	FT-MSL	FEET	UNK
Time Date				
0100 01MAY1983	-	888.900	.000	M
0200 01MAY1983	-	889.000	.000	M
0300 01MAY1983	-	M	.000	M
0400 01MAY1983	-	M	.000	M
0500 01MAY1983	-	M	.000	M
0600 01MAY1983	-	M	.000	M
0700 01MAY1983	-	889.430	.000	M
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
0600 05MAY1983	-	892.770	.000	M
0700 05MAY1983	-	892.770	.000	M
0800 05MAY1983	.000	892.730	.000	.000
0900 05MAY1983	-	892.660	.042	.000
1000 05MAY1983	-	892.520	.083	.000
1100 05MAY1983	-	892.540	.125	.000
1200 05MAY1983	-	892.520	.167	.000
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
0500 06MAY1983	-	892.150	.875	.000
0600 06MAY1983	-	892.160	.917	.000
0700 06MAY1983	-	892.100	.958	.000
0800 06MAY1983	1.000	892.060	1.000	1.000
0900 06MAY1983	-	892.050	.995	1.000
1000 06MAY1983	-	892.040	.991	1.000
1100 06MAY1983	-	891.980	.986	1.000
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
0500 24MAY1983	-	888.290	-.986	1.000
0600 24MAY1983	-	888.280	-.991	1.000
0700 24MAY1983	-	888.280	-.995	1.000
0800 24MAY1983	-1.000	888.280	-1.000	-1.000
0900 24MAY1983	-	888.270	-1.000	-1.000
1000 24MAY1983	-	888.270	-1.000	-1.000
1100 24MAY1983	-	888.270	-1.000	-1.000
1200 24MAY1983	-	888.260	-1.000	-1.000
1300 24MAY1983	-	888.260	-1.000	-1.000
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
2300 26MAY1983	-	888.170	-1.000	-1.000
2400 26MAY1983	-	888.170	-1.000	-1.000



Name: *SQRT* *Square root function*
Use: *CO TY=SQRT(TX)*

Compute the square root of TX and store in TY. TX and TY can be the same variable. TX can be scalar, time series or paired data. If a value of TX is undefined, resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged. If a value in TX is negative, the value in TY will be set to missing.

Name: *SS* *Straddle Stagger routing function*
Use: *CO TY=SS(TX,NAVG,LAG,NR)*

Route the uniform time series variable TX by the Straddle Stagger hydrologic routing method and store it in variable TY. Note: Variables TY and TX cannot be the same variable and variable TX is permanently modified by being lagged by the SS function. Variable NAVG is the number of ordinates to average. Variable LAG is the number of ordinates to lag hydrograph. Variable NR is the number of subreaches to use. The "IF" compute option has no effect on this function.

Name: *SSW* *Wilmington District Straddle Stagger routing function*
Use: *CO TY=SSW(TX,NAVG,LAG,NR)*

Route the uniform time series variable TX by the Straddle Stagger hydrologic routing method and store it in variable TY. Note: Variables TY and TX cannot be the same variable. This function is similar to function SS except that LAG is usually zero and the result of averaging NAVG values is stored in the NAVGed value. Variable NAVG is the number of ordinates to average. Variable LAG is the number of ordinates to lag hydrograph. Variable NR is the number of subreaches to use. For example, if NAVG is 7, the results for ordinates 13 through 19 is stored in ordinate 19. For the same example, if LAG is set to 2, the result is stored in ordinate 21. The "IF" compute option has no effect on this function.

Example: The following example uses the TSHIFT and SQRT functions to compute the lagged geometric mean of an observed flow. The same observed flow is then routed using the Straddle Stagger functions SS and SSW. Note: These examples are not real life applications and only serves to demonstrate

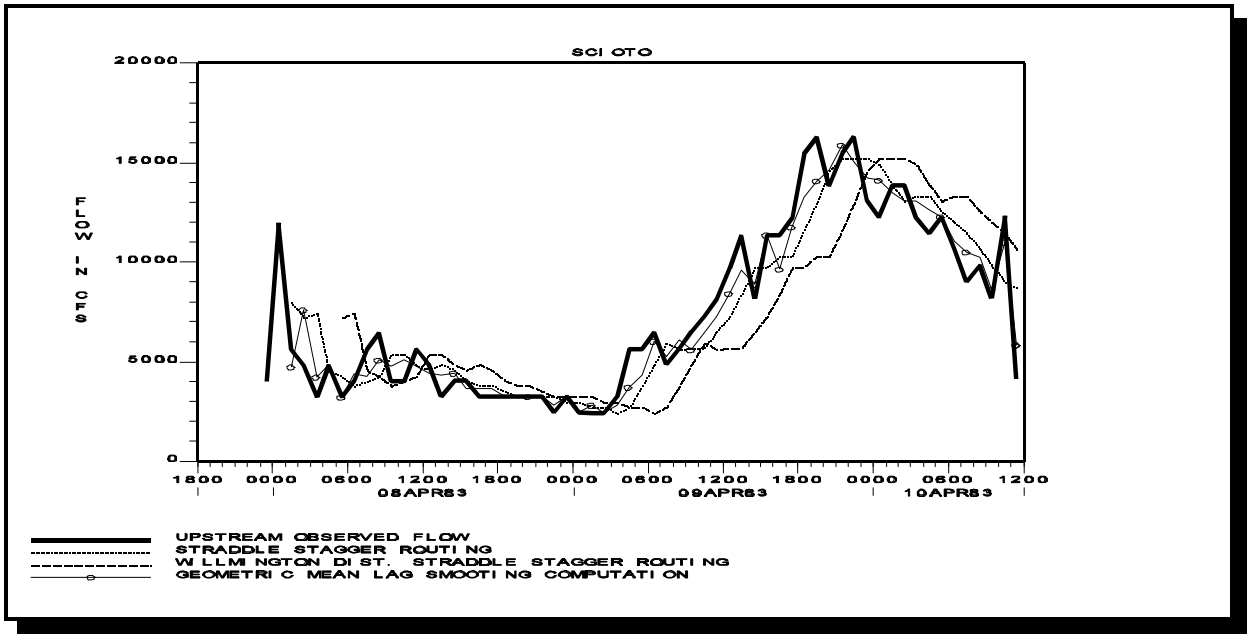
the use and results of the functions.

Macro:

```
 MACRO TSS
CL ALL
TIME 07APR1983 0100 10APR1983 1200
GET FLOW=testdb.dss:/SCIOTO/ACSE3/FLOW/01APR1983/1HOUR/OBS/
COMP TX=TSHIFT(FLOW,2H)
COMP TX=FLOW*TX
COMP TX=SQRT(TX)
PUT.A TX=B=TXSS,F=CAL
** COMPUTE A STRADDLE STAGGER ROUTING
COMP RFLOW=SS(FLOW,3,2,1)
PUT.A RFLOW=B=TSS,F=CAL
** COMPUTE A WILMINGTON DIST. STRADDLE STAGGER ROUTING
COMP RWFLOW=SSW(FLOW,3,2,1)
PUT.A RWFLOW=B=TWSS,F=CAL
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
!R TSS
ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                      Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.    3:  /SCIOTO/ACSE3/FLOW/01APR1983/1HOUR/OBS/
WARNING STARTING TIMES ARE NOT THE SAME
WARNING -- EXISTING VARIABLE RE-DEFINED
WARNING -- EXISTING VARIABLE RE-DEFINED
-----DSS---ZWRITE Unit  71; Vers.    2:  /SCIOTO/TXSS/FLOW/01APR1983/1HOUR/CAL/
Warning - Variable:  FLOW        Has been lagged
-----DSS---ZWRITE Unit  71; Vers.    7:  /SCIOTO/TSS/FLOW/01APR1983/1HOUR/CAL/
Warning - Variable:  FLOW        Has been lagged
-----DSS---ZWRITE Unit  71; Vers.    7:  /SCIOTO/TWSS/FLOW/01APR1983/1HOUR/CAL/
```



Name: TS1 Interpolated data at regular intervals
Use: CO TY=TS1(TX,DT,TOFF)

Derive a new, regular-interval time series TY from TX. TY will have a time span defined by the current time window, a time interval of DT, and a time interval offset of TOFF. TOFF is time from the beginning of the standard interval to the actual time of the data. DT and TOFF are expressed as units of time as nnT, where T is M for minutes, H for hours, and D for days. TX may be a regular or irregular time series. Units and type are preserved. An IF condition has no effect.

Name: TS2 Period averages at regular intervals
Use: CO TY=TS2(TX,DT,TOFF)

Derive a new, regular-interval time-series TY from the instantaneous values in TX with a time span defined by the current time window, a time interval of DT, and an interval offset of TOFF. DT and TOFF are expressed as units of time as nnT, where T is M for minutes, H for hours, and D for days. TOFF is time from the beginning of the standard interval to the actual time of the data. TX may be an irregular or regular time-series, but must be of type INST-VAL. TY will be typed PER-AVER. Example of use: deriving daily average flow from hourly observed values. An IF condition has no effect.

Name: TS3 Period mins or maxs at regular intervals
Use: CO TY=TS3(TX,DT,TOFF,EXT)

Finds either the minima or maxima in TX at DT intervals, with interval offsets at TOFF, and puts the result in TY. EXT is used to indicate the extreme of interest: 'MIN' or 'MAX'. DT and TOFF are expressed as units of time as nnT, where T is M for minutes, H for hours, and D for days. TOFF is time from the beginning of the standard interval to the actual time of the data. TY is typed PER-EXTR. Example of use: finding daily minima and maxima from hourly instantaneous observations. An IF condition has no effect.

Example: The following examples use the TS1, TS2, and TS3 functions to derive a new 6-hour regular interval from a 1-hour regular time interval based on linear

interpolation, period average, and maximum and minimum values. Note: These examples are not real life applications and only serves to demonstrate the use and results of the functions.

Macro:

```
MACRO TTS1-3
CL ALL
TIME 09APR1983 1200 10APR1983 1200
GET FLOW=testdb.dss:/SCIOTO/ACSE3/FLOW/01APR1983/1HOUR/OBS/
COMP TS1=TS1(FLOW,6H,0M)
COMP TS2=TS2(FLOW,6H,0M)
COMP TS3MX=TS3(FLOW,6H,0M,MAX)
COMP TS3MN=TS3(FLOW,6H,0M,MIN)
TAB.F FLOW TS1 TS2 TS3MX TS3MN
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
I>!R TTS1-3
  ALL VARIABLES HAVE BEEN CLEARED
    -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                       Unit:  71;  DSS Version: 6-IA
 -----DSS--- ZREAD Unit  71; Vers.    3:  /SCIOTO/ACSE3/FLOW/01APR1983/1HOUR/OBS/
```

```
Variables ----    FLOW         TS1         TS2         TS3MX       TS3MN
Units     ----    CFS          CFS         CFS         CFS         CFS
Time   Date
1130 09APR1983    8146.000      -          -          -          -
1200 09APR1983      -       8953.501      M          M          M
1230 09APR1983    9761.000      -          -          -          -
1330 09APR1983   11380.000      -          -          -          -
1430 09APR1983    8169.000      -          -          -          -
1530 09APR1983   11398.000      -          -          -          -
1630 09APR1983   11403.000      -          -          -          -
1730 09APR1983   12217.000      -          -          -          -
1800 09APR1983      -       13841.500   10755.380   12217.000    8169.000
1830 09APR1983   15466.000      -          -          -          -
1930 09APR1983   16282.000      -          -          -          -
2030 09APR1983   13872.000      -          -          -          -
2130 09APR1983   15501.000      -          -          -          -
2230 09APR1983   16321.000      -          -          -          -
2330 09APR1983   13093.000      -          -          -          -
2400 09APR1983      -       12685.500   15004.500   16321.000   13093.000
0030 10APR1983   12278.000      -          -          -          -
0130 10APR1983   13902.000      -          -          -          -
0230 10APR1983   13908.000      -          -          -          -
0330 10APR1983   12287.000      -          -          -          -
0430 10APR1983   11479.000      -          -          -          -
0530 10APR1983   12294.000      -          -          -          -
0600 10APR1983      -       11481.500   12674.460   13908.000   11479.000
0630 10APR1983   10669.000      -          -          -          -
0730 10APR1983    9043.000      -          -          -          -
0830 10APR1983    9863.000      -          -          -          -
0930 10APR1983    8234.000      -          -          -          -
1030 10APR1983   12324.000      -          -          -          -
1130 10APR1983    4156.000      -          -          -          -
1200 10APR1983      -          M          M          M          M
```

Name: *TS4* *Interpolated data at irregular intervals*

Use: *CO TY=TS4(TX,TZ)*

Derive a new, irregular-interval time series TY from TX at the times for TZ (ie., TZ provides the time pattern). Units and type are preserved. An IF condition has no effect.

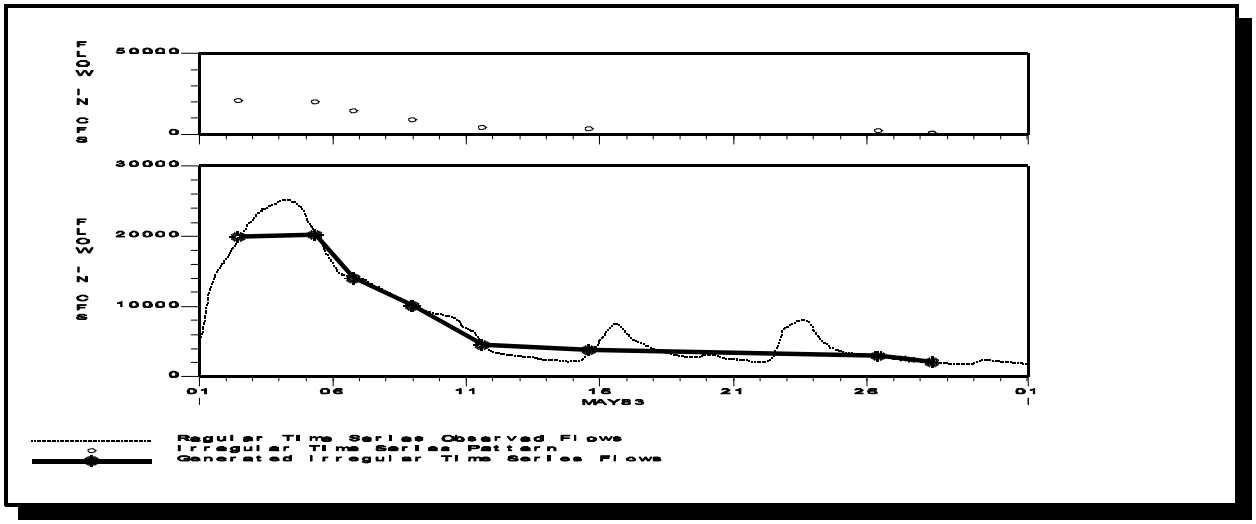
Example: The following example uses the TS4 function to generate an irregular time series data set from a regular interval data set based on a pattern irregular time series data set. Note: This example is not a real application and only serves to demonstrate the use and results of the function.

Macro:

```
 MACRO TTS4
CL ALL
TI 01MAY1983 0100 31MAY83 2400
GET TX=testdb.dss:/SCIOTO/CISG3/FLOW/01MAY1983/1HOUR/OBS/
GET TZ=testdb.dss:/SCIOTO/CISG3/FLOW/01JAN1983/IR-YEAR/MEAS/
COM TS4=TS4(TX,TZ)
PUT.A TS4=B=TTS4,F=CAL
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
I>!R TTS4
 ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                    Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.   22:  /SCIOTO/CISG3/FLOW/01MAY1983/1HOUR/OBS/
-----DSS---ZWRITE Unit  71; Vers.    2:  /SCIOTO/TTS4/FLOW/01JAN1983/IR-YEAR/CAL/
```


50000
0
30000
20000
10000
0
01
06
11
16
21
26
01
MAY83
Regular Time Series Observed Flows
Irregular Time Series Pattern
Generated Irregular Time Series Flows

Name: *TSCYCL* *Time series cyclic analysis*
Use: *CO XX=TSCYCL(TX,TREAT,STATFILE,DSSFILE)*

Derive a set of statistics from cyclic regular time series TX. TX must be regular data at a 1HOUR, 1DAY, or 1MONTH interval. The time series TX may be, for example, 30 years of 1DAY data. For each interval, (e.g., 1st day of the year, 2nd day of the year, ... , 365th day of the year), statistics would be determined. The 14 statistics determined for each interval are: maximum, minimum, average, standard deviation, 5%, 10%, 25%, 50% (median), 75%, 90%, 95% percentiles, date of maximum, date of minimum, and number of values processed. These results are written to a specified DSS file, and two text output files. The variable XX is ignored.

TREAT controls the treatment of newly generated statistic values when missing data occur in the interval. It must be expressed in either the form nn# or nn%. If nn# is used nn gives the number of missing data values in the interval that will be accepted in the computation of the new value. If nn% is used nn gives the percent of missing data values in the interval that will be accepted in the computation of the new value. If the criteria is met a data value will be generated, if not a missing value will occur.

.

STATFILE is the base name, not more than 6 characters, of two files that will be written containing the same data written to the DSS file. The first file will contain all of the statistics generated. It will be in the form of a wide table. Its name will be 'statfile.sts' or 'statfile.t' where 'statfile' is the portion of the name the user specifies. This file is wide, so it is hard to view on some systems. The second file is a subset of the first named 'statfile.sum', or 'statfile.s'. The second file is limited to 80 columns wide for easy viewing. The second file contains: maximums with their dates, minimums with their dates, average, and the 50% percentile(median) values.

DSSFILE is the DSS file into which the cyclic results will be written. The results will be, for example, a record of all the maximums for each day of the year over the 30 year period. Fourteen records will be added to the DSS file specified, each containing one of the derived statistics. The records will use the pseudo year 3000 for storing the statistics

Example: The following example uses the TSCYCL function to compute flow statistics of daily regular interval data at a reservoir gage. The amount of statistics generated is quite large and only a select sample of output is presented. Both statistics files "--.sts" and "--.sum" are presented with a couple of statistics plotted. This example was executed on the CD-4300 workstation because of the large amount of data being processed.

Macro:

```
MACRO BB
!RUN B1    LEHIGH  WALTER FLOW  1927  1975
$CONTINUE INCASE OF ERROR
ENDMACRO
MACRO B1 $BASIN $LOC $PARM $BYR $EYR
CL ALL
TIME 01JAN$BYR 2400 31DEC$EYR 2400
GE TX=lhprdss.dss:/$BASIN/$LOC/$PARM/01JAN1990/1DAY/OBS/
COM SZ = TSCYCL ( TX,10%, $LOC, $LOC )
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
I>!R BB
 ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: LHPRDSS.DSS
                    Unit:  71;  DSS Version: 6-GJ
 -----DSS--- ZREAD Unit  71; Vers.    1:  /LEHIGH/WALTER/FLOW/01JAN1927/1DAY/OBS/
 -----DSS--- ZREAD Unit  71; Vers.    1:  /LEHIGH/WALTER/FLOW/01JAN1928/1DAY/OBS/
 -----DSS--- ZREAD Unit  71; Vers.    1:  /LEHIGH/WALTER/FLOW/01JAN1929/1DAY/OBS/
 -----DSS--- ZREAD Unit  71; Vers.    1:  /LEHIGH/WALTER/FLOW/01JAN1930/1DAY/OBS/
 -----DSS--- ZREAD Unit  71; Vers.    1:  /LEHIGH/WALTER/FLOW/01JAN1931/1DAY/OBS/
 -----DSS--- ZREAD Unit  71; Vers.    1:  /LEHIGH/WALTER/FLOW/01JAN1932/1DAY/OBS/

 -----DSS--- ZREAD Unit  71; Vers.    1:  /LEHIGH/WALTER/FLOW/01JAN1975/1DAY/OBS/
  ------ Begin CYCLIC analysis ------
         Interval: 1DAY
        Begin date: 01JAN1927
    Number of data:        3180
 Missing permitted:          0
    Table filename: walter.sts
         Pathname: /LEHIGH/WALTER/FLOW/01JAN1990/1DAY/OBS/
   -----DSS---ZOPEN:  Existing File Opened,  File: WALTER.DSS
                    Unit:  72;  DSS Version: 6-IB
 -----DSS---ZWRITE Unit  72; Vers.    2:  /LEHIGH/WALTER/COUNT-FLOW/01JAN3000/1DAY/OBS[01JAN1927-31DEC1975]/
 -----DSS---ZWRITE Unit  72; Vers.    2:  /LEHIGH/WALTER/FLOW-AVER/01JAN3000/1DAY/OBS[01JAN1927-31DEC1975]/
 -----DSS---ZWRITE Unit  72; Vers.    2:  /LEHIGH/WALTER/FLOW-MAX/01JAN3000/1DAY/OBS[01JAN1927-31DEC1975]/
 -----DSS---ZWRITE Unit  72; Vers.    2:  /LEHIGH/WALTER/FLOW-MIN/01JAN3000/1DAY/OBS[01JAN1927-31DEC1975]/
 -----DSS---ZWRITE Unit  72; Vers.    2:  /LEHIGH/WALTER/FLOW-SD/01JAN3000/1DAY/OBS[01JAN1927-31DEC1975]/
 -----DSS---ZWRITE Unit  72; Vers.    2:  /LEHIGH/WALTER/FLOW-P05/01JAN3000/1DAY/OBS[01JAN1927-31DEC1975]/
 -----DSS---ZWRITE Unit  72; Vers.    2:  /LEHIGH/WALTER/FLOW-P10/01JAN3000/1DAY/OBS[01JAN1927-31DEC1975]/
 -----DSS---ZWRITE Unit  72; Vers.    2:  /LEHIGH/WALTER/FLOW-P25/01JAN3000/1DAY/OBS[01JAN1927-31DEC1975]/
 -----DSS---ZWRITE Unit  72; Vers.    2:  /LEHIGH/WALTER/FLOW-P50/01JAN3000/1DAY/OBS[01JAN1927-31DEC1975]/
 -----DSS---ZWRITE Unit  72; Vers.    2:  /LEHIGH/WALTER/FLOW-P75/01JAN3000/1DAY/OBS[01JAN1927-31DEC1975]/
 -----DSS---ZWRITE Unit  72; Vers.    2:  /LEHIGH/WALTER/FLOW-P90/01JAN3000/1DAY/OBS[01JAN1927-31DEC1975]/
 -----DSS---ZWRITE Unit  72; Vers.    2:  /LEHIGH/WALTER/FLOW-P95/01JAN3000/1DAY/OBS[01JAN1927-31DEC1975]/
 -----DSS---ZWRITE Unit  72; Vers.    2:
/LEHIGH/WALTER/DATE-FLOW-MAX/01JAN3000/1DAY/OBS[01JAN1927-31DEC1975]/
 -----DSS---ZWRITE Unit  72; Vers.    2:
/LEHIGH/WALTER/DATE-FLOW-MIN/01JAN3000/1DAY/OBS[01JAN1927-31DEC1975]/
```

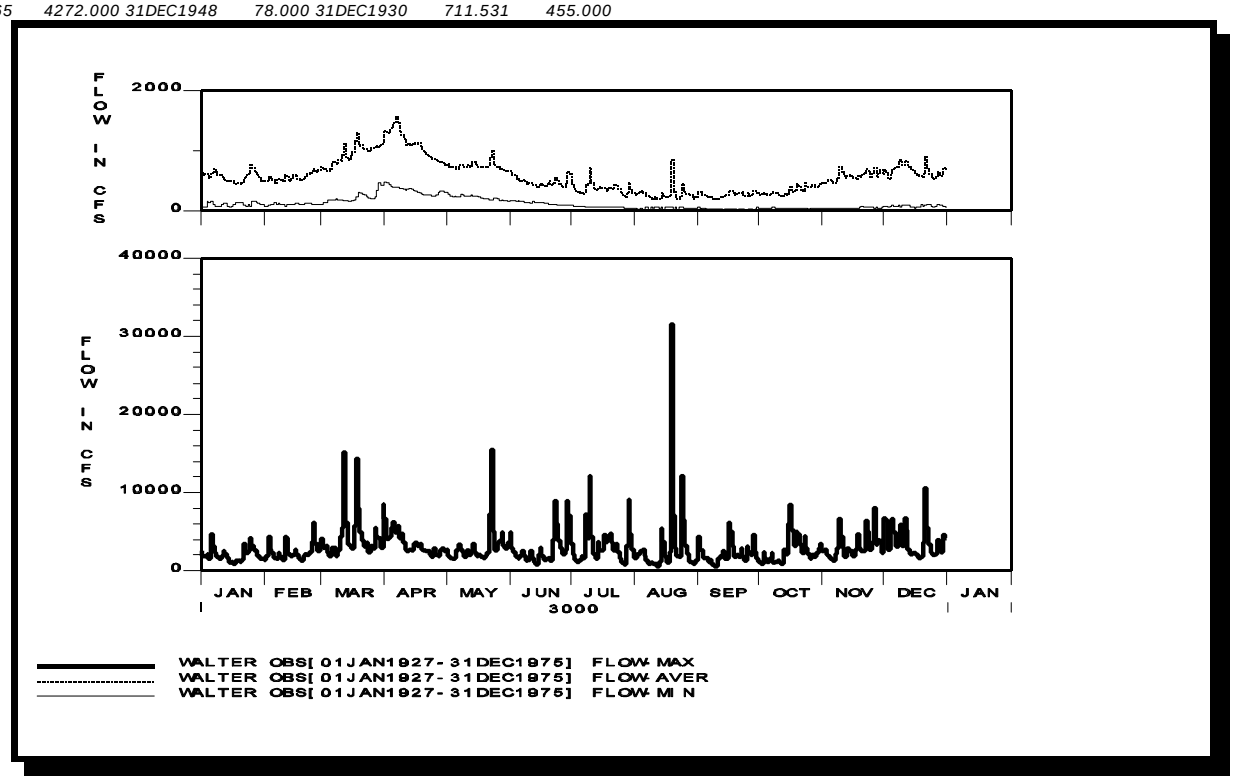
**Statistics File "WALTER.STS"

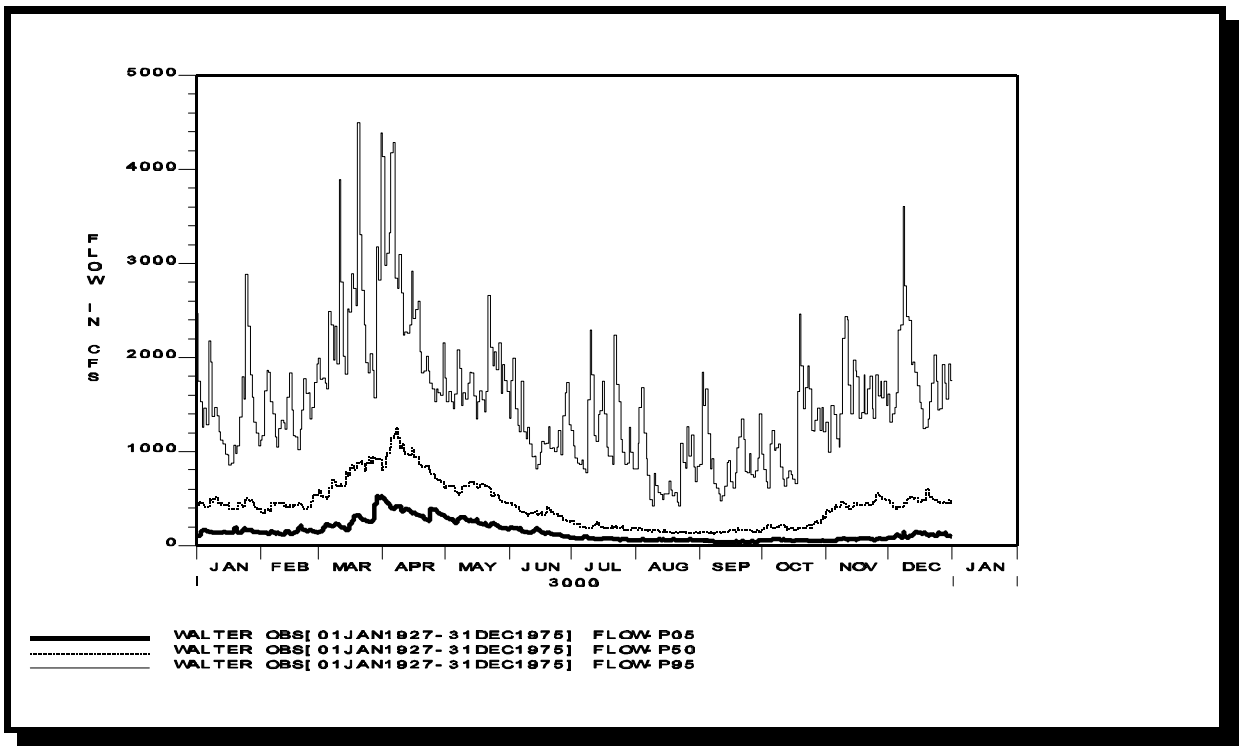
```
* Table of Statistics ***
*** Pathname> /LEHIGH/WALTER/FLOW/01JAN1990/1DAY/OBS/
*** Time window> 01JAN1927 - 31DEC1975
*** Number of bins>   365
Bin Co  Sum      Aver   Xmax Xmin    SD    P05 P10 P25 P50  P75 P90    P95    Dmax   Dmin  Dmax      Dmin     Qflag  Use  Rej
  1 48 31005   645.937  2557   69  578.305  90 183 331 441  780 1250   2471  17898  11323 01JAN1949 01JAN1931    0   48    1
  2 48 29907   623.062  1966   65  434.971 113 183 317 473  730 1211   1746  26665  11324 02JAN1973 02JAN1931    0   48    1
  3 48 30326   631.792  1970   78  443.625 159 201 296 470  754 1433   1539  21917  11325 03JAN1960 03JAN1931    0   48    1
  4 48 28673   597.354  2250  159  406.050 169 229 320 450  784 1155   1263  21918  11326 04JAN1960 04JAN1931    0   48    1
  5 48 27070   563.958  1610  156  363.564 174 219 319 427  719 1030   1469  21919  11692 05JAN1960 05JAN1932    0   48    1
  6 48 31279   651.646  4724  165  679.607 165 210 332 429  831 1111   1290  17903  11328 06JAN1949 06JAN1931    0   48    1
  7 48 34332   715.250  3233  110  641.665 156 210 345 509  832 1263   2183  17904  11329 07JAN1949 07JAN1931    0   48    1
  8 48 31813   662.771  2290   87  526.880 156 232 322 478  768 1093   1957  17905  11330 08JAN1949 08JAN1931    0   48    1
  9 48 29521   615.021  1808   83  396.767 147 230 325 507  744 1178   1379  17906  11331 09JAN1949 09JAN1931    0   48    1
 10 48 29384   612.167  1603   83  381.140 147 210 325 529  750 1134   1478  16811  11332 10JAN1946 10JAN1931    0   48    1
 11 48 28261   588.771  1968  110  393.346 138 229 301 464  741 1021   1388  27404  11333 11JAN1975 11JAN1931    0   48    1
 12 48 26442   550.875  2643  138  409.048 147 256 306 450  655  904   1209  27405  14621 12JAN1975 12JAN1940    0   48    1
 13 48 24871   518.146  2246  129  345.814 147 238 310 459  581  739   1120  27406  11335 13JAN1975 13JAN1931    0   48    1
 14 48 24198   504.125  1688   92  305.915 151 238 300 437  581  788   1084  27407  11336 14JAN1975 14JAN1931    0   48    1
 15 48 24525   510.937  1171   74  262.069 147 238 290 437  635  904    976  27408  11337 15JAN1975 15JAN1931    0   48    1
 16 48 24513   510.687  1302   92  269.332 147 243 300 455  680  869    965  22661  11338 16JAN1962 16JAN1931    0   48    1
 17 48 22599   470.812  1003  110  217.819 142 238 281 409  627  741    856  22662  11339 17JAN1962 17JAN1931    0   48    1
 .
351 49 30342   619.224  2156   65  431.526 135 183 300 489  818 1102   1458  20805  11308 17DEC1956 17DEC1930    0   49    0
352 49 29144   594.776  1692   78  363.753 142 179 315 491  818 1200   1247  20806  11309 18DEC1956 18DEC1930    0   49    0
353 49 28986   591.551  1808  117  362.247 125 167 301 617  777  988   1263  20076  11310 19DEC1954 19DEC1930    0   49    0
354 49 31525   643.367  3730   88  562.783 111 210 283 528  852 1120   1352  21173  11311 20DEC1957 20DEC1930    0   49    0
355 49 44876   915.837 10700   95 1793.876 120 210 280 528  849 1012   1531  21174  11312 21DEC1957 21DEC1930    0   49    0
356 49 35687   728.306  5622  112  960.743 119 210 276 509  720  940   1729  27019  23732 22DEC1973 22DEC1964    0   49    0
357 49 31383   640.469  3485  110  616.542 110 221 280 491  680  895   2033  27020  11314 23DEC1973 23DEC1930    0   49    0
358 49 29063   593.122  2377   87  472.976 109 220 298 473  647 1102   1746  27021  11315 24DEC1973 24DEC1930    0   49    0
359 49 27560   562.449  2138   74  416.612 147 196 301 473  581  949   1450  15334  11316 25DEC1941 25DEC1930    0   49    0
360 49 28690   585.510  2240   92  457.691 130 200 305 463  638 1164   1460  21179  11317 26DEC1957 26DEC1930    0   49    0
361 49 32719   667.735  4051  124  730.906 129 219 328 471  629 1204   1924  27024  11683 27DEC1973 27DEC1931    0   49    0
362 49 30526   622.980  3349   98  567.862 141 210 319 469  722 1006   1728  27025  11684 28DEC1973 28DEC1931    0   49    0
363 49 29548   603.020  2487  108  459.409 110 201 319 456  715 1218   1560  27026  11685 29DEC1973 29DEC1931    0   49    0
364 49 35359   721.612  4644   92  841.856 116 203 301 496  667 1388   1937  17896  11321 30DEC1948 30DEC1930    0   49    0
365 49 34865   711.531  4272   78  834.734 100 220 274 455  707 1260   1764  17897  11322 31DEC1948 31DEC1930    0   49    0
END OF FILE
```

**Statistics Summary File "WALTER.SUM"

```
 *** Table of Statistics ***
*** Pathname> /LEHIGH/WALTER/FLOW/01JAN1990/1DAY/OBS/
*** Time window> 01JAN1927 - 31DEC1975
Bin     Max        Date        Min       Date        Aver        P50
  1     2557.000 01JAN1949      69.000 01JAN1931      645.937      441.000
  2     1966.000 02JAN1973      65.000 02JAN1931      623.062      473.000
  3     1970.000 03JAN1960      78.000 03JAN1931      631.792      470.000
  4     2250.000 04JAN1960     159.000 04JAN1931      597.354      450.000
  5     1610.000 05JAN1960     156.000 05JAN1932      563.958      427.000
  6     4724.000 06JAN1949     165.000 06JAN1931      651.646      429.000
  7     3233.000 07JAN1949     110.000 07JAN1931      715.250      509.000
  8     2290.000 08JAN1949      87.000 08JAN1931      662.771      478.000
  9     1808.000 09JAN1949      83.000 09JAN1931      615.021      507.000
 10     1603.000 10JAN1946      83.000 10JAN1931      612.167      529.000
 11     1968.000 11JAN1975     110.000 11JAN1931      588.771      464.000
 12     2643.000 12JAN1975     138.000 12JAN1940      550.875      450.000
 13     2246.000 13JAN1975     129.000 13JAN1931      518.146      459.000
 14     1688.000 14JAN1975      92.000 14JAN1931      504.125      437.000
 15     1171.000 15JAN1975      74.000 15JAN1931      510.937      437.000
 16     1302.000 16JAN1962      92.000 16JAN1931      510.687      455.000
 17     1003.000 17JAN1962     110.000 17JAN1931      470.812      409.000
 .
348     2263.000 14DEC1927      76.000 14DEC1930      718.673      516.000
349     2308.000 15DEC1956      65.000 15DEC1930      691.367      479.000
350     2352.000 16DEC1956      60.000 16DEC1930      646.918      491.000
351     2156.000 17DEC1956      65.000 17DEC1930      619.224      489.000
352     1692.000 18DEC1956      78.000 18DEC1930      594.776      491.000
353     1808.000 19DEC1954     117.000 19DEC1930      591.551      617.000
354     3730.000 20DEC1957      88.000 20DEC1930      643.367      528.000
355    10700.000 21DEC1957      95.000 21DEC1930      915.837      528.000
356     5622.000 22DEC1973     112.000 22DEC1964      728.306      509.000
357     3485.000 23DEC1973     110.000 23DEC1930      640.469      491.000
358     2377.000 24DEC1973      87.000 24DEC1930      593.122      473.000
359     2138.000 25DEC1941      74.000 25DEC1930      562.449      473.000
360     2240.000 26DEC1957      92.000 26DEC1930      585.510      463.000
```

361	4051.000	27DEC1973	124.000	27DEC1931	667.735	471.000
362	3349.000	28DEC1973	98.000	28DEC1931	622.980	469.000
363	2487.000	29DEC1973	108.000	29DEC1931	603.020	456.000
364	4644.000	30DEC1948	92.000	30DEC1930	721.612	496.000
365	4272.000	31DEC1948	78.000	31DEC1930	711.531	455.000




5000
4000
3000
2000
1000
0
FLOW IN CFS
JAN
FEB
MAR
APR
MAY
JUN
JUL
AUG
SEP
OCT
NOV
DEC
JAN
3000
WALTER OBS[01JAN1927-31DEC1975] FLOW: P05
WALTER OBS[01JAN1927-31DEC1975] FLOW: P50
WALTER OBS[01JAN1927-31DEC1975] FLOW: P95

Name: *TSHIFT* *Shift time series in time*
Use: *CO TY=TSHIFT(TX,DT)*

Derive a time series TY by shifting times in TX by DT. TX may be regular or irregular. DT is specified in the form nnT, where nn is the number of time units T and T may be M for minutes, H for hours, or D for days. Units and type are preserved. An IF condition has no effect.

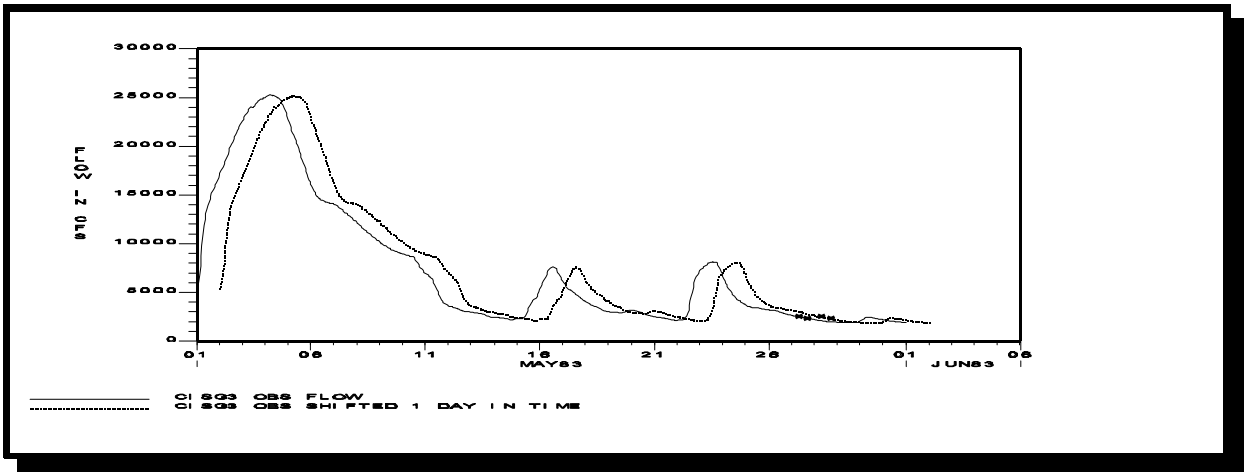
Example: The following example uses the TSHIFT function to shift a regular time series record and shift it in time by one day.

Macro:

```
MACRO TTSHIFT
!=R TTSHIFT
CL ALL
TI 01MAY1983 0100 31MAY83 2400
GET T1=testdb.dss:/SCIOTO/CISG3/FLOW/01MAY1983/1HOUR/OBS/
COMP TS=TSHIFT(T1,1D)
PUT.A TS=B=TTSHIFT,F=CAL
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
I>!R TTSHIFT
 ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                 Unit:  71;  DSS Version: 6-IA
 -----DSS--- ZREAD Unit  71; Vers.   22:  /SCIOTO/CISG3/FLOW/01MAY1983/1HOUR/OBS/
 -----DSS---ZWRITE Unit  71; Vers.    2:  /SCIOTO/TTSHIFT/FLOW/01MAY1983/1HOUR/CAL/
 -----DSS---ZWRITE Unit  71; Vers.    2:  /SCIOTO/TTSHIFT/FLOW/01JUN1983/1HOUR/CAL/
```



Name: *TSNAP* *Snap Irregular times to nearest Regular period*
Use: *CO TY=TSNAP(TX,DT,TOFF,TBACK,TFORWARD)*

Derive a regular time series from existing time series TX. TX may be regular or irregular. The new time series will be at a time interval of DT, interval offset of TOFF, backward time window of TBACK, and forward time window of TFORWARD. Parameters: DT, TOFF,TBACK, and TFORWARD are expressed as units of time "nnT", where "nn" is a number and "T" is MIN (minutes), H (hours), D (days), W (weeks), MON(months), or Y (years). An IF condition has no effect.

Example: The following example uses the TSNAP function to convert irregular time series data that is reported close to the hour to regular time series at the hour. Take notice of the two irregular time series data values that do not fall within five minutes from the hour.

Macro:

```
MACRO TTSNAP
CL ALL
TIME 01MAY1983 0100 02MAY1983 0100
GET FLOW=testdb.dss:/SCIOTO/BMRI2/FLOW/01MAY1983/IR-MONTH/OBS-I/
COM FLOWR=TSNAP(FLOW,1H,0MIN,05MIN,05MIN)
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
I>!R TTSNAP
  ALL VARIABLES HAVE BEEN CLEARED
    -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                    Unit:  71;  DSS Version: 6-IA
```

```
Variables ----    FLOW        FLOWR
 Units    ----    CFS         CFS
Time   Date
0100 01MAY1983     100.735     100.735
0104 01MAY1983     100.735       -
0200 01MAY1983       -         100.085
0204 01MAY1983     100.085       -
0300 01MAY1983       -          99.436
0304 01MAY1983      99.436       -
0400 01MAY1983       -           M
0500 01MAY1983       -          98.140
0504 01MAY1983      98.140       -
0600 01MAY1983       -           M
0700 01MAY1983       -           M
0730 01MAY1983      96.844       -
0800 01MAY1983       -          96.196
0804 01MAY1983      96.196       -
0900 01MAY1983       -          95.873
0904 01MAY1983      95.873       -
1000 01MAY1983       -           M
1100 01MAY1983       -          97.491
1104 01MAY1983      97.491       -
1200 01MAY1983       -         102.035
1204 01MAY1983     102.035       -
1300 01MAY1983       -         106.597
1304 01MAY1983     106.597       -
1400 01MAY1983       -           M
1500 01MAY1983       -         109.211
1504 01MAY1983     109.211       -
1600 01MAY1983       -         111.504
1604 01MAY1983     111.504       -
1700 01MAY1983       -         112.815
1704 01MAY1983     112.815       -
1800 01MAY1983       -           M
1900 01MAY1983       -         115.443
1904 01MAY1983     115.443       -
2000 01MAY1983       -           M
2014 01MAY1983     115.443       -
2100 01MAY1983       -         114.457
2104 01MAY1983     114.457       -
2200 01MAY1983       -           M
2300 01MAY1983       -         112.815
2304 01MAY1983     112.815       -
2400 01MAY1983       -         112.487
0004 02MAY1983     112.487       -
0100 02MAY1983       -           M
```

Name: *TTSR* *Transform time series to regular*
Use: *CO TY=TTSR(TX,DT,TOFF,FUNCT,TREAT,TYPE)*

Derive a new regular time series from existing time series TX. TX may be regular or irregular. The new time series will be at a time interval of DT, and an interval offset of TOFF. DT and TOFF must each be expressed in the form nnT, where T may be MIN, HOUR, DAY, WEEK,TRI, MON, or YEAR (e.g. 1DAY). FUNCT is one of the following:

INT - Interpolation at end of interval
MAX - Maximum over interval
MIN - Minimum over interval
AVE - Average over interval
ACC - Accumulation over interval
ITG - Integration over interval
NUM - Number valid data over interval

TREAT controls the treatment of the new generated data value when missing data occur in the interval. It must be expressed in either the form nn# or nn%. If nn# is used nn gives the number of missing data values in the interval that will be accepted in the computation of the new value. If nn% is used nn gives the percent of missing data values in the interval that will be accepted in the computation of the new value. If the criteria is met a data value will be generated, if not a missing value will occur.

TYPE provides the user control to override how the interpolation and processing of data occurs. TYPE must be one of the following: DEFAULT, INST-VAL, PER-AVER, or PER-CUM. If DEFAULT is used, processing depends on the data type stored in the

DSS data record. Otherwise processing will be performed as if the data type were as given by TYPE. Data type INST-VAL considers the data to change linearly from the previous data value to the current data value over the interval. Data type PER-AVER considers the data to be constant at the current data value over the interval. Data type PER-CUM considers the data to increase from zero (0.0) up to the current value over the interval.

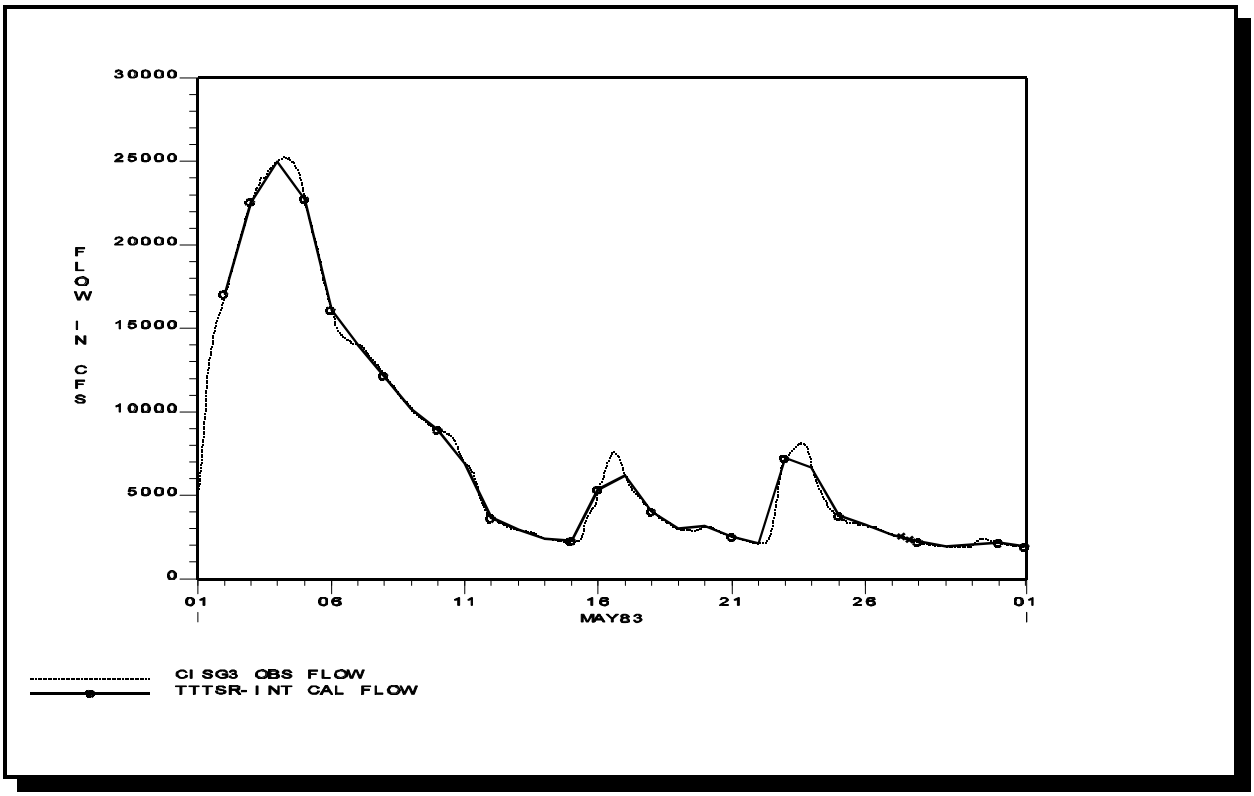
Example: The following example uses the TTSR function to derive a new regular time series from an existing time series at one hour intervals to a daily interval using six of the FUNCT parameters. Note: This example is not a real application and only serves to demonstrate the use and results of the function.

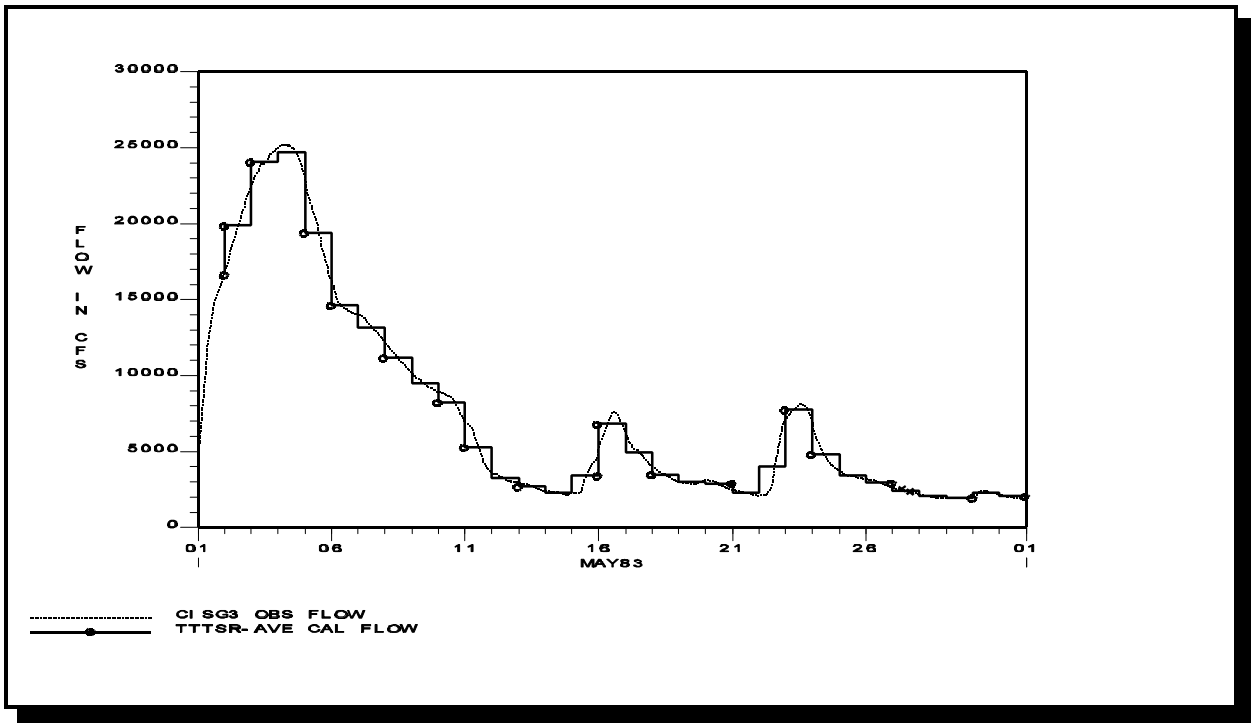
Macro:

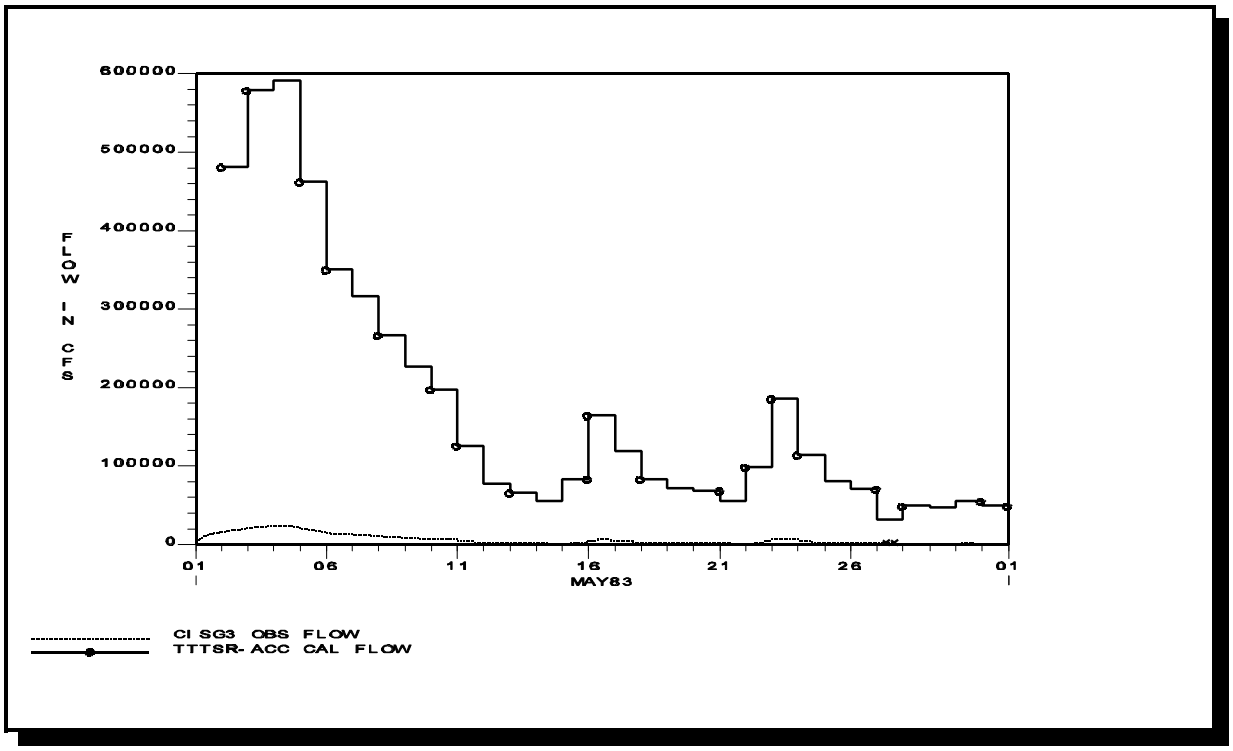
```
MACRO TTTSR
CL ALL
TI 01MAY83 0100 31MAY83 2400
GET TX=testdb.dss:/SCIOTO/CISG3/FLOW/01MAY1983/1HOUR/OBS/
COMP TS=TTSR(TX,24H,0H,INT,20#,DEFAULT)
PUT.A TS=B=TTTSR-INT,E=1DAY,F=CAL
COMP TS=TTSR(TX,24H,0H,AVE,20#,DEFAULT)
PUT.A TS=B=TTTSR-AVE,E=1DAY,F=CAL
COMP TS=TTSR(TX,24H,0H,ACC,20#,DEFAULT)
PUT.A TS=B=TTTSR-ACC,E=1DAY,F=CAL
COMP TS=TTSR(TX,24H,0H,ITG,20#,DEFAULT)
PUT.A TS=B=TTTSR-ITG,E=1DAY,F=CAL
COMP TS=TTSR(TX,24H,0H,MAX,20#,DEFAULT)
PUT.A TS=B=TTTSR-MAX,E=1DAY,F=CAL
COMP TS=TTSR(TX,24H,0H,MIN,20#,DEFAULT)
PUT.A TS=B=TTTSR-MIN,E=1DAY,F=CAL
$CONTINUE INCASE OF ERROR
ENDMACRO
```

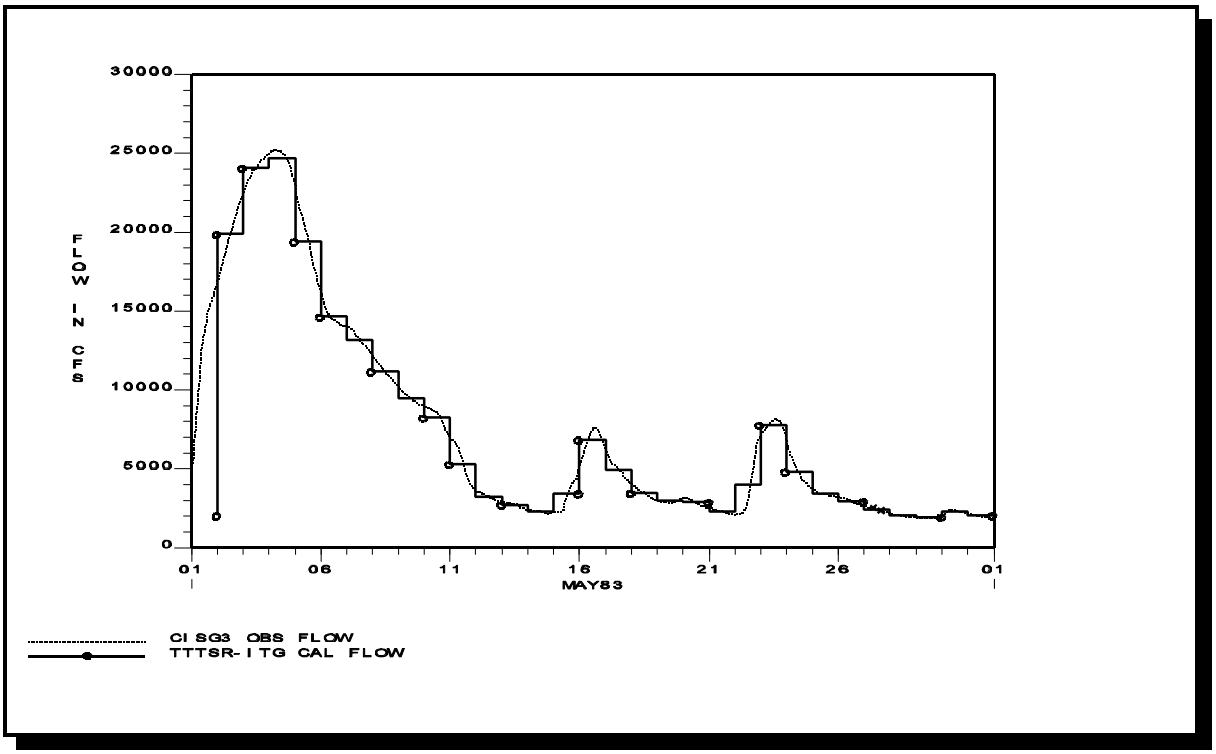
Execution:

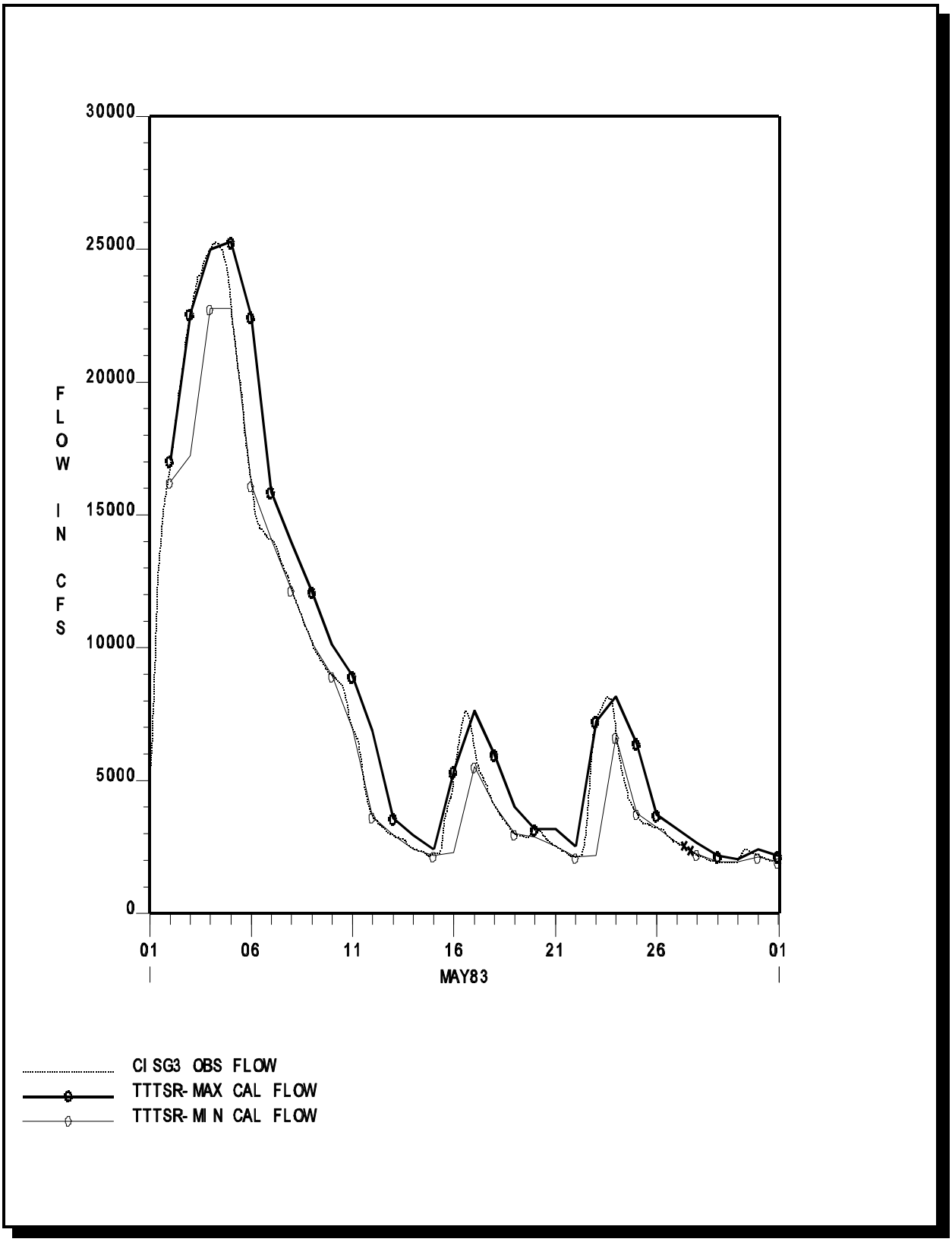
```
I>!R TTTSR
 ALL VARIABLES HAVE BEEN CLEARED
   -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                      Unit:  71;  DSS Version: 6-IA
-----DSS--- ZREAD Unit  71; Vers.   22:  /SCIOTO/CISG3/FLOW/01MAY1983/1HOUR/OBS/
-----DSS---ZWRITE Unit  71; Vers.    4:  /SCIOTO/TTTSR-INT/FLOW/01JAN1983/1DAY/CAL/
WARNING -- EXISTING VARIABLE RE-DEFINED
-----DSS---ZWRITE Unit  71; Vers.    4:  /SCIOTO/TTTSR-AVE/FLOW/01JAN1983/1DAY/CAL/
WARNING -- EXISTING VARIABLE RE-DEFINED
-----DSS---ZWRITE Unit  71; Vers.    5:  /SCIOTO/TTTSR-ACC/FLOW/01JAN1983/1DAY/CAL/
WARNING -- EXISTING VARIABLE RE-DEFINED
-----DSS---ZWRITE Unit  71; Vers.    4:  /SCIOTO/TTTSR-ITG/FLOW/01JAN1983/1DAY/CAL/
WARNING -- EXISTING VARIABLE RE-DEFINED
-----DSS---ZWRITE Unit  71; Vers.    4:  /SCIOTO/TTTSR-MAX/FLOW/01JAN1983/1DAY/CAL/
WARNING -- EXISTING VARIABLE RE-DEFINED
-----DSS---ZWRITE Unit  71; Vers.    4:  /SCIOTO/TTTSR-MIN/FLOW/01JAN1983/1DAY/CAL/
```


30000
25000
20000
15000
10000
5000
0
FLOW IN CFS
01
06
11
16
21
26
01
MAY83
CI SG3 OBS FLOW
TTTSR-INT CAL FLOW


30000
25000
20000
15000
10000
5000
0
FLOW IN CFS
01
06
11
16
21
26
01
MAY83
CI SG3 OBS FLOW
TTTSR-AVE CAL FLOW


600000
500000
400000
300000
200000
100000
0
FLOW IN CFS
01
06
11
16
21
26
01
MAY83
CI SG3 OBS FLOW
TTTSR-ACC CAL FLOW


30000
25000
20000
15000
10000
5000
0
FLOW IN CFS
01
06
11
16
21
26
01
MAY83
CI SG3 OBS FLOW
TTTSR-ITG CAL FLOW


30000
25000
20000
15000
10000
5000
0
FLOW IN CFS
01
06
11
16
21
26
01
MAY83
CI SG3 OBS FLOW
TTTSR-MAX CAL FLOW
TTTSR-MI N CAL FLOW

Name: *TTSI* *Transform time series to irregular*
Use: *CO TY=TTSI(TX,TZ,FUNCT,TREAT,TYPE)*

Derive a new irregular time series from existing time series TX. TX may be regular or irregular. TZ is an existing time series at the desired new spacing of TY. All other parameters are as shown for the function TTSR.

Example: The following example uses the TTSI function to take hourly regular data and generate an irregular time series based on an irregular time series pattern record. Note: This example is not a real application and only serves to demonstrate the use and results of the function.

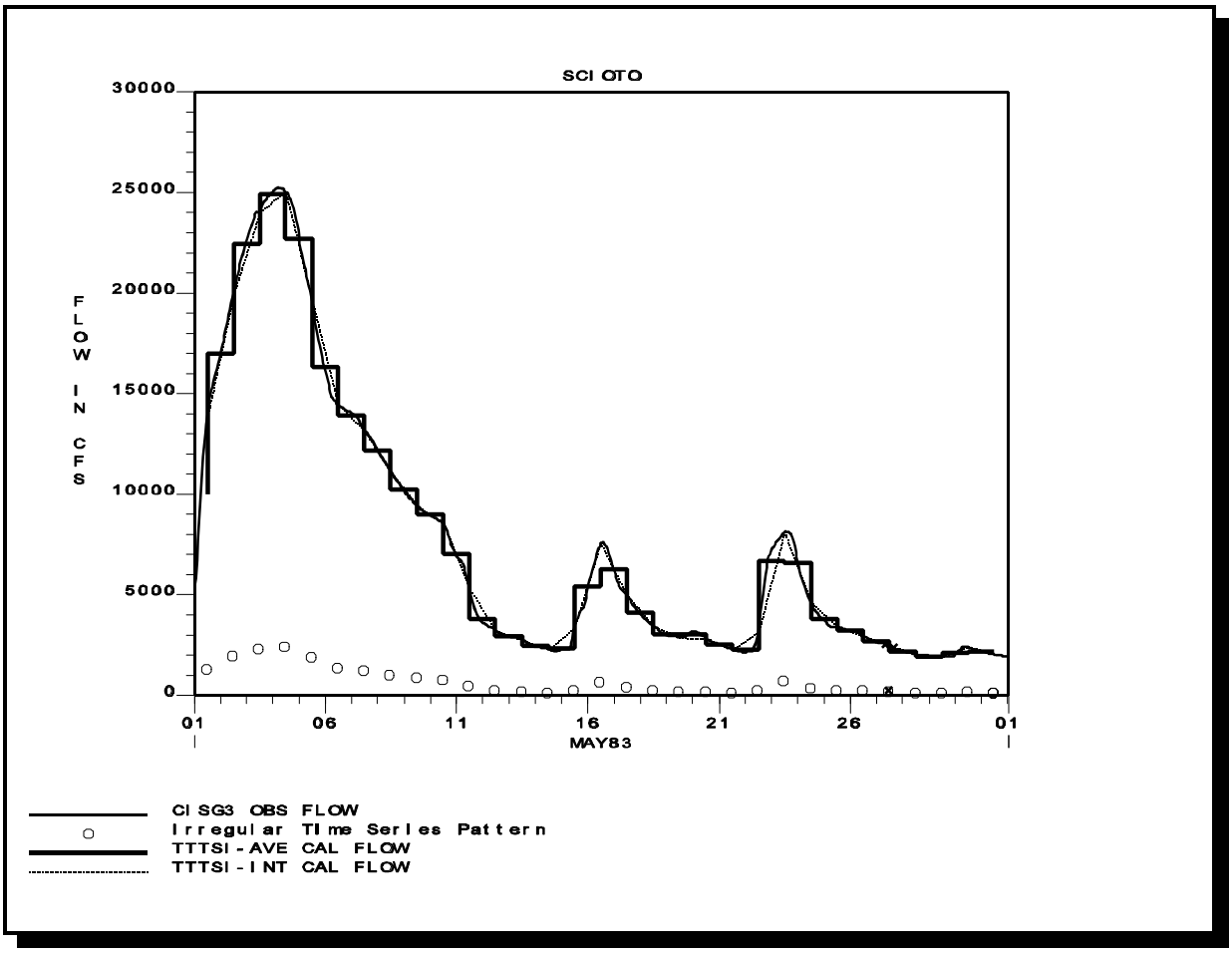
Macro:

```
MACRO TTTSI
!R TTTSI5 AVE
!R TTTSI5 INT
ENDMACRO

MACRO TTTSI5 $FUN
CL ALL
TI 01MAY1983 0100 31MAY83 2400
GET TX=testdb.dss:/SCIOTO/CISG3/FLOW/01MAY1983/1HOUR/OBS/
GET TZ=/TZ/TTTS-PATTERN/FLOW/01MAY1983/IR-MONTH/NEW/
COMP TS=TTSI(TX,TZ,$FUN,12#,DEFAULT)
PUT.A TS=B=TTTSI-$FUN,E=IR-MONTH,F=CAL
$CONTINUE INCASE OF ERROR
ENDMACRO
```

Execution:

```
I>!R TTTSI
  ALL VARIABLES HAVE BEEN CLEARED
    -----DSS---ZOPEN:  Existing File Opened,  File: TESTDB.DSS
                    Unit:  71;  DSS Version: 6-IA
 -----DSS--- ZREAD Unit  71; Vers.   22:  /SCIOTO/CISG3/FLOW/01MAY1983/1HOUR/OBS/
 -----DSS---ZWRITE Unit  71; Vers.    4:  /TZ/TTTSI-AVE/FLOW/01MAY1983/IR-MONTH/CAL/
  ALL VARIABLES HAVE BEEN CLEARED
 -----DSS--- ZREAD Unit  71; Vers.   22:  /SCIOTO/CISG3/FLOW/01MAY1983/1HOUR/OBS/
 -----DSS---ZWRITE Unit  71; Vers.    3:  /TZ/TTTSI-INT/FLOW/01MAY1983/IR-MONTH/CAL/
```



TIME	DATE	CISG3 OBS FLOW CFS INST-VAL	TTTSI-AVE CAL FLOW CFS PER-AVER	TTTSI-INT CAL FLOW INST-VAL
0100	01MAY1983	5432.000	-	-
0200	01MAY1983	5840.000	-	-
0300	01MAY1983	6710.000	-	-
0400	01MAY1983	7747.000	-	-
0500	01MAY1983	8783.000	-	-
0600	01MAY1983	9819.000	-	-
0700	01MAY1983	10855.000	-	-
0800	01MAY1983	11892.000	-	-
0900	01MAY1983	12497.000	-	-
1000	01MAY1983	13077.000	-	-
1100	01MAY1983	13526.000	-	-
1200	01MAY1983	13946.000	10039.550	13946.000
1300	01MAY1983	14298.000	-	-
1400	01MAY1983	14617.000	-	-
1500	01MAY1983	14903.000	-	-
1600	01MAY1983	15191.000	-	-
1700	01MAY1983	15408.000	-	-
1800	01MAY1983	15626.000	-	-
1900	01MAY1983	15845.000	-	-
2000	01MAY1983	16064.000	-	-
2100	01MAY1983	16285.000	-	-
2200	01MAY1983	16544.000	-	-
2300	01MAY1983	16767.000	-	-
2400	01MAY1983	17065.000	-	-
0100	02MAY1983	17253.000	-	-
0200	02MAY1983	17479.000	-	-
0300	02MAY1983	17705.000	-	-
0400	02MAY1983	17933.000	-	-
0500	02MAY1983	18200.000	-	-
0600	02MAY1983	18436.000	-	-
0700	02MAY1983	18674.000	-	-
0800	02MAY1983	18952.000	-	-
0900	02MAY1983	19231.000	-	-
1000	02MAY1983	19552.000	-	-
1100	02MAY1983	19794.000	-	-
1200	02MAY1983	20037.000	17033.900	20037.000