WATER SURFACE PRESSURE GRADIENT

HYDRAULIC ANALYSIS COMPUTER PROGRAM F0515P DOCUMENTATION

LOS ANGELES COUNTY FLOOD CONTROL DISTRICT

APRIL 1979

FORTRAN PROGRAMS F0515A AND F0515B

Edit and Computation Programs for the Water Surface and Pressure Gradient Hydraulic Analysis System.

PROGRAMMER'S GUIDE

May 1979

Prepared By: Richard A. Schaeffer
Flood Control Data Center

and

Saman J. Shahin, and
Garvin J. Pederson
Design Systems and Standards Group

Los Angeles County Flood Control District

2250 Alcazar Street, Los Angeles
Telephone: (213) 226-4111

P.O. Box 2418 Terminal Annex
Los Angeles, CA 90054

FORWARD

This program was written for use by the Los Angeles County Flood Control District or by its Contractors on District projects.

Any other persons or agenecies utilizing these programs should understand that the Los Angeles County Flood Control District makes no warranties (expressed or implied) concerning its accuracy, completeness, reliability, usability, or suitability and the District shall be under no liability for any use made thereof.

This program is the property of the District. Therefore, any recipient further agrees not to assert any proprietary rights to this program or modification of or to represent it to anyone as other than a District program.

ACKNOWLEDGEMENT

A special thank you is merited by the Design Division Stenographers Rita Lopez and Deborah Mount whose pleasantness, skill, understanding and tolerance were largely responsible for the completion of this report. Appreciation is also given to the Design and Water Conservation Divisions for allowing the time to complete the documentation for this system.

TABLE OF CONTENTS

* Input card layouts and report layouts are discussed in the Water Surface Profile User's Guide.

** Subroutine flowcharts are presented in the order shown for Main Routine under Computation Program in the Table of Contents. Functions are the only routines not flowcharted.

F0515 SYSTEM DESCRIPTION

The Water Surface Profile System has been designed to caculate depths in a channel given the channel definition and flow rates. Several flow rates can be inputted for a single job (the channel definition and elements remaining the same) to determine the performance of the channel under various flow rate circumstances.

The system is composed of two programs. F0515A is the editing program which insures the validity and sequence of the input data. F0515B is the computation program which manipulates the data and produces the final water surface profile.

The programs were written in the FORTRAN language for the IBM 370 system. The programs were written by Richard A. Schaeffer and designed by Garvin Pederson and Saman Shahin of the Los Angeles County Flood Control District. Any problems or questions regarding the system should be referred to the above at (213) 226-4085.

1.1)

The editing program reads the input card data and checks to see if the data is valid. When submitting the input the cards must be in the following order:

1. The trace swith card.
2. The title cards.
3. The element cards.
4. The channel definition cards.
5. The cross section points cards.

If the input is not in the proper sequence then error messages indicating this will be printed.

1.2)

The trace switch card has either a zero or a one in the first column. If it has a one then the element file will be printed at the end of the editing phase. Since the file is printed in an unformatted manner the data will have to be interpreted from the record layouts according to the record code (the fourth position of each record).

1.3)

The title and element cards are then read and placed on a temporary file for later processing. The end of the element cards is detected upon the reading of the system headworks card. The element input data always begins with the one system outlet card and proceeds in an upstream direction ending with the one system headworks card.

1.4)

The channel definition cards are then read and edited for valid data according to the channel type they describe. There must be at least one channel definition card per run as there must be at least one channel description per element. If there are more than one they may be in any order. The channel definition data is printed along with the appropriate error messages if any of the data is in error. The channel definitions that are valid are written to a temporary file according to the section number they describe, this will enable several elements to access the same channel definition data by just supplying the section number containing the description.

1.5)

The cross section points cards are read last. If there are no irregular channels being used it is not necessary to have any cross section points cards. Since there can be from 3 to 99 coordinates to describe a section

EDIT FILES AND COMMON BLOCKS

1.1)

The following is a description of the use of the files used in the editing phase. The list is in order by the number of the particular file being described:

1) The element file - this is an output file which contains the combined data from the element cards and channel definition cards to produce the channel description of the various elements used. The file is random access and is in element number order from the system outlet to the system headworks, there can be a maximum of 200 element records; record 201 is for title data and record 202 is for the number of elements being processed.

4) The cross section points file - this is an output file which contains the coordinate data and pier base data for irregular sections. It is random access and is in section number sequence, there can be a maximum of 200 records.

5) The input card file - this file contains the following data in order: the trace switch card, the title data, the element type data, the channel definition data and the irregular section coordinate data.

6) The output print file - this is the printing of the input data along with any editing errors which may have occured.

8) The temporary element file - this file holds the title and element card data until after the channel definition data and irregular cross section point data is processed, then the element data is processed. It is in the sequential order of the inputted title and element card data and there can be a maximum of 202 records (3 title cards and 200 element cards).

9) The temporary channel definition file - this file contains the data from the channel definition cards and is used whenever an element card references it by section number to be used as part of the element file data. It is random access in section number order and can contain a maximum of 200 records.

1.2)

The following is a description of the use of the common blocks used in the editing phase:

FILEBK) This contains the key data for reading and writing the random access disk files.

MAIN ROUTINE

1.1)

The MAIN routine defines all the files to be used and controls all the editing processes of the element cards, channel definition cards, and cross section point cards. The input data must be in the following order:

1) Trace switch card,
2) Title cards,
3) Element cards (from System Outlet to System Headworks),
4) Channel definition cards,
5) Cross Section points cards.

The trace switch card has a 0 in column 1 for regular printing or a 1 in column 1 to print the element file at the end of the editing. The remaining cards are checked for valid card codes indicating the card type before any editing processing is done on the card.

1.2)

The title and element cards are read and stored on a temporary disk file for later processing, the system headworks card signifies the end of this procedure. If any channel definitions or cross section points are encountered before the system headworks card an error is detected and a message is printed.

1.3)

The channel definition cards are processed next and at least one card must be submitted. If there are no channel definition cards an error is detected and a message is printed. If any other card type is encountered between the system headworks and the channel definitions it is also an error; however, if it is a cross section points card it will be edit checked regardless of its position. Once the channel definition card is recognized it will be edited according to the channel type specified on the card as follows:

1) Open trapezoidal channel,
2) Open rectangular channel,
3) Closed trapezoidal or rectangular channel,
4) Pipe channel,
5) Open irregular channel,
6) Closed irregular channel,

if the channel type is not one of the above no editing can be performed and an error is detected on the channel type.

CDTYP2

1.1)

Subroutine CDTYP2 edits the required channel definition data for an open rectangular channel section.

1.2)

The following edit checks are made:

1) Section number must be between 1 and 200,
2) If the number of piers is given it must be between 1 and 10,
   a) Pier width must be greater than zero,
   b) Total pier width must be less than channel width,
3) Channel height must be greater than zero,
4) Channel width must be greater than zero.

If there are no piers then pier width is not checked.

1.3)

The left and right slopes are set to zero.

1.4)

If there were any errors the appropriate error messages are written, otherwise, the channel definition data is written to a temporary disk file according to the section number and is used later is the editing of the element data.

CDTYP4

1.1)

Subroutine CDTYP4 edits the required channel definition data for pipe channel sections.

1.2)

The following edit checks are made:

1) Section number must be between 1 and 200,
2) The pipe diameter must be greater than zero.

1.3)

If there were any errors the appropriate error messages are written, otherwise, the channel definition data is written to a temporary disk file according to the section number and is used later in the editing of the element data.

PTREAD

1.1)

Subroutine PTREAD is used to edit and print the cross section points data for irregular channels. A table is set up to handle a maximum of 99 points per irregular section.

1.2)

The first points card for a section is read and it is checked to see how many points are to be read to describe the section. The number of points to describe the section must be between 3 and 99 or it is invalid. The five points from the card (if the number of points is greater than 4) or the number of points decribing the channel (if it is less than 5) are placed into the table in the order they are read.

1.3)

If the number of points is greater than 5 the successive cards (5 points per card) are read until all the points describing the channel are read and they are placed into the table. On all these successive cards describing the same channel the number of points must be blank to distinguish them from the first card describing the next irregular section. If a card is encountered which is not a cross section points card or one which has a number of points indicated on it before all the indicated points for the current section are read, the data is assumed to be invalid and an error message is printed.

1.4)

Once all the cross section points are read for a section they are printed out. If no errors were encountered in reading the points then subroutine IRROPE or IRRCLO (depending if the section is open or closed) is called to check the validity of the order of the points.

1.5)

When the order of the points has been checked control is transferred back to see if there are any more cross section points to be read. If there are more points the processing repeats itself, otherwise, control is returned to the MAIN routine for further processing.

IRRCLO

1.1)

Subroutine IRRCLO completes the editing of the irregular cross section points for a closed channel. The points are adjusted to the X-axis and the pier base values (if they exist) are also adjusted to the X-axis.

1.2)

The cross section points are read in and the X coordinate values are tested to make sure they are in a counter-clockwise order from the minimum X value. X coordinates must continually increase until the maximum X is reached and then continually decrease to the last point describing the channel. As the points are processed the minimum and maximum Y coordinate values are selected (minimum Y if more than one exists is the one with the greatest X coordinate value and maximum Y if more than one exists is the one with the smallest X coordinate value).

1.3)

The cross section points are then rearranged so the point where maximum Y exists is the first point and the remaining points follow in the same counter-clockwise order.

1.4)

The minimum Y value is then subtracted from all the Y coordinates so the bottom of the channel will sit on the X-axis so if there is no depth it will equal zero. If there are any piers in the section then minimum Y is subtracted from their pier bases so they will still coincide with the section after it has been adjusted.

1.5)

If any errors were encountered the appropriate messages are written. Otherwise, the adjusted cross section points and pier values are written to the cross section points file according to the section number. This file is then used throughout the water surface profile computation to compute pressure, area, and wetted perimeter in this irregular section.

EDTELM

1.1)

Subroutine EDTELM controls the editing of the element cards and title cards. Title cards are processed first in subroutine TITLE and then the element cards are read from the system outlet to the system headworks. The data given on the element card is for the U/S end of an element, the data at the D/S end is obtained from the previous element.

1.2)

The edit checks made here are only on the card codes to make sure the code specifies a legitimate element type, then processing is done according to that element type. When the system headworks card is encountered and processed then the editing phase is complete. If there were any errors then the water surface profile will not be computed until the errors are corrected, otherwise, the computation of the water surface profile will begin.

1.3)

The legitimate card codes for the elements are as follows:

SO = SYSTEM OUTLET (only one per run), see READSO for editing,
R = REACH, see READR for editing,
JX = JUNCTION, see READJS for editing,
TS = TRANSITION, see READTS for editing,
BE = BRIDGE ENTRANCE, see READBE for editing,
BX = BRIDGE EXIT, see READBX for editing,
WE = WALL ENTRANCE, see READBE for editing,
WX = WALL EXIT, see READBX for editing,
SH = SYSTEM HEADWORKS (only one per run), see RDHDWK for editing.

1.1)

Subroutine READSO does the editing of the system outlet card. This should be the first card following the title cards, if it is not an error will have occurred and processing of the outlet will terminate.

1.2)

The system outlet card data is printed as it was read. The card code is checked for either an SO or an So, if it is not an error has occurred and outlet processing is terminated with the appropriate message.

1.3)

Edit checks on the outlet data which result in errors are:

1) If the Invert Elevation is blank,
2) If the Section Number is not between 1 and 200,
3) If the Channel Type is not between 1 and 6,
4) If there is no data on the channel definition file for the section number indicated,
5) If the channel is irregular and there is no cross section points data for the section number indicated.

If the Invert Elevation is given and it is not greater than zero a warning message is printed.

1.4)

If any editing errors existed the appropriate error messages are written and control is returned to subroutine EDTELM. If there were no edit errors the data from the system outlet card and the corresponding data from the channel definition file are combined into the system outlet record. The outlet record is written to the first record of the element file with a record code of 1 to indicate a system outlet and control is given back to subroutine EDTELM to process the remaining element cards.

Note: there can only be one system outlet record per run, otherwise, it is an error.

READJS

1.1)

Subroutine READJS does the editing of the junction cards. The data from the junction card is printed as it is read. The element file key is incremented by one so this would be next record to be written to the element file.

1.2)

The following edit check are made on the junction card data to see if it is valid:

1) The Invert for the U/S end must be given,
2) The Station given for the U/S end of the junction must be greater than the station given at the previous element,
3) The Section Number for the U/S end must be between 1 and 200,
4) There must be data on the channel definition file for the section number given at the U/S end,
5) If the channel type at the U/S end is irregular there must be data on the cross section points file for that section number,
6) If there is a first lateral the Section Number for the first lateral must be between 1 and 200,
7) If there is a first lateral there must be data on the channel definition file for the section number of the first lateral,
8) If the first lateral is irregular there must be data of the cross section points file for that section number,
9) If there is a first lateral indicated the Q for the first lateral must be given,
10) If there is a first lateral indicated the Angle of the first lateral must be given,
11) If there is a second lateral the Section Number for the lateral must be between 1 and 200,
12) If there is a second lateral there must be data on the channel definition file for that section number,
13) If the second lateral is irregular there must be data on the cross section points file for that section,
14) If there is a second lateral indicated the Q for the second lateral must be given,
15) If there is a second lateral indicated the Angle of the second lateral must be given.

If the Invert given for the U/S end is less than the invert given in the previous element a warning message is written. If the Inverts for existing laterals are not given they are computed as the average of the inverts in the U/S and D/S ends. If Manning's Factor is not given it is assumed to be .014.

1.3)

If any editing errors existed the appropriate error messages are written and control is returned to subroutine EDTELM. If there were no errors the

READTS

1.1)

Subroutine READTS does the editing of the transition cards. The data from the transition card is printed as it is read. The element file key is incremented by one so this would be the next record to be written to the element file.

1.2)

The following edit checks are made on the transition card data to see if it is valid:

1) The Invert for the U/S end must be given,
2) The Station given for the U/S end must be greater than the station at the U/S end of the previous element,
3) The Section Number for the U/S end must be between 1 and 200,
4) There must be data on the channel definition file for the section number given,
5) If the channel is irregular there must be data on the cross section points file for the section number given.

If the Invert given for the U/S end is less than the invert given in the previous element a warning message is written. If Manning's Factor is not given it is assumed to be .014.

1.3)

If any editing errors existed the appropriate error messages are written and control is returned to subroutine EDTELM. If there were no errors the data from the transition card, the corresponding data from the previous element, and the data from the channel definition file are combined into the transition record. This record is then written to the element file with a record code of 7 to indicate a transition and control is given back to subroutine EDTELM to process the remaining element cards.

NOTE: Transitions can be submitted as junctions with no laterals if calculations are desired by the momentum and pressure methods instead of the adapted Bernoulli's formula method.

READBX

1.1)

Subroutine READBX does the editing of the bridge exit and wall exit cards. The data from the exit card is printed as it is read. The element file key is incremented by one so this would be the next record to be written to the element file.

1.2)

The following edit checks are made on the exit card data to see if it is valid:

1) If the Station is given for the exit it must be the same as the station given in the previous element,
2) If the Invert is given for the exit it must be the same as the invert given in the previous element,
3) There must not have been any pier data given in the previous element,
4) The Section Number given must be between 1 and 200,
5) There must be data on the channel definition file for the section number given,
6) If the channel is irregular there must be data on the cross section file for the section number given,
7) If this is a bridge exit there must be Pier Data given for the channel,
8) If this is a bridge exit the channel definition for the exit must be the same as the channel definition data for the previous element except for channel height and pier data,
9) If this is a bridge exit the section cannot be a pipe.

If the Station or Invert is not given it is set equal to its respective value given in the previous element.

1.3)

If any editing errors existed the appropriate error messages are written and control is returned to subroutine EDTELM. If there were no errors the data from the exit card, the corresponding data from the previous element, and the data from the channel definition file are combined into the exit record. This record is then written to the element file with a record code of 5 (a bridge exit) or a record code of 9 (a wall exit) and control is returned to subroutine EDTELM to process the remaining element cards.

ERRI, WARN, and PRINTO

1.1)

Subroutine ERRI prints the error message according to the type of error encountered while editing the element card data. This is done by having the editing subroutine send a value to indicate the number of the message which should be printed and a value to indicate the specific variable which is in error. If more than 20 errors have occurred processing is stopped and the existing errors to this point must be corrected and the job resubmitted.

2.1)

Subroutine WARN prints the warning messages for the editing of the element cards. This is done by the editing subroutine sending a value to indicate the number of the error message which is to be printed.

3.1)

Subroutine PRINTO prints out the element file after it has been created so you can see if each element contains the correct data. This is only done if the trace switch equals 1. The trace switch is set by the first input card.

ERROR MESSAGES IN SEQUENCE CHECKING THRU CHANNEL DEFINITION DATA

1) NO SYSTEM HEADWORKS CARD - CANNOT TELL WHERE THE START OF CHANNEL DEFINITION DATA IS - NO PROCESSING

There must be a system headworks card at the end of the element cards just preceeding the channel definition cards. Check input data.

2) CHANNEL DEFINITION DATA (CD) DID NOT FOLLOW THE SYSTEMS HEADWORKS CARD - CONTINUING TO LOOK FOR CD OR PTS

There must be at least one CD card following the system headworks card, and all CD cards follow the system headworks and come before the cross section points (PTS) cards. Check input data.

3) NO CHANNEL DEFINITION (CD) OR CROSS SECTION POINT CARDS (PTS) WERE RECOGNIZED - CHECK DATA

There must be at least one channel definition card following the system headworks card. Check input data card code columns.

4) NO CHANNEL DEFINITION CARDS BEFORE CROSS SECTION POINT CARDS - CHECK DATA

Check order of input cards. Element cards ending with system headworks must be followed by at least one channel definition card. Cross section point cards follow the channel definition cards.

5) INVALID CHANNEL TYPE ON CHANNEL DEFINITION CARD ITYPE = X SECT = XXX

ITYPE is the channel type requested and SECT is the section number the channel type is specified to define. Channel type must be a number between 1 and 6. Check and correct this CD card.

6) NO CROSS SECTION POINTS ENCOUNTERED - ASSUME NO IRREGULAR CHANNELS

No irregular channels or cross section points are indicated for this problem. This is a warning message. Processing will continue.

7) INVALID CARD CODE ENCOUNTERED WHILE PROCESSING CD AND PTS CARDS CODE = XXX

After the first CD card a card was found which did not have a code of CD or PTS. CODE indicates the invalid card code which should be corrected or placed in the correct order.

ERROR MESSAGES IN CHANNEL DEFINITION PROCESSING

1) SECTION NUMBER INVALID OR MISSING, DATA CANNOT BE WRITTEN TO THE OUTPUT FILE

There was an invalid section number on a CD or PTS card. Section number must be between 1 and 200. Correct invalid data.

2) INVALID VALUE FOR THE NUMBER OF PIERS - MUST BE BETWEEN 0 AND 10 IF GIVEN

The number of piers on the CD card is invalid. Correct the invalid data.

3) AVERAGE WIDTH OF PIERS IS INVALID OR NOT GIVEN WHEN THERE IS A VALUE FOR NUMBER OF PIERS IN THE CHANNEL

When number of piers is given there must be a value for average width of piers. Correct whichever field is wrong.

4) CHANNEL HEIGHT IS INVALID OR IS NOT GIVEN

Correct the height data in the channel definition.

5) CHANNEL DIAMETER IS INVALID OR IS NOT GIVEN

Correct the width data in the channel definition.

6) CHANNEL WIDTH IS INVALID OR IS NOT GIVEN

Correct the width data in the channel definition.

7) THERE IS A DIFFERENCE BETWEEN THE NO. OF PIERS AND THE NUMBER OF VALUES FOR PIER DEPTHS

If fewer depths are given for piers in an irregular section than the number of piers indicated, then the remaining pier base values must be added even if they are 0.0. If more depths are given only the amount up to the number of piers declared will be considered.

ERROR MESSAGES IN CROSS SECTION POINT PROCESSING

1) ENCOUNTERED A POINT WHERE X = 0 AND Y = 0 BEFORE ALL THE INDICATED POINTS WERE PROCESSED - ASSUMING ERROR

   Only the first coordinate of the cross section points can be 0, 0 - otherwise the program cannot distinguish between blanks and zeroes. If point desired is 0, 0 use .01, .01 for approximate data. Correct invalid data.

2) THE CROSS SECTION POINTS ARE OUT OF SEQUENCE FOR AN IRREGULAR OPEN SECTION - MUST BE COUNTER - CLOCKWISE FROM MINIMUM X

   Check the sequence of points on the cross section point cards for the data which is out of order.

3) THE CROSS SECTION POINTS ARE OUT OF SEQUENCE FOR AN IRREGULAR SECTION - MUST BE COUNTER - CLOCKWISE FROM MINIMUM X

   Check the sequence of points on the cross section point cards for the data which is out of order. When maximum X is reached the following X values must continually decrease.

4) MAXIMUM Y IS NOT AT EITHER SIDE OF AN OPEN IRREGULAR CHANNEL ASSUMED BAD DATA AND PROCESSING IS STOPPED

   For some reason maximum Y was not at the end of an open irregular channel. Check input data and correct if wrong, otherwise see programmer.

9) INVALID SECTION NUMBER ON ELEMENT CARD

The section number given for this element is not between 1 and 200. Correct input with correct channel number reference.

10) SECTION NUMBER HAD NO DATA FOR CHANNEL DEFINITION

The section number on the element card refers to a section that was not defined or that was labeled as being in error in editing the channel definition cards. Check the channel number and the results from the channel definition editing.

11) SECTION NUMBER HAD NO DATA FOR CROSS SECTION

Same as Message No. 10 only for cross section data instead of channel definition data.

12) THE CHANNEL DEFINITION REFERENCED DID NOT CONTAIN THE REQUIRED DATA TO BE USED IN THIS ELEMENT

There is a conflict between the data in the channel definition used to describe this element and the type of element being described. Check the restrictions for this element and make sure the channel definition selected has applicable data.

13) THE PREVIOUS SECTION OR CHANNEL DEFINITION DID NOT COINCIDE WITH THE DATA UTILIZED IN THIS ELEMENT

There is a conflict between the data in the channel defintions of the previous element and the current element being described. Check the restrictions for the element type and the channel definition data used.

1.1)

The water surface profile program computes and prints the composite profile of a designed channel. The first card read is the trace switch card which controls program printing (See main routine). The Q value is supplied on a card and then beginning at the system headworks the Q value and controlling depths (critical, normal and maximum open flow depths) are computed for each element and written to the element file (the file created in the editing program containing all the elements in the channel design).

1.2)

The downstream profile is computed next. The downstream processing is done from the system headworks to the system outlet. Inside the elements processing is done from the upstream end to the downstream end. The depth in the upstream end is set to the lesser value of the critical depth in the upstream end or the depth from the downstream end of the previous element; unless the depth from the downstream end of the previous element equals zero, then the critical depth in the upstream end is assumed. The data at the upstream end is computed and written to the upstream profile. The depth in the downstream end of the element is computed by formulae which alter the depth according to the element type. The data at the downstream end is then computed and written to the downstream profile.

1.3)

When downstream processing is complete the upstream processing begins. The upstream processing is done from the system outlet to the system headworks. Inside the elements processing is done from the downstream end to the upstream end. The depth in the downstream end is set to the greater value of the critical depth at the downstream end or the depth at the upstream end of the previous element. The data at the downstream end is computed and written to the upstream profile. The depth in the upstream end is computed by formulae according to the element type. The data at the upstream end is then computed and written to the upstream profile.

1.4)

After the upstream procesing is completed the upstream and downstream profiles are compared and a composite profile is established. The profiles of the individual elements are compared by force values. If the forces at both ends on the upstream profile are greater than the forces at both ends of the downstream profile in any element then the upstream profile controls. If the forces at both ends on the downstream profile are greater than the forces at both ends on the upstream profile in any element then the downstream profile controls. If the upstream force is greater or equal to the downstream force at only one end of the element then a hydraulic jump occurs and the upstream profile controls at the end where the upstream force is greater and the downstream profile controls at the end where the downstream force is greater.

VARIABLE DEFINITIONS USED IN WRITE-UPS

1.1)

In the write-ups preceeding the various subroutines abbreviations are used for certain variables. Below is an explanation of these abbreviations.

A:	Cross sectional area of flow.
BP:	Average pier width.
D:	Depth of flow in a section.
DC:	Critical depth of flow in a section.
DELL:	Length of an element or distance between stations.
DH:	Maximum open flow depth in a section: Ten feet above channel walls in an open channel, the height of the channel in a closed or covered channel (start of pressure flow).
DIAM:	Diameter of a pipe section.
DN:	Normal depth of flow in a section.
DP:	Depth of flow in the previous element: From the upstream end of the previous element in upstream processing and from the downstream end of the previous element in downstream processing. The upstream end is closest to the headworks element and the downstream end is closest to the outlet element.
E:	Energy value for the flow in a section.
EGL:	Energy grade line for the flow in a section.
F:	Force of flow in a section.
G:	Gravity, 32.2 feet per second.
H:	The channel height of a section.
HV:	Velocity head for the flow in a section.
INV:	Invert elevation in a section.
M:	Momentum of flow in a section.
P:	Pressure of flow in a section.
Q:	Flow rate in a section or lateral.
SC:	The critical slope in a reach.
SF:	The friction slope in an element.
SO:	The channel slope in a reach.
V:	Velocity of flow in a section.
W:	Base width of a channel section.
WP:	The wetted perimeter of a section.
W.S.:	Water surface elevation in a section.
ZL:	Left side slope of a trapezoidal section.
ZR:	Right side slope of a trapezoidal section.

These variables may contain one of two suffixes to identify their locations as follows:

1: Identifies the variable at the upstream end of the element.
2: Identifies the variable at the downstream end of the element.

For example:

DC1 is the critical depth at the upstream end of the element.
SF2 is the friction slope at the downstream end of the element.

In all cases except DP the variable being used will refer to the upstream or downstream end of the element being processed. It should also be noted that throughout the documentation:

U/S will always stand for upstream, and
D/S will always stand for downstream,

whether referring to a particular end of an element or referring to the processing direction.

COMPUTATION FILES AND COMMON BLOCKS

1.1)

The following is a description of the use of the files used in the computation phase. The list is in order by the number of the particular file being described:

1) The element file - this is an input output file which contains the descriptions of the elements to be processed. The file is random access and is in element number order from the system outlet to the system headworks, there can be a maximum of 200 element records; record 201 is for the title data and record 202 is for the number of elements being processed.

2) The upstream file - this is an input/output file which contains the data computed in the upstream direction for the elements. The file is random access and is in element number sequence with each element starting 50 records apart to allow for a maximum of 50 intermediate points in a reach element. There is a maximum of 10,000 records (50 times the 200 maximum elements).

3) The downstream file - this is an input/output file which contains the data computed in the downstream direction for the elements. See file 2.

4) The cross section points file - this is an input file which contains the coordinate data and pier base data for irregular sections. It is random access and is in section number sequence, there are a maximum of 200 records.

5) The input card file - this file consists of the trace switch card followed by the Q data cards.

6) The output print file - this is the printing and plotting of the computed data and any messages which may be caused to appear.

8) The plot file - this file consists of the data which is to be plotted after the composite profile is printed. It is in element number sequence and can have a maximum of 10,000 records (as computed for file 2).

1.2)

The following is a description of the use of the common blocks used in the computation phase:

CONSTS) This contains the gravity value, the trace switch value, the limit for intermediate reach records, and the depth from the previous element. This block is used by all subroutines and functions.

WRITBK) This contains file key data and processing codes from the previous element. This block is used by all subroutines and functions.

PTBLCK) This contains the coordinate data and pier base data for irregular shaped sections. This block is used by all subroutines and functions.

RCHBRN) This contains the channel description data of a reach element to be used in subroutines REACH and BERNLI to compute intermediate and ending points in the reach.

RCH12B) This contains the channel description of the upstream and downstream ends of a junction or transition for use in functions DEPSMP, SUMM, and SUMP for computing momentum and pressure curves for the junction and in TRANSB for computing the energy in the transition.

RCH34B) This contains the channel description of the laterals in a junction for use in functions DEPSMP, SUMM, and SUMP for computing momentum and pressure curves.

WALLBK) This contains the channel description for the upstream and downstream ends of a wall or bridge entrance to use in function FWALL to compute the area of obstruction.

REACHX) This contains the description of the reach element to use in subroutine PPMDEP to compute the station and force of a hydraulic jump.

1.1)

There are five files used in the computation of the water surface profile. Their uses and descriptions follow.

1.2)

The element file contains 202 records. The first 200 records contain the element data to be processed. This includes the location of the element, its length, the data defining the sections at both the upstream and downstream ends, and the controlling depths (critical, normal, and maximum open flow depths where required) in the element. These elements are arranged from the system outlet (record number one) to the system headworks (record number from 3 to 200). The 200 represents the maximum number of elements which can be processed in a single computation. Inbetween the outlet and headworks elements the following element types may occur: reach, junction, transition, bridge entrance, bridge exit, wall entrance, and wall exit. There must be at least one of these elements between the outlet and headworks. Record number 201 contains any title data to be used in the printout, and record number 202 contains the number of elements which are to be processed (between 3 and 200) and also is the key pointer to the headworks which is the last element in the channel to be processed.

1.3)

The upstream file contains the data computed for the upstream profile. This data includes the channel definition for the particular end of the element, the controlling depths (as described above) and the cross section area flow, depth, force, and friction slope for the particular end of the channel. The outlet and headworks elements can only produce one upstream profile record. The junction, transition, bridge entrance, bridge exit, wall entrance, and wall exit elements can produce two (one at the upstream end and one at the downstream end of the element) upstream profile records. The reach element can produce up to fifty (over fifty will cause processing to stop) upstream profile records. Since it is not known when a reach record may occur enough space is allocated to write fifty records for each element. This allows for 10,000 records (50 * 200 (maximum number of elements)) on the upstream file. The key to read or write upstream records is therefore found by multiplying fifty times the quantity (element number minus one) and then the record wanted is added to that value.

1.4)

The downstream file is the same as the upstream file only it contains the downstream profile data.

1.5)

The plot file contains the key data from the composite profile. This data includes the controlling depths (as described for the element file), channel height, water surface elevation, invert elevation, station location, and depth of flow. This data is then plotted after the composite profile is written.

1.6)

The cross section points file contains the cross section points and Y coordinate values for pier bases for irregular shaped sections. There can be up to 99 points to describe the section and up to ten pier values in the section. There are 200 records in this file allowing for a different set of points for each possible element. The key for this file is the section number on the element file for the upstream end, downstream end or lateral (junction only) in the element being processed.

STANDARD CALCULATIONS

A : see function AREAC

D : see function DEPTH (for lower limit depth)

E : (D + HV)

H : INV1 - INV2

M : (Q **2) / (A * G)

Q : from Q card (laterals are added in a junction)

R : (A / WP)

V : (Q / A)

DC : see function DCRIT

DN : see function DNORM

DH : channel height in closed sections, channel height plus ten in open sections

HV : ((Q / A) **2) / (2 * G)

SF : see function SF

SO : (INV1 - INV2) / DELL

WP : see function WETP

WS : (INV + D)

VSQ : ((Q / A) **2)

DELL : (STA1 - STA2)

EGL : (INV + D + HV)

X : where * is a multiplication sign, and ** is an exponential sign.

MAIN

1.1)

The main routine controls all the processing and printing to obtain the final water surface profile. There are nine major steps to accomplish this which are as follows:

1) Read the first input card for the trace switch (JDEBUG) which controls the output printing according to the following values:

   (0) - Prints and plots the composite profile.

   (1) - Prints computational flow, upstream, downstream, and composite profiles and plots the composite profile.

   (2) - Prints upstream, downstream, and composite profiles and plots the composite profile.

2) Read the Q card data which consists of the design Q at the system headworks and the Q factor which is a multiplier to adjust the Q values in the junction laterals to their desired values.

3) Call subroutine ELMCHG to update all the channel elements with new Q values and to compute normal depth, critical depth, maximum open flow depth, and channel slope in the elements where they are applicable.

4) Call subroutine DWNSTM to compute the downstream profile data and write it to the downstream file.

5) Call subroutine UPSTRM to compute the upstream profile data and write it to the upstream file.

6) Call subroutine JUMPP to select the controlling profile for each element and to determine when and where any hydraulic jumps may occur (create the composite profile).

7) Call subroutine WRITEN to print the composite profile and build the plot file.

8) Call subroutine PLOTR to plot the key values of the composite profile.

9) Check to see if any more Q cards are present. If more Q cards exist steps 2 through 9 are performed again, otherwise the run is complete and processing is stopped.

GENERAL DOWNSTREAM PROCESSING PROCEDURES

1.1)

In downstream processing, computation begins at the system headworks and proceeds to the system outlet. Inside the individual elements processing is always done at the upstream end first (the end closest to the system head-works) and the downstream end last (the end closest to the system outlet). The controlling depth at the upstream end of each element is set equal to the depth at the downstream end of the element which was processed immediately prior to the current element. If the downstream depth from the preceeding element was not computed or is invalid then the critical depth at the upstream end of the current element becomes the controlling depth.

1.2)

The depth of flow in the downstream direction is never permitted to go above the critical depth at that point for which the depth is being computed. If it does go above critical depth no computation will be done beyond that point in that element. The depth of flow is also never permitted to reach zero or become negative. If the depth does reach zero no computation will be done beyond that point in that element.

1.3)

Each time a depth has been computed or a controlling depth has been assumed: the area, force, and friction slope (if it exists in the element) are computed for that depth. The data at this point will then be added to the downstream profile by writing the data to the downstream file.

1.4)

Special energy calculations and processing have been added to the transition and junction downstream processing routines. These calculations are used to insure that total energy will not be allowed to rise from the upstream end of the element to the downstream end of the element.

1.1)

Subroutine DWNSTM controls the downstream computation and builds the D/S file which contains the data for the D/S profile. Record number 202 of the element file is read to obtain the number of elements to be processed (stored there in the edit program) and this number is also the element file key which points to the location of the system headworks element. D/S processing is then computed from the system headworks to the system outlet and DP is the depth from the D/S end of the previous element passed to the U/S end of the current element.

1.2)

Subroutine HDWKDS is called to do the D/S computation on the headworks element. The element file key is then decremented by one to read the next element to be processed. The record is read to obtain the element type and processing is then done according to that type as follows:

Element Type 1: Call subroutine OTLTDS to do the D/S computation on the system outlet element.

Element Type 2: Call subroutine RCHDS to do the D/S computation on the reach element.

Element Type 3: Call subroutine XJNCDS to do the D/S computation on the junction element.

Element Type 4: Call subroutine BRENDS to do the D/S computation on the bridge entrance element.

Element Type 5: Call subroutine BREXDS to do the D/S computation on the bridge exit element.

Element Type 6: This represents a headworks element, which was previously processed.

Element Type 7: Call subroutine TRNSDS to do the D/S computation on the transition element.

Element Type 8: Call subroutine WENTDS to do the D/S computation on the wall entrance element.

Element Type 9: Call subroutine WEXTDS to do the D/S computation on the wall exit element.

When the system outlet record has been processed the D/S profile is complete. Otherwise, the next record on the element file is read and processed according to element type in the same manner.

1.3)

After the D/S profile has been computed a check is made to see if a request has been made to print the D/S profile. If no printing is required, the D/S processing is complete and control is returned to the MAIN routine so it can initiate U/S processing. If the D/S profile print is requested, the D/S profile is printed in reverse order starting from the system outlet and ending with the system headworks. This is done so it will be in element number sequence starting with element number one. If there was no D/S computation for any element a message will be printed to indicate this. When the D/S profile has been printed control of processing is returned to the MAIN routine.

1.1)

Subroutine HDWKDS does the D/S computation on the system headworks. Read the headworks record. There is no SO, SF, or DN in the headworks, and the D/S end of the element is assumed in D/S processing.

1.2)

If there was no initial WS inputted or if the inputted WS is less than the INV then the controlling depth is set equal to DC. If there was a valid WS inputted, then the controlling depth is set equal to WS minus INV. If the controlling depth is greater than DC set the depth equal to DC, otherwise set the depth equal to the controlling depth.

1.3)

Compute the data for the headworks, write the data to the D/S file, and return processing control to subroutine DWNSTM.

1.1)

Subroutine RCHDS does the D/S computation on the reach element. The reach record is read. If SO is not greater than zero and if DP was not computed or equals DC, there is no processing for this reach, the D/S processing codes are set to invalid, and processing control is returned to subroutine DWNSTM.

1.2)

Compute the data at the U/S end of the reach. If D1 equals DC and this is a mild reach (SO less than SC), then there is no further computation in this reach and processing control is returned to subroutine DWNSTM. Otherwise, write the data computed at the U/S end of the reach to the U/S file.

1.3)

If D1 equals DC in a critical reach (SO equals SC) or if D1 equals DN in a steep reach (SO greater than SC), then set D2 equal to D1 (the depth does not change throughout the entire reach), compute the data at the D/S end of the reach, write the data to the D/S file, and return processing control to subroutine DWNSTM.

1.4)

If D1 equals DC and this is a steep reach (SO greater than SC) then the depth in the reach should fall. Compute a new SF1 at a depth slightly lower than DC (so in case DC was at DH in a closed section then the effect from the top of the channel will be omitted in the new computation of SF1, and the SF average between this point and the next point of computation in the reach will not be distorted).

1.5)

If D1 is less than DC then select the appropriate reach factor according to whether the reach is critical, mild, or steep. If the depth of flow will rise set the reach factor equal to 1/1.1, otherwise depth will drop and reach factor is set equal to 1.1, where:

> Reach factor is a value which is multiplied times the last known HV to determine intermediate points in a reach. Every time HV varies by ten per cent of its current value, computation is done at that point and the data is written to the D/S file as part of the D/S profile. If the change in HV extends the length of channel processed beyond the D/S end, then the D/S end is processed and computation is completed (see subroutine REACH).

Call subroutine REACH to do the processing for the intermediate points within a reach and to process the D/S end of the reach.

1.6)

If the computation was valid through the D/S end of the reach return processing control to subroutine DWNSTM. If DC was reached before the D/S end in a reach with a mild, zero, or negative invert slope then set the D/S processing codes to valid (the data processed is valid) but set D2 equal to zero (because processing was not complete) and return processing control to subroutine DWNSTM. If DC was not reached and the D/S end was not processed, an error occurred in the processing so return control to subroutine DWNSTM.

1.1)

Subroutine XJNCDS does the D/S computation on the junction elements. The junction record is read and the controlling depth is set. Compute the data in the U/S end and write the data to the D/S file (the U/S end of the junction is always retained as part of the D/S profile). Compute M1 and E1 where E1 is energy computed by the following formula:

E1 = D1 + ((Q1 / A1) **2 / (2 * G)) + H - ((DELL * SF1) / 2).

1.2)

If D1 equals DC1 then subtract .001 from D1 to lower the depth in the section in case DC1 equals DH1 in a closed section (this prevents the top of the channel from distorting D/S computation). Subtract .001 from DC2 for the same reason as above. Compute M2C (critical momentum at the D/S end using DC2 as the depth for the basis of computation).

1.3)

Compute D3 (the depth in the first lateral) as follows:

D3 = ((D1 + INV1 + DC2 + INV2) / 2) - INV3,
where INV3 is the invert elevation of the lateral.

If D3 is greater than the height of the first lateral then set D3 equal to the height of the lateral. Compute M3COS (the momentum effect on the junction caused by the first lateral) as follows:

M3COS = ((Q3 **2) / (A3 * G)) * COS (ANG3),
where Q3 is the rate of flow in the lateral,
A3 is the cross section area of flow in the lateral,
COS represents the function cosine,
ANG3 is the angle of confluence of the lateral at the junction.

If there is a second lateral in the junction do the same processing as for the first lateral only use:

D4 = ((D1 + INV1 + DC2 + INV2) / 2) - INV4, and
M4COS = (Q4 **2 / (A4 * G)) * COS (ANG4).

Compute SUMPT (the pressure effect on the momentum in the junction) by calling function SUMP. Compute MSUM (the upper limit momentum at the D/S end) as follows:

MSUM = M1 + M3COS + M4COS + SUMPT.

1.4)

If M2C equals MSUM then set D2 equal to DC2 and go to paragraph 1.5. If M2C is greater than MSUM then no computation can be done at the D/S end because D2 would have to exceed DC2 so set D2 equal to zero, and return processing control to subroutine DWNSTM. Otherwise, M2C is less than MSUM so compute a lower limit area (divide Q2 by 200) and use function DEPTH to compute a lower limit depth for the D/S end. Then call function DEPSMP to iterate between the lower limit depth and DC2 until a depth at the D/S end is found where pressure and momentum curves intersect. If no intersection could be found within this range of depths then no computation can be done at the D/S end so processing control is returned to subroutine DWNSTM. Otherwise, set D2 equal to the depth where the momentum and pressure curves intersect.

1.5)

Compute the data at the D/S end of the junction. Compute E2 (energy at the D/S end of the junction) as follows:

```
E2 = D2 + ((Q2 / A2) **2 / (2 * G)) + ((DELL * SF2) / 2)
```

If El is not less than E2 the data is written to the D/S file and processing control is returned to subroutine DWNSTM. If E1 is less than E2 it means there will be a drop in energy in the junction which is in error so function FORCEJ is used to iterate depths between the previous D2 and a depth equal to E1 until a D2 is found which produces an energy in the D/S end equal to the energy in the U/S end. If FORCEJ could not compute D2 then there is no computation at the D/S end so control is returned to subroutine DWNSTM. If FORCEJ did compute D2 then the data is recomputed in the D/S end and written to the D/S file and processing control is returned to subroutine DWNSTM.

1.1)

Subroutine TRNSDS does the D/S computation on the transition elements. The transition record is read and .001 is subtracted from the critical depths in both the U/S and D/S ends to insure the top of a closed channel will not be reached. The controlling D1 is set and the data at the U/S end is computed and written to the D/S file (the U/S end of the transition is always retained as part of the D/S profile). Compute B1 (energy for Bernoulli's equation at the U/S end of the transition) as follows:

B1 = D1 + HV1 + H - ((DELL * SF1) / 2)

Compute B2C (the critical energy for Bernoulli's equation at the D/S end using DC2 as the depth for the basis of the computation) as follows:

B2C = DC2 + HV2C + (K + (HV2C - HV2)) + ((DELL + SF2C) / 2)
where K equals 0.1 if V2C is greater than V1 and K equals -0.2 if V2C is less than V1.

1.2)

If B2C is greater than B1 go to paragraph 1.5 to determine if D2 can be computed. If B2C equals B1 then set D2 equal to DC2 and go to paragraph 1.4. Otherwise B2C is less than B1.

1.3)

Use function DEPTH to compute a lower limit depth in the D/S end based on an area equal to Q2 divided by 200. Use function TRANSB to iterate between the lower limit depth and DC2 to find an energy in the D/S end which matches the energy in the U/S end using the formula in paragraph 1.1 for B2C only for the iterated depth instead of DC2. If TRANSB could not solve for the depth to produce an equal energy then there is no computation in the D/S end and control is returned to subroutine DWNSTM. Otherwise set D2 equal to the depth computed in function BRANSB.

1.4)

Compute the data at the D/S end of the transition. Write the data to the D/S file and return control to subroutine DWNSTM.

1.5)

Set a test depth equal to a depth just slightly lower than DC2 and compute a test energy in the D/S end using the same formula in paragraph 1.1 for the computation of B2C only using the test depth instead of DC2. If the test energy is greater than B2C then the energy needed in the D/S end is outside the nose of the energy curve and there can be no computation at the D/S end so processing control is returned to subroutine DWNSTM. Otherwise the energy needed to equal the U/S energy is in the nose of the energy curve.

1.6)

If the test energy is not greater than B1 this test depth is considered the correct D2 and computation continues at paragraph 1.4. Otherwise function TRANSB is used to iterate depths in the nose of the energy curve decreasing from DC2 as done in paragraph 13.

1.1)

Subroutine BRENDS does the D/S computation on the bridge entrance elements. The bridge entrance record is read and SO, DN, SF, and the number of piers at the U/S end are set to zero because they do not exist. If the D/S end of the previous element was not computed or if DP equals DC1 then no processing is done for this bridge entrance (because depth rises going through a bridge entrance but it may not exceed DC in D/S processing) and processing control is returned to subroutine DWNSTM. Otherwise, set D1 equal to DP.

1.2)

Compute the data in the U/S end in the bridge entrance. Compute M1, AP1 (the area of the piers under water if the piers at the D/S end were in the U/S end), and PP1 (the pressure for the piers if the piers at the D/S end were in the U/S end). AP1 and PP1 are computed as follows:

AP1 = NOP2 * BP2 * D1
PP1 = NOP2 * BP2 * (D1 **2 / 2)
where NOP2 is the number of piers at the D/S end,
BP2 is the average base of piers at the D/S end.

Compute FMI (the desired force to have at the D/S end of the bridge entrance) as follows:

FMI = (M1 * (A1 - (FP * AP1)) / A1) + (P1 - PP1)
where FP is the pier factor inputted by the user for processing in bridge entrances.

Compute F2C (the critical force at the D/S end using DC2 as the depth for which the computation is made).

1.3)

If FMI is not greater than F2C then the D2 needed to achieve a force equal to FMI would have to be greater than DC2; this cannot happen so no computation at the D/S end can be done and processing control is returned to subroutine DWNSTM.

1.4)

If FMI is greater than F2C then function FORCEF is used to iterate between D1 and DC2 to find a depth at the D/S end which will produce an F2 equal to FMI. If a depth is not found within these limits then no computation can be done at the D/S end and processing control is returned to subroutine DWNSTM. Otherwise, D2 is set equal to the depth which produced a force equal to FMI.

1.5)

Compute the data at the D/S end of the bridge entrance. Write the data at the U/S end and the data at the D/S end to the D/S file and return processing control to the subroutine DWNSTM.

BREXDS

1.1)

Subroutine BREXDS does the D/S computation on the bridge exit elements. The bridge exit record is read and SO, DN, SF, and the number of piers at the D/S end are set to zero because they do not exist. The controlling depth is set in the U/S end and the depth will drop going through a bridge exit.

1.2)

Compute the data in the U/S end and write the data to the D/S file (the U/S end of the bridge exit is always retained on the D/S profile). Set FMI (the desired force at the D/S end) equal to the force at the U/S end. Compute a lower limit area at the D/S end (this is done by dividing Q2 by 200) and use function DEPTH to compute a lower limit depth based on the lower limit area.

1.3)

Insert the piers from the U/S end into the D/S end of the bridge exit and use function FORCEM to iterate between the lower limit depth and DC2 to find a depth at the D/S end which will produce a force which equals FMI. If a depth is not found within these limits then no computation can be done at the D/S end and processing control is returned to subroutine DWNSTM. Otherwise, set D2 equal to the depth which produced the force equal to FMI.

1.4)

Compute the data in the D/S end of the bridge exit, write the data to the D/S file, and return processing control to subroutine DWNSTM.

1.1)

Subroutine WENTDS does the D/S computation on the wall entrance elements. The wall entrance record is read and SO, DN, SF, and the number of piers at the U/S end are set equal to zero because they do not exist. If the D/S end of the previous element was not computed or if DP equals DC1 then no computation can be done in the wall entrance (because depth rises going through a wall entrance but it cannot exceed DC) and processing control is returned to subroutine DWNSTM. Otherwise, set the controlling depth in the U/S end.

1.2)

Compute the data in the U/S end of the wall entrance. Compute A2T (area at the D/S end using D1 as the depth on which the calculation is made). Compute AOBST (area of obstruction) by subtracting A2T from A1. If AOBST is not greater than zero an error has occurred, a message will be printed, and processing will be stopped.

1.3)

Compute M1 and compute F2T (force at the D/S end using D1 as the depth on which the calculation is made). Compute P2T (pressure at the D/S end based on F2T and A2T) as follows:

P2T = F2T - (Q2 **2 / (A2T * G)).

Compute FMI (the desired force at the D/S end) as follows:

FMI = M1 * ((A1 - AOBST) / A1) + P2T.

Compute F2C (the force at the D/S end using DC2 as the depth on which the calculations are made).

1.4)

If FMI is not greater than F2C then the depth needed to produce a force equal to FMI would have to be greater than DC2. This cannot happen in D/S processing so the D/S end cannot be computed and processing control is returned to subroutine DWNSTM.

1.5)

If FMI is greater than F2C then compute a lower limit area at the D/S end (divide Q2 by 200) and use function DEPTH to compute a lower limit depth based on the lower limit area. Use function FWALL to iterate between the lower limit depth and DC2 to find a depth at the D/S end which will produce a force equal to FMI. If a depth is not found within these limits, then there is no computation at the D/S end and processing control is returned to subroutine DWNSTM. Otherwise, set D2 equal to the depth which produced the force equal to FMI.

1.6)

Compute data at the D/S end of the wall entrance, write the data at the U/S end and the data at the D/S end to the D/S file and return control to subroutine DWNSTM.

1.1)

Subroutine WEXTDS does the D/S computation on the wall exit elements. The wall exit record is read and SO, DN, SF, and the number of piers at the D/S end are set equal to zero because they do not exist. Set the controlling depth in the U/S end. The depth should drop when going through a wall exit.

1.2)

Compute the data in the U/S end of the wall exit and write the data to the D/S file (the U/S end of the wall exit is always retained on the D/S profile). Compute V1 and HV1 and compute FMI (the desired energy value at the D/S end) by adding HV1 and D1.

1.3)

Compute a lower limit area for the D/S end (divide Q2 by 200). Compute a lower limit depth at the D/S end by using function DEPTH to compute a depth based on the lower limit area. Use function FORCEW to iterate between the lower limit depth and DC2 to find a depth at the D/S end which will produce a value equal to FMI in the following formula:

FMI = DT + HVT - ABS ((V1 **2 / (2 * G)) - (VT **2 / (2 * G)))
where T indicated values based on the iterated depth,
and ABS means absolute value.

If a depth is not found within the limits which will produce a value equal to FMI then no computation can be done at the D/S end so processing control is returned to subroutine DWNSTM. Otherwise, set D2 equal to the depth which produced the value equal to FMI.

1.4)

Compute the data in the D/S end of the wall exit, write the data to the D/S file, and return processing control to subroutine DWNSTM.

OTLTDS

1.1)

Subroutine OTLTDS does the D/S computation on the system outlet element. Only the U/S end of the outlet is assumed in D/S processing. Read the outlet record and set SO, DN, and SF equal to zero because they do not exist. If the D/S end of the previous element was not computed there is no computation and control is returned to subroutine DWNSTM. Otherwise, set the controlling depth in the system outlet.

1.2)

Compute the data in the outlet, write the data to the D/S file, and return control to subroutine DWNSTM.

GENERAL UPSTREAM PROCESSING PROCEDURES

1.1)

In upstream processing computation begins at the system outlet and proceeds to the system headworks. Inside the individual elements processing is always done at the downstream end first (the end closest to the system outlet) and at the upstream end last (the end closest to the system headworks). The controlling depth at the downstream end of each element is set equal to the depth at the upstream end of the element which was processed immediately prior to the current element. If the upstream depth from the preceeding element was not computed or is invalid then the critical depth at the downstream end of the current element becomes the controlling depth.

1.2)

The depth of flow in the upstream direction is never permitted to go below the critical depth at that point for which the depth is being computed. If it does go below critical depth no computation in the upstream direction will be done beyond that point in that element. The depth of flow in an open channel (no top) is restricted to ten feet above the channel walls. If the depth exceeds this value at any time, then the section is inadequate to carry the flow and all processing will stop. In a closed (covered) section there is no restriction on how high the depth of flow may rise because once the height of the channel is surpassed the pressure flow routines will be able to calculate the resulting depth.

1.3)

Each time a depth has been computed or a controlling depth has been assumed: the area, force, and friction slope (if it exists in the element) are computed for that depth. The data at this point will then be added to the upstream profile by writing the data to the upstream file.

1.4)

Special energy calculations and processing have been added to the transition and junction upstream processing routines. These calculations are used to insure that total energy will not be allowed to drop from the downstream end of the element to the upstream end of the element.

UPSTRM

1.1)

Subroutine UPSTRM controls the upstream computation and builds the upstream file which contains the data for the upstream profile. Processing in the U/S direction is computed from the system outlet to the system headworks. The system outlet is record number one on the element file. Set the element file key to one and call subroutine OTLTUS to do the U/S computation in the outlet (element type 1).

1.2)

Increment the element file key by one and read the next element record to determine what processing is to be done. The processing routine to be used is dependent upon the type of element to be processed as follows:

Element Type 2: Call subroutine RCHUS to do the U/S computation on the reach element.

Element Type 3: Call subroutine XJCTUS to do U/S computation on the junction element.

Element Type 4: Call subroutine BRENUS to do U/S computation on the bridge entrance element.

Element Type 5: Call subroutine BREXUS to do U/S computation on the bridge exit element.

Element Type 6: Call subroutine HDWKUS to do U/S computation on the system headworks element.

Element Type 7: Call subroutine TRNSUS to do U/S computation on the transition element.

Element Type 8: Call subroutine WENTUS to do U/S computation on the wall entrance element.

Element Type 9: Call subroutine WEXTUS to do U/S computation on the wall exit element.

When the system headworks element has been processed the U/S computation has been completed then go to paragraph 1.3. Otherwise, read the next record and process it according to element type as done above.

1.3)

A check is made to see if a request has been made to print the U/S profile. If no printing is required the U/S processing is complete and processing control is returned to the MAIN routine so it can initiate hydraulic jump processing. If the U/S profile print is requested it is printed in element number sequence starting with element number one which is the system outlet. If there was no U/S computation for any element a message will be printed to indicate this. When the U/S profile has been printed control of the processing is returned to the MAIN routine.

1.1)

Subroutine OTLTUS does the U/S computation on the system outlet. Read the outlet record and set DN, SO, and SF equal to zero because they do not exist. The U/S end of the outlet is assumed in U/S processing.

1.2)

Subtract INV. from the inputted WS to obtain the controlling depth. If the controlling depth is less than DC then set the controlling depth equal to DC. Set D1 equal to the controlling depth.

1.3)

If D1 is greater than DH1 and this is in an open section then the WS is too high for the channel, an error message is printed, and all processing is stopped.

1.4)

Compute the data in the outlet. If there was no D/S processing in the outlet or if F1 computed in D/S processing is less than F1 computed in U/S processing go to paragraph 1.5. Otherwise no U/S profile is computed for the outlet and processing control is returned to subroutine UPSTRM.

1.5)

Write the data computed in the outlet to the U/S file and return processing control to subroutine UPSTRM.

1.1)

Subroutine RCHUS does the U/S computation on the reach elements. See note after paragraph 5.3 if pressure calculations are utilized. The reach record is read and the controlling depth is set in the D/S end. If D2 equals DC and if the D/S profile is complete with the depth in the D/S end less than DC (the D/S profile controls) then, there is no U/S processing in the reach and processing control is returned to subroutine UPSTRM.

1.2)

If D2 is not less than DH in an open section then WS is too high in an open channel, an error message is printed, and all processing is stopped.

1.3)

Compute the data in the D/S end of the reach and write the data to the U/S file. If D2 is not less than DH in a closed section then go to pressure flow calculations at paragraph 4.1.

1.4)

The section is in open flow processing. If SO is not greater than zero go to paragraph 5.1 to calculate the U/S profile for negative or zero slopes. If D2 equals DC and this reach is steep (SO is greater than SC) no more computation can be done since the depth will drop below DC in U/S processing, so control is returned to subroutine UPSTRM.

1.5)

If D2 equals DC and the reach is critical (SO equals SC) then the depth remains at DC throughout the reach so set D1 equal to DC, or if the channel is mild (SO is less than SC) and D2 is equal to DN then the depth remains at DN throughout the entire reach so set D1 equal to DN, then compute the data in the U/S end, write the data to the U/S file, set U/S processing codes to valid and return control to subroutine UPSTRM. Otherwise go to paragraph 2.1

RCHUS
(OPEN FLOW CALCULATIONS)

2.1)

If the depth of flow will rise in the reach, set the reach factor equal to 1/1.1, otherwise depth will drop and the reach factor is set equal to 1.1.

Reach factor is a value which is multiplied times the last known HV to determine intermediate points in a reach. Every time HV varies by ten percent of its current value computation is done at that point and the data is written to the U/S file as part of the U/S profile. If the change in HV extends the length of reach covered beyond the U/S end, then the U/S end is processed and computation is completed (see subroutine REACH).

2.2)

Call subroutine REACH to do the processing for the intermediate points within a reach through the U/S end or through the point where pressure flow begins in the reach.

2.3)

If the processing in the REACH subroutine was computed through to the U/S end then return processing control to subroutine UPSTRM. If the processing in the REACH subroutine was not complete through the U/S end because the depth dropped to DC before the end of a steep reach, then set D1 equal to zero, and return control to subroutine UPSTRM. If the processing in the REACH subroutine was not complete because pressure flow was entered (depth reached DH in a closed section) then go to paragraph 3.1. Otherwise return control of processing to subroutine UPSTRM with an indication that the computation in the reach is invalid.

RCHUS
(PRESSURE FLOW IN U/S END)

3.1)

Pressure flow was encountered in processing through the middle of a reach. Compute DELLP (the remaining length of the reach which is under pressure) by subtracting DELLO (the length of the reach processed in the REACH subroutine to the point where pressure flow begins) from DELL (the entire length of the reach). Compute ANGP (the effect of the angle of bend in DELLP) as follows:

ANGP = ANG * (DELLP / DELL)
where ANG is the angle of bend in the reach.

Compute NMHP (the number of manholes in DELLP) as follows:

NMHP = NMH * (DELLP / DELL)
where NMH is the number of manholes in the reach.

Compute HF (the head loss due to friction in DELLP) by multiplying DELLP times SFT (the SF at the point where pressure flow was entered).

Compute HB (the head loss due to the bend in DELLP) as follows:

HB = .2 * HV2 * (ANGP / 90) **0.5)

Compute HM (the head loss due to manholes) as follows:

HM = .05 * HVT * NMHP
where HVT is the velocity head at the point where pressure flow was entered.

Compute WS1 as follows:

WS1 = WS2 + HF + HB + HM.

Then compute D1 by subtracting INV1 from WS1.

3.2)

Compute the data at the U/S end, write the data to the U/S file, and return processing control to subroutine UPSTRM.

RCHUS
(PRESSURE FlOW IN D/S END)

4.1)

The D/S end of the reach was under pressure. Compute HF (the head loss due to friction in the reach) by multiplying SF2 by DELL (the length of the reach). Compute HM (the head loss due to the number of manholes in the reach) as follows:

HM = .05 * HV2 * NMH
where NMH is the number of manholes in the reach.

Compute APT (the head loss due to the angle point in the reach) as follows:

APT = ANGPT * HV2 * .0033
where ANGPT is the angle point in the reach.

If ANG (the angle of the bend in the reach) is not greater than zero then set HB (the head loss due to the bend in the reach) equal to zero, otherwise compute HB as follows:

HB = .2 * HV2 *((ANG / 90) **.5).

Compute WS1 as follows:

WS1 = WS2 + HF + HM + HB + APT.

4.2)

Compute D1 by subtracting INV1 from WS1. Compute SOFFIT (the height the WS can reach at the U/S end without being under pressure) by adding INV1 to DH. If WS1 is less than SOFFIT then pressure flow broke seal in the middle of the reach, go to paragraph 4.3 to find where open flow begins. Otherwise the U/S end of the reach is also under pressure so D1 is a good value. Compute the data in the U/S end, write the data to the U/S file, and return processing control to subroutine UPSTRM.

4.3)

Set the values for solving the quadratic equation to find the point where pressure flow ends in the reach. Compute AQ (the A value in the quadratic equation) as follows:

AQ = SO - SF2 - (.05 * HV2 * (NMH / DELL))
where NMH is the number of manholes in the reach.

If AQ is not positive set AQ equal to SO minus SF2. Compute CQ (the C value in the quadratic equation) as follows:

CQ = D2 + APT - DH.

If ANG (the angle of the bend in the reach) is not positive set BQ (the B value in the quadratic equation) equal to zero, otherwise compute BQ as follows:

BQ = .2 * HV2 * ((ANG / (90 * DELL)) **.5).

Put these values into the quadratic equation to compute DELLX (the length of the reach under pressure) as follows:

DELLX = (2 * AQ * CQ + BQ ** 2 + (((2 * AQ * CQ + (BQ **2)) **2) - (4 * (AQ **2) * (CQ **2)) **.5)) / (2 * (AQ ** 2)),

if BQ equals zero use:

DELLX = (2 * AQ * CQ) / (2 * (AQ **2)).

4.4)

Compute STAX (the station or point where pressure flow breaks seal) by adding DELLX to STA2 (the station or point at the D/S end of the reach). Set DX (the depth at the end of pressure flow) equal to DH. Compute INVX (invert at the end of pressure flow) as follows:

INVX = (SO * DELLX) + INV2.

Compute DELLO (the length of the reach left which is not under pressure) by subtracting DELLX from DELL. Substitute DELLO for DELL and all the values computed for the point where pressure flow ends for the values at the D/S end of the reach, hence:

4.5)

STA2 = STAX, D2 = DX, INV2 = INVX and DELL = DELLO.

Compute the data at the point where pressure flow breaks seal and write the data to the U/S file. Subtract.001 from D2 and recalculate SF2 so the top of the channel will not distort the SF average between this point and the next point to be computed. Go compute the remainder of the reach for open flow processing at paragraph 2.1.

5.1)

Compute DJI (the depth at the next point in the reach to be computed) by multiplying the current DJI (D2 the first time through) by 1.05. If DJI is greater than DH set DJI equal to DH. Compute EJI (the specific energy for DJI) by adding DJI with HVJI (velocity head at this intermediate point). If DJI equals DH subtract .001 from DJI so the top of the channel will not distort the computation of SFJI (the friction slope at this intermediate point). Compute SFJI. If DJI was lowered to compute SFJI then set DJI back equal to DH. Compute SFAVE (the average between SFJI and SFJIP (the friction slope computed at the previous point)). Compute DELLJI (the length of reach processed between the previous point and the current point) as follows:

DELLJI = (EJI - EJIP) / (SO - SFAVE)
where EJIP is the specific energy at the previous point computed.

Add DELLJI to DELLT (the summation value of all the DELLJI which have been computed thus far).

5.2)

If DELLT is not greater than DELL (the length of the reach to be processed) the end of the reach has not been passed so go to paragraph 5.3 to check for pressure flow. If DELLT exceeds DELL the current DJI is past the end of the reach. Compute DELLB (the remaining length of reach to be processed from the last intermediate point written to the U/S file) by subtracting DELLTP (the previous DELLT) from DELL. Call subroutine BERNLI to compute the depth which will be at the U/S end of the reach by use of Bernoulli's equation. Set DJI equal to the depth computed in subroutine BERNLI and set DELLT equal to DELL.

5.3)

Compute the data at this point and write it to the U/S file. If DELLT equals DELL this was the U/S end of the reach so return control to subroutine UPSTRM. If DELLT is less than DELL and DJI is less than DH then go back to paragraph 5.1 to compute the next intermediate point. Otherwise DJI equals DH at this point and depth is rising. If this is not a closed section then print an error message because WS has gone too high in an open section and stop all processing. Otherwise this is where pressure flow begins in a closed section so substitute all the values computed at this intermediate point for the values at the D/S end, hence:

D2 = DJI, INV2 = INVJI, SF2 = SFJI, HV2 = HVJI and WS2 = WSJI.

Then go to compute pressure flow calculations in a reach at paragraph 3.1 using DELLO equal to DELLT.

Note:

The same reach cannot enter pressure or open flow twice. It can all be under pressure, or all under open flow, or it can go from pressure to open flow, or it can go from open flow to pressure. It cannot go from pressure to open flow and back to pressure, or from open flow to pressure and back to open flow.

1.1)

Subroutine XJCTUS does the U/S computation in junction elements. The junction record is read and the controlling depth is set in the D/S end. If D2 equals DC2 and if the D/S profile is complete with the depth in the D/S end less than DC2 (the D/S profile controls), then there is no U/S processing in this junction, and control is returned to subroutine UPSTRM.

1.2)

If D2 is not less than DH2 and this is an open section, then an error message is printed because WS is too high for the section and all processing is stopped.

1.3)

Compute the data in the D/S end and write it to the U/S file.

Compute M2 and compute E2 (energy value at the D/S end) as follows:

E2 = D2 + (((Q2 / A2) **2) / (2 * G)) + ((DELL * SF2) / 2)

If there are no laterals set M3COS and M4COS equal to zero and go to paragraph 1.4.

Compute D3 (the depth in the first lateral) as follows:

D3 = ((D2 + INV2 + DC1 + INV1) / 2) - INV3,
where INV3 is the invert elevation of the lateral.

If D3 is greater than the height of the lateral set D3 equal to the height of the lateral. If D3 is not greater than zero then set M3COS (the momentum in the first lateral) equal to zero, otherwise compute M3COS as follows:

M3COS = ((Q3 **2) / (A3 * G)) * COS(ANG3)
where Q3 is the rate of flow in the lateral,
A3 is the cross section area of flow in the lateral,
COS represents the function cosine,
ANG3 is the angle of confluence of the lateral at the junction.

If there is a second lateral do the same processing as for the first lateral only use:

D4 = ((D2 + INV2 +DC1 +INV1) / 2) - INV4, and

M4COS = ((Q4 **2) / (A4 * G)) * COS(ANG4)
where 4 represents references to the second lateral as 3 represented references to the first lateral.

If there is no second lateral set M4COS equal to zero.

1.4)

If D2 is not less than DH2 go to pressure flow processing at paragraph 2.1. Otherwise compute SUMPT (the pressure effect on the momentum in the junction) by calling function SUMP. Compute M1C (the critical momentum at the U/S end using DC1 as the depth at which the calculations are made). Compute MSUM (the lower limit momentum at the U/S end) as follows:

MSUM = M1C + M3COS + M4COS + SUMPT.

1.5)

If M2 equals MSUM then set D1 equal to DC1 and go to the next paragraph. If M2 is less than MSUM there is no computation at the U/S end (but the D/S end is always retained in U/S processing), and processing control is returned to subroutine UPSTRM. Otherwise M2 is greater than MSUM. Use function DEPSMP to iterate between DC1 and DH1 until a depth is found at the U/S end where pressure and momentum curves intersect. If the depth returned from DEPSMP is less than DH1 and no intersection was found then there was an error in processing and no computation can be done at the U/S end and processing control is returned to subroutine UPSTRM. If the intersection of pressure and mementum curves was above DH1 then set the depth equal to DH1 and go to paragraph 1.7 for pressure flow calculations. Otherwise set D1 equal to the depth where the momentum and pressure curves intersected.

1.6)

Compute the data in the U/S end. Compute E1 (energy value at the U/S end) as follows:

E1 = D1 + (((Q1 / A1) **2) / (2 * G)) + H - ((DELL * SF1) / 2).

If E1 is not less than E2 write the data to the U/S file and return processing control to subroutine UPSTRM. Otherwise energy dropped in the junction which is not allowed.

1.7)

If D1 is greater than DH1 and this is an open section then an error message is printed because WS is too high in the section, and all processing is stopped. If D1 is greater than DH1 in a closed section (pressure) then compute a new D1 as follows:

D1 = E2 - H - (((Q1 / A1) **2) / (2 * G)) + (DELL * ((SF1) / 2)

and go back to paragraph 1.6 to recompute the data in the U/S end. Otherwise D1 is less than DH1 with a decreasing energy. Use function FORCEJ to iterate between the previously computed D1 and a depth equal to E2 to find a depth which will produce an energy value equal to E2. Set D1 equal to the depth which produces this energy and go back to paragraph 1.6 to recompute the data in the U/S end.

2.1)

Set the depths in the existing laterals equal to the heights of the laterals and recompute the momentum in the laterals (as done in open flow). Use function SUMP to find SUMPT (the pressure effect on the momentum in the junction). Compute the data in the U/S end to obtain M1H (the momentum in the U/S end using DH1 as the depth for the basis of the calculations). Compute D1 as follows:

D1 = ((2 * (M2 - M1H - M3COS - M4COS)) / (A1H + A2))

+ (DELL * ((SF1H + SF2) / 2)) + D2 - H.

2.2)

If D1 is not less than DH1 (the entire junction is under pressure) then compute the data in the U/S end, write the data to the U/S file, and return control to subroutine UPSTRM. If DC1 equals DH1 then no computation can be done at the U/S end, and processing control is returned to subroutine UPSTRM. Otherwise compute M1C (critical momentum at the U/S end using DC1 as the depth for the basis of computation).

Compute MSUM (the lower limit momentum at the U/S end) as follows:

MSUM = M1C + M3COS + M4COS + SUMPT.

Go back to paragraph 1.5 to compare M2 and MSUM to determine processing.

1.1)

Subroutine TRNSUS does the U/S computation on transition elements. Read the transition record and set the controlling depth in the D/S end. If D2 equals DC2 and if the D/S profile is complete with the depth in the D/S end less than DC2 (the D/S profile controls), then there is no U/S computation so return processing control to subroutine UPSTRM.

1.2)

If D2 is not less than DH2 and this is an open section then an error message is printed because WS is too high in this section, and all processing is stopped.

1.3)

Compute the data in the D/S end and write the data to the U/S file (the D/S end is always retained in U/S processing). Compute B2 (energy for Bernoulli's equation at the U/S end using DC1 as the depth for the basis of the computation) as follows:

B2 = D2 + HV2 + (DELL * SF2) / 2)

Compute B1C (the critical energy for Bernoulli's equation at the U/S end using DC1 as the depth for the basis of the computation) as follows:

B1C = DC1 + HV1C + H + (K * (HV1C - HV2) - ((DELL * SF1C) / 2))
where K equals 0.1 if V2 is greater than V1C
or K equals -0.2 if V2 is less than V1C.

1.4)

If B2 is less than B1C go to paragraph 1.7 to compute the U/S depth in the nose of the energy curve. If B2 equals B1C then D2 equals DC1 so go to paragraph 1.6 to compute the data in the U/S end. Otherwise B2 is greater than B1C. Compute B1H (the maximum open flow energy in the U/S end) using the same formula in paragraph 1.3 to compute B1C only using depth DH1 instead of DC1. If B1H is less than B2 go to paragraph 1.8 because the U/S end is under pressure.

1.5)

Use function TRANSB to iterate between DC1 and DH1 to find the depth in the U/S end which produces an energy equal to the energy in the D/S end using the same formula used to compute B1C in paragraph 1.3 only using the iterated depth instead of DC1. If the depth computed in TRANSB is less than DC1 there is no computation in the U/S end processing control is returned to subroutine UPSTRM. If the depth computed in TRANSB is greater than DH1 go to paragraph 1.8 because the U/S end is under pressure. Otherwise set D1 equal to the depth computed in TRANSB.

1.6)

Compute the data in the U/S end. Write the data to the U/S file. Return processing control to subroutine UPSTRM.

1.7)

The depth at the U/S end may produce an energy which is in the nose of the energy curve. Set a test depth just greater than DC1. Compute a test energy using the same formula to compute B1C in paragraph 1.3 only using the test depth instead of DC1. If the test energy is greater than B1 the depth needed to produce an energy in the U/S end equal to the energy in the D/S end cannot be computed so processing control is returned to subroutine UPSTRM. Otherwise D2 can be computed. If the test energy is not greater than B2C it is assumed that D2 is equal to this test depth and control is sent to paragraph 1.6. If the test depth is greater than DH1 in the nose of the energy curve no computation can be done and control is returned to subroutine UPSTRM. Otherwise function TRANSB is used to iterate depths increasing from DC1 in the nose of the energy curve as done in paragraph 1.5.

1.8)

The U/S end of the transition is under pressure. If this is an open channel the WS is too high for this section, and error message is printed, and all processing is stopped. Otherwise compute D1 as follows:

D1 = B2 - HVT - H + ((DELL * SFT) /2) + (K * (HV2 - HVT))
where K equals 0.1 if V2 is greater than VT or
K equals -0.2 if V2 is less than VT and subscript T
represents data computed for the test depth used in
paragraph 1.4 or paragraph 1.7.

If D1 is not greater than DH1 print a warning message because it should have been established that D1 should be greater than DH1 to get to this processing. Go to paragraph 1.6 to continue processing.

TRNSUS
(PRESSURE IN THE D/S END)

2.1)

Use function SUMP to compute SUMPT (the pressure effect on the momentum in the transition). Compute the data in the U/S end to obtain M1H (the momentum in the U/S end using DH1 as the depth for the basis of the calculations). Compute D1 as follows:

D1 = ((2 * (M2 - M1H)) / (A1H + A2))

+ (DELL * ((SF1H + SF2) / 2)) + D2 - H.

2.2)

If D1 is not less than DH1 go back to paragraph 1.6 to compute the data in the U/S end under pressure. If DC1 equals DH1 then no computation can be done at the U/S end and processing control is returned to subroutine UPSTRM. Otherwise compute M1C (critical momentum at the U/S end using DC1 as the depth for the basis of the calculations). Compute MSUM (the lower limit momentum at the U/S end) by adding M1C and SUMPT and go back to compare the values of M2 and MSUM to determine processing as done for open flow processing at paragraph 1.4.

1.1)

Subroutine BRENUS does the U/S computation for the bridge entrance elements. Read the bridge entrance record and set SO, DN, SF, and the number of piers at the U/S end to zero because they do not exist. Set the controlling depth in the D/S end. If D2 equals DC2 and if the D/S profile is complete with the depth in the D/S end less than DC2, then there is no computation in the U/S direction (D/S profile will control) and processing control is returned to subroutine UPSTRM.

1.2)

If D2 is not less than DH2 and this section is open then print an error message because WS is too high in the section and all processing is stopped.

1.3)

Compute the data in the D/S end and write it to the U/S file. The D/S end is always retained in U/S processing. Set FMI (the desired force at the U/S end) equal to F2.

1.4)

If D2 is not less than DH2 go to paragraph 2.1 for pressure calculations. Otherwise use function FORCEF to iterate between D2 and DH1 (depth rises in a bridge entrance) to find a force at the U/S end (using the piers from the D/S end section inserted into the section at the U/S end, and the pier factor from the input) which is equal to FMI. If a depth was found in function FORCEF which produced a force equal to FM1 then set D1 equal to that depth and go to paragraph 1.5. If function FORCEF could not find a depth within this range of depths and DH1 was not reached then no computation can be done at the U/S end, and processing control is returned to subroutine UPSTRM. Otherwise D1 will be greater than DH1. If this is on open section then WS is too high so an error message is printed and processing is stopped. Otherwise compute HV1H (the HV at the U/S end using DH1 as the depth for the basis of the calculations). Compute EGL2 and then compute EGL1H as follows:

EGL1H = EGL2 + (0.5 * (ABS (HV1H - HV2)))
where ABS means absolute value. Compute D1 by subtracting HV1H and INV1 from EGL1H.

1.5)

Compute the data in the U/S end, write the data to the U/S file, and return processing control to subroutine UPSTRM.

BRENUS
(PRESSURE IN THE D/S END)

2.1)

If D2 is not less than DH1 go to paragraph 2.2. Use function FWALL to iterate between D2 and DH1 (depth rises in a bridge entrance) to find a force at the U/S end (using the area of obstruction between the U/S and D/S ends) which is equal to FMI. If FWALL could not compute a depth because of an error in computation then no computation can be done at the U/S end, and processing control is returned to subroutine UPSTRM. Otherwise set D1 equal to the depth found in function FWALL and go to paragraph 2.3.

2.2)

The depth in the U/S end is greater than DH1. If this is an open section print an error message because WS is going to be too high in the section, and stop all processing. If the channel is closed compute HV1H (the HV at the U/S end using DH1 as the depth for the basis of the calculations). Compute EGL2 and then compute EGL1H as follows:

EGL1H = EGL2 + (0.5 * ABS (HV1H - HV2)).
where ABS means absolute value.

Compute WS1 by subtracting HV1H from EGL1H. Compute D1 by subtracting INV1 from WS1.

2.3)

Compute the data in the U/S end, write it to the U/S file, and return processing control to subroutine UPSTRM.

1.1)

Subroutine BREXUS does the U/S computation on the bridge exit elements. Read the bridge exit record and set SO, DN, SF, and the number of piers at the D/S end equal to zero because they do not exist. Set the controlling depth in the D/S end. If D2 equals DC2 and if the D/S profile is complete with the depth in the D/S end less then DC2 (the D/S profile controls), then there is no U/S processing and processing control is returned to subroutine UPSTRM.

1.2)

If D2 is not less than DH2 and the section is open, then print an error message because WS is too high in the section, and stop all processing.

1.3)

Compute the data in the D/S end. If D2 is greater than DH1 (bridge exit will be under pressure at both ends) then set D1 equal to D2 and go to paragraph 1.5. Otherwise compute PP2 (the effect on pressure if the piers at the U/S end were inserted in the D/S end) as follows:

PP2 = (NP1 * BP1 * (D2**2)) /2
where NP1 is the number of piers at the U/S end, and
BP1 is the average width of piers at the U/S end.

Compute FMI (the desired force at the U/S end) by subtracting PP2 from F2. Compute F1C (the critical force at the U/S end using DC1 as the depth for the basis of the calculations). If FMI equals F1C set D1 equal to DC1 and go to paragraph 1.5. If FMI is greater than F1C go to paragraph 1.4 to iterate for D1. Otherwise FMI is less than F1C and no computation can be done at the U/S end because D1 would have to be less than DC1, so return control to subroutine UPSTRM.

1.4)

Use function FORCEM to iterate between DC1 and D2 (depth drops in a bridge exit) to find a depth which will produce a force at the U/S end equal to FMI. If a depth was found which produced a force equal to FM1 then set D1 equal to that depth and go to paragraph 1.4. Otherwise no computation can be done at the U/S end and processing control is returned to subroutine UPSTRM.

1.5)

Compute the data in the U/S end, write the data at the D/S end and the data at the U/S end to the U/S file and return control to subroutine UPSTRM.

1.1)

Subroutine WENTUS does the U/S computation on the wall entrance elements. Read the wall entrance record and set SO, DN, SF, and the number of piers in the U/S end equal to zero because they do not exist. Set the controlling depth in the D/S end. If D2 equals DC2 and if the D/S profile is complete with the depth in the D/S end less than DC2 (the D/S profile controls), then there is no U/S computation in the wall entrance and control is returned to subroutine UPSTRM.

1.2)

If D2 is not less than DH2 and the section is open then print an error message because WS went to high in an open section, and stop all processing.

1.3)

Compute the data in the D/S end and write the data to the U/S file. Set FMI (the desired Force at the U/S end) equal to F2. If D2 is not less than DH1 go to paragraph 1.6 for pressure flow.

1.4

Use function FWALL to iterate between D2 and DH1 (depth of water rises) using the area of obstruction effect to find a depth which will produce a force value at the U/S end which is equal to FMI. If a depth was found which produced a force equal to FMI then set D1 equal to that depth and go to paragraph 1.5. If the depth reached DH1 before FMI could be reached (pressure flow) go to paragraph 1.6. Otherwise the U/S end can not be computed and processing conrol is returned to subroutine UPSTRM.

1.5

Compute the data in the U/S end, write the data to the U/S file, and return processing control to subroutine UPSTRM.

1.6)

The wall entrance is under pressure. Compute HV1H (the HV at the U/S end using DH1 as the depth for the basis of the calculations). Compute EGL2 and compute EGL1H as follows:

EGL1H = EGL2 + (FKC * (ABS (HV1H - HV2)))
where FKC is the wall factor inputted for this wall entrance and ABS means absolute value.

Compute WS1 by subtracting HV1H from EGL1H. Compute D1 by subtracting INV1 from WS1 and go back to paragraph 1.5 to compute data in the U/S end.

1.1)

Subroutine WEXTUS does the U/S processing in the wall exit elements. Read the wall exit record and set SO, DN, SF, and the nuber of piers at the D/S end equal to zero because they do not exist. Set the controlling depth in the D/S end. If D2 equals DC2 and if the D/S profile is complete with the depth in the D/S end less than DC2 (D/S profile controls), then there is no computation for the wall exit and processing control is returned to subroutine UPSTRM.

1.2)

If D2 is not less than DH2 and the section is open write an error message because WS is too high for this open section, and stop all processing.

1.3)

Compute the data in the D/S end. Compute V1C and HV1C (the V and HV at the U/S end using DC1 as the depth for the basis of the calculations). Compute MN1C (the lower limit specific energy value at the U/S end) as follows:

MN1C = DC1 + HV1C - (ABS ((V2 **2) / (2 * G)) - ((V1C **2) / (2 * G))), where ABS means absolute value.

Compute MN2 (the specific energy value at the D/S end) by adding D2 and HV2.

1.4)

If MN2 equals MN1C set D1 equal to DC1 and go to paragraph 1.5. If MN2 is less than MN1C no computation can be done at the U/S end (D1 would have to be below DC1) and processing control is returned to subroutine UPSTRM. Otherwise MN2 is greater than MN1C. Use function FORCEW to iterate between DC1 and D2 (depth is dropping at the U/S end) to find a depth which will compute an energy (the same way as computed for MN1C) which equals MN2. If a depth is found which computes an energy equal to MN2 then set D1 equal to that depth and go to paragraph 1.5. Otherwise computation at the U/S end can not be done and processing control is returned to subroutine UPSTRM.

1.5)

Compute the data at the U/S end, write the data at the D/S end and the data at the U/S end to the U/S file, and return processing control to subroutine UPSTRM.

HDWKUS

1.1)

Subroutine HDWKUS does the U/S computation on the system headworks element. The D/S end of the headworks is assumed. Read the headworks record and set SO, DN, and SF equal to zero because they do not exist. If the U/S end of the previous element was not computed, then no processing can be done in the headworks processing codes to invalid, and processing control is returned to subroutine UPSTRM.

1.2)

Set the controlling depth in the headworks. If D2 is not less than DH2 and the section is open then WS is too high for the open section so print on error message and stop all processing.

1.3)

Compute the data in the headworks, write the data to the U/S file, and return processing control to subroutine UPSTRM so it can initiate hydraulic jump processing.

1.1)

Subroutine JUMPP controls the processing to compute the composite profile by finding any hydraulic jumps which occur and by selecting whether the U/S or D/S profile has control in the element. Only the profile data which is in control at the time is kept and the remaining data from the other profile is discarded. The processing in JUMPP runs from the system outlet to the system headworks.

1.2)

Set the element file key to one which is the position of the system outlet on the file. Call subroutine OTLTJP to do the processing on the outlet (Element Type 1).

1.3)

Increment the element file key by one. Read the next element record to find out which element type is to be processed. Do the following processing according to the element type:

Element Type 2: Call subroutine RCHJP to do the jump processing on a reach element.

Element Type 3: Call subroutine XJCTJP to do the jump processing on a junction element.

Element Type 4: Call subroutine BENTJP to do the jump processing in a bridge entrance element.

Element Type 5: Call subroutine BEXTJP to do the jump processing on a bridge element.

Element Type 6: Call subroutine HDWKJP to do the jump processing on the headworks element.

Element Type 7: Call subroutine TRNSJP to do the jump processing on a transition element.

Element Type 8: Call subroutine WENTJP to do the jump processing on a wall entrance element.

Element Type 9: Call subroutine WEXTJP to do the jump processing on a wall exit element.

1.4)

When the system headworks element has been processed the hydraulic jump phase has been completed and processing control is returned to the MAIN routine. Otherwise, go back to paragraph 1.3 and start the processing for the next element.

1.1)

Subroutine OTLTJP does the hydraulic jump processing in the system outlet element. Read the outlet record to get the U/S and D/S processing codes. If there was no U/S and no D/S processing an error message is written (there should always be U/S processing) and processing control is returned to subroutine JUMPP.

1.2)

If there was U/S processing and no D/S processing go to paragraph 1.3. If there was D/S processing and no U/S processing go to paragraph 1.4. Otherwise, there is both U/S and D/S computation. If the U/S force is greater than the D/S force go to paragraph 1.3. If the U/S force is less than the D/S force go to paragraph 1.4. Otherwise, the U/S force equals the D/S force. If the U/S depth is not greater than DC go to paragraph 1.4. If the next element is a bridge entrance, bridge exit, wall entrance, or wall exit then go to paragraph 1.3. Otherwise, a hydraulic jump occurs in the outlet. Set the jump code to valid and keep both U/S and D/S profiles. Return processing control to subroutine JUMPP.

1.3)

The U/S profile controls so delete the D/S profile data. Set the jump code to invalid (no hydraulic jump occurs). Return processing control to subroutine JUMPP.

1.4)

The D/S profile controls so delete the U/S profile data. Set the jump code to invalid (no hydraulic jump occurs). Return processing control to subroutine JUMPP.

1.1)

Subroutine RCHJP does the hydraulic jump processing in the reach element. Read the reach record to get the U/S and D/S processing codes. If there was no U/S processing go to paragraph 1.7. If there was no D/S processing go to paragraph 1.8. Otherwise, there is both U/S and D/S processing. Read the first and last points on the U/S and D/S profiles. These values will be assumed to be at the U/S and D/S ends of the reach, however, D/S processing may not have reached the D/S end and U/S processing may not have reached the U/S end.

1.2)

If the U/S forces are greater than the D/S forces at both the U/S and D/S ends of the reach go to paragraph 1.8. If the D/S forces are greater than the U/S forces at both the U/S and D/S ends of the reach go to paragraph 1.7. If the U/S force is greater at one end and the D/S force is greater at the other end then the U/S and D/S force curves intersected in the reach so go to paragraph 2.1 to compute for a hydraulic jump.

1.3)

The U/S and D/S forces are equal at one or both ends of the reach. If the reach slope (SO) is not greater than zero or if this is a mild reach (SO is less than SC) then go to paragraph 1.5. If this is a critical reach (SO equals SC) go to paragraph 1.4. Otherwise, this is a steep reach (SO greater than SC). If the U/S and D/S forces are equal in the D/S end there is a hydraulic jump at the D/S end so delete the U/S profile except for the D/S end, keep all the D/S profile, and return processing control to subroutine JUMPP. Otherwise, the U/S and D/S forces are equal in the U/S end. If the U/S processing was complete and the depth in the U/S end is greater than DC go to paragraph 1.8. If the U/S processing was not complete and the depth in the U/S end is greater than DC then go to paragraph 2.1 to compute the location of a hydraulic jump. Otherwise, the U/S depth in the U/S end equals DC. If the next element is a bridge entrance, bridge exit, wall entrance, or wall exit go to paragraph 1.8. Otherwise, there is a jump at the U/S end of the reach so delete all the D/S profile except the U/S end, keep all the U/S profile, and return processing control to subroutine JUMPP.

1.4)

Processing for critical reaches. If the depth did not change in U/S processing go to paragraph 1.7. If the depth did not change in D/S processing go to paragraph 1.8. If the U/S force at the U/S end is greater than the D/S force at the U/S end go to paragraph 1.8. If the U/S and D/S forces at the D/S end are equal then there is a jump at the D/S end so the U/S profile is deleted except for the D/S end, all the D/S profile is kept, and control is returned to subroutine JUMPP. Otherwise, the U/S and D/S forces are equal at the U/S end. If the next record is a bridge entrance, bridge exit, wall entrance, or wall exit go to paragraph 1.8. Otherwise, there is a jump at the U/S end so delete the D/S profile except for the U/S end, keep all the U/S profile, and return processing control to subroutine JUMPP.

1.5)

Processing for zero or negative channel slopes. If the U/S force is greater than the D/S force in the U/S end go to paragraph 1.8. If the U/S force is less than the D/S force at the U/S end go to paragraph 1.6. Otherwise, the U/S and D/S forces are equal in the U/S end. If the next element is a bridge entrance, bridge exit, wall entrance, or wall exit go to paragraph 1.8. Otherwise, there is a jump at the U/S end so delete the D/S profile except for the U/S end, keep all the U/S profile, and return processing control to subroutine JUMPP.

1.6)

The U/S and D/S forces are equal in the D/S end of a reach with a negative or zero slope. If the D/S depth in the D/S end is less than DC then there is a jump in the D/S end so delete the U/S profile except for the D/S end, keep all the D/S profile, and return processing control to subroutine JUMPP. If the D/S profile was complete go to paragraph 1.7. If the D/S profile was not complete through the entire reach go to paragraph 2.1 to compute the location of the hydraulic jump.

1.7)

The D/S profile is in control. Delete the entire U/S profile. Keep all the D/S profile. Return processing control to subroutine JUMPP.

1.8)

The U/S profile is in control. Delete the entire D/S profile. Keep all the U/S profile. Return processing control to subroutine JUMPP.

RCHJP
(HYDRAULIC JUMP IN MIDDLE OF REACH)

2.1)

Call subroutine JUMPR to find the intermediate points between which the hydraulic jump occurs and the station and the force at the point of the hydraulic jump. Read the U/S and D/S profiles for the closest intermediate points upstream and downstream of the area of the intersection of the force curves. Use the following terminology for the remainder of the processing:

D1US: depth on the U/S profile just downstream of the intersection.

D2US: depth on the U/S profile just upstream of the intersection.

D1DS: depth on the D/S profile just downstream of the intersection.

D2DS: depth on the U/S profile just upstream of the intersection.

DU: upper limit depth.

D2: lower limit depth.

DEPTHU: depth at the point of intersection on the U/S profile.

DEPTHD: depth at the point of intersection on the D/S profile.

2.2)

If D1US equals D2US set DEPTHU equal to D1US and go to paragraph 2.3. If D1US is less than D2US then set DU equal to D2US and DL equal to D1US. If D1US is greater than D2US then set DU equal to D1US and DL equal to D2US. Call subroutine PPMDEP to iterate between DL and DU to find the D which produces the force at the point of intersection. Set DEPTHU equal to the D which produces the appropriate force.

2.3)

If D1DS equals D2DS set DEPTHD equal to D1DS and go to paragraph 2.4. If D1DS is greater than D2DS then set DU equal to D1DS and set DL equal to D2DS. If D1DS is less than D2DS then set DU equal to D2DS and DL equal to D1DS. Call subroutine PPMDEP to iterate between DL and DU to find the D which produces the force at the point of intersection. Set DEPTHD equal to the D which produces the appropriate force.

2.4)

Delete all the data on the U/S profile that is upstream of the hydraulic jump. Compute the data based on DEPTHU and write it to the U/S end of the remaining U/S profile. Delete all the data on the D/S profile downstream of the hydraulic jump. Compute the data based on DEPTHD and write it to the D/S end of the remaining D/S profile. Set the jump code to valid and return control of processing to subroutine JUMPP.

1.1)

Subroutine XJCTJP does the hydraulic jump processing for the junction elements. Read the junction record to obtain the U/S and D/S processing codes. If there was no U/S computation go to paragraph 1.6. If there was no D/S computation go to paragraph 1.5. Otherwise, there is both U/S and D/S processing. Read the D/S end from the U/S profile. If there was no U/S processing at the U/S end then set the U/S force at the U/S end equal to zero, otherwise, read the U/S end from the U/S profile. Read the U/S end from the D/S profile. If there was D/S processing at the D/S end read the D/S end from the D/S profile, otherwise, set the D/S force and depth at the D/S end equal to zero.

1.2)

If the U/S forces are greater than the D/S forces at both the U/S and D/S ends go to paragraph 1.5. If the D/S forces are greater than the U/S forces at both the U/S and D/S ends go to paragraph 1.6. If the U/S force is greater than the D/S force in one end and the D/S force is greater than the U/S force in the other end then the force curves crossed so go to paragraph 1.7 for hydraulic jump processing.

1.3)

If the U/S force is greater than the D/S force in the U/S end go to paragraph 1.5. If the U/S and D/S forces are equal in the U/S end go to paragraph 1.4. Otherwise, the U/S and D/S forces are equal in the D/S end. If the D/S depth in the D/S end equals DC2 go to paragraph 1.6. Otherwise, there is a jump at the D/S end so delete the U/S end of the U/S profile, keep all the D/S profile, and return processing control to subroutine JUMPP.

1.4)

The U/S and D/S forces are equal in the U/S end. If the D/S force is greater than the U/S force in the D/S end go to paragraph 1.6. If the D/S depth in the U/S end is less than DC1 and the next element is a bridge entrance, bridge exit, wall entrance, or wall exit go to paragraph 1.5. Otherwise if the D/S depth in the U/S end is less than DC1 there is a jump at the U/S end so delete the D/S end of the D/S profile, keep all the U/S profile and return processing control to subroutine JUMPP. Otherwise the U/S and D/S forces are also equal in the D/S end. If the D/S depth at the D/S end equals DC2 go to paragraph 1.5. Otherwise, there is a jump at the D/S end so delete the U/S end of the U/S profile, keep all the D/S profile, and return processing control to subroutine JUMPP.

1.5)

The U/S profile controls the junction. Delete the D/S profile. Retain the U/S profile. Return processing control to subroutine JUMPP.

1.6)

The D/S profile controls the junction. Delete the U/S profile. Retain the D/S profile. Return processing control to subroutine JUMPP.

1.7)

There is a hydraulic jump in the middle of the junction. Delete the U/S end of the U/S profile. Delete the D/S end of the D/S profile. Retain the U/S end of the D/S profile and the D/S end of the U/S profile. Return processing control to subroutine JUMPP.

1.1)

Subroutine TRNSJP does the hydraulic jump processing in the transition elements. Read the transition record to obtain the U/S and D/S processing codes. If there was no U/S computation go to paragraph 1.6. If there was no D/S computation go to paragraph 1.5. Otherwise, there is both U/S and D/S computation. Read the D/S end of the U/S profile and the U/S end of the D/S profile. If there was no U/S processing at the U/S end then set U/S force at the U/S end equal to zero, otherwise, read the U/S end from the U/S profile. If there was D/S processing at the D/S end then read the D/S end from the D/S profile, otherwise, set the D/S force and depth in the D/S end equal to zero.

1.2)

If the U/S forces are greater than the D/S forces at both the U/S and D/S ends go to paragraph 1.5. If the D/S forces are greater than the U/S forces at both the U/S and D/S ends go to paragraph 1.6. If the U/S force is greater than the D/S force in one end and the D/S force is greater than the U/S force in the other end then the force curves crossed so go to paragraph 1.7 for hydraulic jump processing.

1.3)

If the U/S force is greater than the D/S force in the U/S end go to paragraph 1.5. If the U/S and D/S forces are equal in the U/S end go to paragraph 1.4. Otherwise, the U/S and D/S forces are equal in the D/S end. If the D/S depth in the D/S end equals DC2 go to paragraph 1.6. Otherwise, there is a jump at the D/S end so delete the U/S end of the U/S profile, keep all the D/S profile, and return processing control to subroutine JUMPP.

1.4)

The U/S and D/S forces are equal in the U/S end. If the D/S force is greater than the U/S force in the D/S end go to paragraph 1.6. If the D/S depth in the U/S end is less than DC1 and the next element is a bridge entrance, bridge exit, wall entrance, or wall exit go to paragraph 1.5. Otherwise, if the D/S depth in the U/S end is less than DC1 there is a jump at the U/S end so delete the D/S end of the D/S profile, keep all the U/S profile, and return processing control to subroutine JUMPP. Otherwise the U/S and D/S forces are also equal in the D/S end. If the D/S depth at the D/S end equals DC2 go to paragraph 1.5. Otherwise, there is a jump at the D/S end so delete the U/S end of the U/S profile, keep all the D/S profile, and return processing control to subroutine JUMPP.

1.5)

The U/S profile controls the transition. Delete the D/S profile. Retain the U/S profile. Return processing control to subroutine JUMPP.

1.6)

The D/S profile controls the transition. Delete the U/S profile. Retain the D/S profile. Return processing control to subroutine JUMPP.

1.7)

There is a hydraulic jump in the middle of the transition. Delete the U/S end of the U/S profile and the D/S end of the D/S profile. Retain the U/S end of the D/S profile and the D/S end of the U/S profile. Return processing control to subroutine JUMPP.

1.1)

Subroutine BENTJP does the hydraulic jump processing in the bridge entrance elements. Read the bridge entrance record to obtain the U/S and D/S processing codes. If there was no U/S processing go to paragraph 1.6. If there was no D/S processing go to paragraph 1.5. Otherwise, there is both U/S and D/S processing. Read the U/S end from both the U/S and D/S profiles.

1.2)

Compute AP1 (the effect on area if the piers at the D/S end were in the U/S end) and PP1 (the effect on pressure if the piers at the D/S end were in the U/S end) for U/S processing as follows:

AP1 = BP2 * NOP2 * UD1
PP1 = BP2 * NOP2 * (UD1 **2) / 2
where BP2 is the average width of piers in the D/S end,
NOP2 is the number of piers in the D/S end, and
UD1 is the U/S depth in the U/S end.

Compute UM1 and UP1 (U/S momentum and pressure in the U/S end). Compute FMIU (the desired force at the D/S end of the U/S profile) as follows:

FMIU = (UM1 * ((UA1 - (FP * AP1)) / UA1) + (UP1 - PP1))
where FP is the pier factor inputted for this bridge entrance.

1.3)

Compute AP1 and PP1 for D/S processing as follows:

AP1 = BP2 * NOP2 + DD1
PP1 = BP2 * NOP2 * (DD1 **2) / 2
where AP1, PP1, BP2, NOP2, and DD1 are the same as in paragraph 1.2 only for D/S instead of U/S processing.

Compute FMID (the desired force at the D/S end of the D/S profile) as follows:

FMID = (DM1 * ((DA1 - (FP * AP1)) / DA1) + (DP1 - PP1))
where FP is the pier factor inputted for this bridge entrance.

1.4)

If FMIU is greater than FMID go to paragraph 1.5. If FMIU is less than FMID go to paragraph 1.6. Otherwise, FMIU equals FMID. There is a jump at the U/S end of the bridge entrance. Delete the D/S end of the D/S profile. Retain the U/S profile and the U/S end of the D/S profile. Return processing control to subroutine JUMPP.

1.5)

The U/S profile controls. Delete the D/S profile and retain the U/S profile. Return processing control to subroutine JUMPP.

1.6)

The D/S profile controls. Delete the U/S profile and retain the D/S profile. Return processing control to subroutine JUMPP.

1.1)

Subroutine BEXTJP does the hydraulic jump processing for the bridge exit elements. Read the bridge exit record to obtain the U/S and D/S processing codes. If there was no U/S processing go to paragraph 1.5. If there was no D/S processing go to paragraph 1.6. Otherwise, there is both U/S and D/S processing. Read the D/S end of both the U/S and D/S profiles.

1.2)

Compute UM2 and UP2 (the U/S momentum and pressure at the D/S end). Compute PP2 (the effect on pressure if the piers at the U/S end were in the D/S end) as follows:

PP2 = NOP1 * BP1 * (UD2 **2) / 2
where NOP1 is the number of piers at the U/S end,
BP1 is the average width of piers in the U/S end, and
UD2 is the U/S depth at the D/S end.

Compute FMIU (the desired force at the U/S end of the U/S profile) as follows:

FMIU = UP2 + UM2 - PP2.

1.3)

Compute DM2 and DP2 (the D/S momentum and pressure in the D/S end). Compute PP2 as follows:

PP2 = NOP1 * BP1 * (DD2 **2) / 2
where PP2, NOP1, BP1, and DD2 are described the same as in paragraph 1.2 except for the D/S profile instead of the U/S profile.

Compute FMID (the desired force at the U/S end of the D/S profile) as follows:

FMID = DP2 + DM2 - PP2.

1.4)

If FMIU is greater than FMID go to paragraph 1.6. If FMIU is less than FMID go to paragraph 1.5. Otherwise, FMIU equals FMID. Read the U/S end of the U/S profile. If the U/S depth at the U/S end equals DC1 go to paragraph 1.6. Otherwise, there is a jump at the U/S end of the bridge exit. Delete the D/S end of the D/S profile. Retain the U/S profile and the U/S end of the D/S profile. Return processing control to subroutine JUMPP.

1.5)

The D/S profile controls. Delete the U/S profile and retain the D/S profile. Return processing control to subroutine JUMPP.

1.6)

The U/S profile controls. Delete the D/S profile and retain the U/S profile. Return processing control to subroutine JUMPP.

1.1)

Subroutine WENTJP does the hydraulic jump processing on the wall entrance elements. Read the wall entrance record to obtain the U/S and D/S processing codes. If there was no U/S processing go to paragraph 1.6. If there was no D/S processing go to paragraph 1.5. Otherwise, there was both U/S and D/S processing. Read the U/S end and the D/S end from the U/S profile. Read the U/S end and the D/S end from the D/S profile.

1.2)

If the U/S forces are greater than the D/S forces at both the U/S and D/S ends go to paragraph 1.5. If the D/S forces are greater than the U/S forces at both the U/S and D/S ends go to paragraph 1.6. If the U/S force is greater than the D/S force in one end and the D/S force is greater than the U/S force in the other end then go to paragraph 1.7 because there is a hydraulic jump.

1.3)

If the D/S force is less than the U/S force in the U/S end go to paragraph 1.5. If the U/S and D/S forces are equal in the U/S end go to paragraph 1.4. Otherwise, the D/S force is greater than the U/S force in the U/S end and the U/S and D/S forces are equal in the D/S end. If the D/S depth in the D/S end equals DC2 go to paragraph 1.6. Otherwise, there is a jump at the D/S end so delete the U/S end of the U/S profile, keep all the D/S profile, and return processing control to subroutine JUMPP.

1.4)

The U/S and D/S forces are equal in the U/S end. If the D/S force is greater than the U/S force in the D/S end go to paragraph 1.6. Otherwise, the U/S and D/S forces are also equal in the D/S end (U/S and D/S forces were equal throughout the wall entrance). If the D/S depth in the U/S end is less than DC1 then there is a jump at the U/S end so delete the D/S end of the D/S profile, keep all the U/S profile and return processing control to subroutine JUMPP. If the D/S depth in the D/S end is less than DC2 then there is a jump in the D/S end so delete the U/S end of the U/S profile, keep all the D/S profile, and return processing control to subroutine JUMPP.

1.5)

The U/S profile controls. Delete the D/S profile and retain the U/S profile. Return processing control to subroutine JUMPP.

1.6)

The D/S profile controls. Delete the U/S profile and retain the D/S profile. Return control of the processing to subroutine JUMPP.

1.7)

The U/S and D/S force curves intersected in the middle of the wall entrance. Delete the U/S end of the U/S profile and the D/S end of the D/S profile. Retain the D/S end of the U/S profile and the U/S end of the D/S profile. Return processing control to subroutine JUMPP.

WEXTJP

1.1)

Subroutine WEXTJP does the hydraulic jump processing on the wall exit elements. Read the wall exit record for the U/S and D/S processing codes. If there was no U/S processing to to paragraph 1.6. If there was no D/S processing go to paragraph 1.5. Read the U/S end and the D/S end from the U/S profile. Read the U/S end and the D/S end from the D/S profile.

1.2)

If the U/S forces are greater than the D/S forces at both the U/S and D/S ends go to paragraph 1.5. If the D/S forces are greater than the U/S forces at both the U/S and D/S ends go to paragraph 1.6. If the U/S force is greater than the D/S force in one end and the D/S force is greater than the U/S force in the other end then go to paragraph 1.7 because there is a hydraulic jump.

1.3)

If the D/S force is less than the U/S force in the U/S end go to paragraph 1.5. If the U/S and D/S forces are equal in the U/S end go to paragraph 1.4. Otherwise, the D/S force is greater than the U/S force in the U/S end and the U/S and D/S forces are equal in the D/S end. If the D/S depth in the D/S end equals DC2 go to paragraph 1.6. Otherwise, there is a jump at the D/S end so delete the U/S end of the U/S profile, keep all the D/S profile, and return processing control to subroutine JUMPP.

1.4)

The U/S and D/S forces are equal in the U/S end. If the D/S force is greater than the U/S force in the D/S end go to paragraph 1.6. Otherwise, the U/S and D/S forces are also equal in the D/S end (U/S and D/S forces were equal throughout the wall exit). If the D/S depth in the U/S end is less than DC1 then there is a jump at the U/S end so delete the D/S end of the D/S profile, keep all the U/S profile, and return processing control to subroutine JUMPP. If the D/S depth in the D/S end is less than DC2 then there is a jump in the D/S end so delete the U/S end of the U/S profile, keep all the D/S profile, and return processing control to subroutine JUMPP.

1.5)

The U/S profile controls. Delete the D/S profile and retain the U/S profile. Return processing control to subroutine JUMPP.

1.6)

The D/S profile controls. Delete the U/S profile and retain the D/S profile. Return processing control to subroutine JUMPP.

1.7)

The U/S and D/S force curves cross in the middle of the wall exit. Delete the U/S end of the U/S profile and the D/S end of the D/S profile. Retain the D/S end of the U/S profile and the U/S end of the D/S profile. Return processing control to subroutine JUMPP.

1.1)

Subroutine HDWKJP does the hydraulic jump processing on the system headworks element. Read the headworks record for the U/S and D/S processing codes. If there was no U/S processing go to paragraph 1.4. If there was no D/S processing go to paragraph 1.3. Otherwise, there was both U/S and D/S processing. There is only one U/S profile record and one D/S profile record for the headworks. Read the U/S profile record and the D/S profile record.

1.2)

If the U/S force is greater than the D/S force go to paragraph 1.3. If the U/S force is less than the D/S force go to paragraph 1.4. Otherwise, the U/S force equals the D/S force. If the D/S depth equals DC go to paragraph 1.4. Otherwise, there is a jump in the headworks. Retain the U/S profile record and the D/S profile record and return processing control to subroutine JUMPP.

1.3)

The U/S profile controls. Delete the D/S profile and retain the U/S profile. Return processing control to subroutine JUMPP.

1.4)

The D/S profile controls. Delete the U/S profile and retain the D/S profile. Return processing control to subroutine JUMPP.

REACH

1.1)

Subroutine REACH does the processing of the intermediate points in the reach elements. This routine is used by RCHDS and RCHUS subroutines for D/S and U/S processing. This routine alters HV by ten per cent (the reach factor either raises or lowers it) and computes a new point unless the end of the reach is encountered. Depth in the D/S direction is allowed to rise no higher than DN in critical and steep reaches and DC in mild reaches. Depth in the U/S direction is allowed to drop no lower than DN in mild and critical reaches and DC in steep reaches. If this is a D/S mild reach or an U/S steep reach then DN is set equal to DC to make HVN (velocity head based on DN) equal to HVC (velocity head based on DC) so in processing DC and HVC will appear to be DN and HVN.

1.2)

Compute HVN. If this is U/S processing set DJI, WSJI, and INVJI (the JI suffix will mean the current intermediate point being processed, JIP will mean the previous intermediate point processed) equal to D2, WS2, and INV2 and set station factor (a value either +1 or -1 to add or subtract the length of reach from the last station processed) to +1 because stations will be increasing in value while going from the D/S end to the U/S end of the reach in this subroutine. If this is D/S processing set DJI, WSJI, and INVJI equal to D1, WS1, and INV1 and set station factor to -1 because stations will be decreasing in value while going from the U/S end to the D/S end of the reach in this subroutine. Set DJIP equal to DJI, HVJIP equal to HVJI, and STAJIP equal to STAJI (the station at the intermediate point). Calculate HVJIP, SFJIP, and EJIP. Compute RJIP (hydraulic radius based on DJIP) by dividing AJIP by WPJIP.

1.3)

Compute a new HVJI (to establish a new intermediate point) by multiplying HVJIP by the reach factor. If the reach factor is 1.1 and HVJI is less than HVN go to the next paragraph. If the reach factor is 1/1.1 and HVJI is greater than HVN go to paragraph 1.4. Otherwise, HVJI has dropped below HVN causing DJI to rise higher than DN in D/S processing or HVJI has risen above HVN causing DJI to drop below DN in U/S processing. This cannot occur in the reach. Set DJI equal to DN and HVJI equal to HVN. Go to paragraph 1.5 and do the processing on the last intermediate point in the reach.

1.4)

Compute VJIX (velocity) as follows:

VJIX = (HVJI * 2 * G) **.5.

Compute AJIX (area based on VJIX) by dividing Q by VJIX. Use function DEPTH to compute the real DJI based on AJIX. This DJI is now the correct depth at the new intermediate point. Compute EJI. If DJI equals DN in U/S processing then lower DJI by .001 so the top (if the channel is closed) will not distort the computation of SFJI. Compute SFJI and RJI. If DJI was lowered by .001 reset DJI equal to DN. Compute SFAVE as the average of SFJI and SFJIP. Compute DELLJI (the length of reach covered between two consecutive intermediate points on the W.S. profile) as follows:

DELLJI = ((EJI - EJIP) / (SO - SFAVE)).

Add DELLJI to DELLT (the total length of all the DELLJI calculated so far). If DELLT is not greater than DELL (the intermediate point is within the limits of the reach) go to paragraph 1.5. Otherwise, the intermediate point to be computed went past the end of the channel. Compute DELLB by subtracting the DELLTP (DELLT at the previous intermediate point) from DELL. Set the last known depth equal to DJIP. Compute the remaining length of the reach by taking the absolute value of DELL minus DELLTP. Compute the station at the end of the reach as follows:

STAJI = STAJIP + (Station Factor * DELLB)

Use subroutine BERNLI to calculate the depth at the end of the channel section. Set DJI equal to the depth calculated in subroutine BERNLI. Compute AJI, FJI, and SFJI. Write the data to the appropriate U/S or D/S file. This was either the D/S end of the reach in D/S processing or the U/S end of the reach in U/S processing. Go to paragraph 1.8 to finish processing.

1.5)

Compute the station for this HVJI as follows:

STAJI = STAP + (DELLJI * Station Factor)
where STAP is the station at the previous HVJI.

Compute INVJI as follows:

INVJI = INVP + (SO * DELLJI * Station Factor).

Compute FJI (force based on the depth DJI). Write the data to the U/S or D/S file depending on whether this is U/S or D/S processing. If there have been 50 intermediate points processed in this reach (the U/S and D/S files are keyed so 50 is the maximum number of records which can be written to the U/S or D/S file for a single element) then there is an error message printed and processing is stopped. If DJI equals DN and DELLT equals DELL (the end of the reach) go to paragraph 1.8. If DJI equals DN and DELLT is less than DELL (this is not the end of the reach but an intermediate point) then go to paragraph 1.6. Otherwise, DJI is less than DN. Set HVJIP, DJIP, SFJIP, and EJIP equal to HVJI, DJI, SFJI, and EJI, respectively. Go to paragraph 1.3 to start the processing for the next intermediate point.

1.6)

The last intermediate point computed had DJI equal to DN. If this is U/S processing in a steep reach or if this is D/S processing in a mild reach go to paragraph 1.7. Otherwise, the depth will remain at DN through the remainder of the reach unless pressure flow is encountered. Compute the following data for the end of the reach with DJI equal to DN as follows:

DELLJI = (DELL - DELLT), the length left in the reach,
STAJI = STAJIP + (DELLJI * Station Factor),
INVJI = INVJIP + (DELLJI * SO * Station Factor),
WSJI = INVJI + DJI,
HVJI = ((Q / AJI) **2) / (2 * G),
where JI is the D/S end in D/S processing, and
JI is the U/S end in U/S processing.

If this is a closed section in U/S processing going under pressure (DN is not less than DH) then compute SFJI for pressure (using the top of the channel in its calculation), substitute the data at the intermediate point as the D/S end of the reach and return control to subroutine RCHUS to do the computation of the remainder of the reach under pressure flow. Otherwise, write the data to the appropriate U/S or D/S file and go to paragraph 1.8 to finish processing.

1.7)

This is either a mild reach in D/S processing and DJI has risen to DC before the end of the reach or this is a steep reach in U/S processing and DJI has dropped to DC before the end of the reach. The processing is stopped in the middle of the reach but the data computed up to this point is to be retained on the appropriate U/S or D/S file. If this is D/S processing set D2 equal to zero and return processing control to subroutine RCHDS. If this is U/S processing set D1 equal to zero and return processing control to subroutine RCHUS.

1.8)

The end of the reach has been computed and written to the appropriate U/S or D/S file. If this is U/S processing set D1 equal to DJI and return control to subroutine RCHUS. If this is D/S processing set D2 equal to DJI and return control to subroutine RCHDS.

1.1)

Subroutine BERNLI is called from subroutines RCHUS, RCHDS, and REACH to use Bernoulli's equation to calculate a depth at the end of the reach given the last known depth (DJI) before the end of the reach and DELLB (the length from the point of DJI to the end of the reach). If intermediate points were calculated in the reach then D2 will equal DJI in U/S processing and D1 will equal DJI in D/S processing. Bernoulli's uses iteration so any values ending with the suffix T will be those values based on the iterated depth (DT). In U/S processing DT is set to D2 and in D/S processing DT is set to D1. FACT will be the station factor (to establish whether station numbers are increasing or decreasing through this type of reach processing) which is equal to +1 in U/S processing and -1 in D/S processing.

1.2)

Compute AT, WPT, and SFT. Compute VSQT (velocity squared for DT) as follows:

VSQT = (Q / AT) **2.

Compute CONSTD (the desired value the DT at the end of the reach will produce in Bernoulli's equation) as follows:

CONSTD = DT + (VSQT / (2 * G)) + ((SFT * DELLB * FACT) / 2)
- (DELLB * SO * FACT).

If DT equals DN a message is printed and the depth at the end of the reach (D2 in D/S processing or D1 in U/S processing) is set equal to DN and processing control is returned to the appropriate routine. Otherwise, set DL (the minimum value between DT and DN) and DU (the maximum value between DT and DN). Compute CONSTU (the value derived from Bernoulli's equation based on DU) as follows:

CONSTU = DU + (VSQU / (2 * G)) - ((DELLB * SFU * FACT) / 2)
where variables with suffix U are based on DU.

Compute CONSTL the same way only based on DL. If CONSTL does not equal CONSTU go to paragraph 1.3. Otherwise, CONSTL equals CONSTU so the depth does not fluctuate to the end of the reach. Set D2 equal to D1 in D/S processing or D1 equal to D2 in U/S processing. Return processing control to the appropriate subroutine.

1.3)

If CONSTD is not between CONSTU and CONSTL then go to paragraph 1.6. If CONSTL is greater than CONSTU then CONSTT decreases as DT

DT increases, otherwise, CONSTL is less than CONSTU so CONSTT increases as DT increases. For the following processing the suffix P will mean previous value and T will mean current test value. Set DT equal to DL, CONSTP equal to CONSTL, and set the increment value for iterating depths equal to the quantity DU minus DL divided by ten.

1.4)

Compute AT, WPT, VSQT, and SFT. Compute CONSTT as follows:

CONSTT = DT + (VSQT / (2 * G)) - ((DELLB * SFT * FACT) / 2).

If CONSTT equals CONSTD then the current DT has solved the Bernoulli's equation, so set D1 equal to DT if this is U/S processing or set D2 equal to DT if this is D/S processing, and return processing control to the appropriate subroutine. If CONSTT has crossed the CONSTD value then DT is greater than the DT which will solve Bernoulli's equation so go to paragraph 1.5. Otherwise, CONSTT is still approaching the CONSTD value. Set DP equal to DT and CONSTP equal to CONSTT. Increment DT giving a new DT and go back to the beginning of paragraph 1.4 to process the new DT.

1.5)

Divide the current increment by ten to create a new increment. If the new increment is less than .0001 then assume the last DT has solved the Bernoulli's equation, so set D1 equal to DT if this is U/S processing or set D2 equal to DT if this is D/S processing, and return processing control to the appropriate subroutine. Otherwise, the DT just used is too big to produce a CONSTT equal to CONSTD so set DT equal to DP and CONSTT equal to CONSTP. Increment DT to create a new DT and go back to paragraph 1.4 to start processing on the new DT.

1.6)

The CONSTD needed cannot be computed by any depth between DL and DU. Bernoulli's equation cannot be solved. If this is D/S processing and D1 is less than DN and CONSTD is less than CONSTU then set D2 equal to DN. If this is D/S processing and D1 is greater than DN and CONSTD is greater than CONSTL set D2 equal to DN. If this is U/S processing and D2 is less than DN and CONSTD is greater than CONSTU set D1 equal to DN. If this is U/S processing and D2 is greater than DN and CONSTD is less than CONSTL set D1 equal to DN. If this is D/S processing and D2 equals zero then set D2 equal to D1. If this is U/S processing and D1 equals zero then set D1 equal to D2. Return processing control to the appropriate subroutine.

1.1)

Subroutine JUMPR locates the station and force at the intersection of the U/S and D/S force curves for the hydraulic jump in a reach element. This subroutine is called from subroutine RCHJP. Read the D/S end from the D/S processing and the D/S end from the U/S processing. Read the next point on the U/S file and the last point before the D/S end on the D/S file. Both U/S and D/S files are read for the reach from the D/S end to the U/S end. The prefix D will mean from the D/S file and U will mean from the U/S file. The suffix X will represent the point being tested closest to the D/S end and suffix Y will be the next point which is closest to the U/S end. Thus the first time the D/S end will be referred to as DFX, DSTAX, UFX, and USTAX for the force and station on the U/S and D/S profile. The next point on the files going toward the U/S end will be referred to as DFY, DSTAY, UFY, and USTAY for the force and station on the U/S and D/S profile.

1.2)

Compute the station (STAPT) at the point of intersection based on the stations and forces of the current four points (two from the U/S file and two from the D/S file) being tested. This is done by computing XMUS and XMDS which are as follows:

XMUS = (UFX - UFY) / (USTAX - USTAY), and
XMDS = (DFX - DFY) / (DSTAX - DSTAY).

Then compute CXMUS and CXMDS as follows:

CXMUS = (-XMUS * USTAY) + UFY, and
CXMDS = (-XMDS * DSTAY) + DFY.

If XMUS or XMDS was equal to zero go to paragraph 1.3. Otherwise, compute STAPT as follows:

STAPT = (CXMUS - CXMDS) / (XMDS - XMUS).

If STAPT is greater than USTAY or less than USTAX (STAPT is not between the two current U/S stations being used) go to paragraph 1.3. If STAPT is greater than DSTAY or less than DSTAX (STAPT is not between the two current D/S stations) go to paragraph 1.3. Otherwise, STAPT is between the two U/S stations and the two D/S stations so it is the correct station for the intersection of the force curves. Compute FPT (the force at the intersection of the curves) as follows:

FPT = (XMUS * (STAPT - USTAY) + UFY.

Save the keys for the points on the U/S and D/S files between which STAPT is located. Return processing control to subroutine RCHJP.

1.3)

If USTAY is not less than DSTAY go to paragraph 1.4. Otherwise, another point on the U/S file must be read. Set USTAX equal to USTAY and UFX equal to UFY. If there are no more U/S or D/S file records to go to paragraph 1.5. If there are no more U/S file records go to paragraph 1.4. Otherwise, read the next record toward the U/S end of the reach to obtain new USTAY and UFY values. Go to paragraph 1.2 to continue processing with the new values.

1.4)

Another point on the D/S file must be read. Set DSTAX equal to DSTAY and DFX equal to DFY. If there are no more U/S or D/S file records go to paragraph 1.5. If there are no more D/S file records go to paragraph 1.3. Otherwise, read the next record toward the U/S end of the reach to obtain new DSTAY and DFY values. Go to paragraph 1.2 to continue processing with the new values.

1.5)

No intersection of the forces could be found among the points given on the U/S and D/S files. An error in processing has occurred. A message is printed and all processing is stopped.

1.1)

Subroutine PPMDEP uses FPT (the force at a hydraulic jump) to compute depths by iteration to find U/S and D/S depths which produce forces equal to FPT. This subroutine is called by subroutine RCHJP which supplies FPT, DU (the upper limit depth for the iteration), and DL (the lower limit depth for the iteration). Compute the increment value for the depths by dividing the difference between DU and DL by ten. Compute FU and FL (forces based on depths DU and DL). Use subscript T for values based on DT (the iterated test depth), and subscript P for values from DP (the previous DT). Set DP equal to DL and FP equal to FL. Set DT equal to DL. If FL does not equal FPT and FU does not equal FPT go to paragraph 1.2. Otherwise if FU equals FPT set the depth at the hydraulic jump equal to DU, or if FL equals FPT set the depth at the hydraulic jump equal to DL. Return processing control to subroutine RCHJP.

1.2)

If FPT is not between FL and FU then no processing can be done, an error has occurred and all processing is stopped. Otherwise FPT is between FU and FL. If FU is greater than FL then the force will increase as depth increases, otherwise FU is less than FL so force decreases as depth increases.

1.3)

Compute FT, the test force based on DT. If FT equals FPT then DT is the correct depth at the hydraulic jump and processing control is returned to subroutine RCHJP. If FT has crossed the value of FPT then go to the next paragraph. Otherwise FT is still approaching FPT, so set DP equal to DT and set FP equal to FT because DT is not high enough to produce a force equal to FPT. Add the increment to DT to produce a new DT. If DT is not greater than DU go back to the beginning of this paragraph and process the new DT. Otherwise DT exceeded DU before FT reached FPT. An error in the processing has occurred as the depth to produce a force equal to FPT is not between DL and DU. Print an error message and stop all processing.

1.4)

DT is greater than the depth which will produce the force equal to FPT. Divide the increment by ten giving a new increment. If the new increment is less than .0001 then assume DT as the correct depth at the hydraulic jump and return processing control to subroutine RCHJP Otherwise set DT equal to DP and set FT equal to FP. Add the new increment to DT to get a new DT and go back to paragraph 1.3 to process the new DT.

1.1)

Subroutine ELMCHG establishes Q, DC, DN, DH and SO values for all elements when required. This processing is the first to be done before any U/S or D/S computations can be made. In all elements DH is set to the channel height for closed sections and to the channel height plus ten feet in all open channels. Also in all the elements DC is computed at each end after the Qs at the U/S and D/S ends have been established. The processing in ELMCHG begins at the system headworks and proceeds to the system outlet. All processing codes (U/S, D/S and Jump) are set to invalid (zero) so no computations are assumed. The element file is first read (record number 202) to find the number of elements to be processed, which is also the record key for the headworks record. Add 1 to this number to start processing.

1.2)

Decrement 1 from the element file key. Read the element record to establish the type of element to be processed. Continue processing according to the element type as follows:

Element type 1 : go to paragraph 1.3 for the System Outlet.
Element type 2 : go to paragraph 1.4 for the Reach.
Element type 3 : go to paragraph 1.5 for the Junction.
Element type 4 : go to paragraph 1.6 for the Bridge Entrance.
Element type 5 : go to paragraph 1.7 for the Bridge Exit.
Element type 6 : go to paragraph 1.8 for the System Headworks.
Element type 7 : go to paragraph 1.9 for the Transition.
Element type 8 : go to paragraph 1.6 for the Wall Entrance
Element type 9 : go to paragraph 1.7 for the Wall Exit.

When there are no more element types to be processed (the system outlet should be the last element) go to paragraph 2.1

1.3)

This is the system outlet. Set all processing codes to invalid. If the element previous to this was an outlet or a headwoks (not allowed) print a message and add one to the error counter. Set DH, set Q equal to the Q from the D/S end of the previous element, and call function DCRIT to compute DC. There is no DN or SO in the outlet. Write the outlet record with the updated data. This should be the last element to be processed, so go to paragraph 2.1.

1.4)

This is the reach element. Set all processing codes to invalid. If the element previous to this was an outlet (outlet should be the last element) print a message and add one to the error counter. Set DH. Compute DELL by subtracting the station of the previous element from the station of the reach. Set SO equal to zero. Set Q equal to the Q from the D/S end of the previous element. If DELL is greater than zero compute SO as follows:

SO = (INV - INVP) / DELL
where INVP is the invert from the previous element.

If SO is greater than zero then compute U (the value to base DN on) as follows:

U = (Q * XN) / (1.486 * (SO **.5))
where XN is the Manning's factor in the reach.

If SO is greater than zero call function DNORM to compute DN based on the value U, otherwise set DN equal to channel height. Call function DCRIT to compute DC based on the Q value. If DN or DC is not greater than zero then print a message and add one to the error counter. If DC equals the channel height of a closed section lower DC by .001. Compute SC which is SF based on the depth DC. If DC was lowered add .001 to DC to regain its original value. Write the reach record with the updated data. Go back to paragraph 1.2 to process the next element.

1.5)

This is the junction element. Set all the processing codes to invalid. Set Q1 equal to the Q from the D/S end of the previous element. Reset Q3 and Q4 (the Q values in the laterals if present) by multiplying them by QFACT (which is the lateral Q adjustment from Q input, if it was not given the value 1 is assumed so Q3 and Q4 will not change). Compute Q2 by adding Q1, Q3 and Q4 together. Compute DELL by subtracting the station from the previous element from the station of the junction. If DELL is greater than zero compute SO as follows:

SO = (INV1 - INV2) / DELL,

otherwise set SO equal to zero. Set DH at both ends of the junction. Use function DCRIT to compute DC1 and DC2 (based on Q1 and Q2). There is no DN1 or DN2 in the junction. Write the junction record with the updated data and go back to paragraph 1.2 to process the next element.

1.6)

This is a bridge or wall entrance element. Set all the processing codes to invalid. If the previous element was a bridge entrance, bridge exit, wall entrance or wall exit (not allowed) then write a message and add one to the error counter. Set DH1 and DH2. Set Q1 and Q2 equal to the Q from the D/S end of the previous element. Use function DCRIT to compute DC1 and DC2. There is no DN1 or DN2 in a bridge or wall entrance. Write the bridge or wall entrance record with the updated data and return to paragraph 1.2 to process the next element.

1.7)

This is a bridge or wall exit element. Set all the processing codes to invalid. If the previous element was a bridge entrance, bridge exit, wall entrance, or wall exit (not allowed) then write a message and add one to the error counter. Set DH1 and DH2 values. Set Q1 and Q2 equal to the Q from the D/S end of the previous element. Use function DCRIT to compute DC1 and DC2. Write the bridge or wall exit record with the updated data and return to paragraph 1.2 to process the next element.

1.8)

This is a system headworks element. Set all the processing codes to invalid. If QFACT (the Q factor to adjust the Q3 and Q4 values in all the junction laterals) equals zero set QFACT equal to one. Set Q equal to the Q value supplied on the Q card input. Set the DH value. Use function DCRIT to compute DC. There is no DN or SO in the headworks. Write the headworks record with the updated data and return to paragraph 1.2 to process the next element.

1.9)

This is a transition element. Set all the processing codes to invalid. If the previous element was an outlet (not allowed) print a message and add one to the error counter. Set Q1 and Q2 equal to the Q from D/S end of the previous element. Set DH1 and DH2. Compute DELL by subtracting the station from the previous element from the station of the transition. If DELL is greater than zero compute SO as follows:

SO = (INV1 - INV2) / Dell,

otherwise set SO equal to zero. Use function DCRIT to compute DC1 and DC2. There is no DN1 or DN2 in a transition. Write the transition record with the updated data and return to paragraph 1.2 to process the next element.

2.1)

If the error counter is greater than zero then the element file is incorrect and any processing would be inaccurate so a message is printed and all processing is stopped. If no errors were encountered then processing control is returned to the MAIN routine so D/S computation may begin.

1.1)

Subroutine WRITEN prints the composite profile and creates the plot table for plotting of the profile. The element file is read from the system outlet to the system headworks and is used to obtain the U/S, D/S, and jump processing codes. The U/S and D/S files are read to obtain the computed data at the individual stations (stations continually increasing).

1.2)

The data from the current station and the data from the succeeding station are read. The velocity, velocity head, water surface elevation, and energy grade line (specific energy in a regular open channel reach will also be computed if requested) are computed for the current station and printed. The friction slope average, length, and head loss due to friction slope are computed between the current station and the succeeding station. If this is a reach this data is printed. In any other element type the element type is printed instead of the length, and if this is a hydraulic jump in a reach it is printed instead of the non-existent (at a jump) length. The data is then written to a plot file to be held for plotting. Then the next station is processed.

NOTE: Since U/S and D/S ends are contained for all elements (if there is no processing for an element a message wil be written) there will be duplication from the U/S end of the current element to the D/S end of the succeeding element unless there is a change in profile between elements. Therefore a check is made to see if the station and depths match in succeeding records, and if there is a match then the current record is ignored as it will be assumed to be identical to the next record. This will cause problems when reach, junction, and/or transition elements are adjacent to each other and a different Manning's factor is used for each element. This causes different friction slopes for the same depths at the same station so it should be remembered that only the record in the element closest to the system headworks will be printed. If it is desired to see all the records created use a trace switch card set to 2 so the U/S and D/S profiles will also be printed.

PLOTR

1.1)

Subroutine PLOTR determines the number of points to be plotted and the number of lines to be printed in the plot. If there are not more than two points to be plotted then no plot is written, otherwise subroutine PLOT is called to print the plotted data. After the data is plotted processing control is returned to the MAIN routine to see if there is any more Q data to process another profile.

1.1)

Subroutine PLOT does the plot printing of the water surface profile. The data which is plotted is as follows:

I - Invert elevation,
C - Critical depth,
W - Water surface elevation,
H - Channel height,
E - Energy grade line,
X - Two or more values at the same point,
B - Bridge entrance or exit,
Y - Wall entrance or exit.

The data is printed with depth values for the X axis and station values for the Y axis. When all the data is plotted processing control is returned to subroutine PLOTR.

ERRII, PRINTO, and HEADER

1.1)

Subroutine ERRII prints the error messages encountered in subroutine ELMCHG according to the type of error which exists. This is done by passing a value from subroutine ELMCHG to subroutine ERRII to indicate the error message number to be printed.

2.1)

Subroutine PRINTO prints the element file after the preliminary calculations have been made if the trace switch is equal to 1. The trace switch is set by the first input card into the computation program. The element file will be printed only as 80 characters per line so it will be necessary to utilize the file record layouts to locate any specific values.

3.1)

Subroutine HEADER prints the heading for the combined water surface profile and plot printouts. This data comes fromt he title record of the element file which was created from the title cards in the editing phase.

1.1)

Function FORCE calculates force values according to depths and section types.

Processing is done as follows:

```
Section 1 : Open trapezoid,   see process 2.
Section 2 : Open rectangle,   see process 1.
Section 3 : Closed trapezoid, see process 2.
Section 4 : Pipe,             see process 3.
Section 5 : Open irregular,   see process 4.
Section 6 : Closed irregular, see process 4.
```

where force equals pressure plus momentum.

1) Open rectangular section.

```
P = ((D **2) / 2) * (W - (NOP * BP)),
M = (Q **2) / (A * G),
F = P + M.
```

2) Open or closed trapezoid section.

```
P = ((D **2) / 2) * (W - (NOP * BP)) + (D * ((ZR + ZL) / 3)).
```

If this is a closed trapezoid and D is greater than DH then compute PP for pressure flow adjustment to section pressure as follows:

```
PP = (((D - H) **2) / 2) * (((2 * H) + D) * ((ZL + ZR) / 3) + (W - (NOP * BP))
```

otherwise PP is equal to zero.

```
P = P - PP,
M = (Q **2) / (A * G),
F = P + M.
```

3) Pipe section.

```
R = (W / 2)
C = (D / R)
```

If D is greater than W then

```
P = (D - R) * (3.1416 * (R **2))
```

otherwise

```
P = ((R **3) / 3) * ((((2 * C) - (C **2)) **.5) * ((C **2) - (2 * C) + 3)
     + (3 * (C - 1) * (1.570795 + ARCSIN(C - 1)))).
M = (Q **2) / (A * G),
F = P + M.
```

4) Irregular section.

Use function FORCEI to compute P.

```
M = (Q **2) / (A * G)
F = P + M
```

1.1)

Function FORCEI computes the pressure in irregular shaped channels. A table is built containing all the (X, Y) coordinates for the channel which describe the channel up to the current depth of flow. The number of points in the table is called NPNTS and they are assembled counter clockwise beginning with the coordinate where the maximum Y value is located. If the maximum Y value occurs more than once then the beginning is the maximum Y value with the minimum X value. Then P (pressure) is calculated as the summation of pressure between the individual coordinates from I = 2 to I = NPNTS + 1 in the following formula:

```
P = (((X(I) - X(I-1)) / 2) * ((D - Y(I)) **2))
    + ((X(I) - X(I-1)) * (Y(I) - Y(I-1)) / 2) *
    (D - Y(I) + ((Y(I) - Y(I-1)) / 3).
    where NPNTS + 1 equals the first coordinate.
```

Compute PP (pier pressure) by the summation of the pressure on the individual piers as follows:

```
(((D - YP(I)) **2) * (BOP / 2)
```

for all piers whose base (YP) is beneath the depth of flow, otherwise the pier is not included in computation of PP. Then the total pressure equals as follows: P = P - PP

1.1)

Subroutine FORCEM iterates depths until a desired force is reached for a bridge exit element.

1.2)

If FORCEM is called from BREXDS then depth is continually decremented from DC2 (the upper limit depth) to DL (the lower limit depth) until a force is found which equals the desired force at the D/S end of the bridge exit. If the desired force is crossed the decrement is made smaller and processing is started again at the depth previous to when the desired force was crossed. When the desired force is reached or the decrement becomes insignificantly small then D2 is set equal to the depth at that point and control is returned to subroutine BREXDS. Force is computed for the D/S end in the bridge exit based on inserting the piers from the U/S end into the D/S end as follows (where suffix T means a value bases on D):

A) For an irregular section:

```
M = (Q2 **2) / (AT * G),
P is computed in FORCEI using cross section points
  at the D/S end with pier points from the U/S end,
F = M + P.
```

B) For a regular section:

```
M = (Q2 **2) / (AT * G),
PP = (NOP1 * BP1) * ((D **2) / 2),
P = (((D **2) / 2) * (W2 + (D * ((ZR2 + ZL2) / 3)))),
P = P - PP,
F = M + P.
```

NOTE: the section in a bridge exit cannot be a pipe.

1.3)

If FORCEM is called from BREXUS then depth is continually incremented from DC1 (the lower limit depth) to D2 (the upper limit depth) until a force is found which equals the desired force at the U/S end of the bridge exit. If the desired force is crossed the increment is made smaller and processing is started again at the depth previous to when the desired force was crossed. When the desired force is reached or the increment becomes insignificantly small then D1 is set equal to the depth at that point and processing control is returned to subroutine BREXUS. The computation of force for subroutine BREXUS is done by function FORCE.

FORCEF

1.1)

Function FORCEF iterates depths until a desired force is reached for a bridge entrance element.

1.2)

If FORCEF is called from BRENDS then depth is continually decremented from DC2 (the upper limit depth) to DL (the lower limit depth) until a force is found which equals the desired force at the D/S end of the bridge entrance. If the desired force is crossed then the decrement is made smaller and processing is started again at the depth previous to when the desired force was crossed. When the desired force is reached or the decrement becomes insignificantly small then D2 is set equal to the depth at that point and processing control is returned to subroutine BRENDS. The computation of force for subroutine BRENDS is done by function FORCE.

1.3)

If FORCEF is called from BRENUS then depth is continually incremented from D2 (the lower limit depth) to DH1 (the upper limit depth). The piers from the D/S end of the bridge entrance are inserted into the U/S end and then the force is found. If the desired force is crossed the increment is made smaller and processing restarts at the depth previous to when the desired force was crossed. Iteration continues until the desired force is reached or the increment becomes insignificantly small, then D1 is set equal to that depth and processing control is returned to subroutine BRENUS. The computation of force for the BRENUS subroutine is done as follows (where suffix T means a value based on the iterated depth):

A) For irregular sections:

P is calculated by function FORCEI using the cross section points at the U/S end with the pier points from the D/S end,

M = (Q2 **2) / (AT * G),

The area of the piers is calculated by the summation of the area of the individual piers as follows:

AP = ((D - YP(I)) * BP2)
where YP is the Y value at the base of the pier, if the base of the pier is not under water the pier is ignored,

F = (M * ((AT - (PF * AP)) / AT)) - P
where PF is the inputted pier factor.

B) For regular sections:

```
M  = (Q2 **2) / (AT * G),
P  = (((D **2) / 2) * (W1 + (D * ((ZR1 + ZL1) / 3))),
AP = (NOP2 * BP2) * D
PP = (NOP2 * BP2) * ((D **2) / 2)
F  = (M * ((AT - (PF * AP)) / AT) + (P - PP))
```

where PF is the pier factor inputted for the bridge entrance.

NOTE: the section in a bridge entrance cannot be a pipe.

1.1)

Subroutine FORCEW iterates depths until a desired value is reached for wall exit elements.

1.2)

If FORCEW is called from WEXTDS then depth is continually incremented from DL (the lower limit depth) to DC2 (the upper limit depth) to produce a desired energy value at the D/S end of the wall exit. If the desired energy is crossed the increment is made smaller and processing is started again at the depth just before the previous depth to when the desired energy was crossed (because the energy value curve may double back on itself). When the desired energy is reached or the decrement becomes insignificantly small then D2 is set equal to the depth at that point and processing control is returned to subroutine WEXTDS. The energy value (E) is computed as follows (where suffix T is a value based on the iterated depth):

VT = (Q2 / AT)
HVT = (VT **2) / (2 * G)
E = D + HVT + (ABS((V1 **2) / (2 * G)) - ((VT **2) / (2 * G)))
where ABS means absolute value.

1.3)

If FORCEW is called from WEXTUS then depth is continually incremented from DC1 (the lower limit depth) to D2 (the upper limit depth) to produce a desired energy at the U/S end of the wall exit. If the desired energy is crossed the increment is made smaller and processing is started again at the depth previous to when the desired energy was crossed. When the desired energy is reached or the increment becomes insignificantly small then D1 is set equal to the depth at that point and processing control is returned to subroutine WEXTUS. The energy value (E) is computed as follows (where suffix T is a value based on the iterated depth):

VT = (Q1 / AT),
HVT = (VT **2) / (2 * G),
E = D + HVT + (ABS((V2 **2) / (2 * G)) - ((VT **2) / (2 * G)))
where ABS means absolute value.

1.1)

Function FWALL iterates depths until a desired force is reached in a wall entrance element.

1.2)

If FWALL is called from WENTDS then depth is continually decremented from DC2 (the upper limit depth) to DL (the lower limit depth) to produce a force equal to the desired force at the D/S end of the wall entrance. If the desired force is crossed the decrement is made smaller and processing is started again at the previous depth before the desired force was crossed. When the desired force is reached or the decrement becomes insignificantly small then D2 is set equal to the depth at that point and processing control is returned to subroutine WENTDS. The computation of force for subroutine WENTDS is done in function FORCE.

1.3)

If FWALL is called from WENTUS then depth is continually incremented from D2 (the lower limit depth) to DH1 (the upper limit depth) to produce a force equal to the desired force at the U/S end of the wall entrance. If the desired force is crossed the increment is made smaller and processing is started again at the previous depth before the desired force was crossed. When the desired force is reached or the increment becomes insignificantly small then D1 is set equal to the depth at that point and processing control is returned to subroutine WENTUS. The force for the U/S end of the wall entrance is computed as follows (where AT is the area in the U/S end based on the iterated depth):

M = (Q1 **2) / (AT * G),
A2 = area of flow at the D/S end based on the iterated depth,
F2 = force of flow at the D/S end based on the iterated depth,
AOBST (area of obstruction) = AT - A2,
P = F2 - ((Q2 **2) / (A2 * G),
F = (M * ((AT - AOBST) / AT) + P.

1.1)

Function FORCEJ is used in junction elements when the calculated depths in these elements cause the energy in the U/S end of the element to be less than the energy in the D/S end of the element. Since this cannot be allowed to happen the depths are iterated until the energies in both ends of the element are equal. In D/S processing DU (the upper limit depth) is set equal to DC2, DL (the lower limit depth) is set equal to the previously computed D2, and MU (the desired energy) is set equal to the energy in the U/S end. In U/S processing DU and MU are set equal to the energy in the D/S end and DL is set equal to the previously computed D1. Set DT (the test depth) equal to DL and set the increment value equal to 1. If this is U/S processing go to paragraph 1.3.

1.2)

For D/S processing (energy drops as depth increases in the D/S end) compute MT (the test energy) as follows:

MT = DT + HVT + (DELL * SFT / 2).

If MT equals MU then this is the correct depth in the D/S end so return control to the appropriate subroutine. If MT is greater than MU then DT is not high enough so add the increment to DT to obtain a new DT; and if DT is less than DU go back to the beginning of this paragraph to process the new DT. Otherwise DT is too large so divide the increment by ten to get a new smaller increment. If the new increment is insignificantly small then this DT is assumed to be the correct depth and control is returned to the appropriate subroutine. Otherwise set DT equal to the previous DT and add the new increment to it to produce a new DT and go back to the beginning of this paragraph to process the new DT.

1.3)

For U/S processing (energy rises as depth rises in the U/S end) compute MT (the test enrgy as follows:

MT = DT + HVT + (DELL * SFT / 2) + H.

If MT equals MU then this is the correct depth in the U/S end so return control to the appropriate subroutine. If MT is less than MU then DT is not high enough, add the increment to DT to obtain a new DT; and if DT is less than DU go back to the beginning of this paragraph to process the new DT. Otherwise DT is too large so divide the increment by ten to get a new smaller increment. If the new increment is insignificantly small then this DT is assumed to be the correct depth and control is returned to the appropriate subroutine. Otherwise set DT equal to the previous DT and add the new increment to it to produce a new DT and go back to the beginning of this paragraph to process the new DT.

1.1)

Function TRANSB iterates depths until an energy is found which equals the known energy in the opposite end of a transition element. If TRANSB is called from TRNSUS (U/S transition processing) and it has been determined that the desired energy is in the nose of the energy curve go to paragraph 1.5. If TRANSB is called from TRNSUS and the desired energy is outside the nose of the energy curve, then go to paragraph 1.4. Otherwise TRANSB was called from TRNSDS (D/S transition processing). If the desired energy is in the nose of the energy curve go to paragraph 1.3.

1.2)

D/S processing outside the nose of the energy curve. Depth is continually incremented from DL (the lower limit depth) to DC2 (the upper limit depth) until an energy is found which is equal to the energy in the U/S end of the transition. The test energy continually decreases as the test depth increases. If the desired energy is crossed the increment is made smaller and iteration is started again at the previous test depth before the desired energy was crossed. When the increment becomes insignificantly small or the desired energy is obtained then D2 is set equal to the test depth at that point and processing control is returned to subroutine TRNSDS. The formula used to calculate energy is as follows (with suffix T representing values based on iterated depths):

BT = DT + HVT + (k * (HVT -HV1)) + (DELL * SFT) /2))
where HV1 is the velocity head in the U/S end
and K = 0.1 if the test velocity exceeds the U/S velocity
or K = 0.2 if the test velocity is less than the U/S velocity.

1.3)

In D/S processing in the nose of the energy curve the test depth is set to DC2 and continually decremented. The energy will continually decrease as the test depth decreases. If the test energy starts increasing (leaving the nose of the curve) or if the desired energy is passed, then the decrement is made smaller and iteration resumes with the test depth previous to the previous test depth (because the nose doubles back on itself). If the decrement becomes significantly small and the desired energy was never passed then D2 cannot be computed. Otherwise D2 is set equal to the test depth when the decrement is insignificantly small or the desired energy is reached and processing control is returned to subroutine TRNSDS. The test energy is calculated the same as in paragraph 1.2.

1.4)

U/S processing outside the nose of the energy curve. Depth is continually incremented from DC1 (the lower limit depth) to DH1 (the upper limit depth) and the energy will increase as the test depth increases. If the test energy never reaches the desired energy (energy in the D/S end) before DH1 is reached then control is returned to TRNSUS for pressure flow. If the desired energy is crossed the increment is made smaller and iteration resumes at the test depth previous to where the desired energy was crossed. When the increment becomes insignificantly small or the desired energy is reached then D1 is set equal to the test depth at that point and processing control is returned to subroutine TRNSUS. The formula used to calculate energy is as follows (with suffix T representing values based on the iterated depths):

BT = DT + HVT + H + (K * (HVT - HV2)) - ((DELL * SFT) / 2)
where HV2 is the velocity head in the D/S end
and K=0.1 if the test velocity is less than the D/S velocity
or K = -0.2 if the test velocity exceeds the D/S velocity.

1.5)

In U/S processing in the nose of the energy curve the test depth is set at DC2 and continually incremented. The energy will continually decrease as the test depth increases. If the test energy starts increasing (leaving the nose of the curve) or if the desired energy is passed, then the increment is made smaller and iteration resumes with the test depth previous to the previous test depth (because the nose doubles back on itself). If the test depth reaches DH1 or if the increment becomes insignificantly small and the desired energy was never crossed, then D1 cannot be computed. Otherwise D1 is set equal to the test depth when the increment is insignificantly small or the desired energy is reached and processing control is returned to subroutine TRNSUS. The test energy is calculated the same way as shown in paragraph 1.4.

1.1)

Function AREAC computes the cross section area of flow for all elements according to channel section type and depth of flow.
Processing is done as follows:

```
Section 1 : Open trapezoid,   see process 2.
Section 2 : Open rectangle,   see process 1.
Section 3 : Closed trapezoid, see process 3.
Section 4 : Pipe,             see process 4.
Section 5 : Open irregular,   see process 5.
Section 6 : Closed irregular, see process 5.
```

1) Open rectangular section.

```
BNET (net width of channel) = (W - (NOP * BP))
A = (D * BNET) + (EVAL * BNET / 24)
```

2) Open trapezoid section.

```
BNET (net width of channel) = (W - (NOP * BP))
A = (BNET * D) + ((D **2) * ((ZR + ZL) / 2)
```

3) Closed trapezoid section.

If D is less than channel height see process 2. Otherwise:

```
BNET (net width of channel) = (W - (NOP * BP))
PA (allowance for pier adjuncts at top of channel) = .25 * (1 + NOP)
A = (BNET * H) + ((H **2) * ((ZR + ZL) / 2)) - (PA)
```

4) Pipe section.

```
R = W / 2
```

If D is greater than W

```
A = 3.1416 * (R **2),
```

otherwise

```
B = - R
C = D - R
AP = ((C + (((R **2) - (C **2)) **.5)) + ((R **2) * ARCSIN(C / R))
A = AP - ((B * (((R **2) - (B **2)) **.5)) + ((R **2) * ARCSIN(B / R))
```

5) Irregular sections.

Function AREACI computes A for irregular shaped channels.

1.1)

Function AREACI computes the cross section area of flow in irregular shaped channels. A table is built containing all the (X,Y) coordinates for the channel which describe the channel up to the current depth of flow. The number of points in the table is called NPNTS and they are assembled counter-clockwise beginning with the coordinate where the maximum Y value is located. If the maximum Y value occurs more than once the beginning is the maximum Y value with the minimum X value. Then the area is calculated by the summation of the area between the individual coordinates from I = 2 to I = (NPNTS + 1) as follows:

A = (((X(I) - X(I-1)) * (D - Y(I)))

+ (((X(I) - X(I-1)) * (Y(I) - Y(I-1))) / 2))

where NPNTS + 1 equals the first coordinate.

Compute AP (pier area) by the summation of the area of the individual piers as follows:

AP = (BP * (D - YP(I)))

for all piers whose base (YP) is beneath the depth of flow, otherwise the pier is not included in the computation of AP. Then the total area is: A = A - AP.

1.1)

Function WETP computes the wetted perimeter for all elements according to channel section type and depth of flow. Processing is done as follows:

```
Section 1 : Open trapezoid,    see process 2.
Section 2 : Open rectangle,    see process 1.
Section 3 : Closed trapezoid,  see process 3.
Section 4 : Pipe,              see process 4.
Section 5 : Open irregular,    see process 5.
Section 6 : Closed irregular,  see process 5.
```

1) Open rectangular section:

```
BNET (net width of channel) = (W - (NOP * BP))
WP = (BNET + (2 * D * (NOP + 1)))
     where (NOP + 1) is used to include the side walls.
```

2) Open trapezoid section.

```
BNET (net width of channel) = (W - (NOP * BP))
WP = (BNET + (D * ((2 * NOP) + ((1 + (ZR **2)) **.5) + ((1 + (ZL **2)) **.5))))
     where 1 is added to the ZR and ZL effects to include the side walls.
```

3) Closed trapezoid section:

If D is less than channel height see process 2, otherwise:

```
BNET (net width of channel) = (W - (NOP * BP))
WP = (2 * BNET) + (H * (ZL +ZR)) +
     ((H * ((2 * NOP) + ((1 + (ZR **2)) **.5) + ((1 + (ZL **2)) **.5))))
     where 1 is added to the ZR and ZL effects to include the side walls.
```

4) Pipe section.

DIAM = (.93 * W) because above this point a doubling back effect could occur in computing WP.

If D is greater than the channel height (actual diameter) then:

```
WP = (3.1416 * W)
```

otherwise

```
WP = (W * (ARCSIN(((W / 2) - D) / (W / 2)))).
```

5) Irregular sections.

Wetted perimeter values for irregular shaped channels are computed in function WETPI.

1.1)

Function WETPI computes the wetted perimeter in irregular shaped channels. A table is built containing all the (X, Y) coordinates for the channel which describe the channel up to the current depth of flow. The number of points in the table is called NPNTS and they are assembled counter-clockwise beginning with the coordinate where the maximum Y value is located. If the maximum Y value occurs more than once the beginning is the maximum Y value with the minimum X value. Then the wetted perimeter is computed by the summation of the wetted perimeter between the individual coordinates from I = 2 to I = NPNTS as follows:

WP = (((X(I) - X(I-1)) **2) + ((Y(I) - Y(I-1)) **2) **.5)

If this a closed irregular section and D is at or above the top of the section then:

WP = WP + (((X(1) - X(NPNTS)) **2) + ((Y(1) - Y(NPNTS)) **2) **.5)
to include the top of the channel.

Compute WPP (pier wetted perimeter) by the summation of the wetted perimeter on the individual piers as follows:

WPP = ((2 * (D - YP(I))) - BP)

for all piers whose base (YP) is beneath the depth of flow, otherwise the pier is not included in the computation. Then the total wetted perimeter is: WP = WP + WPP.

1.1)

Function SF computes the friction slope for transition, junction, and reach elements. The computation is done as follows:

R23 (hydraulic radius to the 2/3 power) = ((A / WP) **.66667)

SF = (((Q * XN) / (1.486 * A * R23)) **2)
where XN is Manning's factor in the reach, transition, or junction.

1.1)

Function DNORM iterates depths from zero (the lower limit depth) to DU (the upper limit depth) to find a depth which produces a value equal to Y where:

Y = (Q * XN) / (1.486 * (SO **.5))
where XN is Manning's factor in a reach, junction, or transition.

DU is set equal to DH except in a pipe where DU equals .93 times the diameter. In the following calculations the suffix T represents values computed for the iterated depth and the suffix P represents values computed for the previous iterated depth. Set (the increment for iteration) equal to one and set the first DT equal to .0001.

1.2)

If DT exceeds DU in a pipe at any time then set DN equal to the pipe diameter and return to subroutine ELMCHG. If DT is greater than DU and this is not a pipe then go to paragraph 1.3, otherwise DT is not greater than DU so compute YT as follows:

YT = AT * ((AT / WPT) **.6667).

If YT is not less than YP then YT is still increasing so go to paragraph 1.4. Otherwise YT is less than YP which means the Y curve is doubling back on itself and DN cannot be computed any more accurately. If this is a closed irregular section and the increment is relatively insignificant then set DT equal to DH (Y curve is allowed to double back in this section only) and go to paragraph 1.3. Otherwise if the increment is relatively insignificant then set DN equal to DT and return to subroutine ELMCHG. If the increment is not relatively insignificant print a warning message that DN may be inaccurate and set DN equal to DT and return to subroutine ELMCHG.

1.3)

DT has passed the depth to solve for DN. Divide the increment by ten to make it smaller. Set DT equal to DP plus the increment. If the increment is insignificantly small set DN equal to DT and return to subroutine ELMCHG. Otherwise go back to paragraph 1.2 and process the new DT.

1.4)

If YT equals Y then set DN equal to DT and return to subroutine ELMCHG. If YT is greater than Y go back to paragraph 1.3. Otherwise YT is less than Y so DT is not reached. Set DP equal to DT and set YP equal to YT. Set the new DT equal to DT plus the increment and go back to paragraph 1.2 to process the new DT.

1.1)

Function DCRIT iterates depths from zero (the lower limit depth) to DU (the upper limit depth) to find the critical depth (depth requiring minimum specific energy). Set (the increment for iteration equal to one. Set DU equal to DH. In the following calculations the suffix T represents values computed for the iterated depth and the suffix P represents values computed for the previous iterated depth. Set the first DT equal to .00001. Set SEL (lower limit of specific energy) to 1.E50.

1.2)

Compute VSQT (velocity squared) as follows:

VSQT = ((Q / AT) **2).

Compute SET (specific energy) as follows:

SET = DT + (VSQT / 64.348)

If SET is greater than SEL go to paragraph 1.3. Otherwise SET is less than SEL so set SEL equal to SET. If SET is not greater than SEP go to paragraph 1.3. Otherwise DT is not high enough to be DC. Set DP equal to DT and set SEP equal to SET. Multiply ten times DT to compute a new DT (this method will approximate DC more quickly) and go back to the beginning of this paragraph to process the new DT.

1.3)

If DT is not less than DU then DC cannot be computed any more accurately so print a message saying that DC may be inaccurate, set DC equal to DT and return control to subroutine ELMCHG. Otherwise DT is less than DU. If SET is greater than SEP go to paragraph 1.4. Otherwise SET is not greater than SEP so DC has not been reached yet. Set SEL to SEP, set SEP to SET, set DL to DP and set DP to DT. Add the increment to DT to create a new DT and go back to paragraph 1.2 to process the new DT.

1.4)

DT has gone above the DC value. Set DU equal to DT plus the increment, set DT equal to DL, and set SEP equal to SEL. Divide the increment by ten giving a smaller increment. Set DT equal to DL plus the new increment. If the increment is insignificantly small set DC equal to DT and return to subroutine ELMCHG. Otherwise go back to paragraph 1.2 to process the new DT.

1.1)

Function DEPTH computes a depth in a section when A (the cross section area of flow) is known. The processing is done according to section type as follows:

Section 1 : Open trapezoid, see process 2.
Section 2 : Open rectangle, see process 1.
Section 3 : Closed trapezoid, see process 3.
Section 4 : Pipe, see process 4.
Section 5 : Open irregular, see process 7.
Section 6 : Closed irregular, see process 7.

1) Open rectangular:

BNET (net width of channel) = (W - (NOP * BP)),
D = (A / BNET)

2) Open trapezoid:

BNET (net width of channel) = (W - (NOP * BP)),
D = (- BNET + ((BNET **2) + (2 * (ZR + ZL) * A) **.5)) / (ZR + ZL)

3) Closed trapezoid:

Compute AH (area based on channel being full) as follows:

AH = ((((ZR + ZL) / 2) * H) + BNET) * H)
where BNET (net width of channel) = (W - (NOP * BP)).

If A is less than AH see process 2, otherwise set D equal to H.

4) Pipe:

R (radius) = W / 2
AH1 (full pipe) = (PI * (R **2))
AH2 (half full pipe) = (AH1 / 2)

4.1)

If A is greater than AH1 (the pipe is full) set D equal to W (diameter), and return control to the appropriate subroutine. If A is less than AH2 (the pipe is less than half full) set D equal to zero, otherwise A is between AH2 and AH1 (the pipe is over half full) so set D equal to R. Set DP equal to D and the increment value for iteration equal to one. If D equals zero set A equal to AH2 or if D equals R set A equal to AH1 to use as maximum area for iteration. Set DT equal to D and begin iteration.

4.2)

Compute AT. If AT is less than A then DT is not large enough so set DP equal to DT, add the increment to DT to compute a new DT and go back to paragraph 1.11 to process the new DT.

4.3)

If AT equals A then set D equal to DT and return control to the appropriate subroutine. Otherwise AT is greater than A so DT is too high. Divide the increment by ten. Set DT equal to the previous DT plus the increment. If the increment is insignificantly small set D equal to DT and return control to the appropriate subroutine. Otherwise go to paragraph 1.12 to process the new DT.

5) Irregular section:

5.1)

Set DT equal to one and set the increment equal to one.

5.2)

Compute AT. If AT equals A set D equal to DT and return control to the appropriate subroutine. If AT is less than A, then D is not high enough so set DP equal to DT, set DT equal to DT plus the increment and process the new DT at the start of this paragraph. Otherwise DT is too high. Divide the increment by ten. Set DT equal to the previous DT plus the increment. If the increment is insignificantly small set D equal to DT and return to the appropriate subroutine. Otherwise go process the new DT at the start of this paragraph.

DEPSMP

1.1)

Function DEPSMP is used by junction elements to iterate depths between DL (lower limit depth) and DU (upper limit depth) to find the depth where pressure and momentum curves intersect. Compute the increment value for iteration by subtracting DL from DU and dividing that value by ten. Compute H as the difference in invert elevations between the U/S and D/S ends. Set DT (the iterated depth) equal to DL. Use functions SUMM and SUMP to compute MT (momentum) and PT (pressure) values. Set DIFF (the difference between momentum and pressure values) equal to MT minus PT.

1.2)

Set DIFFP equal to DIFF. Set DP equal to DT. Add the increment to DT to get a new DT. Use functions SUMM and SUMP to compute MT and PT. Compute DIFF by subtracting PT from MT. If DIFF equals zero this is the intersection so set the depth equal to DT and return control to the appropriate subroutine. If DIFF and DIFFP are not both positive or both negative then DT is past the point of intersection so go to paragraph 1.3. Otherwise the point of intersection has not been reached, add the increment to DT and go back to the beginning of this paragraph to process a new DT.

1.3)

DT is past the point of intersection. Divide the increment value by ten to get a new smaller increment. Set DT equal to DP and DIFF equal to DIFFP. If the increment is insignificantly small set depth equal to DT and return to the appropriate subroutine. If DT has passed DH in U/S processing set DT equal to DH, as the intersection cannot be found because of pressure flow, and return to the appropriate subroutine. Otherwise go back to paragraph 1.2 to process a new DT.

1.1)

Function SUMM computes the momentum in the section for a given depth in a junction element. DT is the testing depth in both U/S and D/S processing and variables with the suffix T are based on DT. If this is D/S processing to paragraph 1.3.

1.2)

This is U/S computation of M (momentum). If there are no laterals set M3COS and M4COS equal to zero, otherwise compute M3COS and M4COS (momentums) in the junction laterals as follows:

```
D3 = ((D2 + INV2 + DT + INV1) / 2) - INV3,
D4 + ((D2 + INV2 + DT + INV1) / 2) - INV4,
M3COS = ((Q3 **2) / (A3 * G)) * COS(ANG3),
M4COS = ((Q4 **2) / (A4 * G)) * COS(ANG4),
```

where D3 and D4 are lateral depths,
INV3 and INV4 are lateral inverts,
Q3 and Q4 are lateral flow rates,
A3 and A4 are lateral area flows, and
ANG3 and ANG4 are angles of intersection of the laterals into the junction.

Compute M as follows:

```
M = ((Q2 **2) / (A2 * G)) - ((Q1 **2) / (AT * G)) - M3COS - M4COS,
```

and return control to the appropriate subroutine.

1.3)

This is D/S computation of M (momentum). If there are no laterals set M3COS and M4COS equal to zero, otherwise compute M3COS and M4COS (momentums) in the junction laterals as follows:

```
D3 = ((D1 + INV1 + DT + INV2) / 2) - INV3,
D4 = ((D1 + INV1 + DT + INV2) / 2) - INV4,
M3COS = ((Q3 **2) / (A3 * G)) * COS(ANG3),
M4COS = ((Q4 **2) / (A4 * G)) * COS(ANG4),
```

see U/S computation for variable descriptions.

Compute M as follows:

```
M = ((Q2 **2) / (AT * G)) - ((Q1 **2) / (A1 * G)) - M3COS - M4COS,
```

and return control to the appropriate subroutine.

1.1)

Function SUMP computes the pressure in a section for a given depth in a junction element. DT is the testing depth in both U/S and D/S processing and variables with the suffix T are based on DT. If this is D/S processing then P (pressure) is computed as follows:

```
P = (((A1 + AT)  /2) * (D1 + H - DT))
    - ((DELL / 4) * (A1 + AT) * (SF1 + SFT)),
```

otherwise this is U/S processing and P is computed as follows:

```
P = (((AT + A2) / 2) * (DT + H - D2))
    - ((DELL / 4) * (AT + A2) * (SFT + SF2 )).
```

Processing control is returned to the appropriate subroutine.

COMPUTATION ERROR AND WARNING MESSAGES

1) INVALID RECORD CODE IN XXX, IREC = X, ELEMENT = XXX;

The subroutine, record code, and element number are printed. An invalid record code was encountered, the record code value should be between 1 and 9. This code is set internally in the edit phase so it is a program problem - see programming staff. Processing is stopped.

2) WATER SURFACE ELEVATION GIVEN IS LESS THAN OR EQUALS INVERT ELEVATION IN XXX, W.S. ELEV = INV + DC;

The subroutine name is printed. This is a warning message that there was no water surface elevation inputted for either the headworks or outlet or that the water surface inputted is less than the invert elevation causing DC to be the controlling depth. Processing continues.

3) WENTDS, NO AREA OF OBSTRUCTION IN ELEMENT XXX, A1 = XXX, A2 = XXX;

The element number, area in the U/S end, and area in the D/S end (based on depth from the U/S end) are printed. The area in the U/S end must be greater than the area in the D/S end. Make sure this is supposed to be a wall entrance and that the channel sections are described properly. Processing is stopped.

4) W.S. ELEV. IS 10 FEET OR MORE ABOVE OPEN CHANNEL WALLS IN XXX, STATION = XXX, D = XXX, DH = XXX;

The subroutine, station, depth, and maximum open flow depth are written. Open flow depth reached the maximum limit in the program. Raise the heights of the channel walls at this point and resubmit. Processing is stopped.

5) OVER 50 RECORDS WRITTEN IN XXX ELEMENT = XXX STATION = XXX;

The subroutine, element, and station are printed. The maximum number of 50 intermediate points in a reach element have been processed. Divide this reach element into two or more reaches at the station printed and resubmit. Processing is stopped.

6) CANNOT SOLVE QUADRATIC FORMULA FOR START OF OPEN FLOW IN RCHUS, STATION = XXX;

The station at the D/S end of the reach is printed. The solution to solving the quadratic formula was negative for the length of reach in pressure flow. There is no solution for this problem (it should not occur), if it does this element must be hand calculated and the other elements can be run with the hand calculated control depths. Processing is stopped.

7) OTLTJP HAS ERRONEOUS PROCESSING CODES, U/S CODE = X, D/S CODE = X, D = XXX, DC = XXX;

The U/S and D/S processing codes, depth, and critical depth are written. There must be at least U/S processing in the outlet. This is an internal program error because the processing codes for the outlet are improperly set. See programmer. Processing is stopped.

8) A STATION ON THE U/S OR D/S FILE IS PAST THE END OF THE REACH IN RCHJP, STATION FROM FILE = XXX, REACH STATION = XXX;

The station from the U/S or D/S file and the station at the U/S or D/S end of the reach are printed. The station on the file is not between the U/S and D/S ends of the reach. This is an internal problem, a station computed in U/S or D/S processing is in error or the value for the number of U/S or D/S records written is in error. See programmer. Processing is stopped.

9) THE KNOWN DEPTH EQUALED THE NORMAL DEPTH IN BERNLI, DEPTH = XXX;

The known depth is printed. This is a warning message that normal depth has already been reached. The depth at the end of the reach is set equal to normal depth. Processing continues.

10) THE UPPER AND LOWER LIMIT VALUES CALCULATED IN BERNLI WERE THE SAME, LOWER LIMIT = XXX, UPPER LIMIT = XXX;

The values from Bernoulli's equation based on the lower and upper limit depths are printed. This is a warning message indicating that depth cannot be found by Bernoulli's equation and that the upper and lower limit depths are the same. The depth at the end of the reach is set equal to the current known depth. Processing continues.

11) THE VALUE TO SOLVE FOR DEPTH IN BERNLI IS NOT BETWEEN THE UPPER AND LOWER VALUE LIMITS, DESIRED VALUE = XXX, UPPER LIMIT VALUE = XXX, LOWER LIMIT VALUE = XXX, UPPER LIMIT DEPTH = XXX, LOWER LIMIT DEPTH = XXX;

The value needed to solve Bernoulli's equation, the upper and lower limit values from Bernoulli's equation, and the upper and lower limit depths are printed. This is a warning message indicating that depth to solve Bernoulli's equation cannot be found between the limits where it is expected. Depth at the end of the reach is set to the current known depth or to normal depth depending on whether the desired value to solve Bernoulli's equation is greater or less than the prescribed limits. Processing continues.

12) THE XX FILE DOES NOT HAVE DEPTH AT THE HYDRAULIC JUMP IN JUMPR;

The U/S or D/S file is printed. The station of the hydraulic jump cannot be computed although it is indicated to exist because the U/S and D/S force curves crossed. The actual location of the jump is not included on either the U/S or D/S file. This is an internal problem and should not happen. Check U/S and D/S reach processing to see if they are valid (print switch = 2). See programmer. Processing is stopped.

13) NO INTERSECTION OF FORCE CURVES COULD BE FOUND FOR THE HYDRAULIC JUMP IN JUMPR;

A hydraulic jump was indicated but there was insufficient data on the U/S and D/S files to locate the point of intersection. This is an internal problem and should not occur. Check U/S and D/S reach processing to see if they are valid (print switch = 2). See programmer. Processing is stopped.

14) THE FORCE AT THE HYDRAULIC JUMP IS NOT BETWEEN THE FORCES FROM THE UPPER AND LOWER LIMIT DEPTHS, UPPER LIMIT DEPTH = XXX, LOWER LIMIT DEPTH = XXX, UPPER LIMIT FORCE = XXX, LOWER LIMIT FORCE = XXX, FORCE AT JUMP = XXX IN PPMDEP;

The upper and lower limit depths (depth from either side of indicated hydraulic jump), the upper and lower limit forces, and the force at the hydraulic jump are written. The force at the jump should be equal or between the forces on either side of the jump but this was not the case. Either the force given for the jump or the points given from the U/S or D/S file adjacent to the jump are wrong. Check the U/S and D/S files for valid data (print switch = 2). This is an internal problem. See programmer. Processing is stopped.

15) THE TEST DEPTH EXCEEDED THE UPPER LIMIT DEPTH BEFORE THE FORCE AT THE JUMP WAS REACHED, TEST DEPTH = XXX, UPPER LIMIT DEPTH = XXX, TEST FORCE = XXX, JUMP FORCE = XXX IN PPMDEP;

The iterated depth, upper limit depth, iterated force, and force at the hydraulic jump are printed. The depth causing the force at the hydraulic jump should be equal or between the depths on either side of the jump but this was not the case. Either the force given for the jump or the points given from the U/S or D/S file adjacent to the jump are wrong. Check the U/S and D/S files for valid data (print switch = 2). This is an internal problem. See programmer. Processing is stopped.

16) ELEMENT NO. XXX HAS DNORM OR DCRIT LESS THAN OR EQUAL TO ZERO IN ELMCHG;

The element number is printed. Either normal depth or critical depth could not be computed for this reach. This should not happen, check the channel description for the reach element. Hand calculate normal and critical depths and if they exist in the channel description see a programmer because either function DNORM or function DCRIT is in error. This exists as a warning message and processing will continue but will probably terminate before the end of the run.

17) XXX ERRORS WERE ENCOUNTERED IN SETTING THE PRELIMINARY VALUES IN ELMCHG;

The number of errors in analyzing adjacent elements and flow rates and computing critical and normal depths is written. These errors must be corrected and the program must be rerun before actual processing will start. If this message occurs on the same run for other than the first flow rate there is an internal problem, then see the programmer. Processing is stopped.

18) NO XX RECORDS EXISTED WHERE INDICATED - ELEMENT NO. XXX IN WRITEN;

The U/S or D/S file and the element number are printed. The U/S or D/S processing code indicated the computation for the element was valid but there were no records on that file for the element. This is an internal problem with the processing codes. See programmer. Processing continues with the next element.

19) THERE WAS NO JUMP INDICATED WHEN BOTH U/S AND D/S RECORDS EXISTED FOR ELEMENT XXX IN WRITEN;

The element number is printed. There was a problem in the jump processing for this element. Either one of the profiles should be deleted or a hydraulic jump should be indicated. This is an internal problem, see programmer. Processing continues with the next element.

20) A JUMP WAS INDICATED BUT THERE WERE NOT RECORDS ON BOTH THE U/S AND D/S PROFILES FOR ELEMENT XXX IN WRITEN;

The element number is printed. There was a problem in the jump processing for this element. If the entire U/S or D/S profile is deleted then there cannot be a jump and if there is a jump there must be U/S and D/S profile data. This is an internal problem, see programmer. Processing continues with the next element.

21) THERE WERE NO RECORDS FOR ELEMENT XXX IN WRITEN;

The element number is printed. This is a warning message to indicate there was no U/S or D/S processing for this element. Check the U/S and D/S profiles (print switch = 2) to verify this. If there is data there is an internal problem, if there is no data check the construction of the element. Processing continues with the next element.

22) NO PLOT GENERATED, BAD DATA OR NOT ENOUGH POINTS, 3 OR LESS;

If there are only 3 elements being run no plot will be generated, otherwise there was a problem in processing one of the elements and there is an internal problem. This is a warning message and processing continues.

23) ELEMENT NUMBER XXX HAS ADJACENT ELEMENTS WHICH ARE IN ERROR;

The element number is printed. There is an error in the sequence of elements submitted (such as bridge exits back to back) which are not allowed. Check the sequence of the elements, correct the error, and resubmit the data. This is a user error, sequence checking will continue but actual processing will be stopped.

24) XXX DEPTH COULD NOT BE FOUND IN ELEMENT XXX;

Either normal or critical depth and element number are printed. There is either an error in function DCRIT or DNORM or there is a bad channel description. Hand calculate the value and if it is valid for the channel see a programmer. The elements will continue to be checked but no actual processing will take place until the error is resolved.

25) IRREGULAR XXX VALUES ARE ZERO OR NEGATIVE, SET XXX EQUAL TO ZERO, XXX = XXX, PIER XXX = XXX, IN XXX;

Either force, area, or wetted perimeter values are printed from functions FORCEI, AREACI, or WETPI for irregular sections. The appropriate data could not be computed in this irregular section. The problem is internal. The cross section points used for the computation are probably distorted. See a programmer. The desired value is set to zero and processing continues but will probably be in error and processing will probably terminate before the end of the run.

26) PIER WIDTH IS WIDER THAN CHANNEL WIDTH IN XXX, DEPTH = XXX, PIER WIDTH = XXX;

Either force, area, or wetted perimeter, depth, and average pier width is printed. The width of the number of piers at the given depth is wider than the channel width at that depth. This is a user error. Correct the input data and resubmit. The desired value is set to zero and processing continues but results will probably be erroneous and the program will probably terminate before the end of the run.

27) DEPTH EXCEEDS XXX WITH FORCE TOO LOW IN FORCEM, TEST DEPTH = XXX, TEST FORCE = XXX, XXX = XXX, DESIRED FORCE = XXX;

The iterated depth and force, the maximum or minimum depth, and the desired force value are printed. The desired force in the bridge exit could not be reached within the prescribed depth limits. The desired depth is set to zero and no processing is done in that end of the bridge exit. Processing continues with the next element.

28) DESIRED FORCE IS OUT OF THE RANGE OF DEPTHS IN FORCEF, TEST DEPTH = XXX, TEST FORCE = XXX, XXX = XXX, DESIRED FORCE = XXX;

The iterated depth and force, the maximum or minimum depth, and the desired force are printed. In D/S processing the desired force in the D/S end of the bridge entrance could not be reached within the prescribed limits of depth so the desired depth is set to zero and no computation is done in the D/S end. In U/S processing the bridge entrance was under pressure at the U/S end so pressure flow calculations will be done.

29) DESIRED FORCE IS OUT OF THE RANGE OF DEPTHS IN FWALL, TEST DEPTH = XXX, TEST FORCE = XXX, MINIMUM DEPTH = XXX, DESIRED FORCE = XXX;

See Message 28 for D/S bridge entrance only use wall entrance instead.

30) DEPTH IS OUTSIDE THE RANGE OF THE POINTS DESCRIBING THE CHANNEL IN XXX, DEPTH = XXX, YMIN = XXX, YMAX = XXX;

Either force, area, or wetted perimeter values for depth and minimum and maximum Y values are printed. If the depth is not above maximum open flow depth there is an internal error, see programmer. If the depth exceeds maximum open flow depth raise the channel walls. The desired value is set to zero and processing continues but will probably be in error and processing will probably terminate before the end of the run.

31) UNABLE TO CALCULATE FRICTION SLOPE WITH MANNINGS EQUATION IN SF, AREA = XXX, WETTED PREIMETER = XXX;

The area and wetted perimeter are printed. Either the area or wetted perimeter should be less than or equal to zero. This is an internal problem, see programmer. Processing is stopped.

32) CRITICAL DEPTH MAY BE INACCURATE IN ELEMENT XXX, INCREMENT = XX;

The element number and increment value are printed. If the increment is large then critical depth is probably above the top of the channel but is set equal to the channel height. If the increment is small critical depth is probably pretty accurate but for some reason it cannot be computed precisely. This is a warning and processing continues.

33) Q VALUES IN THE JUNCTION ARE INCORRECT FOR DEPSMP, Q1 = XXX, Q2 = XXX, Q3 = XXX, Q4 = XXX;

The Q values for both U/S and D/S ends and for the laterals are printed. Q2 should equal the sum of the other Q's. If it does not there is an internal error in subroutine ELMCHG, see a programmer. If these Q values are in error, resubmit the input data with the correct Q values. Processing is stopped.

34) A LATERAL ANGLE OF CONFLUENCE IS GREATER THAN 90 DEGREES IN DEPSMP, FIRST ANGLE OF CONFLUENCE = XXX, SECOND ANGLE OF CONFLUENCE = XXX;

The angles of the laterals are printed. This is a user error, check the values inputted for the angles of the laterals, they should be in degrees. Processing is stopped.

35) INVALID PROCESSING CODE WAS ENCOUNTERED IN XXX, PROCESSING CODE = X, AND IT SHOULD BE 1 OR 2;

Function DEPSMP, SUMM, or SUMP is printed with the processing code. The processing code should be 1 for D/S processing and 2 for U/S processing representing the known end of the element. This code is internally set just before the function is called so it is a programming problem. Processing is stopped.

36) MOMENTUM AND PRESSURE CURVES DID NOT CROSS IN DEPSMP, SETTING DEPTH EQUAL TO UPPER LIMIT DEPTH PLUS ONE FOOT, DEPTH = XXX;

The depth is printed. The intersection of the pressure and momentum curves was above the maximum open flow depth. For a closed channel pressure flow calculations will be executed, otherwise, processing will stop because depth is too high in an open channel. This is a warning message.

DEBUG MESSAGE)

NO COMPUTATION IN XXXX - A BREAK IN WATER SURFACE PROFILE, STATION = XXX, XXXX

The station, subroutine, and paragraph number are printed. This is a notification message which identifies whether there was any computation for the U/S or D/S end of an element.

Other messages printed in debug trace have the subroutine or function and paragraph number printed.

PROGRAM SIZE AND CONTROL CONFIGURATION

The Water Surface Profile System was designed for the IBM-370-158 computer under the OS/VS (Release 3.7, MVS) and JES2 (Release 4.1) operating system. The programs were compiled using the IBM FORTRAN-H compiler.

The total length of the edit program is 134A8 in hexadecimal and requires 88K to be executed. The total length of the computation program is 52FCO in hexadecimal and requires 345K to be executed.

In addition to this space both programs require approximately 350K supplied by the system to control the disk files.

If your installation does not meet these requirements see the next page for overlay and file size suggestions.

OVERLAYS AND FILE SIZE

1.1)

The water surface profile computation program requires a large amount of disk for execution. If this is a problem in your installation the following suggestions are offered.

1.2)

This program is written so that all the subroutines called from the MAIN routine can be overlayed. If this still requires too much core all the subroutines called from subroutines UPSTRM, DWNSTM and JUMPP can be overlayed. All the other subroutines and all the functions should remain in core through the entire processing.

1.3)

To save disk space the file sizes can be reduced as follows. The number of intermediate points in a reach can be reduced to a number less than 50 (but if this number is exceeded processing will stop). This will reduce the number of records on the upstream and downstream files. The number of records on these files is computed by multiplying the maximum number of records in a reach by the number of allowable elements in the channel (50 times 200 at present). This change will mean changing the record value in file 2 and file 3 definitions and the value of IC in the MAIN routine.

1.4)

Also the number of elements allowed in the channel can be reduced to less than 200. This makes the upstream, downstream, element and cross section points files smaller. To do this the record values in the file 1, file 2, file 3, and file 4 definitions must be changed (see above paragraph for changing file 2 and file 3). The reference to record numbers 201 and 202 must also be changed in subroutines UPSTRM, DWNSTRM, JUMPP, WRITEN, and PLOT to the new values where the title and the number of elements are to be kept.

MODIFICATIONS PROCEEDURE

1.1)

The water surface profile has been written in modules to make modifications to the program easier. Each individual element in upstream, downstream, and jump processing has its own subroutine to control that processing. Each element type which requires iteration in computation has its own function to do that iteration. All other standard computations are computed in seperate subroutines or functions.

1.2)

The processing in the program is also done modually. The Q values and controlling depths are set for all the elements. Then the entire downstream profile is computed. The next step is the entire computation of the upstream profile. The two profiles are then compared to select the composite profile and compute any hydraulic jumps which may occur. Next the composite profile is printed after which the key values in the composite profile are plotted.

1.3)

If an error in processing is detected the first step should be to set the print switch equal to 2. This will cause the upstream and downstream profiles to be printed. From the profiles the problem should be able to be isolated to the particular element causing the problem. If the error cannot be found by checking the subroutine controlling that computation then run the program with the print switch equal to 1. If the error stops U/S or D/S profile computation then check the routine causing the termination and see the next paragraph.

1.4)

When the print switch equals 1 all the processing in the program is traced. This technique causes a large amount of printing. To avoid the unnecessary print out two things can be done. The first proceedure is to run only the portion of the channel which is in error (running only three to five elements setting controls at the outlet and headworks equal to the actual values of the preceeding elements to where the trouble occurs). The second method would be to recompile the subroutine controlling the invalid processing so that the trace switch is on while processing that element type and reset to off when the processing for that element type has been completed. When recompiling it is only necessary to recompile the individual subroutine and link it with the existing object module, then recompile back after the processing error has been corrected to the user's satisfaction. This can be done using the JCL shown in SETCC only changing the card following card 9 to read INCLUDE UPDATE(F0515B).

PROGRAM LIMITATIONS

1) The constant for gravity throughout the program will be 32.2 ft/sec

2) A maximum of 200 elements are allowed per run.

3) A maximum of 50 intermediate points can be computed in a reach element.

4) Critical depth cannot exceed 100 feet.

5) Program will not compute the water surface profile when the friction slope is at one or greater.

6) Open channel processing is limited to a depth ten feet above the height of the described element.

7) Undulating bottoms cannot be calculated properly in an irregular shaped section unless the depth of flow is above the undulations.

8) Only one value for Manning's factor can be given per element. If Manning's factor varies through the element an average value should be used as input. If Manning's factor varies between adjacent elements, only the results from the more D/S element will appear on the composite profile but calculations for both factors will appear on the U/S and D/S profile if they are requested.

9) The program will not accept vertical drops in invert elevations.

10) Calculations in the Water Surface profile print out may be slightly (to .001) inaccurate due to rounding variables to be contained on the U/S and D/S data files.

JCL EXECUTION CARDS

```
//F0515P      JOB (JOB CARD STATISTICS),REGION=350K                     1
//STEP1       EXEC PGM=F0515A                                           2
//STEPLIB     DD DSN=SYSTEM.LOAD.LIBRARY,DISP=SHR                       3

//FT01F001    DD DSN=&&ELMFIL,DISP=(NEW,PASS),                          4
//            DCB=(LRECL=350,BLKSIZE=350,RECFM=F),                      5
//            UNIT=SYSVIO,SPACE=(CYL,(5,2))                             6
//FT04F001    DD DSN=&&POINTS,DISP=(NEW,PASS),                          7
//            DCB=(LRECL=1262,BLKSIZE=1262,RECFM=F),                    8
//            UNIT=SYSVIO,SPACE=(CYL,(7,2))                             9
//FT05F001    DD * CARD INPUT FOLLOWS                                   10
              TRACE SWITCH CARD
                   ELEMENT CARDS
                        CHANNEL DEFINITION CARDS
                             CROSS SECTION POINT CARDS
//FT06F001    DD SYSOUT=A                                               11
//FT08F001    DD DSN=&&TEMP.ELM,DISP=(NEW,DELETE,DELETE),               12
//            DCB=(LRECL=80,BLKSIZE=80,RECFM=F),                        13
//            UNIT=SYSVIO,SPACE=(CYL,(4,2))                             14
//FT09F001    DD DSN=&&TEMP.CDS,DISP=(NEW,DELETE,DELETE),               15
//            DCB=(LRECL=80,BLKSIZE=80,RECFM=F),                        16
//            UNIT=SYSVIO,SPACE=(CYL,(4,2))                             17
//STEP2       EXEC PGM=F0515B,COND=(1,LE,S1)                            18
//STEPLIB     DD DSN=SYSTEM.LOAD.LIBRARY,DISP=SHR                       19
//FT01F001    DD DSN=&&ELMFIL,DISP=(OLD,DELETE)                         20
//FT02F001    DD DSN=&&UPSTRM,DISP=(NEW,DELETE,DELETE),                 21
//            DCB=(LRECL=120,BLKSIZE=120,RECFM=FB),                     22
//            UNIT=SYSVIO,SPACE=(120,(10000))                           23
//FT03F001    DD DSN=&&DWNSTM,DISP=(NEW,DELETE,DELETE),                 24
//            DCB=(LRECL=120,BLKSIZE=120,RECFM=FB),                     25
//            UNIT=SYSVIO,SPACE=(120,(10000))                           26
//FT04F001    DD DSN=&&POINTS,DISP=(OLD,DELETE)                         27
//FT05F001    DD * CARD INPUT FOLLOWS                                   28
              TRACE SWITCH CARD
                   Q CARD DATA
//FT06F001    DD SYSOUT=A                                               29
//FT08F001    DD DSN=&&PLOT,DISP=(NEW,DELETE,DELETE),                   30
              DCB=(LRECL=48,BLKSIZE=48,RECFM=FB),                       31
              UNIT=SYSVIO,SPACE=(48,(10000))                            32
```

JCL DESCRIPTION

Card 1 is the standard installation JOB card.
Card 2 executes the edit program.
Card 3 tells where the load module resides which should also contain the included subroutine SETCC, see SETCC routine.

Cards 4-6 describe the element file.

Cards 7-9 describe the cross section points file.

Card 10 describes the system card input device.

Cards between 10 and 11 are data input.

Card 11 is the system output printing device.

Cards 12-14 describe the temporary element file.

Cards 15-17 describe the temporary channel definition file.

Card 18 executes the computation program.
Card 19 tells where the load module to be executed resides.
Card 20 refers to the element file.

Cards 21-23 define the upstream profile file.

Cards 24-26 define the downstream profile file.

Card 27 refers to the cross section points file.
Card 28 refers to the system card input device.
Cards between 28 and 29 are data input.

Card 29 refers to the system output printing device.

Cards 30-32 define the plot data file.

SETCC

SETCC is a routine written in assembler language which allows the setting of internal condition codes. The condition code value is set to the number of errors encountered in the editing phase and determines whether the computation program will be executed.

The routine SETCC (included in material being sent) should be compiled and added to the system load library before compilation of the edit and computation programs and linked to the program by way of the card.

The following shows how to include SETCC into the executable load module when the linkage phase is executed in a compile and link under the FORTRAN-H compiler step.

```
//F0515CL      JOB (JOB CARD STATISTICS)                                     1
//JOBLIB       DD DSN=SYSTEM.LOAD.LIB,DISP=SHR,UNIT=XXXX,VOL=SER=XXXX        2
//COMPILE      EXEC PGM=FORTHCL,PARM.FORT=(MAP,ID),COND.LKED=(3,LT,FORT),    3
//             SYSLMOD='SYSTEM.LOAD.LIB(F0515A)',LMODISP=OLD                 4
//FORT.SYSTEM DD * SOURCE DECK FOLLOWS                                       5
               SOURCE DECK
/*                   END OF SOURCE                                           6
//LKED.UPDATE DD DSN=SYSTEM.LOAD.LIB,DISP=SHR                                7
//LKED.SYSTEM DD *                                                           8
     INCLUDE JOBLIB(SETCC)
     ENTRY MAIN
/*                                                                           9
//             END OF JOB                                                   10
```

Card 1 is the standard installation JOB card.
Card 2 is the system load library where SETCC load module resides.
Cards 3 and 4 execute the FORTRAN-H compiler and name the location for the load module to be created.
Cards 5 and 6 define the input source deck to the compiler.
Card 7 allows for input into the linkage editor.
Cards 8 thru 9 brings in SETCC from the JOBLIB definition and attaches it to the load module.
Card 10 ends the job.

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

GX20-1702-0 U/M 025
Printed in U.S.A.

RECORD FORMAT

Application *Temporary Element File (FT08F001)* Record Name *Element Card Image* By *R.A. Schaeffer* Date *May, 1979* Page ____ of ____

FIELD NAME	*Temporary Element File Contains Card Images of Title and Element Cards*
CHARACTERISTICS*	*80(A1) Characters*
RELATIVE POSITION	1–80

File Description ____
Recording Mode ____
Records per Block ____
Characters per Record ____
Label Records are ____
File Identification ____
File Serial Number ____
Reel Sequence Number ____
Creation Date ____
Retention Cycle ____

Remarks ____

SORTING FIELDS
Major 1 ____
2 ____
3 ____
4 ____
5 ____
6 ____
7 ____
8 ____
9 ____
10 ____
11 ____
12 ____
13 ____
14 ____
15 ____
Minor 16 ____

FOOTNOTES

* Characteristics
Alphabetic or Blank ____ A
Alphanumeric ____ X
Numeric ____ 9
Assumed Decimal Point ____ V
Examples of Signed Fields
X9999 9999X
X999V99 9999V9X

REVISIONS

Date	By

Fold to Here

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

GX20-1702-0 U/M 026
Printed in U.S.A.

RECORD FORMAT

Application *Temporary Channel Definition (FT09F001)* Record Name *Edited Channel Definition Data* By *Richard Schaeffer* Date *May, 1979* Page ____ of ____

FIELD NAME	SECTION NO.	CHANNEL	NO. of PIERS	AVERAGE WIDTH of PIERS	HEIGHT or DIAMETER	WIDTH	ZL, LEFT SLOPE	ZR, RIGHT SLOPE	EVAL, INVERT CROSS DROP	Y (1) Irregular Pier Base	Y (2) Pier Base	Y (3) Pier Base	Y (4) Pier Base	Y (5) Pier Base	Y (6) Pier Base	Y (7) Pier Base
CHARACTERISTICS*	I3	I1	I2	F5.2	F5.2	F6.2	F5.2	F5.2	F5.2	F5.2	F5.2	F5.2	F5.2	F5.2	F5.2	F5.2
RELATIVE POSITION	1 2 3	4	5 6	7 8 9 10 1	2 3 4 5 6	7 8 9 20 1 2	3 4 5 6 7	8 9 30 1 2	3 4 5 6 7	8 9 40 1 2	3 4 5 6 7	8 9 50 1 2	3 4 5 6 7	8 9 60 1 2	3 4 5 6 7	8 9 70 1 2

FIELD NAME	Y (8) Pier Base	Y (9) Pier Base	Y (10) Pier Base
CHARACTERISTICS*	F5.2	F5.2	F5.2
RELATIVE POSITION	3 4 5 6 7	8 9 80 1 2	3 4 5 6 7

File Description ____
Recording Mode ____
Records per Block ____
Characters per Record ____
Label Records are ____
File Identification ____
File Serial Number ____
Reel Sequence Number ____
Creation Date ____
Retention Cycle ____

Remarks ____

← Fold to Here

SORTING FIELDS

Major 1 ____
2 ____
3 ____
4 ____
5 ____
6 ____
7 ____
8 ____
9 ____
10 ____
11 ____
12 ____
13 ____
14 ____
15 ____
Minor 16 ____

FOOTNOTES

* Characteristics
Alphabetic or Blank ____ A
Alphanumeric ____ X
Numeric ____ 9
Assumed Decimal Point ____ V
Examples of Signed Fields
X9999 9999X
X999V99 9999V9X

REVISIONS

Date	By

INTERNATIONAL BUSINESS MACHINES CORPORATION

GX20-1702-0 U/M 025
Printed in U.S.A.

RECORD FORMAT

Application *Element File (FT01 F001)* Record Name *SYSTEM OUTLET* By *R.A. Schaeffer* Date *4/79* Page ____ of ____

FIELD NAME	
CHARACTERISTICS*	
RELATIVE POSITION	

ELEMENT	RECORD	WIDTH	HEIGHT	D-HEIGHT	ZR RIGHT SLOPE	ZL LEFT SLOPE	E VALUE	AVE. PIER BASE	NO. PIERS	CHN. TYP.	STATION	INVERT	Q	CRITICAL DEPTH	U/S COUNT	D/S COUNT	W/S CODE	D/S CODE	PUMP
I3	I1	F6.2	F6.2	F7.3	F5.2	F5.2	F5.2	F5.2	I2	I1	F9.2	F7.2	F8.1	F7.3	I3	I3	I1	I1	I1
1 2 3	4	5 6 7 8 9 10	1 2 3 4 5 6	7 8 9 20 1 2 3	4 5 6 7 8	9 30 1 2 3	4 5 6 7 8	9 40 1 2 3	4 5	6	7 8 9 50 1 2 3 4 5	6 7 8 9 60 1 2	3 4 5 6 7 8 9 70	1 2 3 4 5 6 7	8 9 80	1 2 3	4	5	6

WATER SURFACE ELEVATION	NO. C-S POS'NS	SECTION NO.
F7.2	I2	I3
7 8 9 90 1 2 3	4 5	6 7 8

File Description ____
Recording Mode ____
Records per Block ____
Characters per Record ____
Label Records are ____
File Identification ____
File Serial Number ____
Reel Sequence Number ____
Creation Date ____
Retention Cycle ____

Remarks ____

SORTING FIELDS

Major 1 ____
2 ____
3 ____
4 ____
5 ____
6 ____
7 ____
8 ____
9 ____
10 ____
11 ____
12 ____
13 ____
14 ____
15 ____
Minor 16 ____

FOOTNOTES

* Characteristics
Alphabetic or Blank ____ A
Alphanumeric ____ X
Numeric ____ 9
Assumed Decimal Point ____ V
Examples of Signed Fields
X9999 9999X
X999V99 9999V9X

REVISIONS

Date	By

Fold to Here

IBM INTERNATIONAL BUSINESS MACHINES CORPORATION

GX20-1702-0 U/M 025
Printed in U.S.A.

RECORD FORMAT

Application *Element File (FT01F001)* Record Name *REACH* By *R.A. Schaeffer* Date *4/79* Page ___ of ___

FIELD NAME
CHARACTERISTICS*
RELATIVE POSITION

ELE-MENT No.	RECORD	WIDTH	HEIGHT	B-HEIGHT	Z RIGHT SLOPE	Z LEFT SLOPE	E VALUE	AVE. PIER BASE	NO. PIERS	CNTY	STATION (1)	INVERT (1)	STATION (2)	INVERT (2)	Q
I3	I1	F6.2	F6.2	F7.3	F5.2	F5.2	F5.2	F5.2	I2	I1	F9.2	- F7.2	F9.2	F 7.2	F8.1
1 2 3	4	5 6 7 8 9 [1]0	1 2 3 4 5 6	7 8 9 [2]0 1 2 3	4 5 6 7 8	9 [3]0 1 2 3	4 5 6 7 8	9 [4]0 1 2 3	4 5	6	7 8 9 [5]0 1 2 3 4 5	6 7 8 9 [6]0 1 2	3 4 5 6 7 8 9 [7]0 1	2 3 4 5 6 7 8	9 [8]0 1 2 3 4 5 6

RUNNING FACTOR	SECTION LENGTH	CHANNEL SLOPE	NORMAL DEPTH	CRITICAL DEPTH	CRITICAL SLOPE	W/S COUNT	D/S COUNT	1/2 CODE	1/2 SIDE	JUMP LOCATION	ANGLE OF CURVE	RADIUS OF CURVE	ANGLE POINT	NO. OF MAN-HOLES	NO. C-S POINTS	SECT-ION No.
F4.3	F9.2	F8.6	F7.3	F7.3	F7.6	I3	I3	I1	I1	F9.2	F6.2	F7.2	F6.2	I2	I2	I3
7 8 9 [9]0	1 2 3 4 5 6 7 8 9	[10]0 1 2 3 4 5 6 7	8 9 [11]0 1 2 3 4	5 6 7 8 9 [12]0 1	2 3 4 5 6 7 8	9 [13]0 1	2 3 4	5	6	7 8 9 [14]0 1 2 3 4 5	6 7 8 9 [15]0 1	2 3 4 5 6 7 8	9 [16]0 1 2 3 4	5 6	7 8	9 [17]0 1

File Description
Recording Mode
Records per Block
Characters per Record
Label Records are
File Identification
File Serial Number
Reel Sequence Number
Creation Date
Retention Cycle

Remarks

Fold to Here

SORTING FIELDS
Major 1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Minor 16

FOOTNOTES

* Characteristics
Alphabetic or Blank — A
Alphanumeric — X
Numeric — 9
Assumed Decimal Point — V
Examples of Signed Fields
X9999 9999X
X999V99 9999V9X

REVISIONS

Date	By

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

GX20-1702-0 U/M 025 Printed in U.S.A.

RECORD FORMAT

Application: *Element File (FT01F001)* Record Name: *JUNCTION* By: *R.A. Schaeffer* Date ____ Page ____ of ____

FIELD NAME	ELEMENT	RECORD	WIDTH (1)	HEIGHT (1)	D-HEIGHT (1)	B RIGHT (1)	B LEFT (1)	E VALUE (1)	AVE. PIER BASE (1)	NO. PIERS (1)	CHK (1)	STATION (1)	INVERT (1)	Q (1)	NO. C-S POINTS (1)	CRITICAL DEPTH (1)	WIDTH (2)
CHARACTERISTICS*	I3	I1	F6.2	F6.2	F7.3	F5.2	F5.2	F5.2	F5.2	I2	I1	F9.2	F7.2	F8.1	I2	F7.3	F6.2

FIELD NAME	HEIGHT (2)	D-HEIGHT (2)	B RIGHT (2)	B LEFT (2)	E VALUE (2)	AVE. PIER BASE (2)	NO. PIERS (2)	CHK (2)	STATION (2)	INVERT (2)	Q (2)	NO. C-S POINTS (2)	CRITICAL DEPTH (2)	LATERAL WIDTH (3)	LATERAL HEIGHT (3)	LATERAL D-HEIGHT (3)
CHARACTERISTICS*	F6.2	F7.3	F5.2	F5.2	F5.2	F5.2	I2	I1	F9.2	F7.2	F8.1	I2	F7.3	F6.2	F6.2	F7.3

FIELD NAME	LATERAL B RIGHT (3)	LATERAL B LEFT (3)	LATERAL E VALUE (3)	LATERAL AVE. PIER BASE (3)	NO. PIERS (3)	CHK (3)	LATERAL INVERT (3)	LATERAL Q (3)	LATERAL ANGLE (3)	LATERAL WIDTH (4)	LATERAL HEIGHT (4)	LATERAL D-HEIGHT (4)	LATERAL B RIGHT (4)	LATERAL B LEFT (4)	LATERAL E VALUE (4)	LATERAL AVE. PIER BASE (4)	NO. PIERS (4)	CHK (4)
CHARACTERISTICS*	F5.2	F5.2	F5.2	F5.2	I2	I1	F7.2	F8.1	F5.2	F6.2	F6.2	F7.3	F5.2	F5.2	F5.2	F5.2	I2	I1

FIELD NAME	LATERAL INVERT (4)	LATERAL Q (4)	LATERAL ANGLE (4)	MANNING FACTOR	U/S CODE	D/S CODE	JUMP	NO. C-S POINTS (3)	NO. C-S POINTS (4)	CHANNEL SLOPE	U/S COUNT	D/S COUNT	SECTION NO. (1)	SECTION NO. (2)	SECTION NO. (3)	SECTION NO. (4)
CHARACTERISTICS*	F7.2	F8.1	F5.2	F4.3	I1	I1	I1	I2	I2	F8.6	I3	I3	I3	I3	I3	I3

File Description ____
Recording Mode ____
Records per Block ____
Characters per Record ____
Label Records are ____
File Identification ____
File Serial Number ____
Reel Sequence Number ____
Creation Date ____
Retention Cycle ____

Remarks ____

SORTING FIELDS: Major 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, Minor 16

FOOTNOTES

* Characteristics
Alphabetic or Blank — A
Alphanumeric — X
Numeric — 9
Assumed Decimal Point — V
Examples of Signed Fields
X9999 9999X
X999V99 9999V9X

REVISIONS

Date	By

Fold to Here

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

GX20-1702-0 U/M 025
Printed in U.S.A.

RECORD FORMAT

Application *Element File (FT01F001)* Record Name *TRANSITION* By *R.A. Schaeffer* Date *4/79* Page ____ of ____

FIELD NAME	ELEMENT No.	RECORD	WIDTH (1)	HEIGHT (1)	B-HEIGHT (1)	E RIGHT (1)	E LEFT (1)	E VALUE (1)	AVE. PIER BASE (1)	NO. PIERS (1)	CHK FXN (1)	STATION (1)	INVERT (1)	Q (1)	NO. C-S POINTS (1)	CRITICAL DEPTH (1)	WIDTH (2)
CHARACTERISTICS*	I3	I1	F6.2	F6.2	F7.3	F5.2	F5.2	F5.2	F5.2	I2	I1	F9.2	F7.2	F8.1	I2	F7.3	F6.2

FIELD NAME	HEIGHT (2)	B-HEIGHT (2)	E RIGHT (2)	E LEFT (2)	E VALUE (2)	AVE. PIER BASE (2)	NO. PIERS (2)	CHK FXN (2)	STATION (2)	INVERT (2)	Q (2)	NO. C-S POINTS (2)	U/S CODE	D/S CODE	JUMP	CRITICAL DEPTH (2)	MANNING FACTOR	CHANNEL SLOPE
CHARACTERISTICS*	F6.2	F7.3	F5.2	F5.2	F5.2	F5.2	I2	I1	F9.2	F7.2	F8.1	I2	I1	I1	I1	F7.3	F4.3	F8.6

FIELD NAME	U/S COUNT	D/S COUNT	SECTION No. (1)	SECTION No. (2)
CHARACTERISTICS*	I3	I3	I3	I3

File Description ____
Recording Mode ____
Records per Block ____
Characters per Record ____
Label Records are ____
File Identification ____
File Serial Number ____
Reel Sequence Number ____
Creation Date ____
Retention Cycle ____

Remarks ____

SORTING FIELDS
Major 1 ____
2 ____
3 ____
4 ____
5 ____
6 ____
7 ____
8 ____
9 ____
10 ____
11 ____
12 ____
13 ____
14 ____
15 ____
Minor 16 ____

FOOTNOTES

* Characteristics
Alphabetic or Blank ____ A
Alphanumeric ____ X
Numeric ____ 9
Assumed Decimal Point ____ V
Examples of Signed Fields
X9999 9999X
X999V99 9999V9X

REVISIONS

Date	By

Fold to Here

GX20-1702-0 U/M 026
Printed in U.S.A.

RECORD FORMAT

Application Element File (FT01F001) Record Name BRIDGE ENTRANCE By R.A. Schaeffer Date 4/79 Page ___ of ___

FIELD NAME	CHARACTERISTICS*	RELATIVE POSITION
ELEMENT NO.	I3	1–3
RECORD	I1	4
WIDTH (1)	F6.2	5–10
HEIGHT (1)	F6.2	11–16
D-HEIGHT (1)	F7.3	17–23
Z RIGHT (1)	F5.2	24–28
Z LEFT (1)	F5.2	29–33
E VALUE (1)	F5.2	34–38
[illegible] (1)	I1	39
STATION (1)	F9.2	40–48
INVERT (1)	F7.2	49–55
CRITICAL DEPTH (1)	F7.3	56–62
Q (1)	F8.1	63–70
NO. C-S [illegible] (1)	I2	71–72
WIDTH (2)	F6.2	73–78
HEIGHT (2)	F6.2	79–84

FIELD NAME	CHARACTERISTICS*	RELATIVE POSITION
D-HEIGHT (2)	F7.3	85–91
Z RIGHT (2)	F5.2	92–96
Z LEFT (2)	F5.2	97–101
E VALUE (2)	F5.2	102–106
[illegible] (2)	I1	107
STATION (2)	F9.2	108–116
INVERT (2)	F7.2	117–123
AVE. PIER BASE (2)	F5.2	124–128
NO. PIERS (2)	I2	129–130
CRITICAL DEPTH (2)	F7.3	131–137
PIER FACTOR	F5.3	138–142
Q (2)	F8.1	143–150
NO. C-S [illegible] (2)	I2	151–152
U/S CODE	I1	153
D/S CODE	I1	154
JUMP	I1	155
U/S COUNT	I3	156–158
D/S COUNT	I3	159–161
SECTION NO. (1)	I3	162–164
SECTION NO. (2)	I3	165–167

File Description ______
Recording Mode ______
Records per Block ______
Characters per Record ______
Label Records are ______
File Identification ______
File Serial Number ______
Reel Sequence Number ______
Creation Date ______
Retention Cycle ______

Remarks ______

SORTING FIELDS
Major 1 ______
2 ______
3 ______
4 ______
5 ______
6 ______
7 ______
8 ______
9 ______
10 ______
11 ______
12 ______
13 ______
14 ______
15 ______
Minor 16 ______

FOOTNOTES

* Characteristics
Alphabetic or Blank ___ A
Alphanumeric ___ X
Numeric ___ 9
Assumed Decimal Point ___ V
Examples of Signed Fields
X9999 9999X
X999V99 9999V9X

REVISIONS

Date	By

Fold to Here

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

GX20-1702-0 U/M 025
Printed in U.S.A.

RECORD FORMAT

Application: Element File (FT01F001)   Record Name: BRIDGE EXIT   By: R.A. Schaeffer   Date: 4/79   Page ___ of ___

FIELD NAME	ELEMENT No.	RECORD	WIDTH (1)	HEIGHT (1)	D-HEIGHT (1)	Z RIGHT (1)	Z LEFT (1)	E VALUE (1)	CHAN TH (1)	STATION (1)	INVERT (1)	CRITICAL DEPTH (1)	Q (1)	AVE. PIER BASE (1)	NO. PIERS (1)	NO. C-S POINTS (1)	WIDTH (2)
CHARACTERISTICS*	I3	I1	F6.2	F6.2	F7.3	F5.2	F5.2	F5.2	I1	F9.2	F7.2	F7.3	F8.1	F5.2	I2	I2	F6.2
RELATIVE POSITION	1 2 3	4	5 6 7 8 9 1/0	1 2 3 4 5 6	7 8 9 2/0 1 2 3	4 5 6 7 8	9 3/0 1 2 3	4 5 6 7 8	9	4/0 1 2 3 4 5 6 7 8	9 5/0 1 2 3 4 5	6 7 8 9 6/0 1 2	3 4 5 6 7 8 9 7/0	1 2 3 4 5	6 7	8 9	8/0 1 2 3 4 5

FIELD NAME	HEIGHT (2)	D-HEIGHT (2)	Z RIGHT (2)	Z LEFT (2)	E VALUE (2)	CHAN TH (2)	STATION (2)	INVERT (2)	CRITICAL DEPTH (2)	Q (2)	NO. C-S POINTS (2)	U/S CODE	D/S CODE	JUMP	U/S COUNT	D/S COUNT	SECTION NO. (1)	SECTION NO. (2)
CHARACTERISTICS*	F6.2	F7.3	F5.2	F5.2	F5.2	I1	F9.2	F7.2	F7.3	F8.1	I2	I1	I1	I1	I3	I3	I3	I3
RELATIVE POSITION	6 7 8 9 9/0 1	2 3 4 5 6 7 8	9 10/0 1 2 3	4 5 6 7 8	9 11/0 1 2 3	4	5 6 7 8 9 12/0 1 2 3	4 5 6 7 8 9 13/0	1 2 3 4 5 6 7	8 9 14/0 1 2 3 4 5	6 7	8	9	15/0	1 2 3	4 5 6	7 8 9	16/0 1 2

File Description ___
Recording Mode ___
Records per Block ___
Characters per Record ___
Label Records are ___
File Identification ___
File Serial Number ___
Reel Sequence Number ___
Creation Date ___
Retention Cycle ___

Remarks ___

Fold to Here

SORTING FIELDS
Major 1 ___
2 ___
3 ___
4 ___
5 ___
6 ___
7 ___
8 ___
9 ___
10 ___
11 ___
12 ___
13 ___
14 ___
15 ___
Minor 16 ___

FOOTNOTES

* Characteristics
Alphabetic or Blank ___ A
Alphanumeric ___ X
Numeric ___ 9
Assumed Decimal Point ___ V
Examples of Signed Fields
X9999 9999X
X999V99 9999V9X

REVISIONS

Date	By

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

GX20-1702-0 U/M 025
Printed in U.S.A.

RECORD FORMAT

Application *Element File (FT01F001)* Record Name *WALL ENTRANCE* By *R.A. Schaeffer* Date *4/79* Page ___ of ___

FIELD NAME	ELEMENT No.	RECORD	WIDTH (1)	HEIGHT (1)	D-HEIGHT (1)	Z RIGHT (1)	Z LEFT (1)	E VALUE (1)	No. Pier (1)	STATION (1)	INVERT (1)	CRITICAL DEPTH (1)	Q (1)	NO. C-S Pos. (1)	WIDTH (2)	HEIGHT (2)	D-HEIGHT (2)
CHARACTERISTICS*	I3	I1	F6.2	F6.2	F7.3	F5.2	F5.2	F5.2	I1	F9.2	F7.2	F7.3	F8.1	I2	F6.2	F6.2	F7.3

FIELD NAME	Z RIGHT (2)	Z LEFT (2)	E VALUE (2)	No. Pier (2)	STATION (2)	INVERT (2)	AVE. PIER BASE (2)	No. PIERS (2)	CRITICAL DEPTH (2)	PIER, WALL FACTOR	Q (2)	NO. C-S POS. (2)	N/S CODE	B/S CODE	SUMP	N/S COUNT	B/S COUNT	SECTION NO. (1)	SECTION NO. (2)
CHARACTERISTICS*	F5.2	F5.2	F5.2	I1	F9.2	F7.2	F5.2	I2	F7.3	F5.3	F8.1	I2	I1	I1	I1	I3	I3	I3	I3

File Description ___
Recording Mode ___
Records per Block ___
Characters per Record ___
Label Records are ___
File Identification ___
File Serial Number ___
Reel Sequence Number ___
Creation Date ___
Retention Cycle ___

Remarks ___

Fold to Here

SORTING FIELDS

Major 1 ___
2 ___
3 ___
4 ___
5 ___
6 ___
7 ___
8 ___
9 ___
10 ___
11 ___
12 ___
13 ___
14 ___
15 ___
Minor 16 ___

FOOTNOTES

* Characteristics
Alphabetic or Blank ___ A
Alphanumeric ___ X
Numeric ___ 9
Assumed Decimal Point ___ V
Examples of Signed Fields
X9999 9999X
X999V99 9999V9X

REVISIONS

Date	By

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

GX20-1702-0 U/M 025
Printed in U.S.A.

RECORD FORMAT

Application *Element File (FT01F001)* Record Name *WALL EXIT* By *R.A. Schaeffer* Date *4/79* Page ___ of ___

FIELD NAME	ELEMENT NO.	RECORD	WIDTH (1)	HEIGHT (1)	D-HEIGHT (1)	Z RIGHT (1)	Z LEFT (1)	E VALUE (1)	TYPE (1)	STATION (1)	INVERT (1)	CRITICAL DEPTH (1)	Q (1)	AVE. PIER BASE (1)	NO. PIERS (1)	NO. C.S. POSNS (1)	WIDTH (2)	HEIGHT (2)
CHARACTERISTICS*	I3	I1	F6.2	F6.2	F7.3	F5.2	F5.2	F5.2	I1	F9.2	F7.2	F7.3	F8.1	F5.2	I2	I2	F6.2	F6.2

FIELD NAME	D-HEIGHT (2)	Z RIGHT (2)	Z LEFT (2)	E VALUE (2)	TYPE (2)	STATION (2)	INVERT (2)	CRITICAL DEPTH (2)	Q (2)	NO C.S. POSNS (2)	[illegible]	[illegible]	TRAP	B/S COUNT	B/S COUNT	SECTION NO. (1)	SECTION NO. (2)
CHARACTERISTICS*	F7.3	F5.2	F5.2	F5.2	I1	F9.2	F7.2	F7.3	F8.1	I2	I1	I1	I1	I3	I3	I3	I3

File Description ______
Recording Mode ______
Records per Block ______
Characters per Record ______
Label Records are ______
File Identification ______
File Serial Number ______
Reel Sequence Number ______
Creation Date ______
Retention Cycle ______

Remarks ______

SORTING FIELDS: Major 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, Minor 16

FOOTNOTES

* Characteristics
Alphabetic or Blank — A
Alphanumeric — X
Numeric — 9
Assumed Decimal Point — V
Examples of Signed Fields
X9999 9999X
X999V99 9999V9X

REVISIONS

Date	By

Fold to Here

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

GX20-1702-0 U/M 025
Printed in U.S.A.

RECORD FORMAT

Application Element File (FT01F001) Record Name SYSTEM HEADWORKS By R.A. Schaeffer Date 4/79 Page ____ of ____

FIELD NAME	ELEMENT NO.	RECORD	WIDTH	HEIGHT	D-HEIGHT	Z RIGHT	Z LEFT	E VALUE	TYPE	STATION	INVERT	CRITICAL DEPTH	Q	AVE. PIER BASE	NO. PIERS	U/S CODE	D/S CODE	JUMP	WATER SURFACE ELEVATION
CHARACTERISTICS*	I3	I1	F6.2	F6.2	F7.3	F5.2	F5.2	F5.2	I1	F9.2	F7.2	F7.3	F8.1	F5.2	I2	I1	I1	I1	F7.2
RELATIVE POSITION	1-3	4	5-10	11-16	17-23	24-28	29-33	34-38	39	40-48	49-55	56-62	63-70	71-75	76-77	78	79	80	1-7

FIELD NAME	NO. C.S. POINTS	U/S COUNT	D/S COUNT	SECTION NO.
CHARACTERISTICS*	I2	I3	I3	I3
RELATIVE POSITION	8-9	10-12	13-15	16-18

File Description ______
Recording Mode ______
Records per Block ______
Characters per Record ______
Label Records are ______
File Identification ______
File Serial Number ______
Reel Sequence Number ______
Creation Date ______
Retention Cycle ______

Remarks ______

Fold to Here

SORTING FIELDS

Major 1 ______
2 ______
3 ______
4 ______
5 ______
6 ______
7 ______
8 ______
9 ______
10 ______
11 ______
12 ______
13 ______
14 ______
15 ______
Minor 16 ______

FOOTNOTES

* Characteristics
Alphabetic or Blank ____ A
Alphanumeric ____ X
Numeric ____ 9
Assumed Decimal Point ____ V
Examples of Signed Fields
X9999 9999X
X999V99 9999V9X

REVISIONS

Date	By

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

GX20-1702-0 U/M 028
Printed in U.S.A.

RECORD FORMAT

Application *Element File (FT01F001)* Record Name *TITLE RECORD* By *R.A. Schaeffer* Date *4/79* Page ___ of ___

FIELD NAME	
CHARACTERISTICS*	
RELATIVE POSITION	

ELEMENT No. 201 — TITLE DATA FROM TITLE CARD ONE

I3	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A2

TITLE DATA FROM TITLE CARD TWO

A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A2

TITLE DATA FROM TITLE CARD THREE

A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A2

File Description ___
Recording Mode ___
Records per Block ___
Characters per Record ___
Label Records are ___
File Identification ___
File Serial Number ___
Reel Sequence Number ___
Creation Date ___
Retention Cycle ___

Remarks ___

← Fold to Here

SORTING FIELDS: Major 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, Minor 16

FOOTNOTES

* Characteristics
Alphabetic or Blank — A
Alphanumeric — X
Numeric — 9
Assumed Decimal Point — V
Examples of Signed Fields
X9999 9999X
X999V99 9999V9X

REVISIONS

Date	By

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

GX20-1702-0 U/M 025
Printed in U.S.A.

RECORD FORMAT

Application Element File (FT01F001) Record Name NO. OF ELEMENTS TO BE PROCESSED By R.A. Schaeffer Date 4/79 Page ___ of ___

FIELD NAME	ELEMENT NO. 202	NO. ELEMENTS
CHARACTERISTICS*	I3	I3
RELATIVE POSITION	1 2 3	4 5 6

File Description ___
Recording Mode ___
Records per Block ___
Characters per Record ___
Label Records are ___
File Identification ___
File Serial Number ___
Reel Sequence Number ___
Creation Date ___
Retention Cycle ___

Remarks ___

Fold to Here

SORTING FIELDS

Major 1 ___
2 ___
3 ___
4 ___
5 ___
6 ___
7 ___
8 ___
9 ___
10 ___
11 ___
12 ___
13 ___
14 ___
15 ___
Minor 16 ___

FOOTNOTES

* Characteristics
Alphabetic or Blank ___ A
Alphanumeric ___ X
Numeric ___ 9
Assumed Decimal Point ___ V
Examples of Signed Fields
X9999 9999X
X999V99 9999V9X

REVISIONS

Date	By

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

GX20-1702-0 U/M 025
Printed in U.S.A.

RECORD FORMAT

Application U/S and D/S Files (FT02F001, FT03F001) Record Name U/S - D/S Calculated Points By R.A. Schaeffer Date 4/79 Page ____ of ____

FIELD NAME	
CHARACTERISTICS*	
RELATIVE POSITION	

STATION	INVERT	DEPTH	Q	AREA	FORCE	NORMAL DEPTH	CRITICAL DEPTH	SLOPE	FRICTION SLOPE
F9.2	F8.3	F7.3	F9.2	F9.3	F10.3	F7.3	F7.3	F7.5	F7.6

HEIGHT	WIDTH	ZL	ZR	AVE. PIER BASE	NO. PIERS	[illegible]	SEC-TION NO.
F6.2	F6.2	F5.2	F5.2	F5.2	I2	I1	I3

File Description ____
Recording Mode ____
Records per Block ____
Characters per Record ____
Label Records are ____
File Identification ____
File Serial Number ____
Reel Sequence Number ____
Creation Date ____
Retention Cycle ____

Remarks ____

Fold to Here

SORTING FIELDS
Major 1 ____
2 ____
3 ____
4 ____
5 ____
6 ____
7 ____
8 ____
9 ____
10 ____
11 ____
12 ____
13 ____
14 ____
15 ____
Minor 16 ____

FOOTNOTES

* Characteristics
Alphabetic or Blank ____ A
Alphanumeric ____ X
Numeric ____ 9
Assumed Decimal Point ____ V
Examples of Signed Fields
X9999 9999X
X999V99 9999V9X

REVISIONS

Date	By

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

GX20-1702-0 U/M 025
Printed in U.S.A.

RECORD FORMAT

Application: Cross Section Points File (PT04F001) Record Name: Cross Section Points, Pier Points By: R.A. Schaeffer Date: 4/79 Page ___ of ___

FIELD NAME	
CHARACTERISTICS*	
RELATIVE POSITION	

SECTION NO.	NO. POINTS	MAXIMUM Y VALUE	Y MAX LOCATION	MINIMUM Y VALUE	Y MIN LOCATION	X(N) N=1	Y(N) N=1	X(N+1)	Y(N+1)	X(N+2)	Y(N+2)	X(N+3)	Y(N+3)	X(N+4)	Y(N+4)
I3	I2	F6.2	I2	F6.2	I2	F6.2	F6.2	F6.2	F6.2	F6.2	F6.2	F6.2	F6.2	F6.2	F6.2

X(N+5) AND Y(N+5) Repeats To X(N+96) AND	Y(N+96)	X(N+97)	Y(N+97)	X(N+98) = X(99)	Y(N+98) = Y(99)	NO. PIERS	Y(1) PIER BOTTOM	Y(2) PIER BOTTOM	Y(3) PIER BOTTOM	Y(4) PIER BOTTOM	Y(5) PIER BOTTOM	Y(6) PIER BOTTOM	Y(7) PIER BOTTOM	Y(8) PIER BOTTOM	Y(9) PIER BOTTOM	Y(10) PIER BOTTOM	FILLER
F6.2	F6.2	F6.2	F6.2	F6.2	F6.2	I2	F5.2	F5.2	F5.2	F5.2	F5.2	F5.2	F5.2	F5.2	F5.2	F5.2	A

File Description ______
Recording Mode ______
Records per Block ______
Characters per Record ______
Label Records are ______
File Identification ______
File Serial Number ______
Reel Sequence Number ______
Creation Date ______
Retention Cycle ______

Remarks ______

Fold to Here

SORTING FIELDS

Major 1 ______
2 ______
3 ______
4 ______
5 ______
6 ______
7 ______
8 ______
9 ______
10 ______
11 ______
12 ______
13 ______
14 ______
15 ______
Minor 16 ______

FOOTNOTES

* Characteristics
Alphabetic or Blank ___ A
Alphanumeric ___ X
Numeric ___ 9
Assumed Decimal Point ___ V
Examples of Signed Fields
X9999 9999X
X999V99 9999V9X

REVISIONS

Date	By

IBM INTERNATIONAL BUSINESS MACHINES CORPORATION

GX20-1702-0 U/M 025
Printed in U.S.A.

RECORD FORMAT

Application: PLOT FILE (FT08F001) Record Name: VARIABLES TO BE PLOTTED By: R.A. Schaeffer Date: 4/79 Page ___ of ___

FIELD NAME	STATION	HEIGHT	WATER SURFACE ELEVATION	INVERT	ENERGY GRADE LINE	CRITICAL DEPTH	[illegible]
CHARACTERISTICS*	F9.2	F7.2	F8.3	F7.2	F8.3	F8.3	I1
RELATIVE POSITION	1 2 3 4 5 6 7 8 9	10 1 2 3 4 5 6	7 8 9 20 1 2 3 4	5 6 7 8 9 30 1	2 3 4 5 6 7 8 9	40 1 2 3 4 5 6 7	8

File Description ______
Recording Mode ______
Records per Block ______
Characters per Record ______
Label Records are ______
File Identification ______
File Serial Number ______
Reel Sequence Number ______
Creation Date ______
Retention Cycle ______

Remarks ______

Fold to Here

SORTING FIELDS
Major 1 ______
2 ______
3 ______
4 ______
5 ______
6 ______
7 ______
8 ______
9 ______
10 ______
11 ______
12 ______
13 ______
14 ______
15 ______
Minor 16 ______

FOOTNOTES

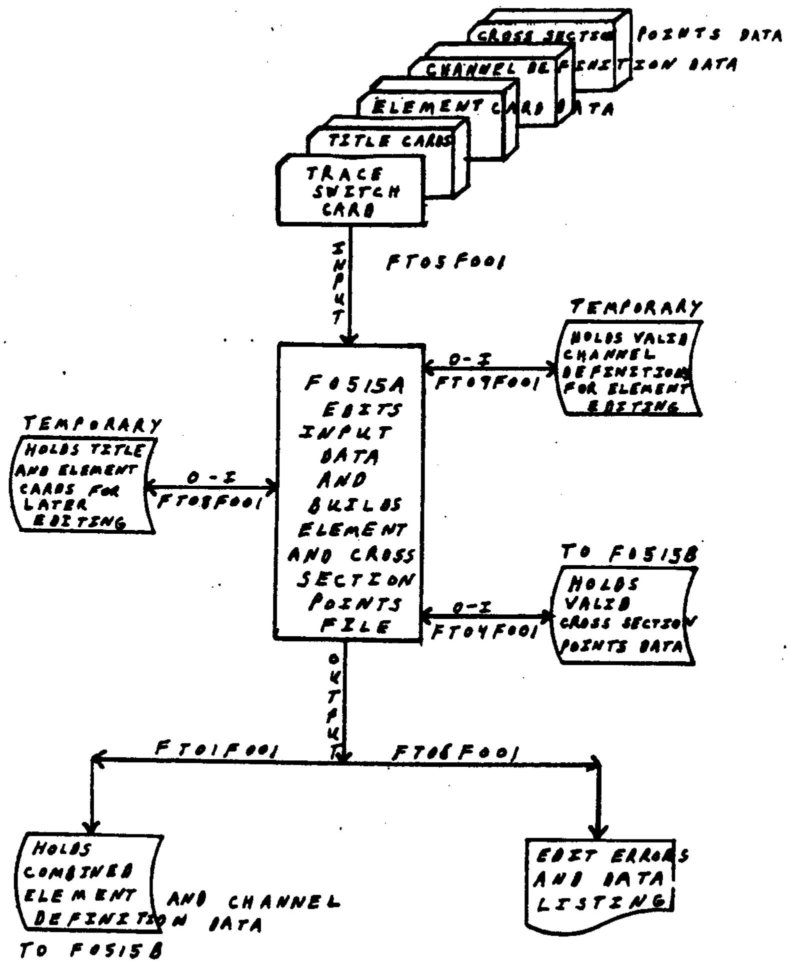
* Characteristics
Alphabetic or Blank ___ A
Alphanumeric ___ X
Numeric ___ 9
Assumed Decimal Point ___ V
Examples of Signed Fields
X9999 9999X
X999V99 9999V9X

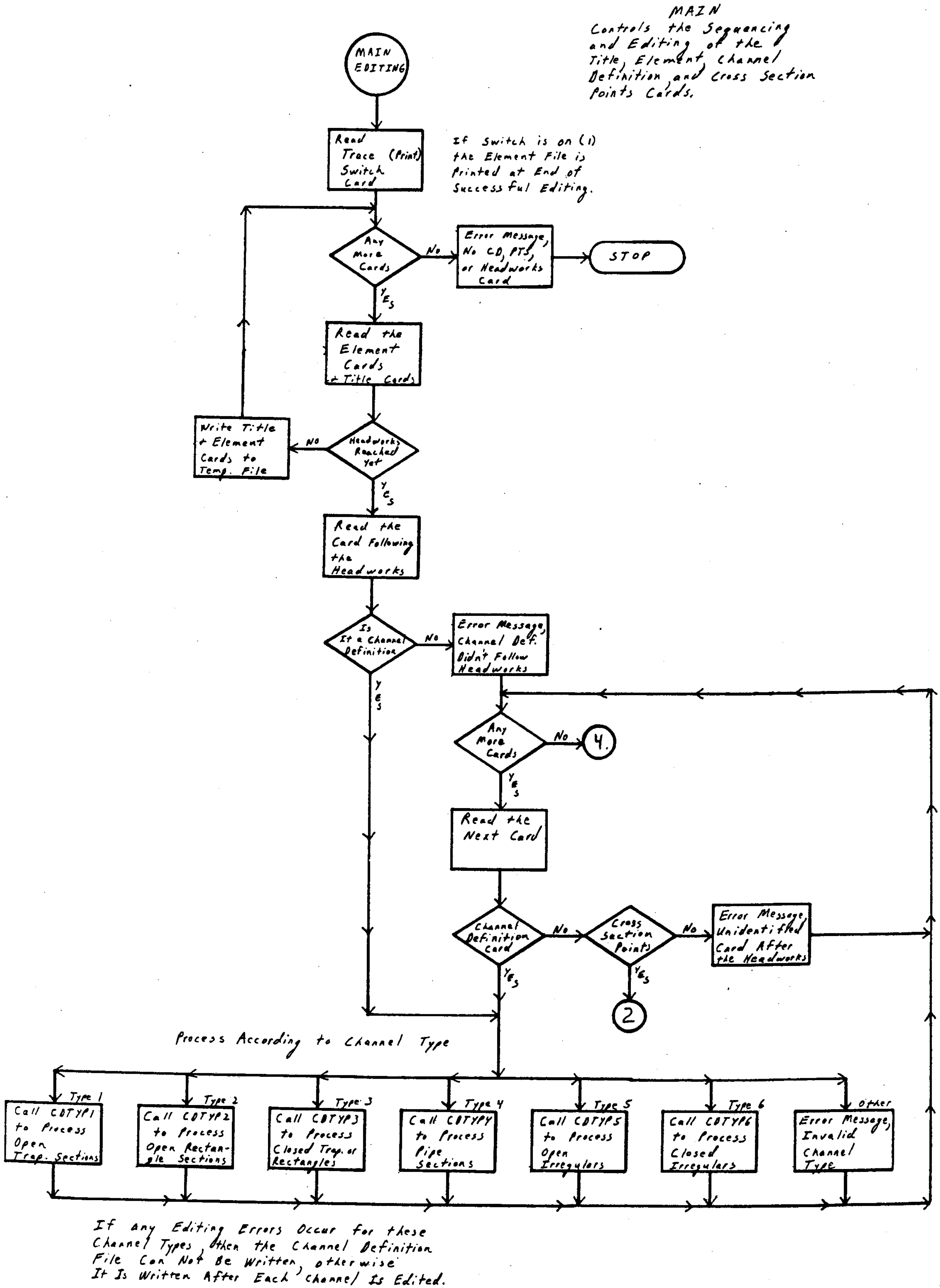
REVISIONS

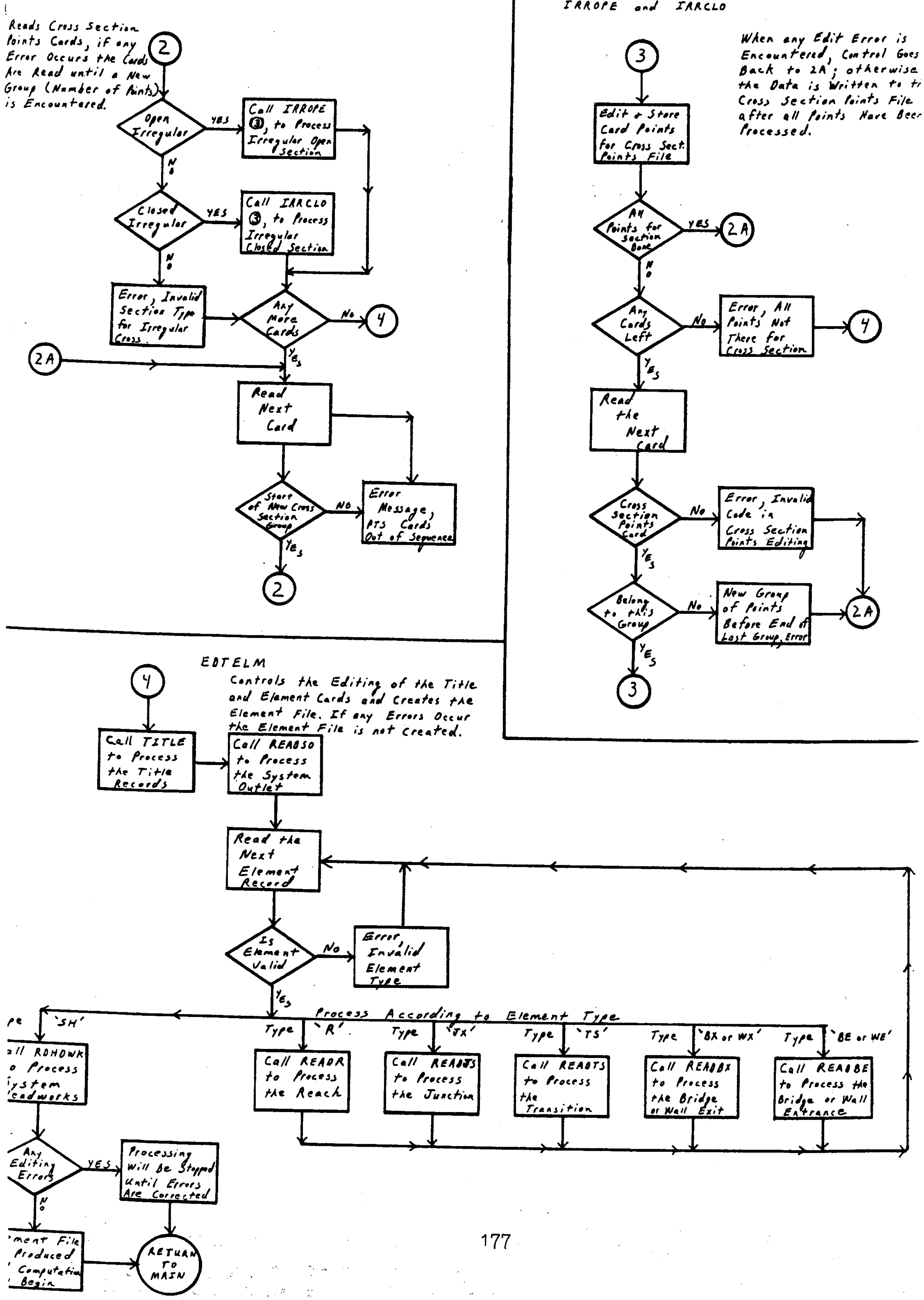
Date	By

F0515A

File Flowchart
General Layout

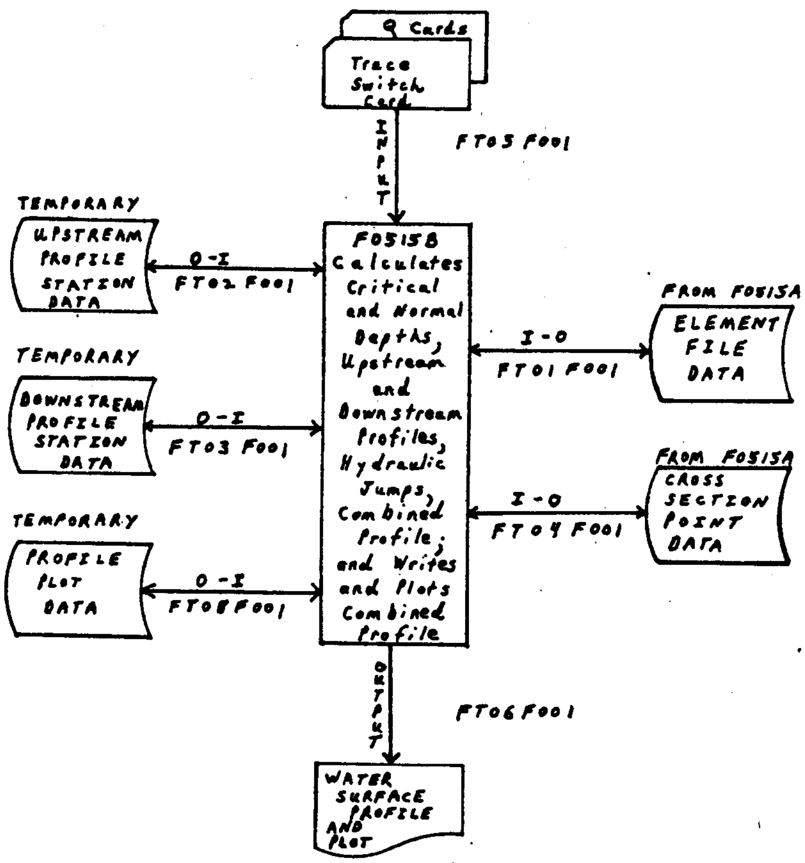



MAIN
Controls the Sequencing and Editing of the Title, Element, Channel Definition and Cross Section Points Cards.
MAIN EDITING
Read Trace (Print) Switch Card
If Switch is on (1) the Element File is Printed at End of Successful Editing.
Any More Cards
No
Error Message, No CD, PTS, or Headworks Card
STOP
Yes
Read the Element Cards + Title Cards
Headworks Reached Yet
No
Write Title + Element Cards to Temp. File
Yes
Read the Card Following the Headworks
Is It a Channel Definition
No
Error Message, Channel Def. Didn't Follow Headworks
Yes
Any More Cards
No
4.
Yes
Read the Next Card
Channel Definition Card
No
Cross Section Points
No
Error Message, Unidentified Card After the Headworks
Yes
2
Yes
Process According to Channel Type
Type 1
Call CDTYP1 to Process Open Trap. Sections
Type 2
Call CDTYP2 to Process Open Rectangle Sections
Type 3
Call CDTYP3 to Process Closed Trap. or Rectangles
Type 4
Call CDTYPY to Process Pipe Sections
Type 5
Call CDTYP5 to Process Open Irregulars
Type 6
Call CDTYP6 to Process Closed Irregulars
Other
Error Message, Invalid Channel Type
If any Editing Errors Occur for these Channel Types, then the Channel Definition File Can Not Be Written, otherwise It Is Written After Each Channel Is Edited.

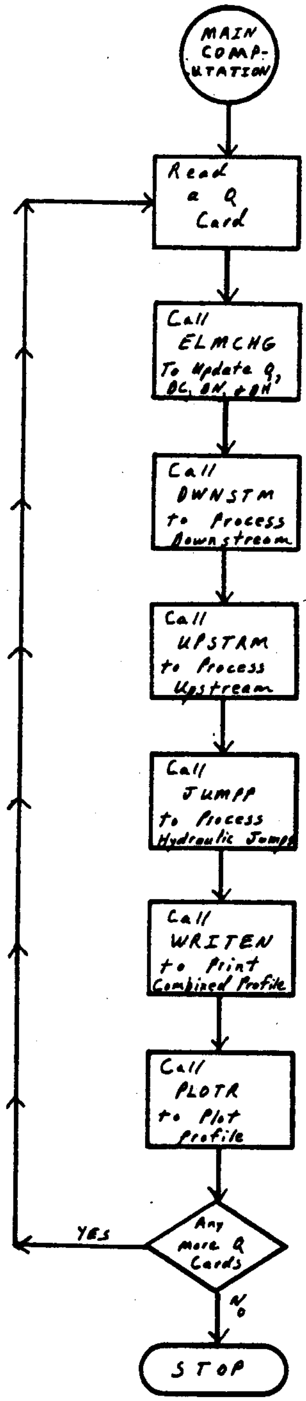

IRROPE and IARCLO
Reads Cross Section Points Cards, if any Error Occurs the Cards Are Read until a New Group (Number of Points) is Encountered.
2
Open Irregular
YES
Call IRROPE ③, to Process Irregular Open Section
NO
Closed Irregular
YES
Call IARCLO ③, to Process Irregular Closed Section
NO
Error, Invalid Section Type for Irregular Cross
Any More Cards
No
4
2A
Yes
Read Next Card
Start of New Cross Section Group
NO
Error Message, PTS Cards Out of Sequence
Yes
2
3
When any Edit Error is Encountered, Control Goes Back to 2A; otherwise the Data is Written to the Cross Section Points File after all Points Have Been Processed.
Edit + Store Card Points for Cross Sect. Points File
All Points for Section Done
YES
2A
NO
Any Cards Left
No
Error, All Points Not There for Cross Section
4
Yes
Read the Next Card
Cross Section Points Card
No
Error, Invalid Code in Cross Section Points Editing
Yes
Belong to this Group
No
New Group of Points Before End of Last Group, Error
2A
Yes
3
EDTELM
Controls the Editing of the Title and Element Cards and Creates the Element File. If any Errors Occur the Element File is not created.
4
Call TITLE to Process the Title Records
Call READSO to Process the System Outlet
Read the Next Element Record
Is Element Valid
No
Error, Invalid Element Type
Yes
Process According to Element Type
'SH'
Type 'R'
Type 'JX'
Type 'TS'
Type 'BX or WX'
Type 'BE or WE'
Call RDHDWK to Process System Headworks
Call READR to Process the Reach
Call READJS to Process the Junction
Call READTS to Process the Transition
Call READBX to Process the Bridge or Wall Exit
Call READBE to Process the Bridge or Wall Entrance
Any Editing Errors
YES
Processing Will Be Stopped Until Errors Are Corrected
NO
Element File Produced Computation Begin
RETURN TO MAIN

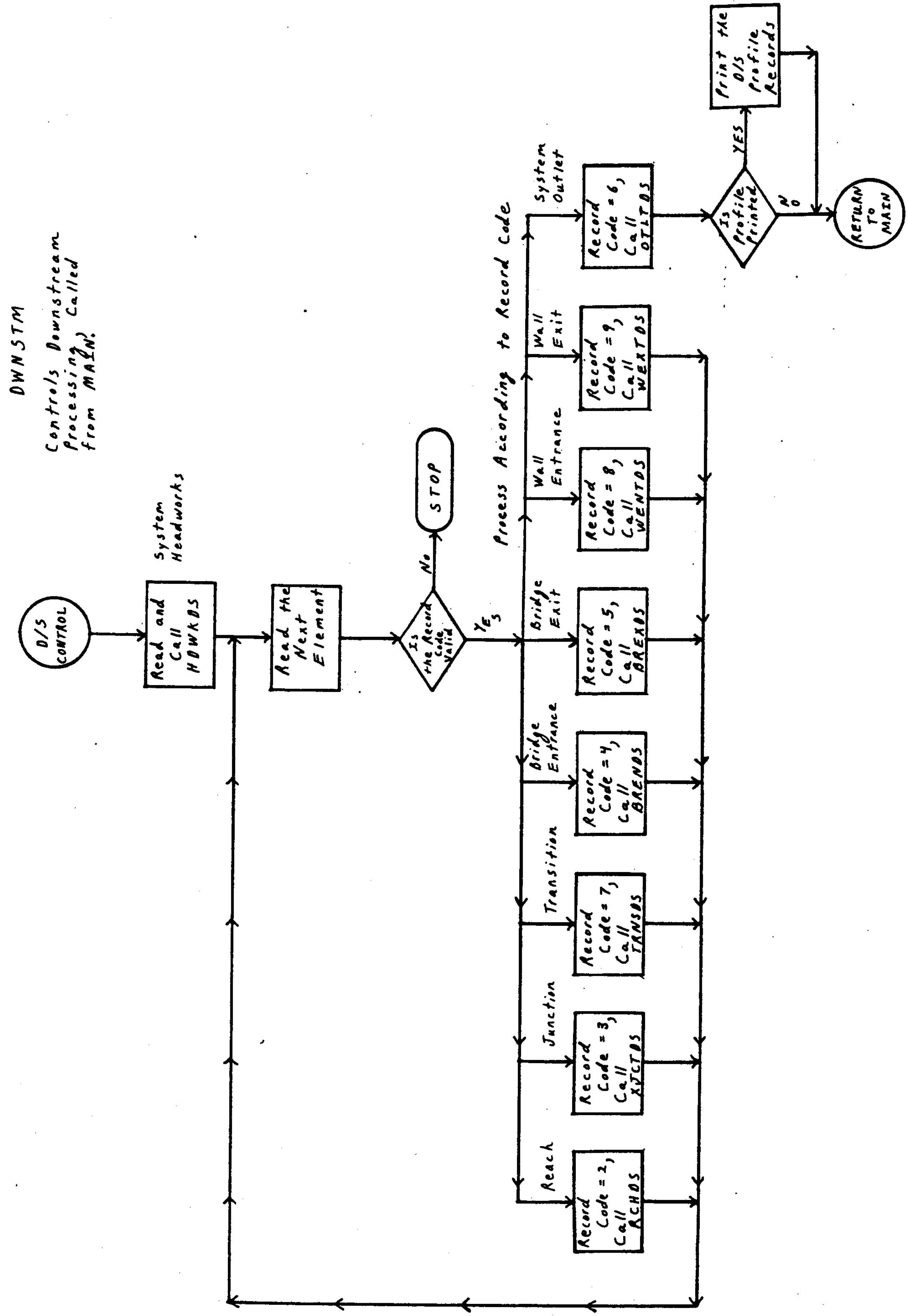
F0515B

File Flowchart
General Layout



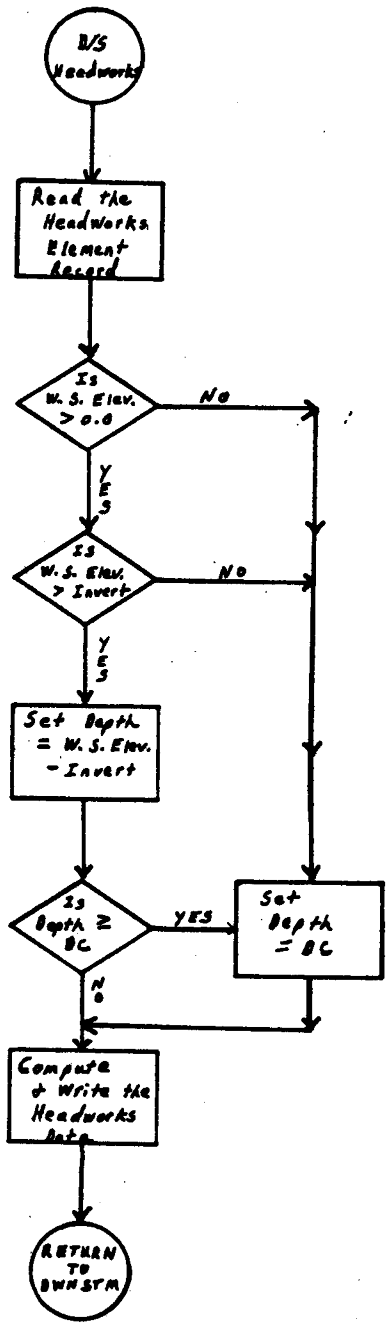
Controls the Computation of the Water Surface Profile.



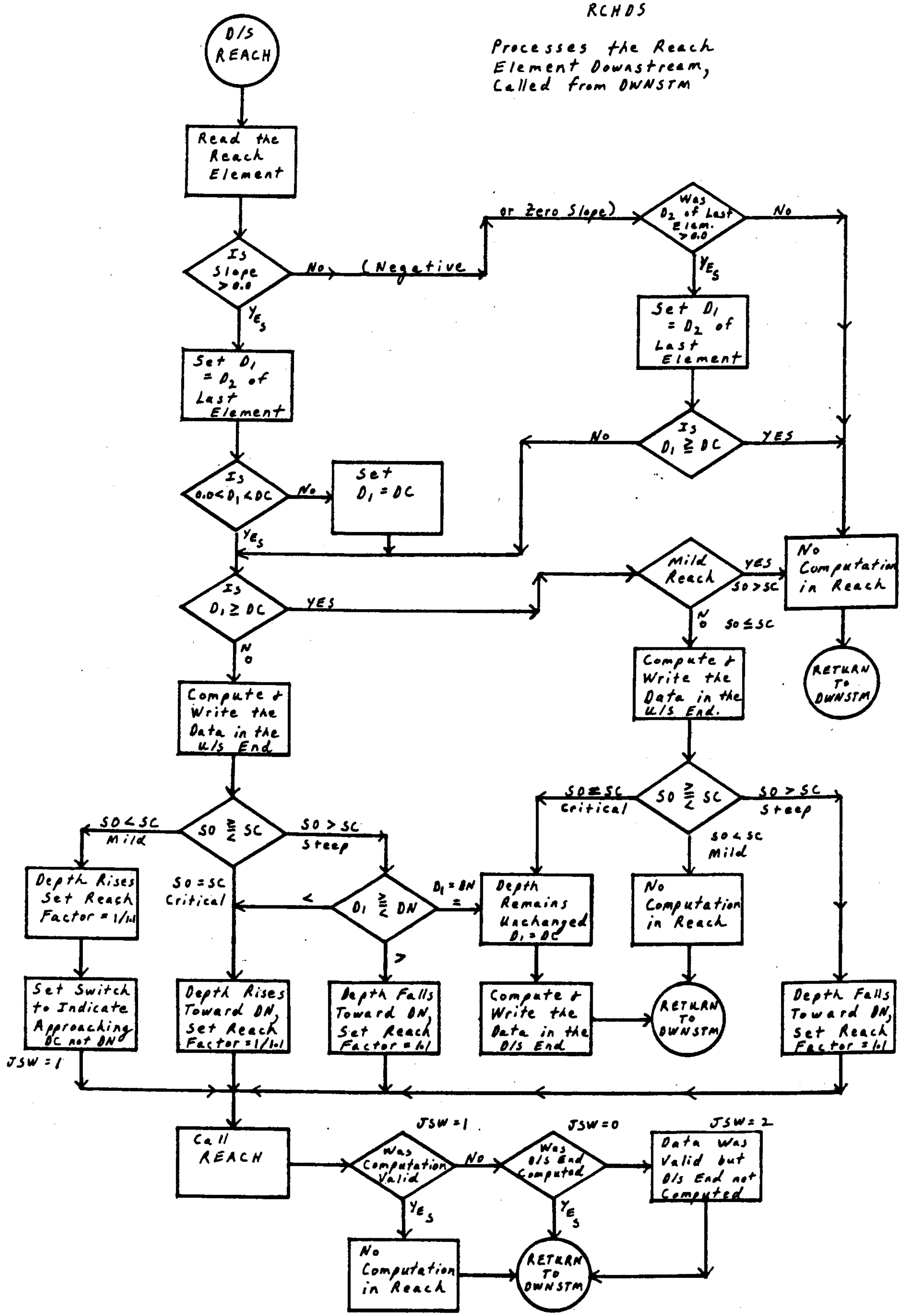

DWNSTM
Controls Downstream Processing, Called from MAIN.
D/S CONTROL
System Headworks
Read and Call HDWKDS
Read the Next Element
Is the Record Code Valid
No
STOP
YES
Process According to Record Code
Reach
Record Code = 2, Call RCHDS
Junction
Record Code = 3, Call XJCTDS
Transition
Record Code = 7, Call TRNSDS
Bridge Entrance
Record Code = 4, Call BRENDS
Bridge Exit
Record Code = 5, Call BREXDS
Wall Entrance
Record Code = 8, Call WENTDS
Wall Exit
Record Code = 9, Call WEXTDS
System Outlet
Record Code = 6, Call OTLTDS
Is Profile Printed
YES
NO
Print the D/S Profile Records
RETURN TO MAIN

HDWKDS

Processes the System Headworks Element Downtstream, Called from DWNSTM.

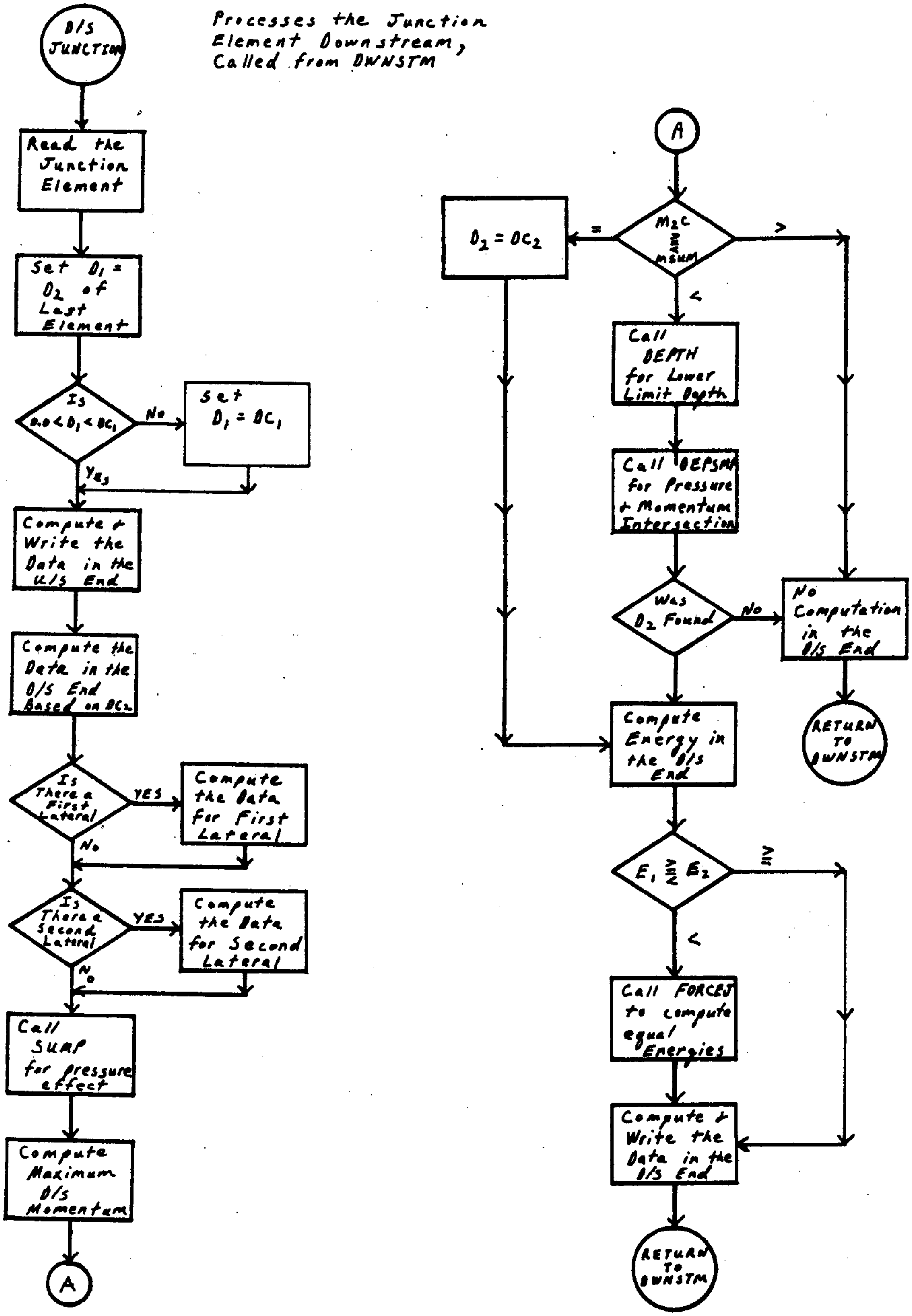


Processes the Reach Element Downstream, Called from DWNSTM



XJCTDS

Processes the Junction Element Downstream, Called from DWNSTM

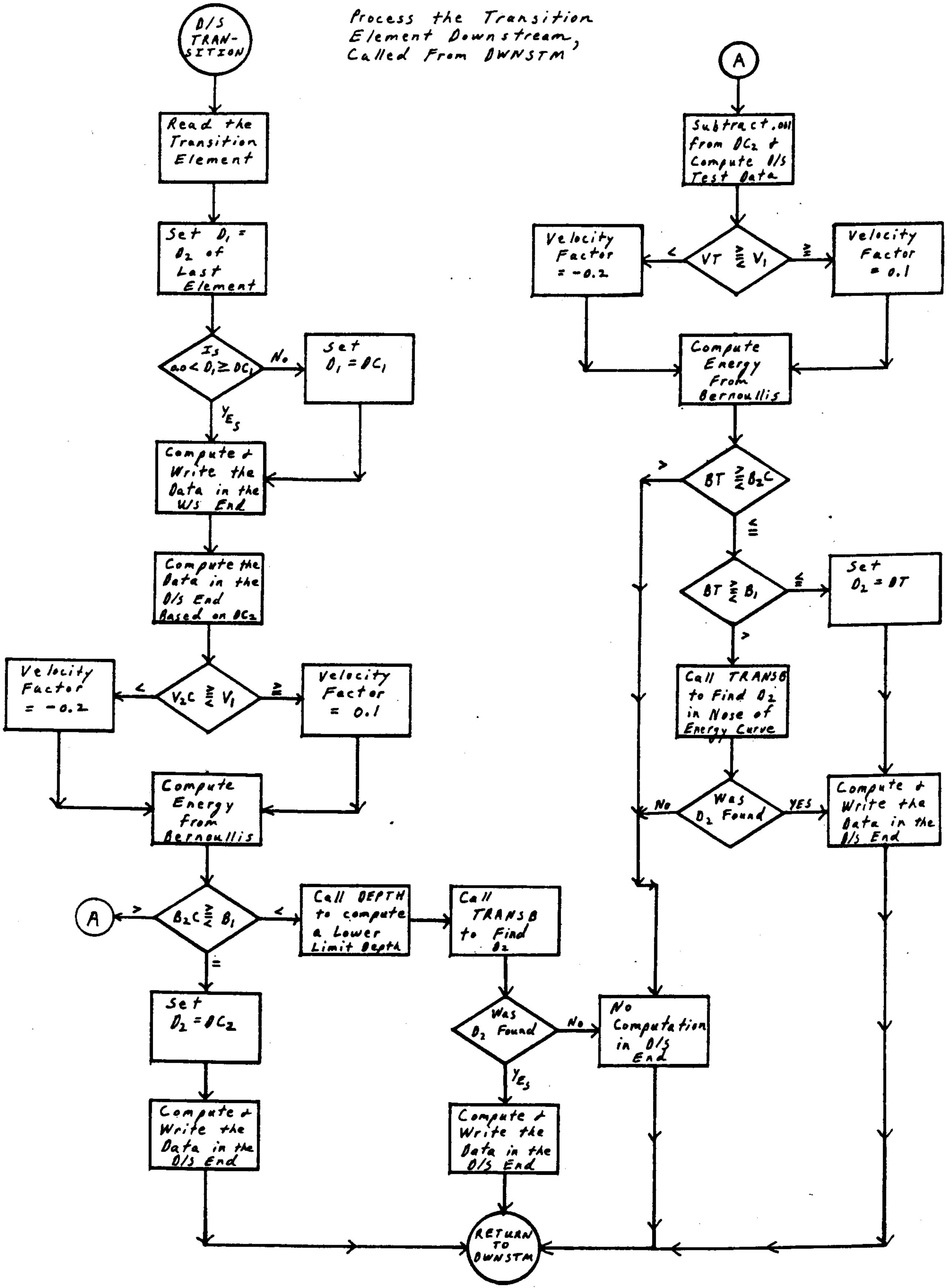


$E_1 = D_1 + (HV_1^2/(2*G)) + H - (DELL * SF_1/2)$

$MSUM = M_1 + M_3 COS + M_4 COS +$ (Pressure from SUMP)

$E_2 = D_2 + (HV_2^2/(2*G)) + (DELL * SF_2/2)$

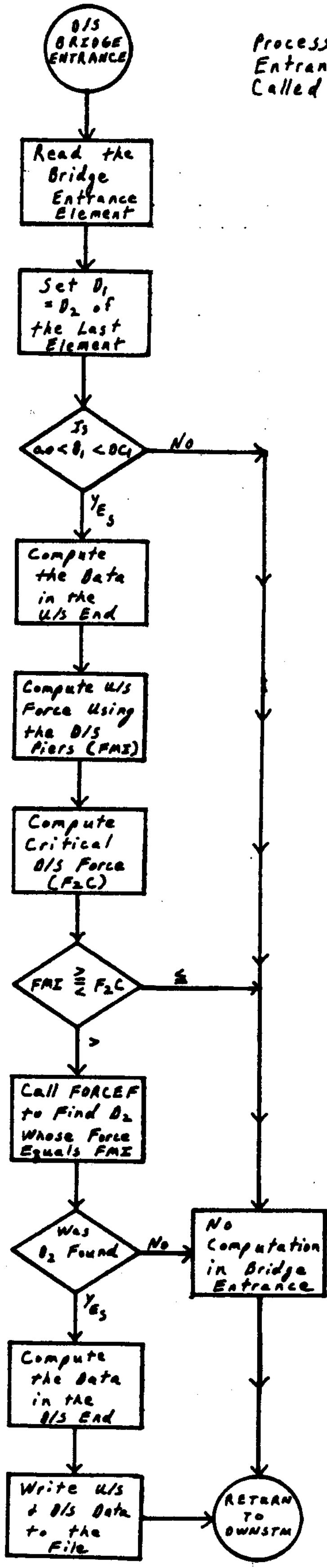
Process the Transition Element Downstream, Called From DWNSTM



$$B1 = D_1 + HV_1 + H - (DELL * SF_1 / 2)$$
$$B_2C = DC_2 + HVC_2 + (K * (HVC_2 - HV_1) + (DELL * SFC_2 / 2))$$
$$BT = DT + HVT + (K * (HVT - HV1) + (DELL * SFT / 2))$$

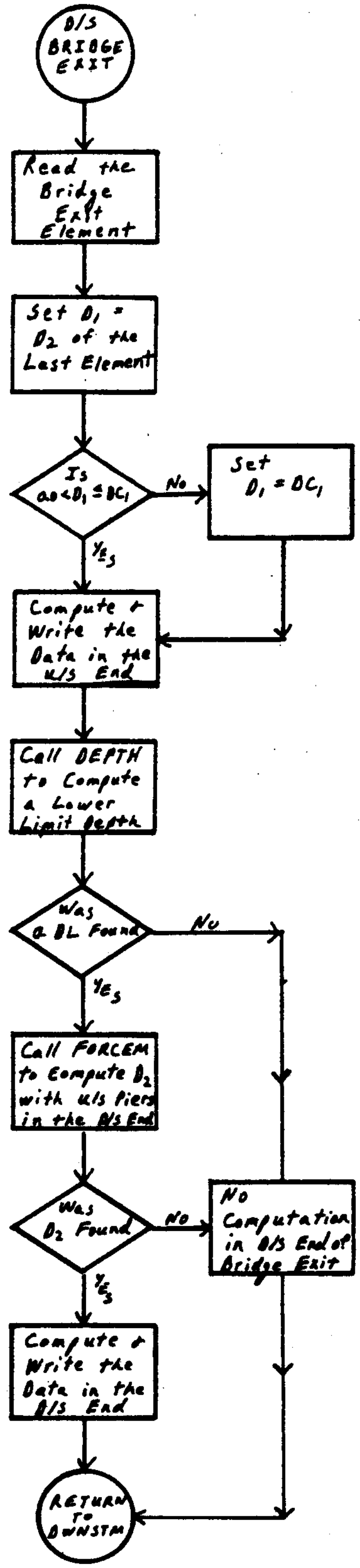
BRENDS

Processes the Bridge Entrance Downstream, Called From DWNSTM.



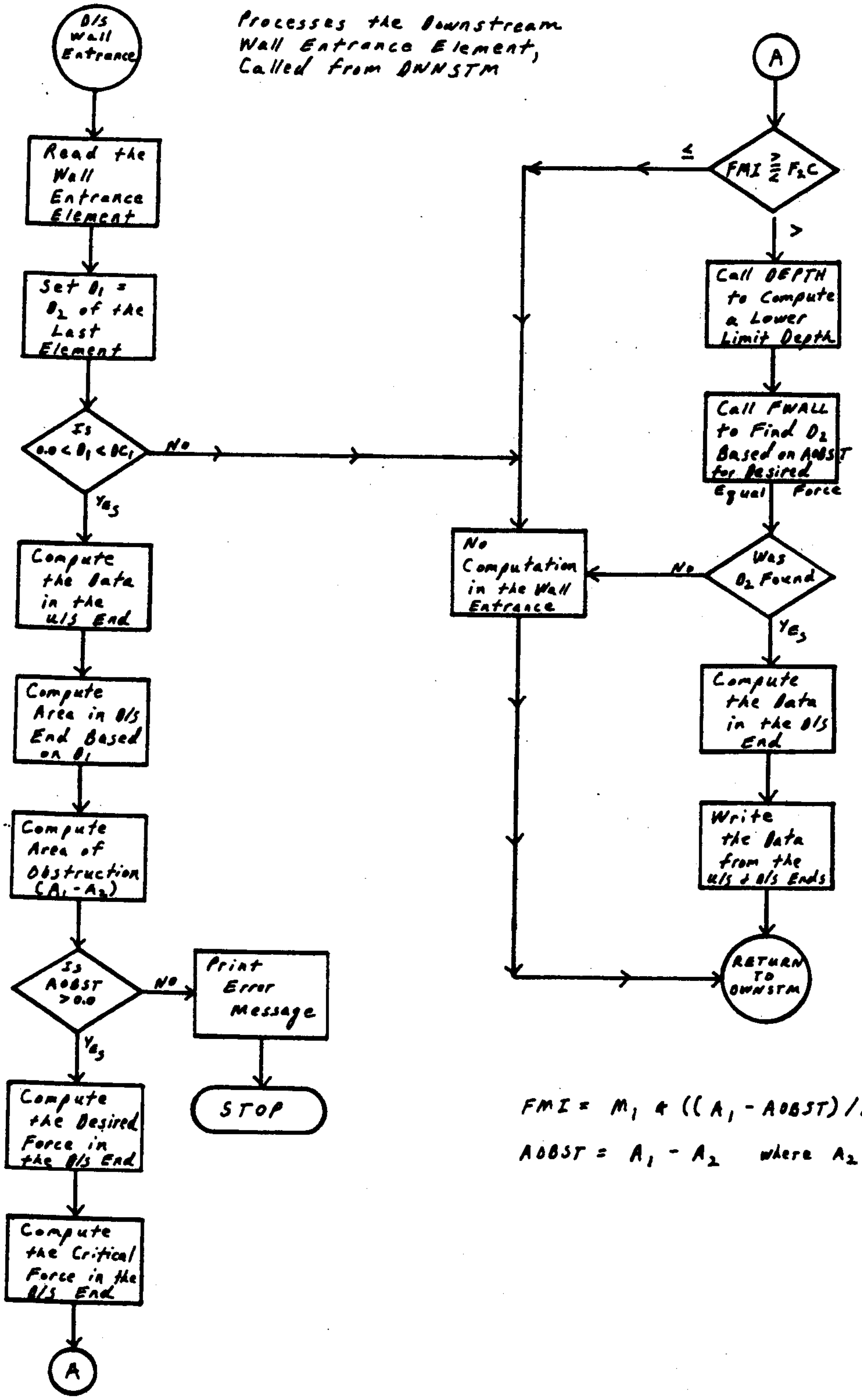
BREXDS

Processes the Downstream Bridge Exit Element, Called From DWNSTM



WENTDS

Processes the Downstream Wall Entrance Element, Called from DWNSTM

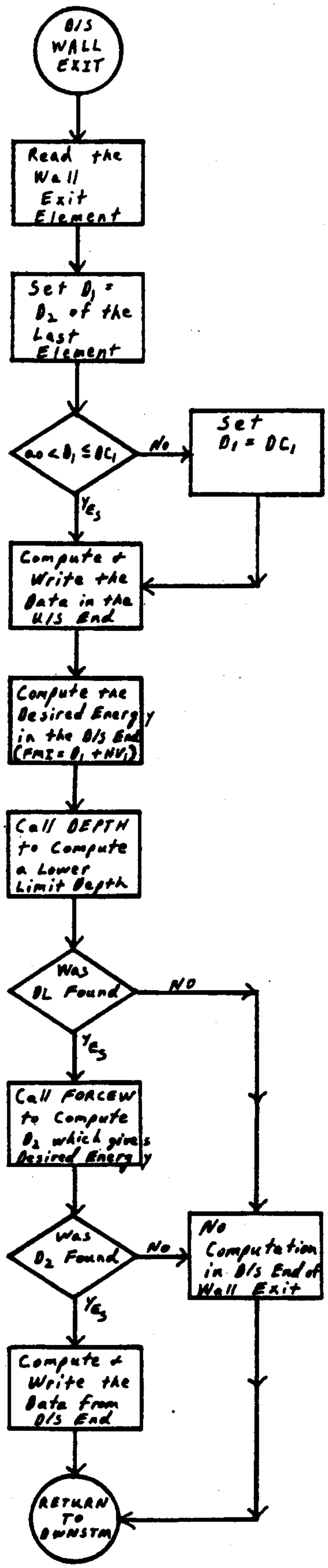


$$FMI = M_1 * ((A_1 - AOBST)/A_1) + P_1$$

$AOBST = A_1 - A_2$ where A_2 is based on D_1

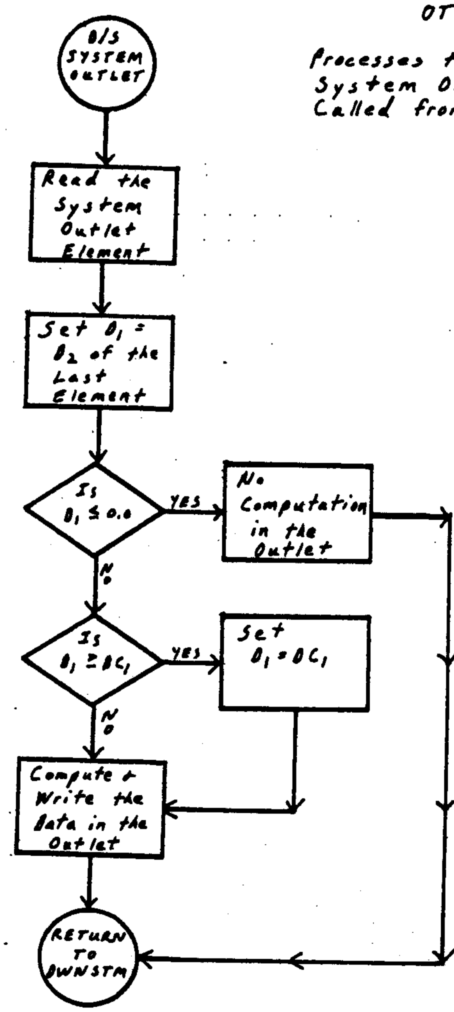
WEXTDS

Processes the Downstream Wall Exit Element, Called from DWNSTM



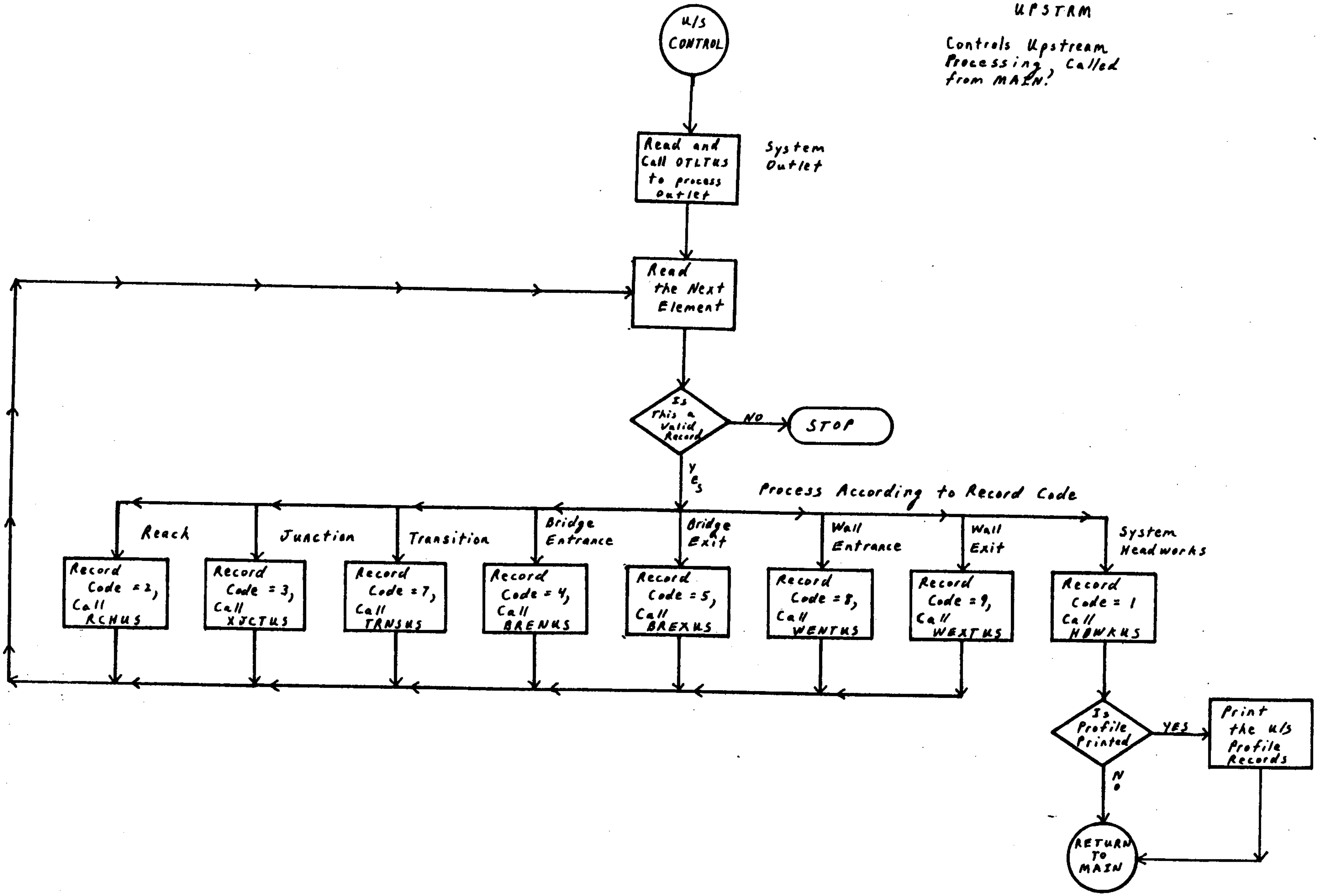
OTLTDS

Processes the Downstream System Outlet Element, Called from DWNSTM



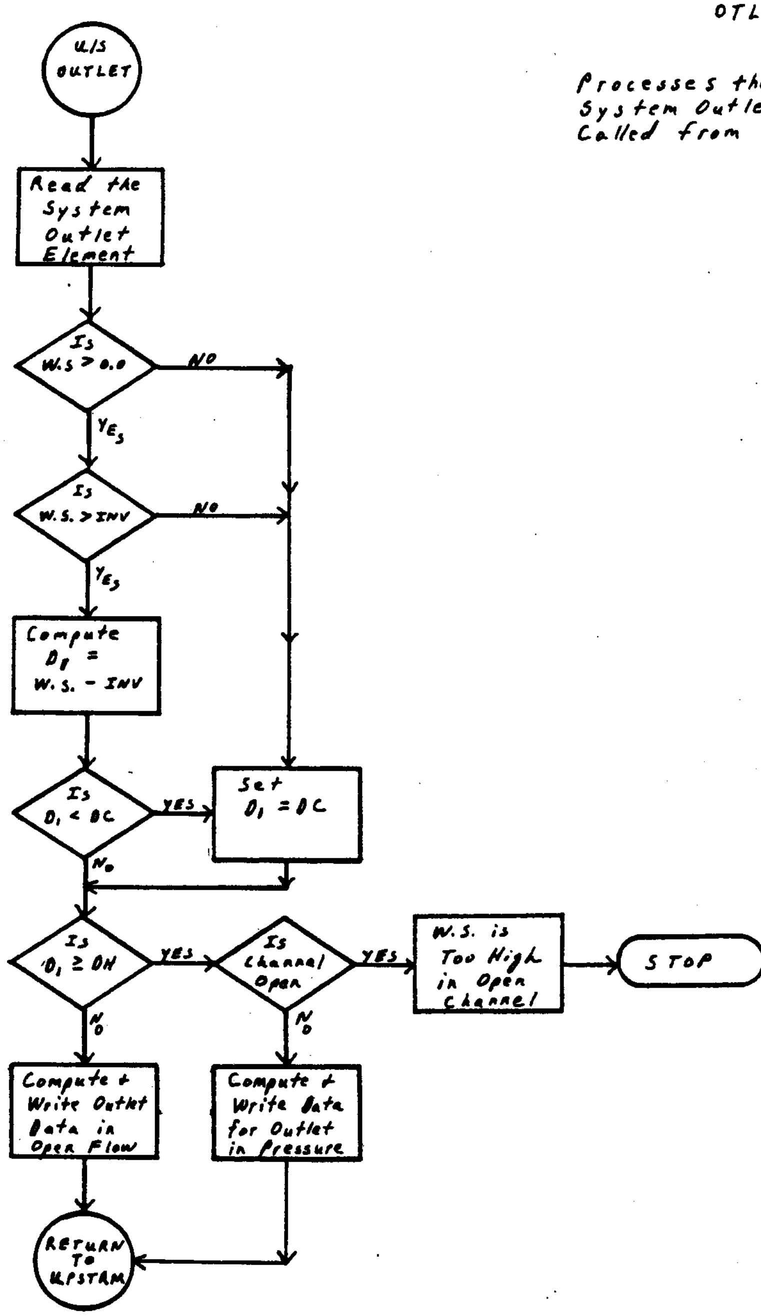
UPSTRM

Controls Upstream Processing, Called from MAIN.



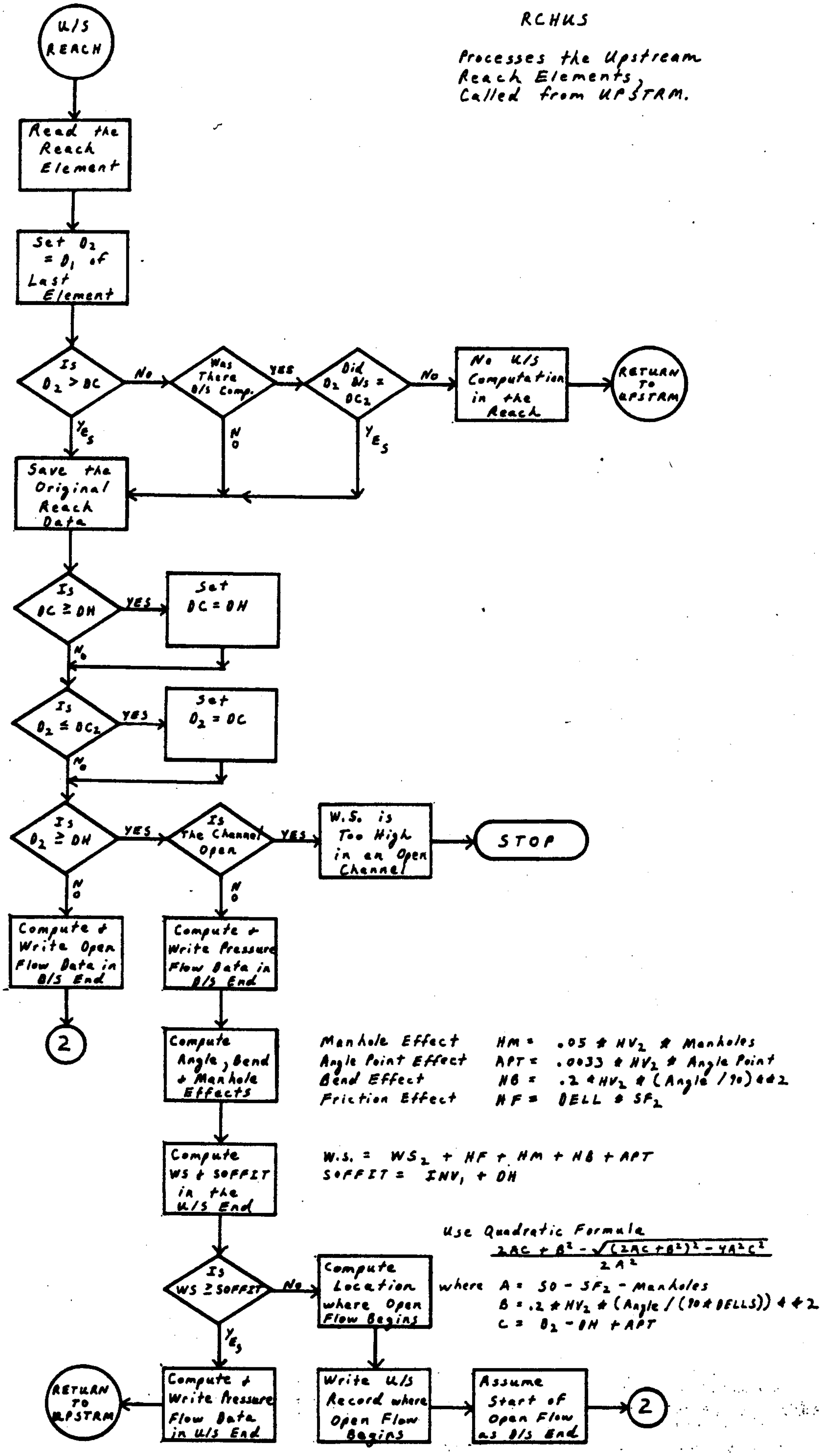
OTLTUS

Processes the Upstream System Outlet Element, Called from UPSTRM



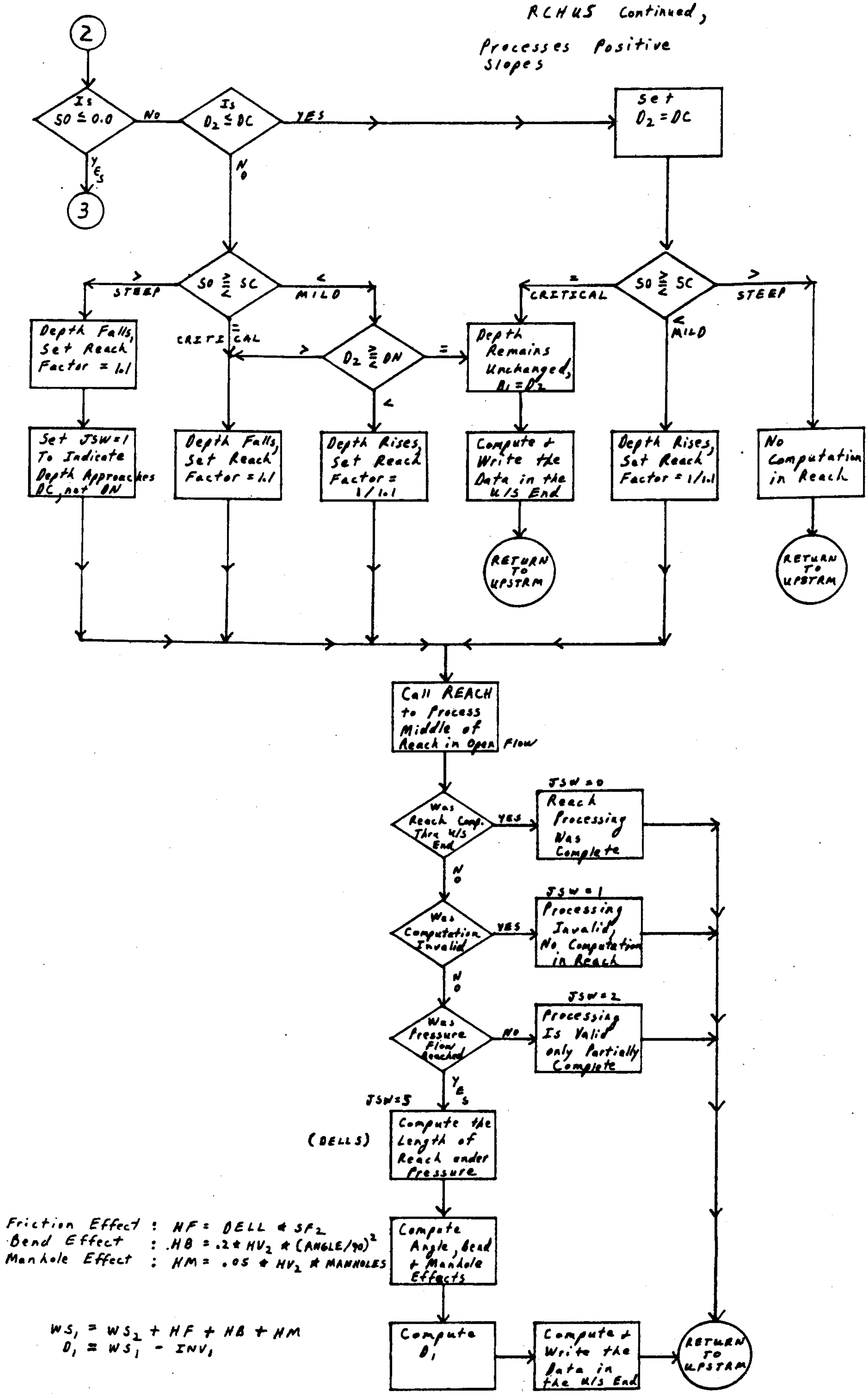
RCHUS

Processes the Upstream Reach Elements, Called from UPSTRM.



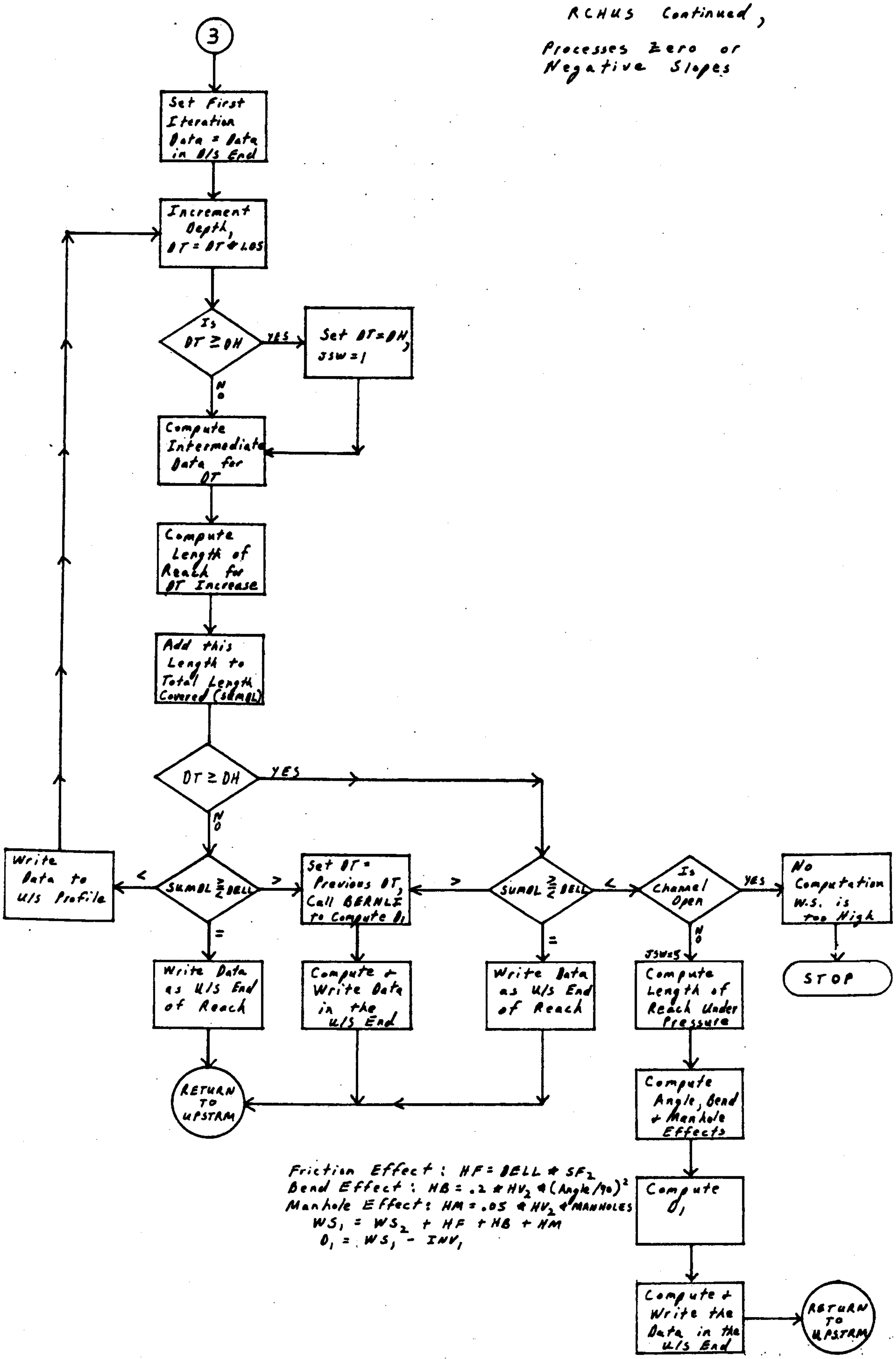
RCHUS Continued,

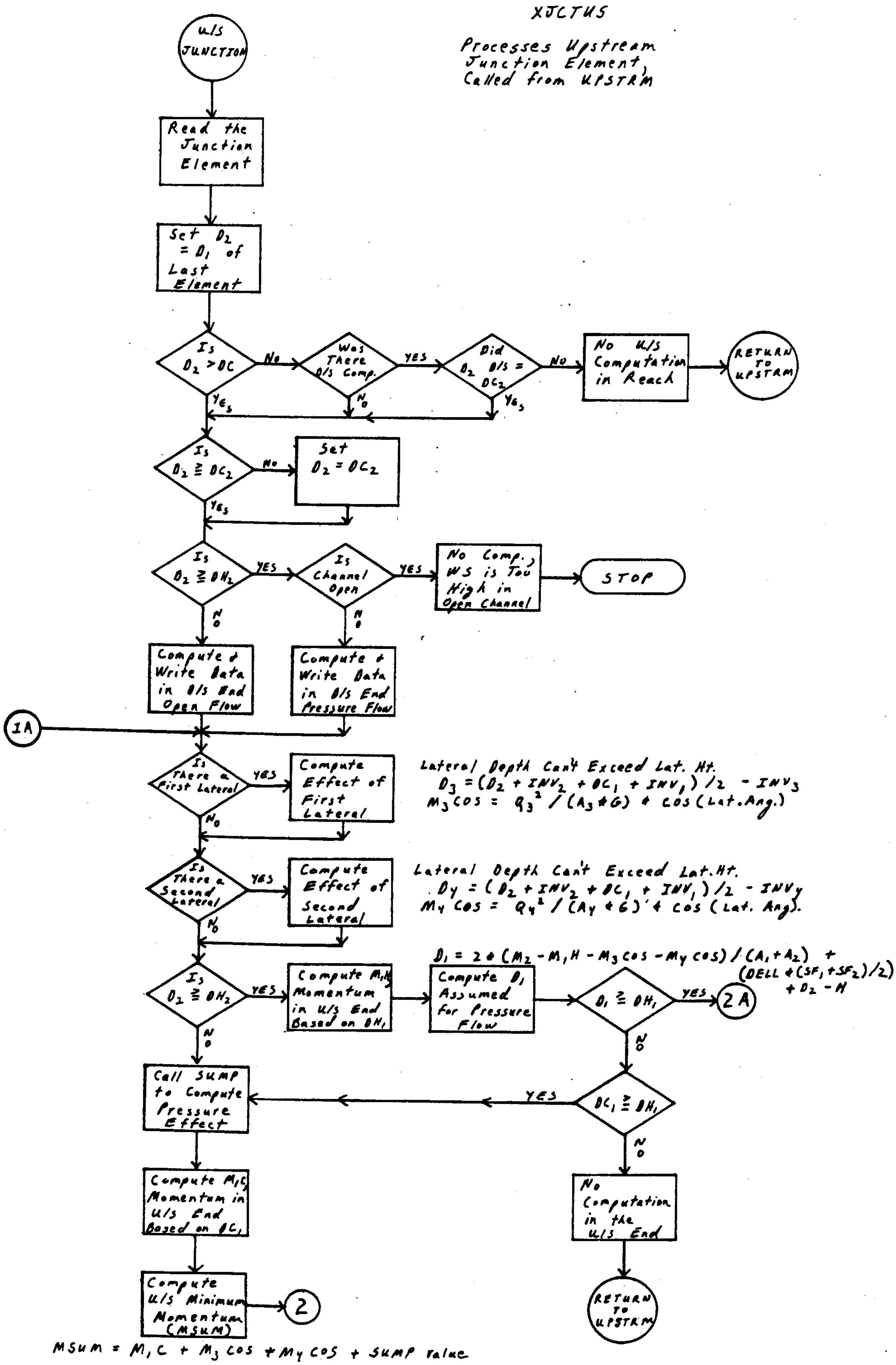
Processes Positive Slopes



RCHUS Continued,

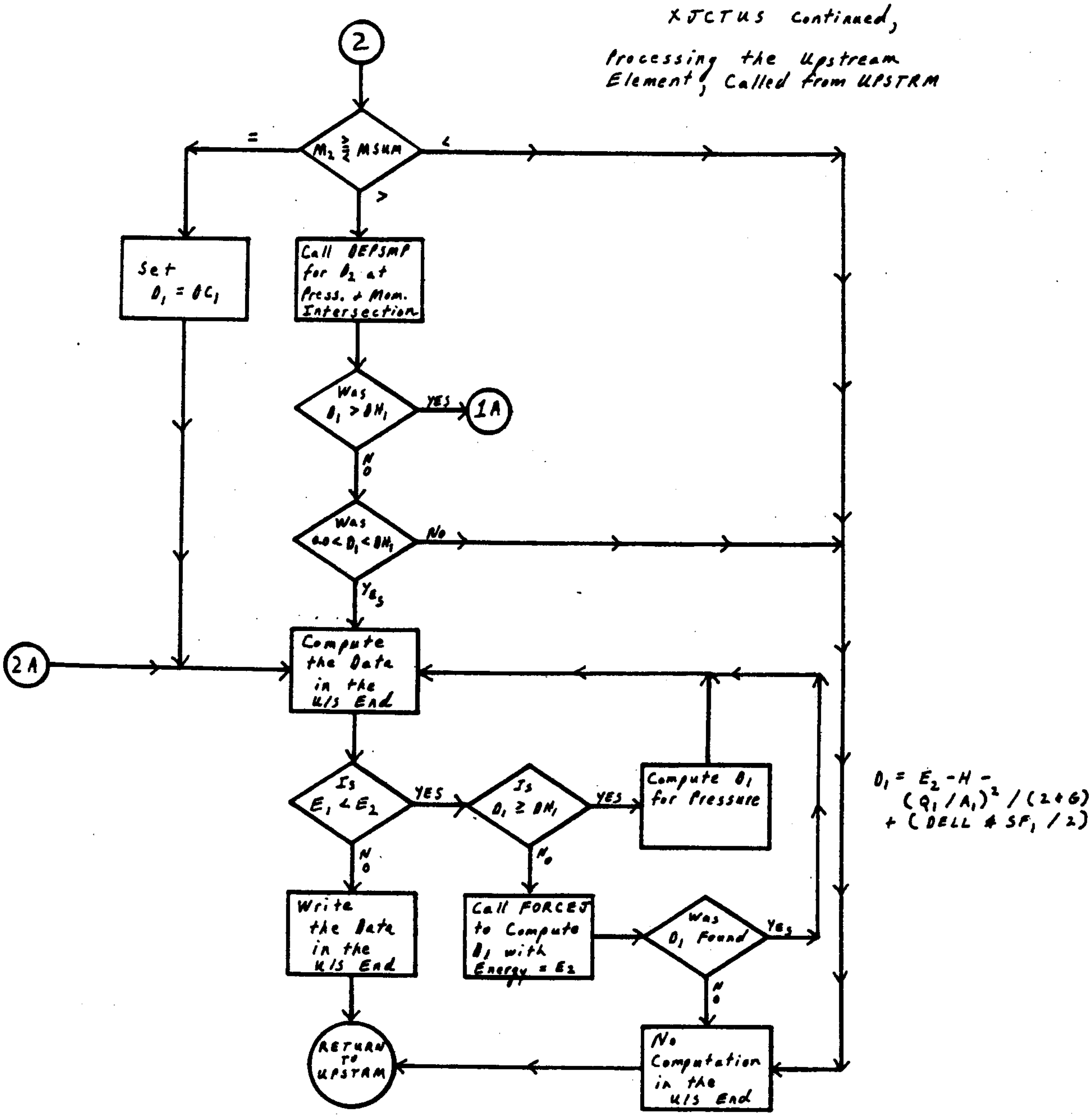
Processes Zero or Negative Slopes




XJCTUS
Processes Upstream Junction Element, Called from UPSTRM
U/S JUNCTION
Read the Junction Element
Set D2 = D1 of Last Element
Is D2 > DC
No
Was There D/S Comp.
YES
Did D2 D/S = DC2
NO
No U/S Computation in Reach
RETURN TO UPSTRM
YES
NO
YES
Is D2 ≧ DC2
NO
Set D2 = DC2
YES
Is D2 ≧ DH2
YES
Is Channel Open
YES
No Comp., WS is Too High in Open Channel
STOP
NO
NO
Compute & Write Data in D/S End Open Flow
Compute & Write Data in D/S End Pressure Flow
1A
Is There a First Lateral
YES
Compute Effect of First Lateral
NO
Lateral Depth Can't Exceed Lat. Ht.
D3 = (D2 + INV2 + DC1 + INV1) / 2 - INV3
M3COS = Q3² / (A3 * G) * COS (Lat. Ang.)
Is There a Second Lateral
YES
Compute Effect of Second Lateral
NO
Lateral Depth Can't Exceed Lat. Ht.
D4 = (D2 + INV2 + DC1 + INV1) / 2 - INV4
M4 COS = Q4² / (A4 + G) * COS (Lat. Ang.)
D1 = 2 * (M2 - M1H - M3 COS - M4 COS) / (A1 + A2) + (DELL * (SF1 + SF2)/2) + D2 - H
Is D2 ≧ DH2
YES
Compute M1H Momentum in U/S End Based on DH1
Compute D1 Assumed for Pressure Flow
D1 ≧ DH1
YES
2A
NO
NO
DC1 ≧ DH1
YES
Call SUMP to Compute Pressure Effect
NO
Compute M1C Momentum in U/S End Based on DC1
No Computation in the U/S End
Compute U/S Minimum Momentum (MSUM)
2
RETURN TO UPSTRM
MSUM = M1C + M3 COS + M4 COS + SUMP Value

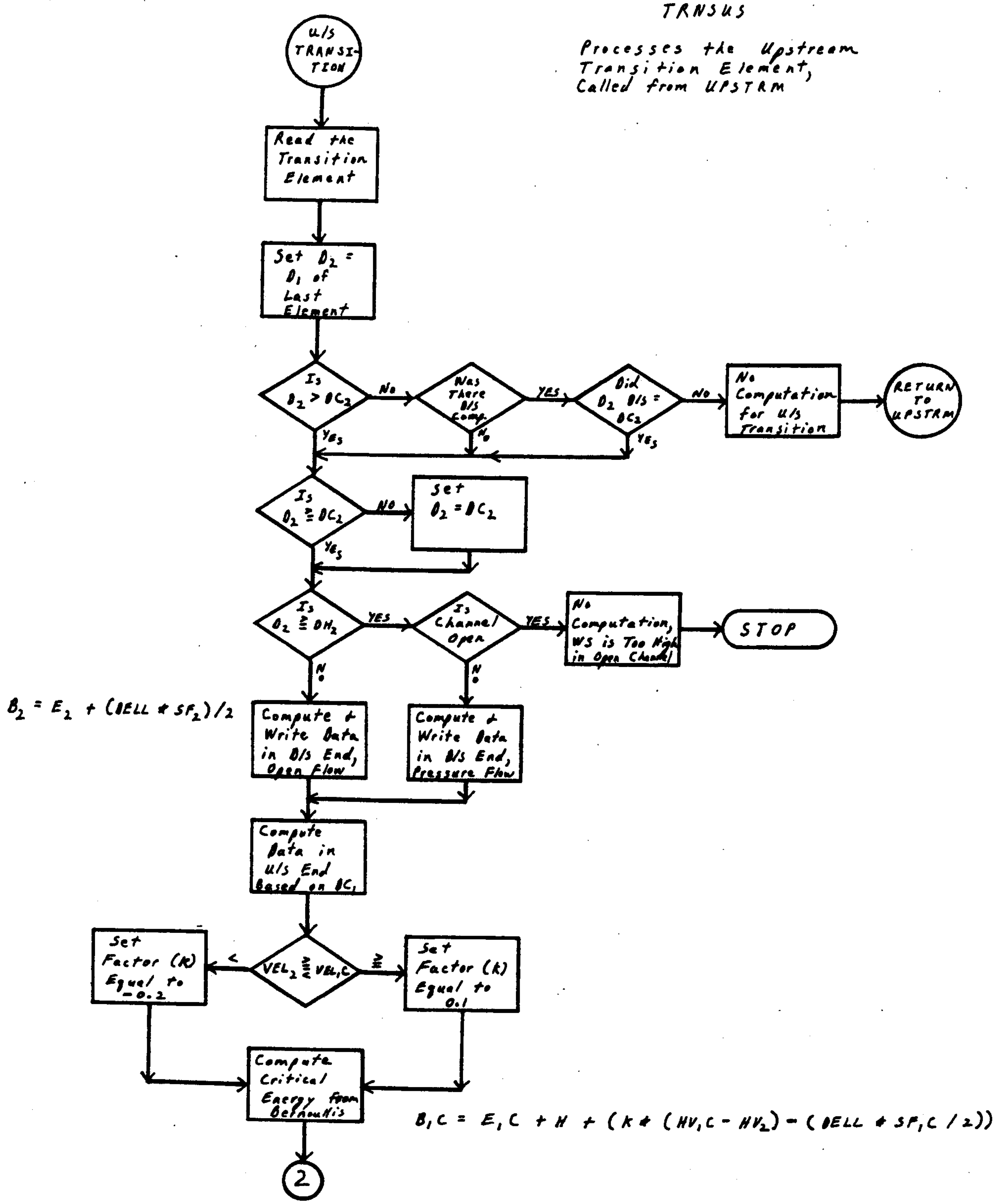
XJCTUS continued,

Processing the Upstream Element, Called from UPSTRM



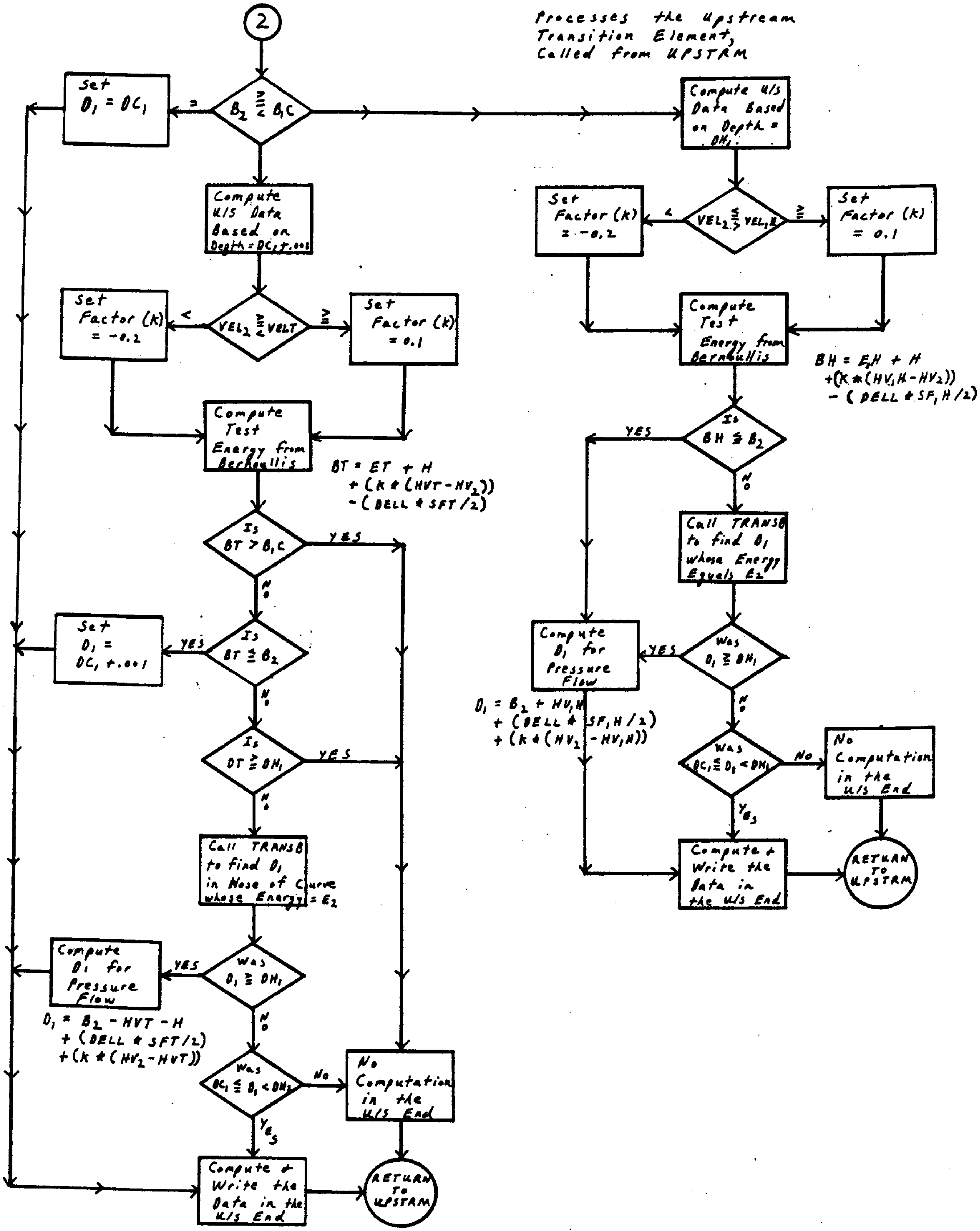
TRNSUS

Processes the Upstream Transition Element, Called from UPSTRM



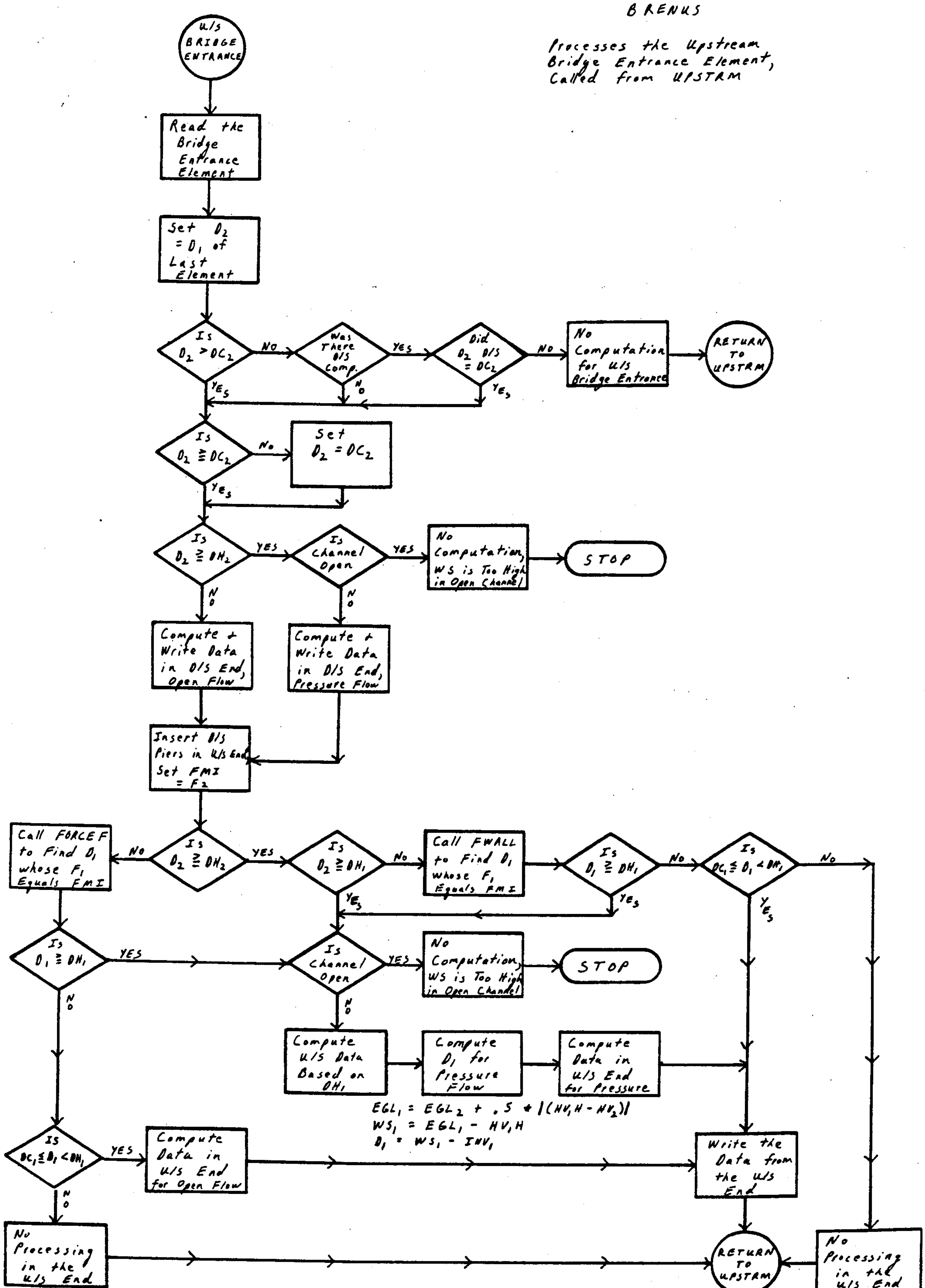
TRNSUS Continued,

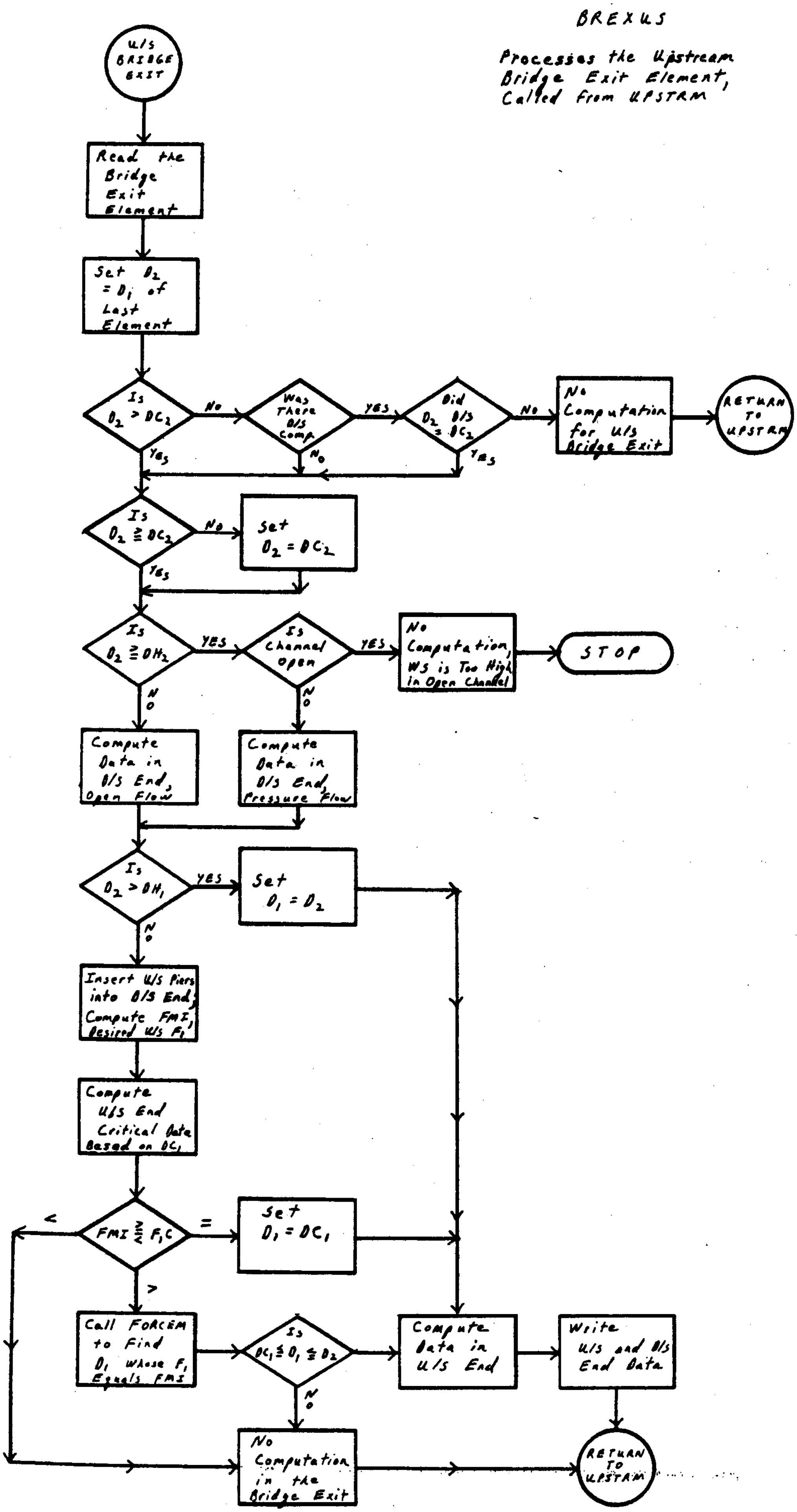
Processes the Upstream Transition Element, Called from UPSTRM



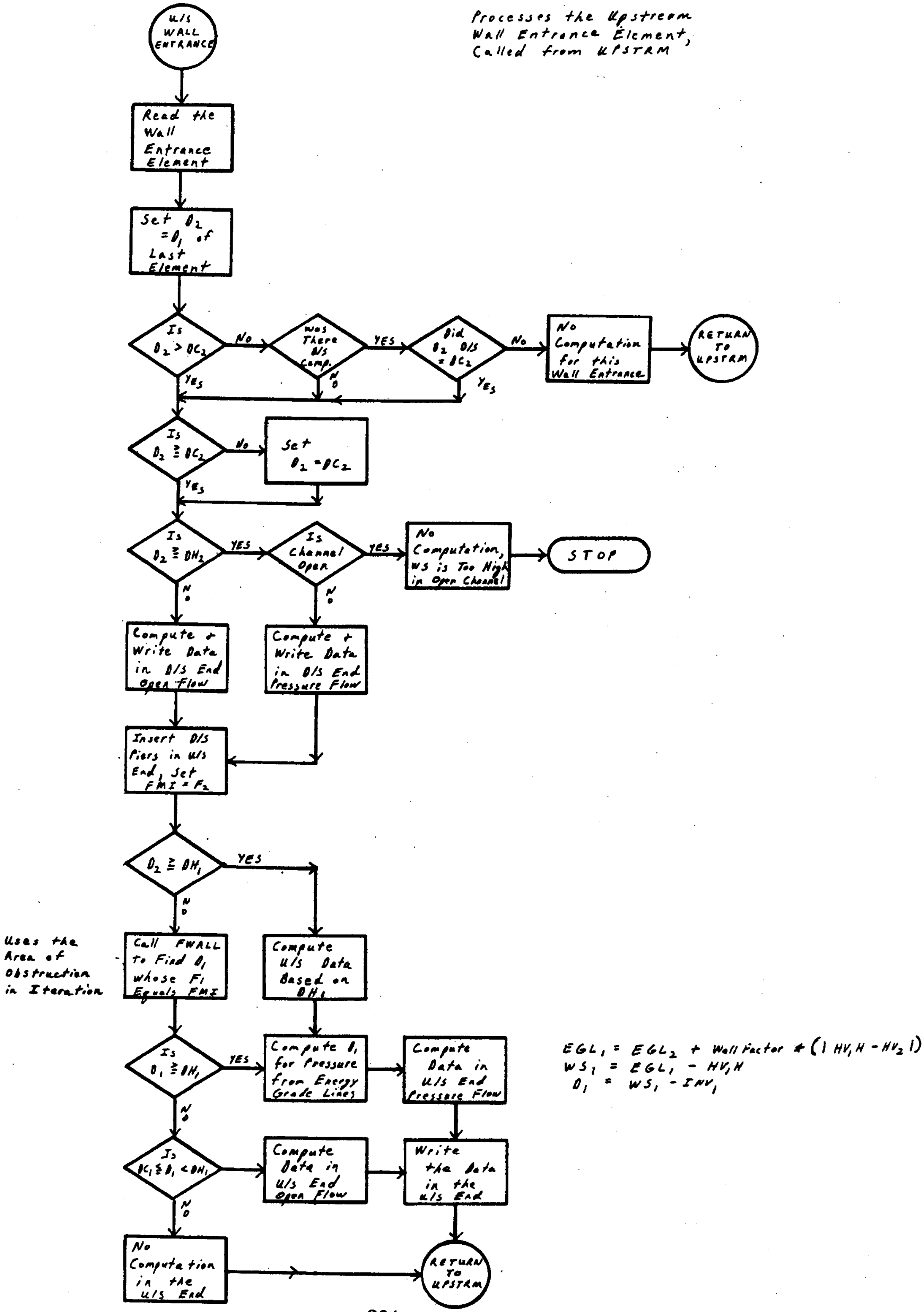
BRENUS

Processes the Upstream Bridge Entrance Element, Called from UPSTRM




BREXUS
Processes the Upstream Bridge Exit Element, Called from UPSTRM
U/S BRIDGE EXIT
Read the Bridge Exit Element
Set D2 = D1 of Last Element
Is D2 > DC2
No
Was There D/S comp.
YES
Did D2 D/S ≥ DC2
No
No Computation for U/S Bridge Exit
RETURN TO UPSTRM
Yes
No
Yes
Is D2 ≧ DC2
No
Set D2 = DC2
Yes
Is D2 ≧ DH2
YES
Is Channel open
YES
No Computation, WS is Too High in Open Channel
STOP
No
No
Compute Data in D/S End, Open Flow
Compute Data in D/S End, Pressure Flow
Is D2 > DH1
YES
Set D1 = D2
No
Insert U/S Piers into D/S End; Compute FMI, Desired U/S F1
Compute U/S End Critical Data Based on DC1
FMI ≧≦ F1C
<
=
Set D1 = DC1
>
Call FORCEM to Find D1 Whose F1 Equals FMI
Is DC1 ≦ D1 ≦ D2
Compute Data in U/S End
Write U/S and D/S End Data
No
No Computation in the Bridge Exit
RETURN TO UPSTRM

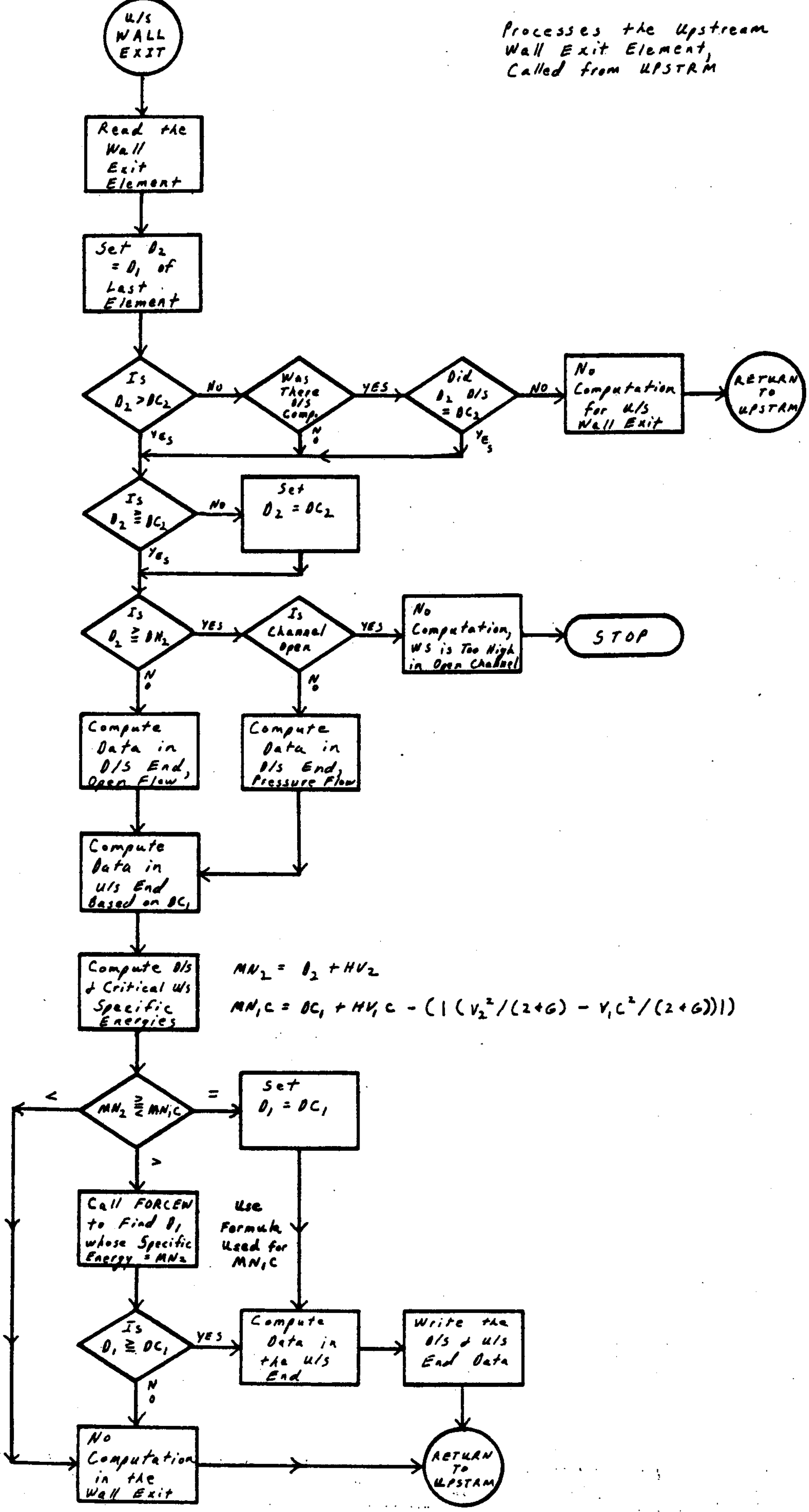
WENTUS



Processes the Upstream Wall Entrance Element, Called from UPSTRM
U/S WALL ENTRANCE
Read the Wall Entrance Element
Set $D_2 = D_1$ of Last Element
Is $D_2 > DC_2$
No
Was There D/S Comp.
YES
Did D_2 D/S $= DC_2$
No
No Computation for this Wall Entrance
RETURN TO UPSTRM
Yes
No
Yes
Is $D_2 \geq DC_2$
No
Set $D_2 = DC_2$
Yes
Is $D_2 \geq DH_2$
YES
Is Channel Open
YES
No Computation, WS is Too High in Open Channel
STOP
No
No
Compute & Write Data in D/S End Open Flow
Compute & Write Data in D/S End Pressure Flow
Insert D/S Piers in U/S End, Set $FMI = F_2$
$D_2 \geq DH_1$
YES
No
Uses the Area of Obstruction in Iteration
Call FWALL To Find D_1 whose F_1 Equals FMI
Compute U/S Data Based on DH_1
Is $D_1 \geq DH_1$
YES
Compute D_1 for Pressure from Energy Grade Lines
Compute Data in U/S End Pressure Flow
$EGL_1 = EGL_2$ + Wall Factor * (| $HV_1H - HV_2$ |)
$WS_1 = EGL_1 - HV_1H$
$D_1 = WS_1 - INV_1$
No
Is $DC_1 \leq D_1 < DH_1$
Compute Data in U/S End Open Flow
Write the Data in the U/S End
No
No Computation in the U/S End
RETURN TO UPSTRM

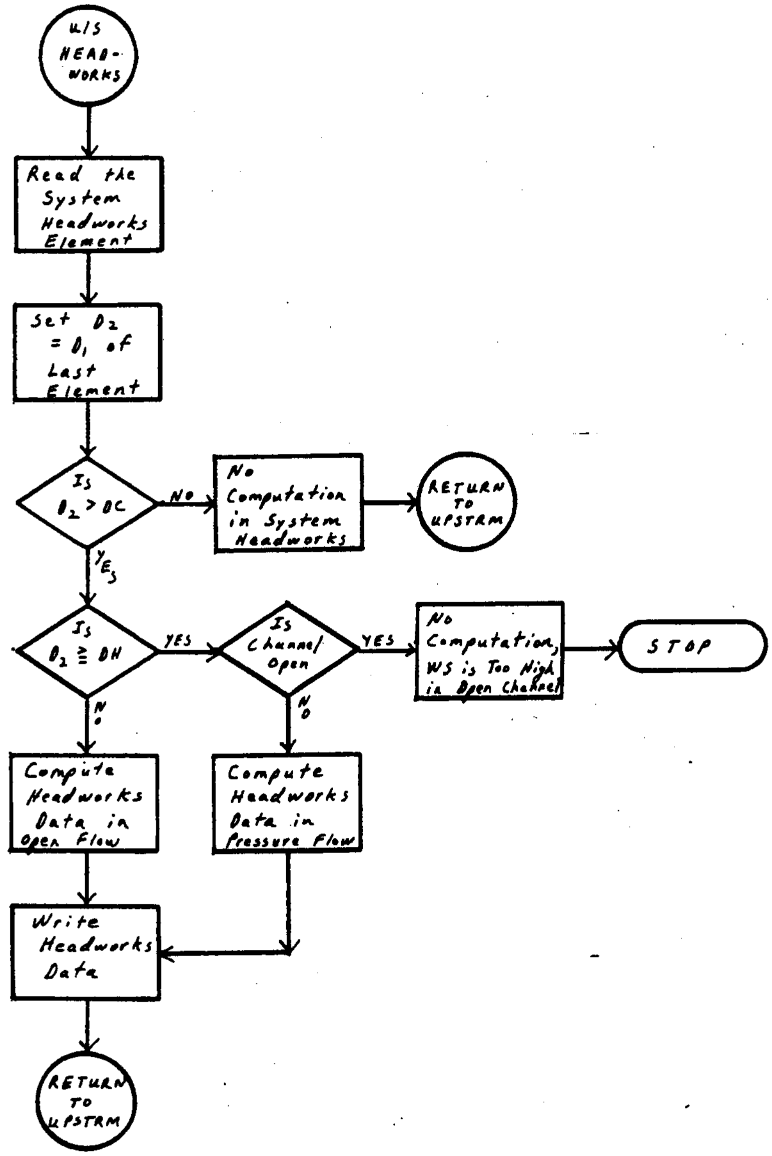
WEXTUS

Processes the Upstream Wall Exit Element, Called from UPSTRM



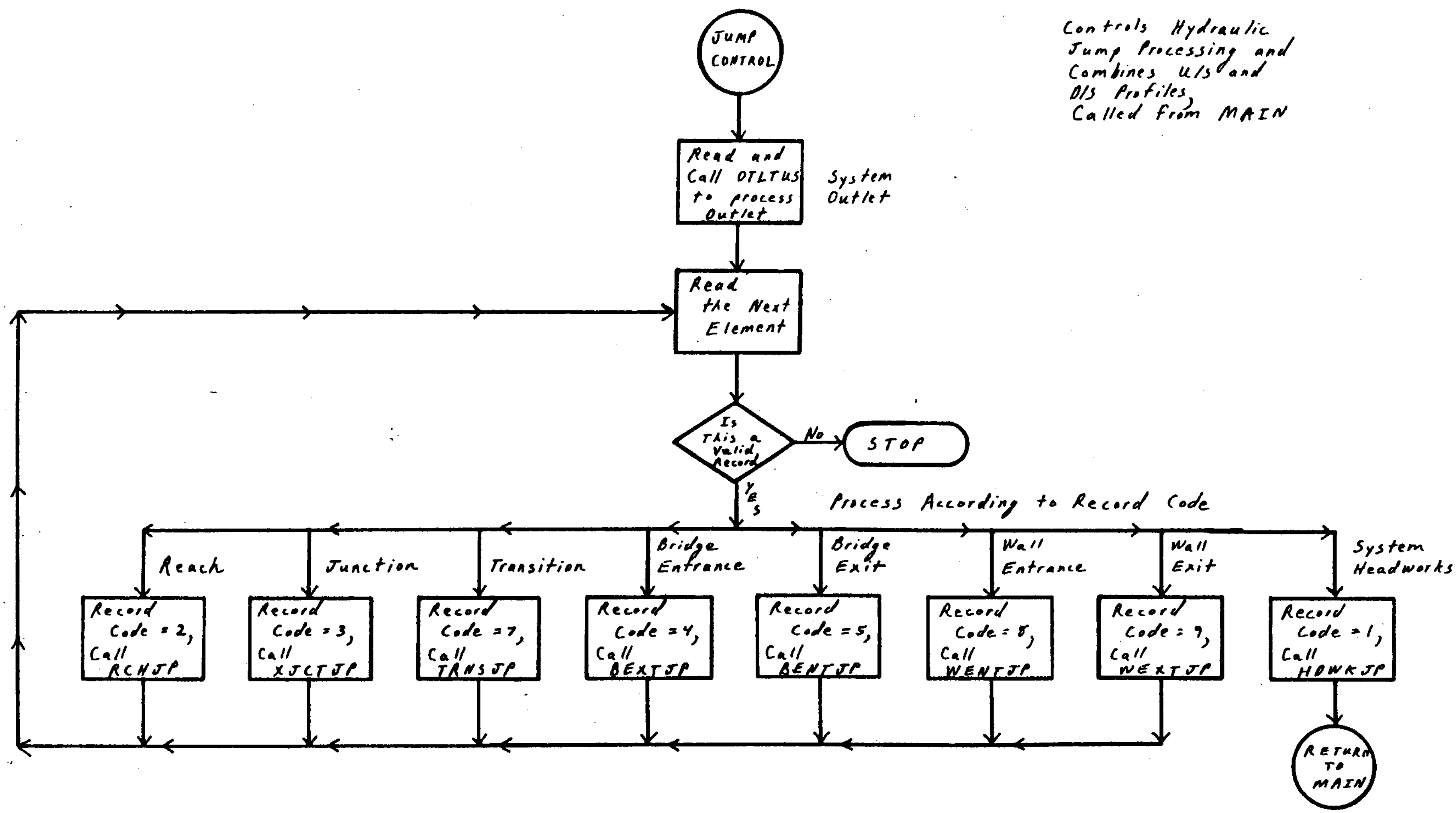
HDWKUS

Processes the Upstream System Headworks Element, Called from UPSTRM



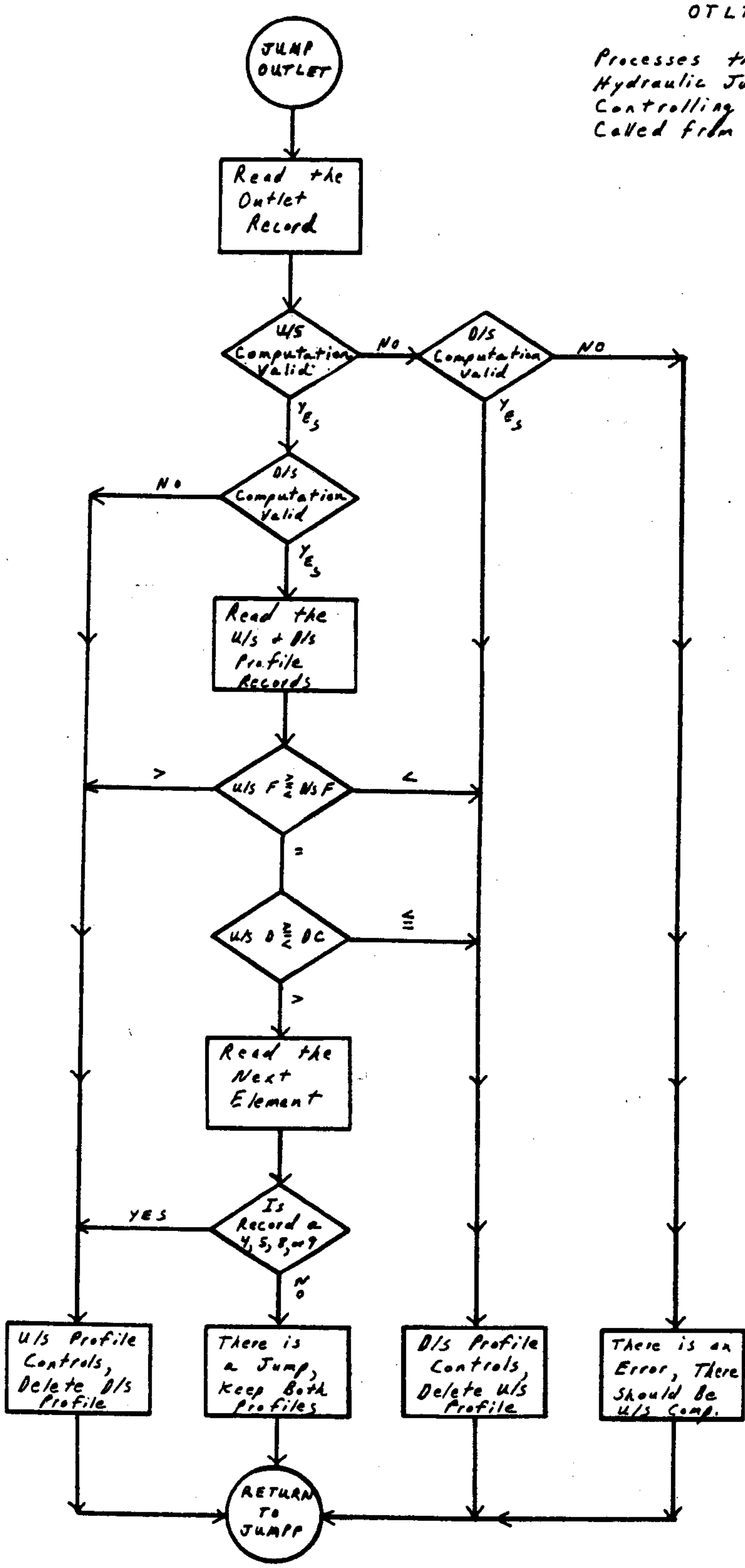
JUMPP

Controls Hydraulic Jump Processing and Combines U/S and D/S Profiles, Called from MAIN



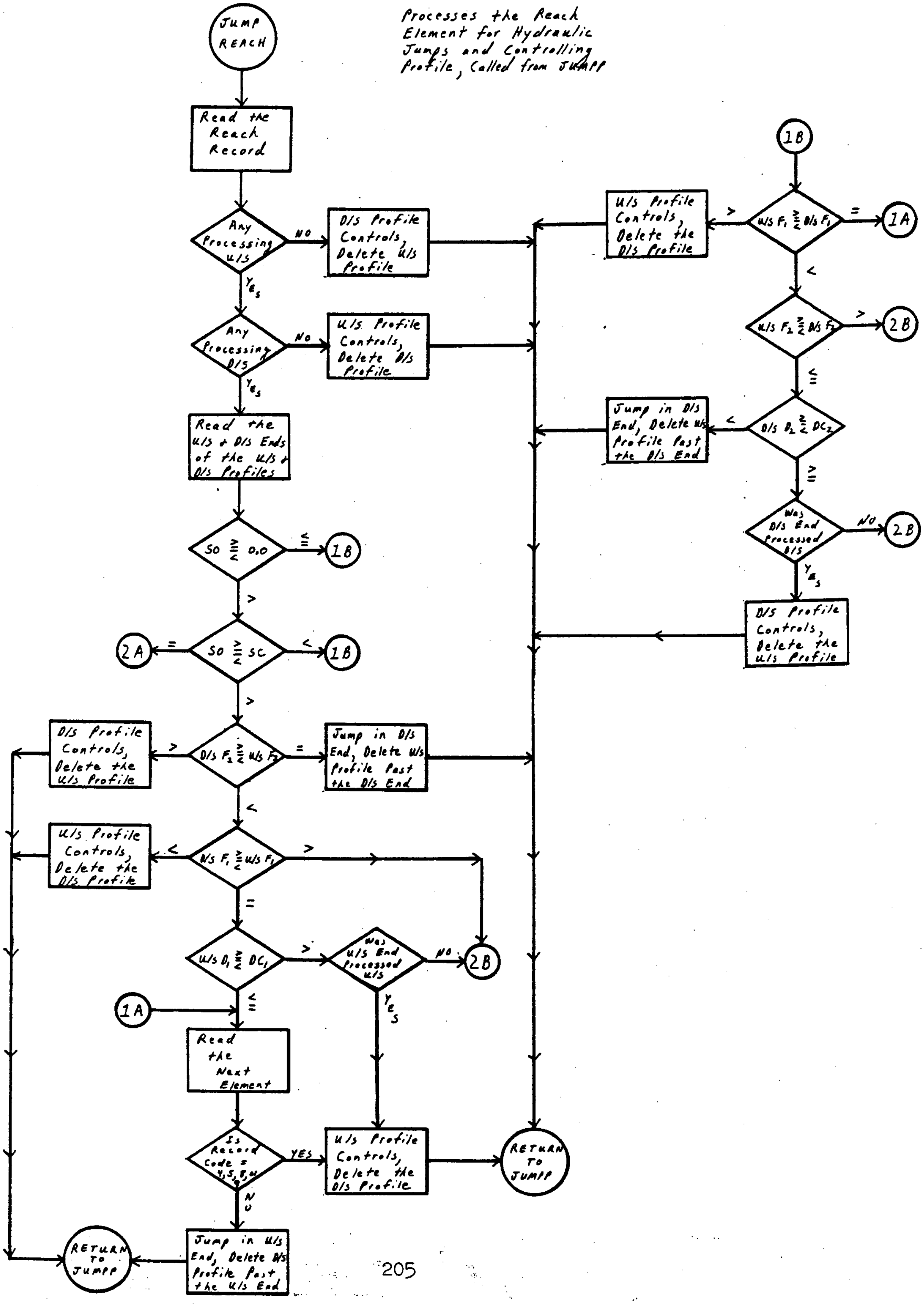
OTLTJP

Processes the Outlet for Hydraulic Jumps and Controlling Profiles, Called from JUMPP



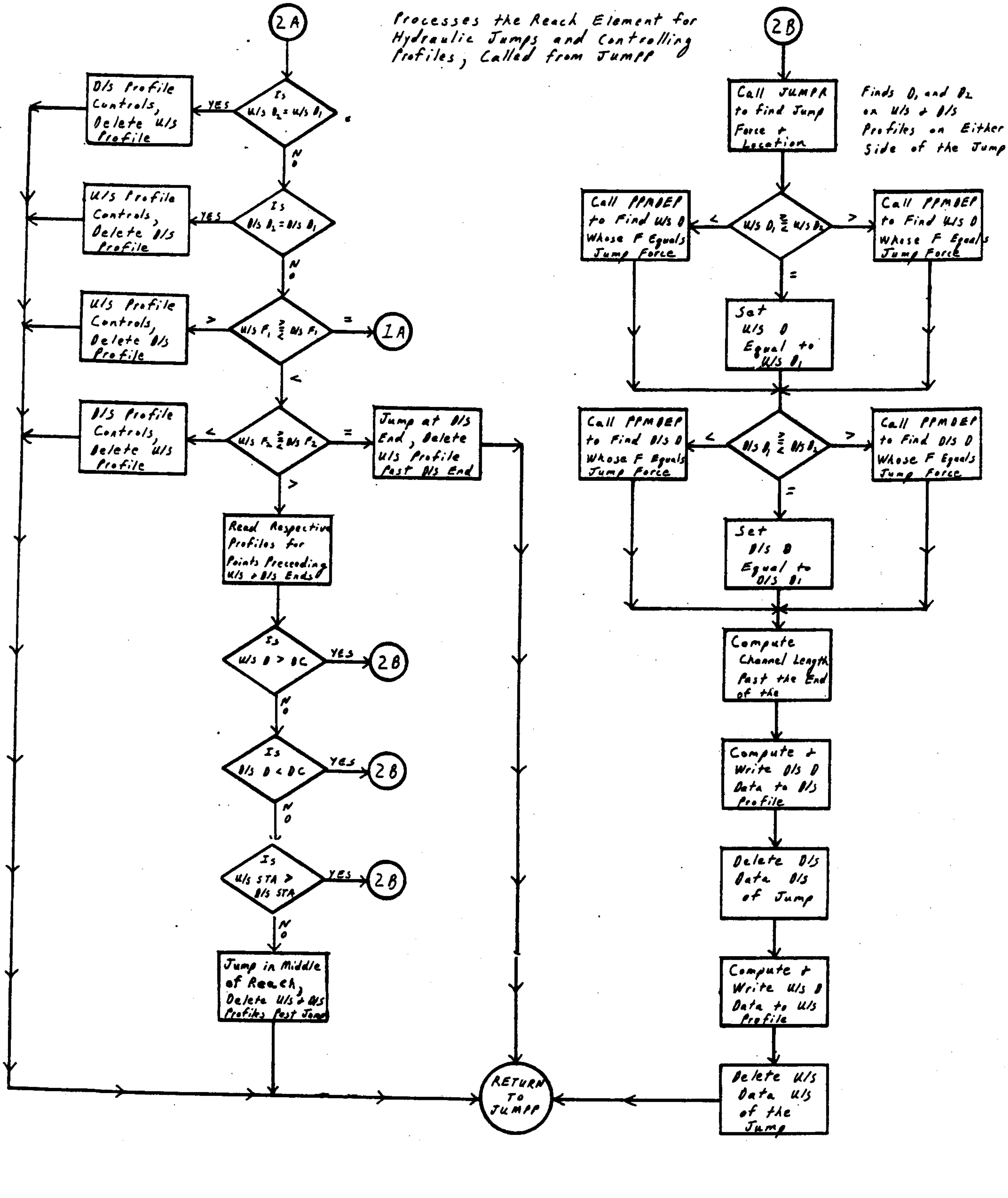
RCHJP

Processes the Reach Element for Hydraulic Jumps and Controlling Profile, Called from JUMPP



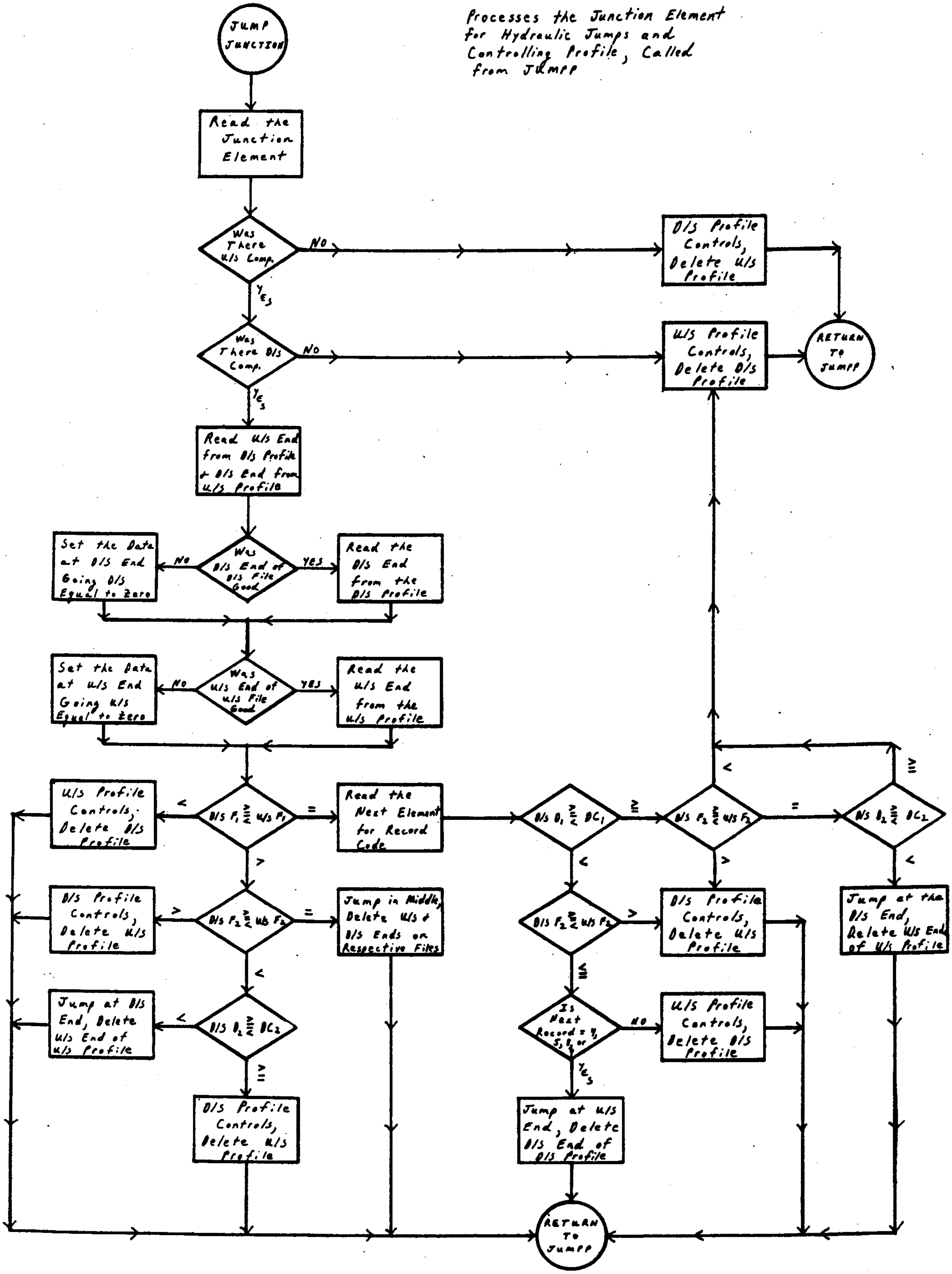
RCHJP Continued,

Processes the Reach Element for Hydraulic Jumps and Controlling Profiles, Called from JUMPP



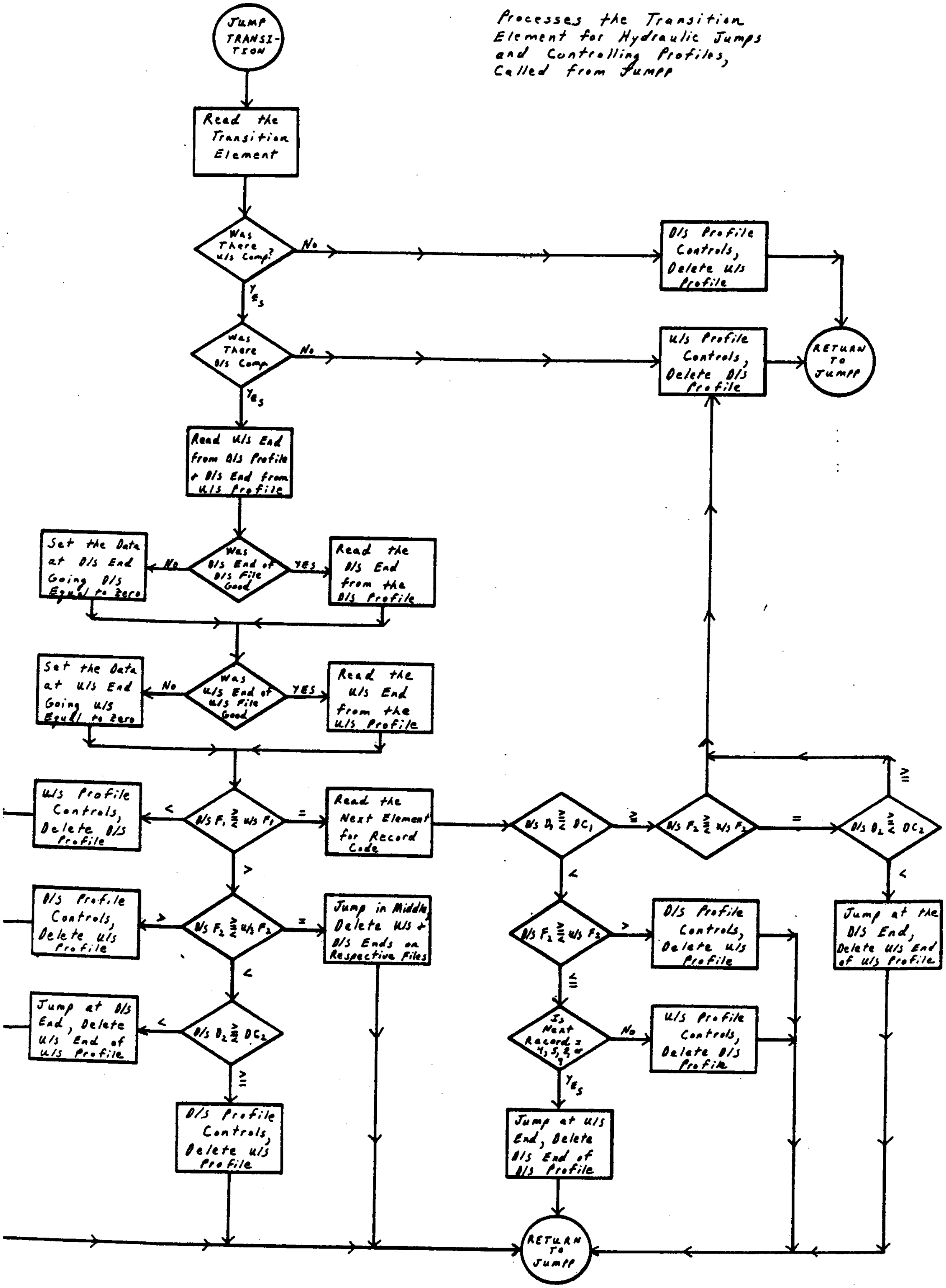
XJCTJP

Processes the Junction Element for Hydraulic Jumps and Controlling Profile, Called from JUMPP



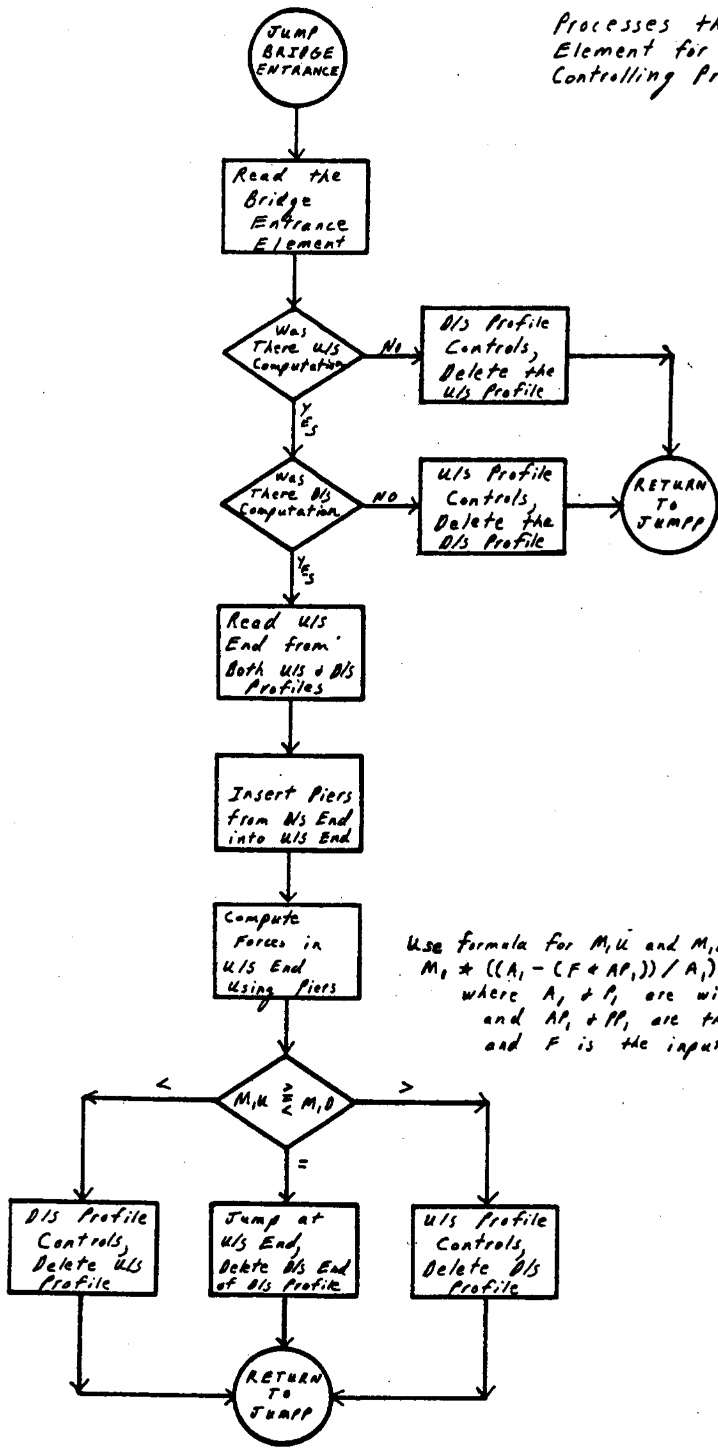
TRNSJP

Processes the Transition Element for Hydraulic Jumps and Controlling Profiles, Called from Jumpp



BENTJP

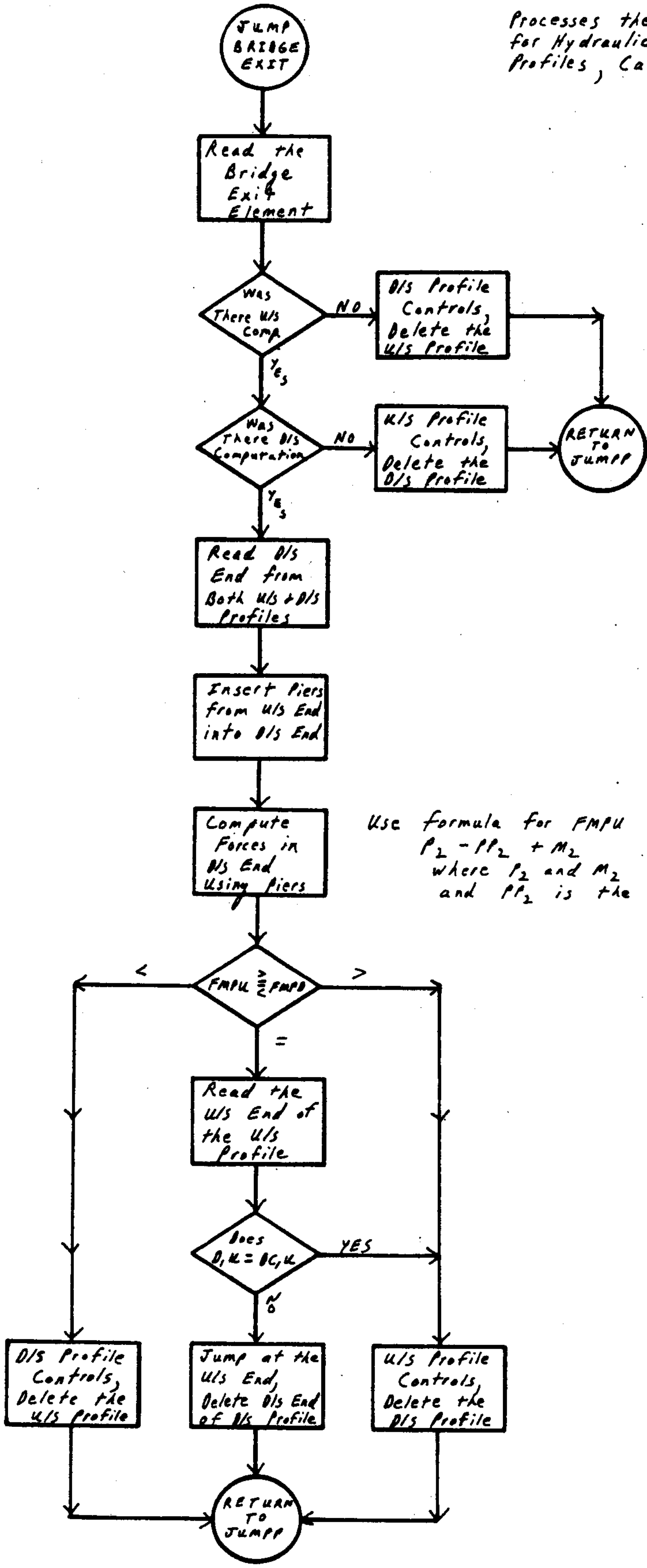
Processes the Bridge Entrance Element for Hydraulic Jumps and Controlling Profiles, Called from JUMPP



Use formula for M_1U and M_1D as follows:
$M_1 * ((A_1 - (F * AP_1)) / A_1) + (P_1 - PP_1)$
where A_1 + P_1 are without Piers
and AP_1 + PP_1 are the Pier effects
and F is the inputted Pier Factor.

BEXTJP

Processes the Bridge Exit Element for Hydraulic Jumps and Controlling Profiles, Called from JUMPP.

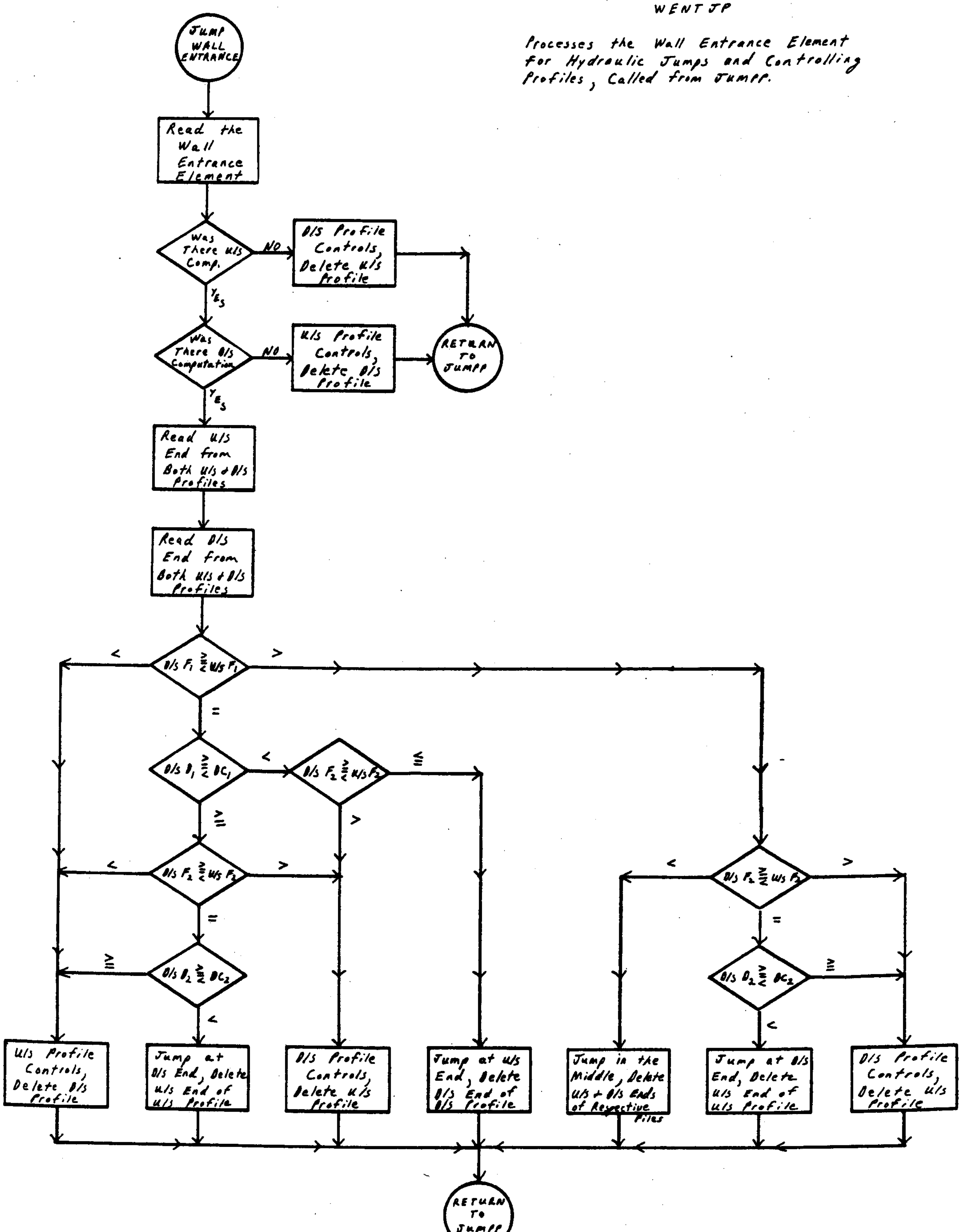


Use formula for FMPU and FMPD as follows:

$P_2 - PP_2 + M_2$

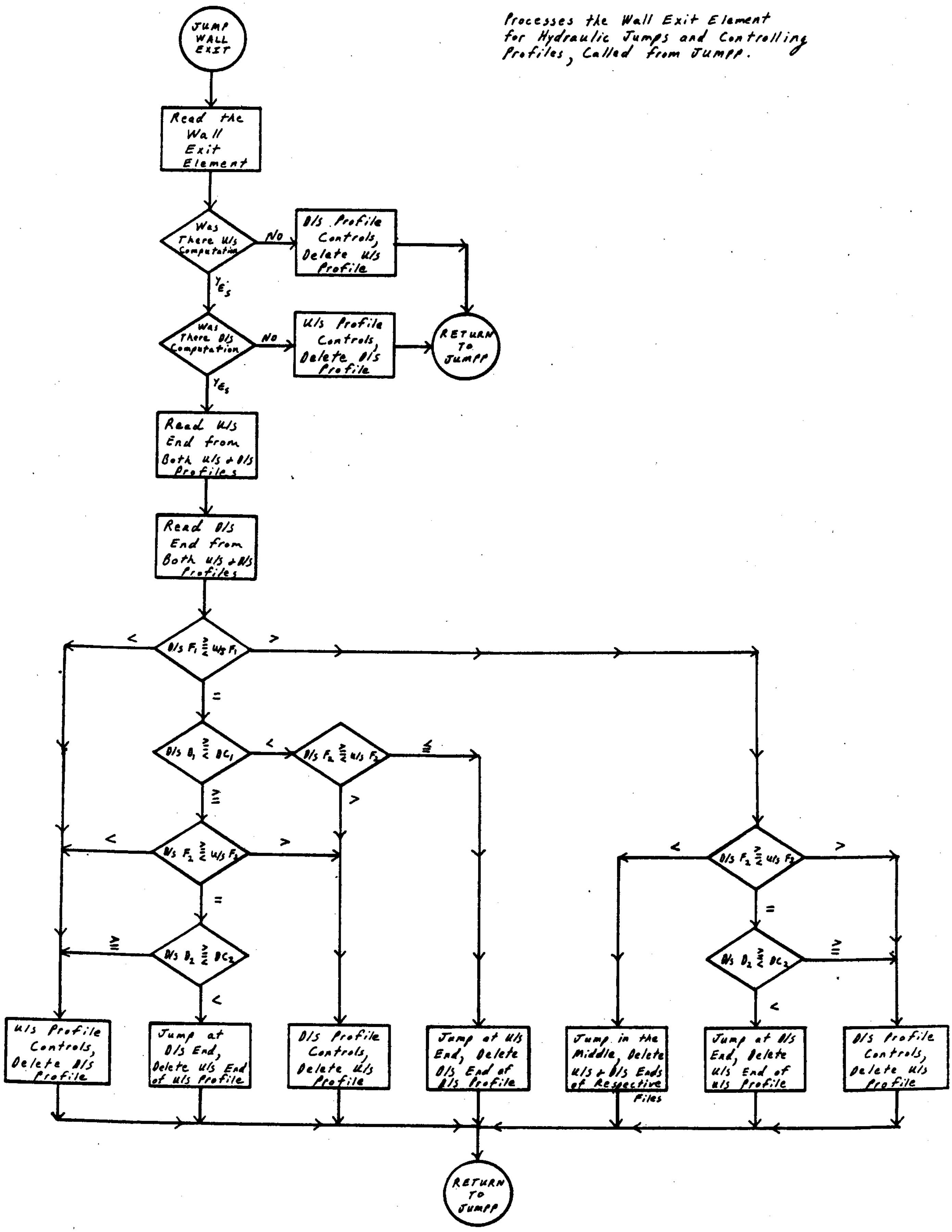
where P_2 and M_2 are without Piers and PP_2 is the Pier Pressure Effect.

Processes the Wall Entrance Element for Hydraulic Jumps and Controlling Profiles, Called from JUMPP.

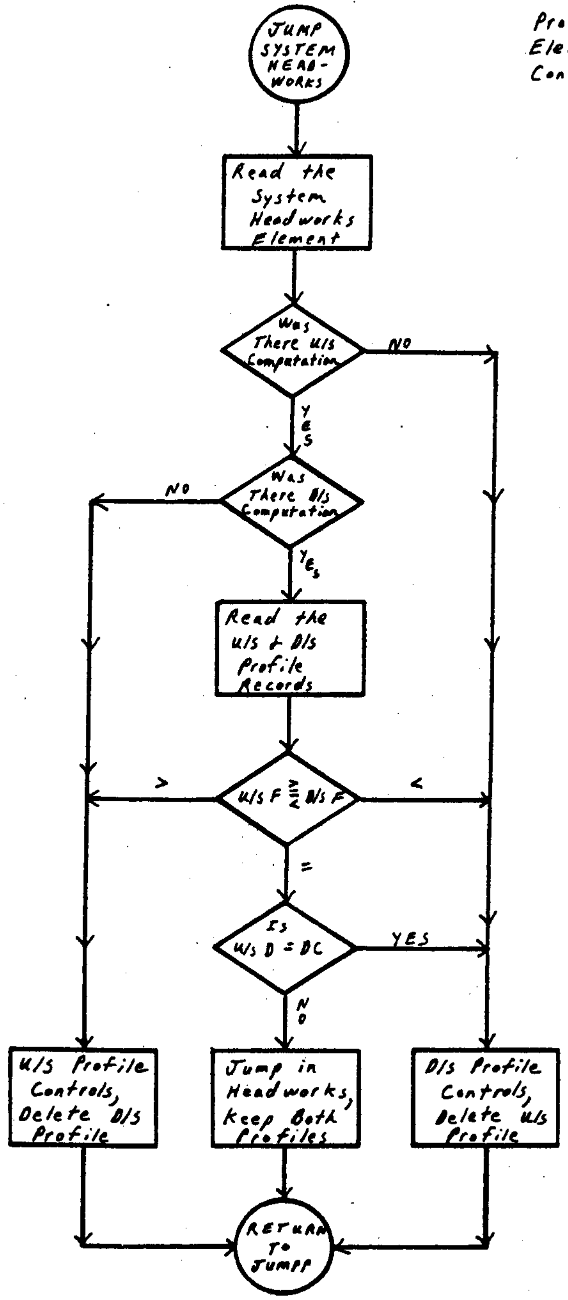


WEXTJP

Processes the Wall Exit Element for Hydraulic Jumps and Controlling Profiles, Called from JUMPP.

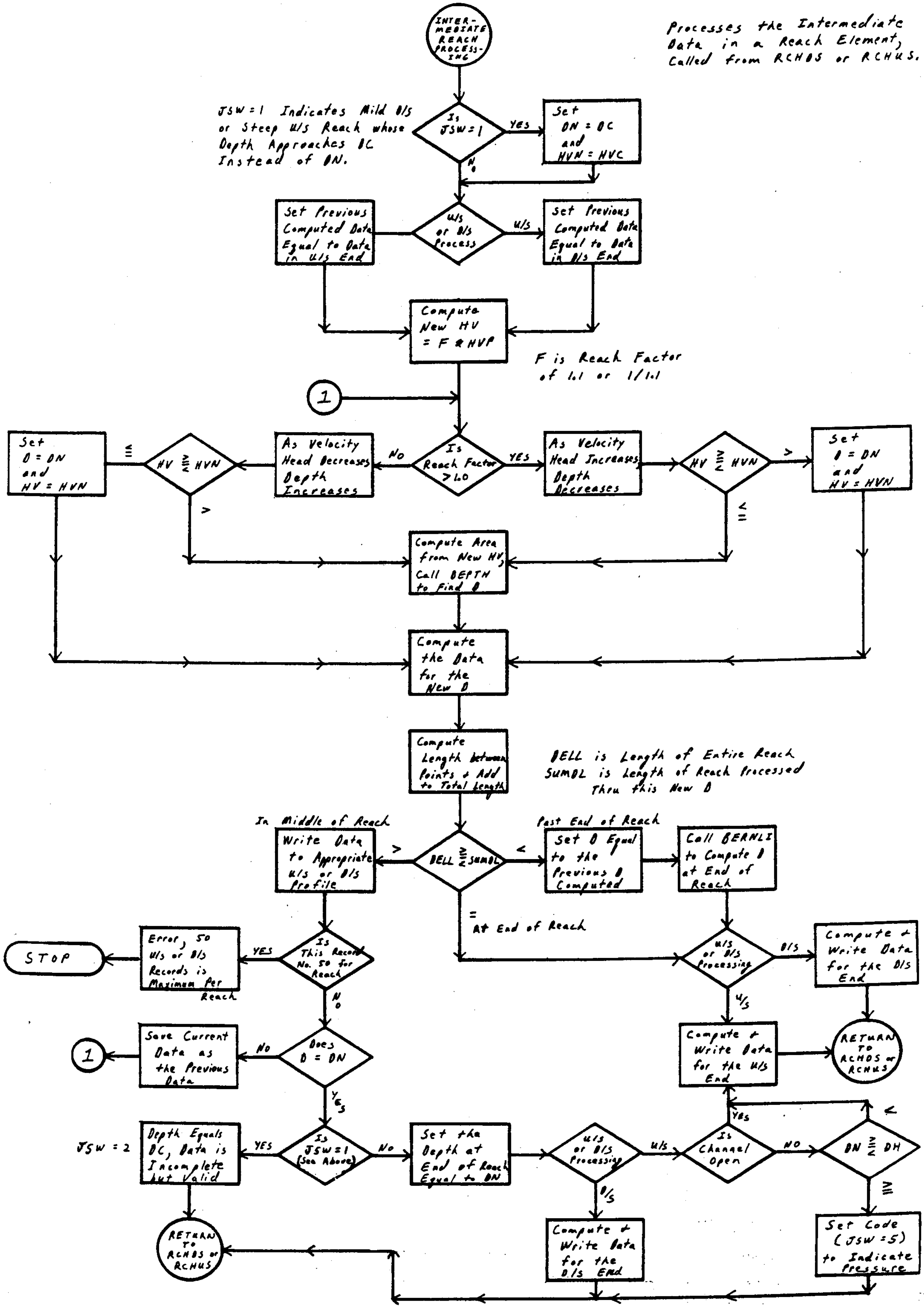


Processes the System Headworks Element for Hydraulic Jumps and Controlling Profiles, Called from JUMPP.



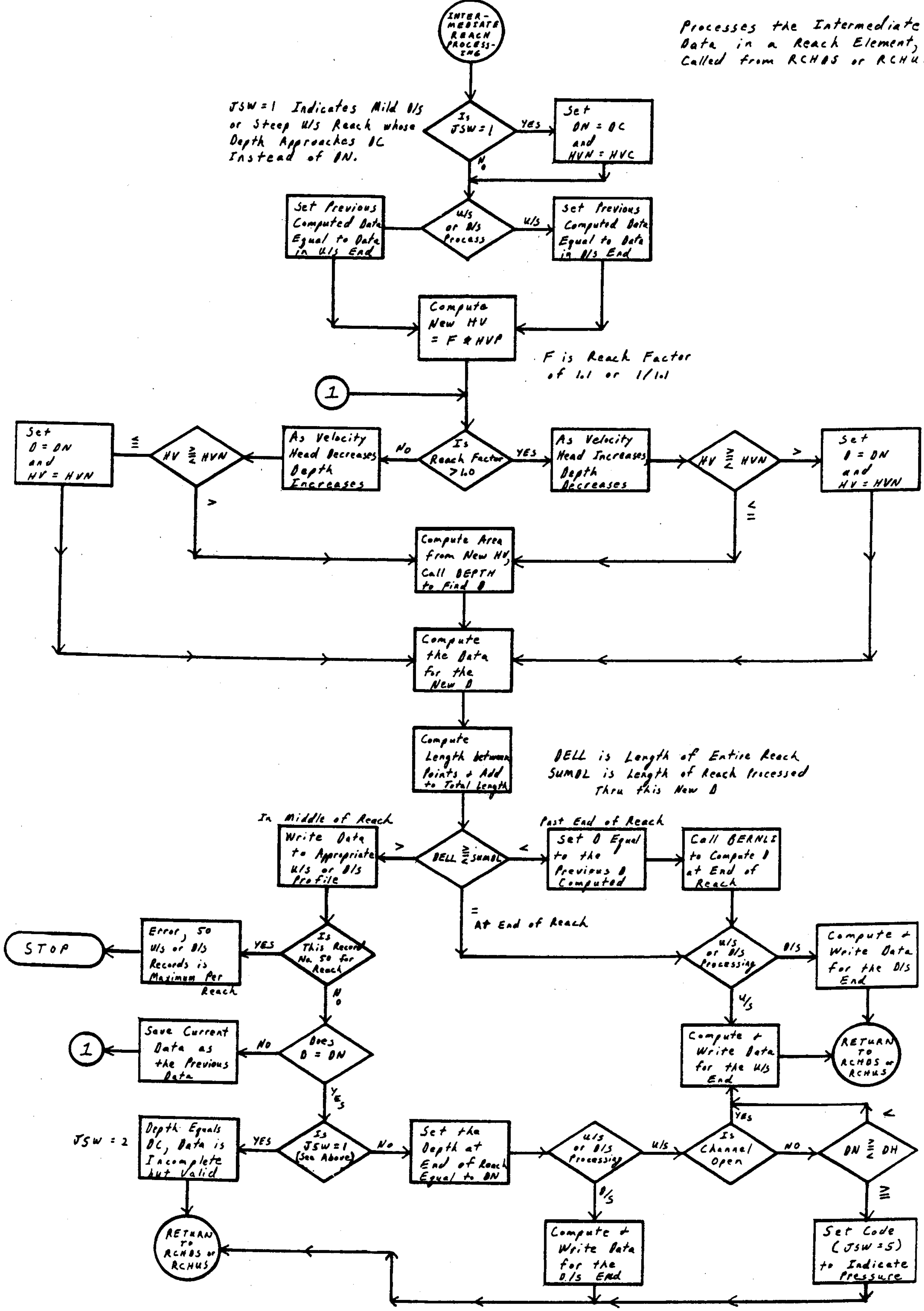
REACH

Processes the Intermediate Data in a Reach Element, Called from RCHDS or RCHUS.

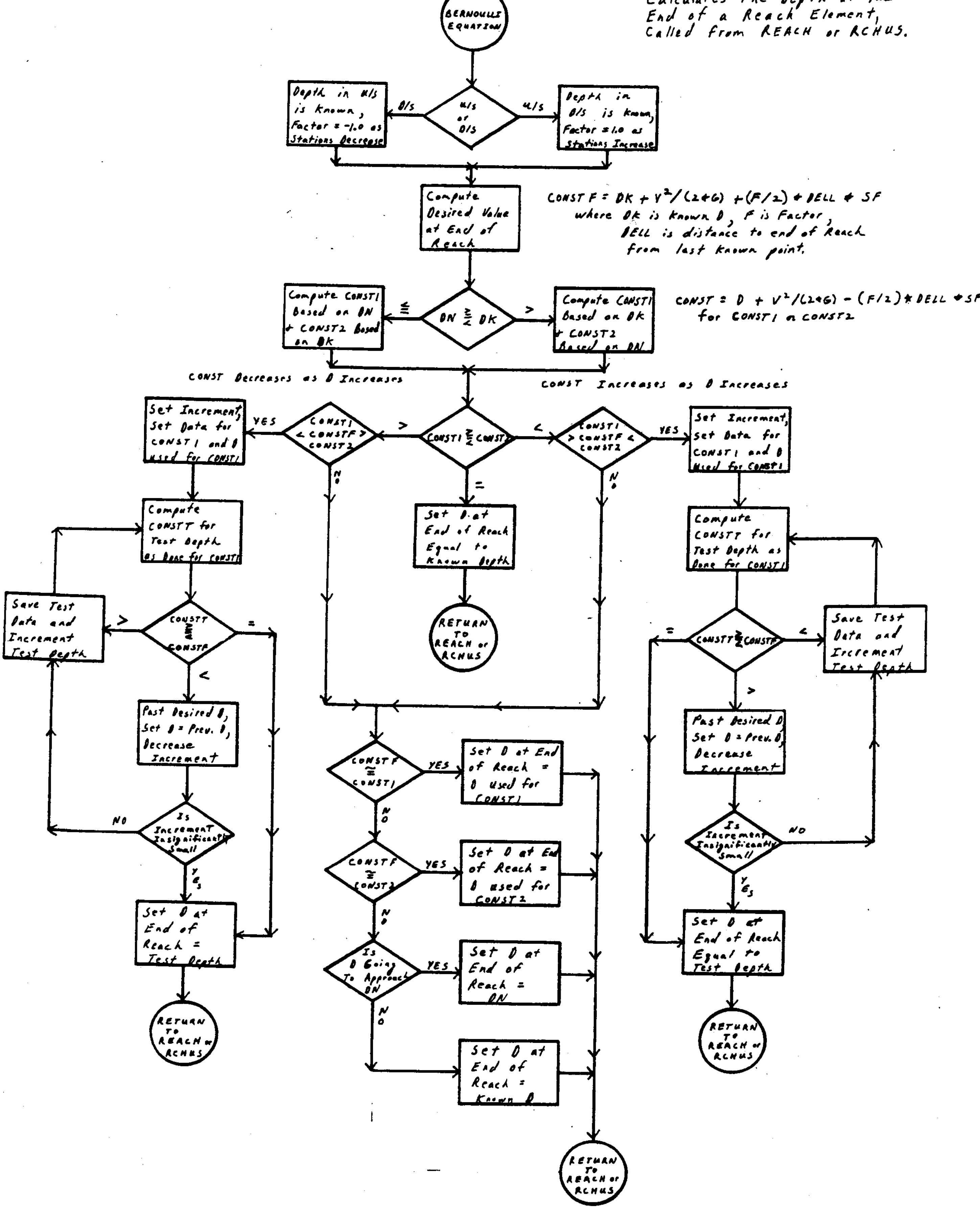


REACH

Processes the Intermediate Data in a Reach Element, Called from RCHDS or RCHUS.

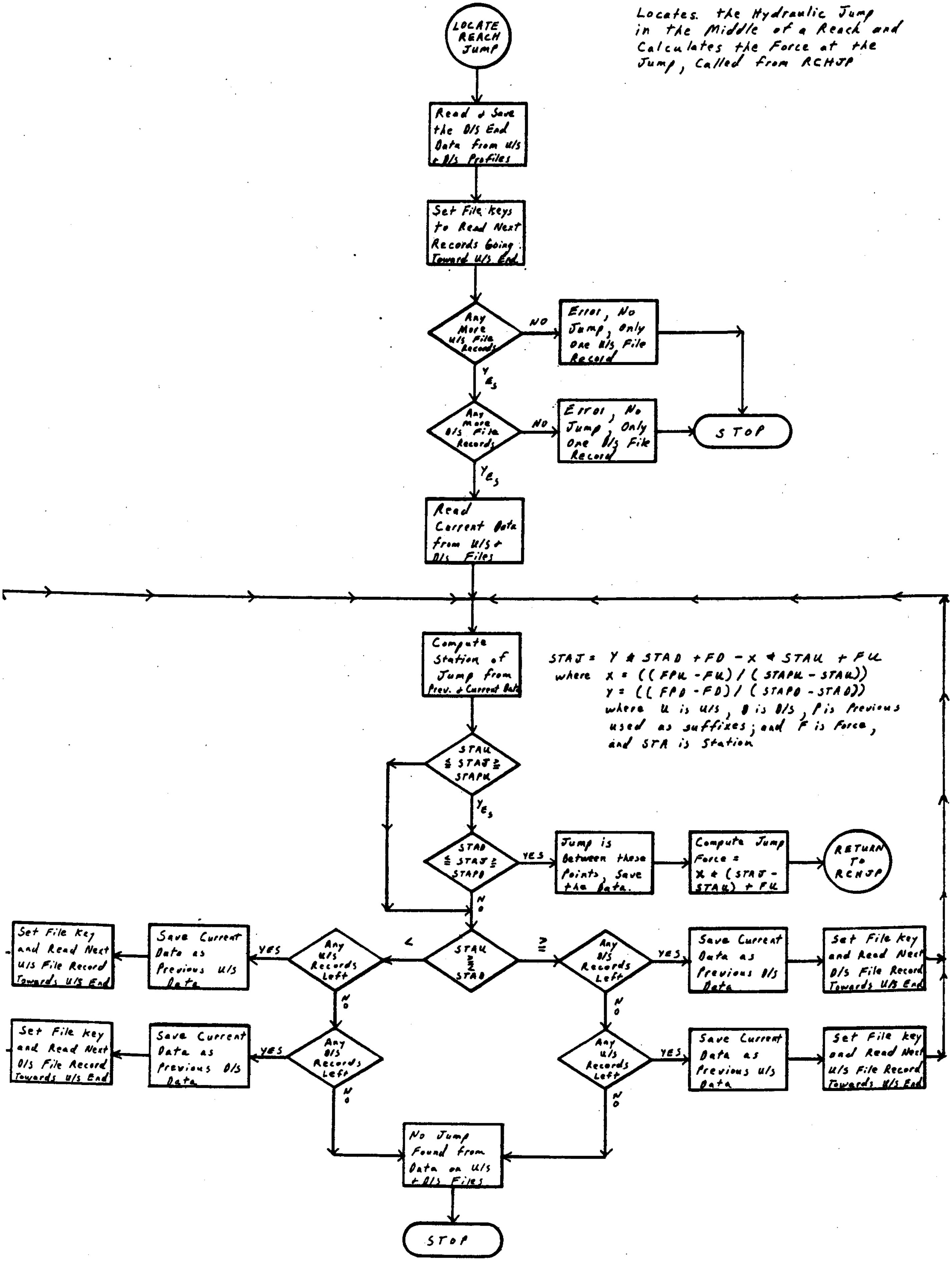


Calculates the Depth at the End of a Reach Element, Called from REACH or RCHUS.



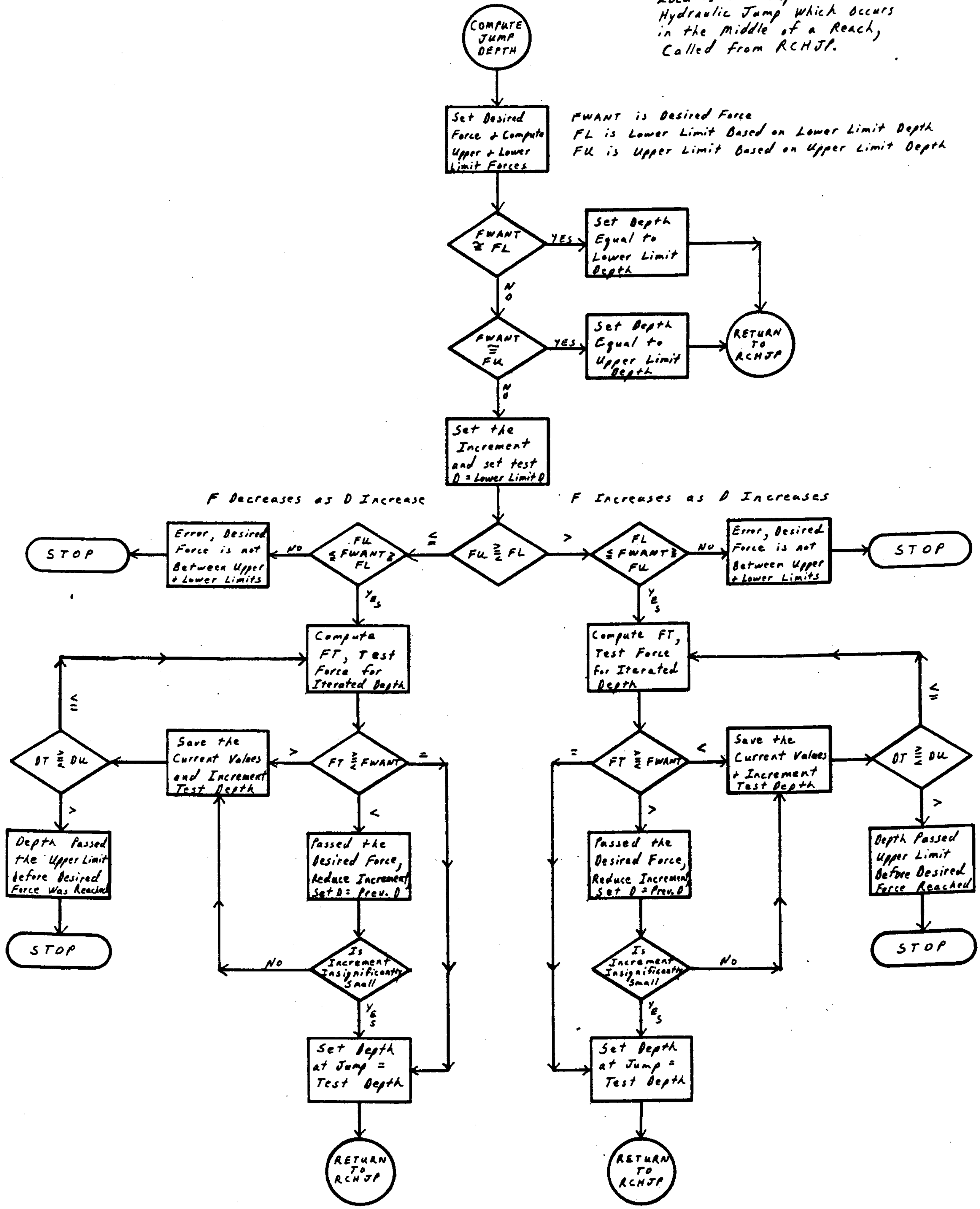
JUMPR

Locates the Hydraulic Jump in the Middle of a Reach and Calculates the Force at the Jump, Called from RCHJP



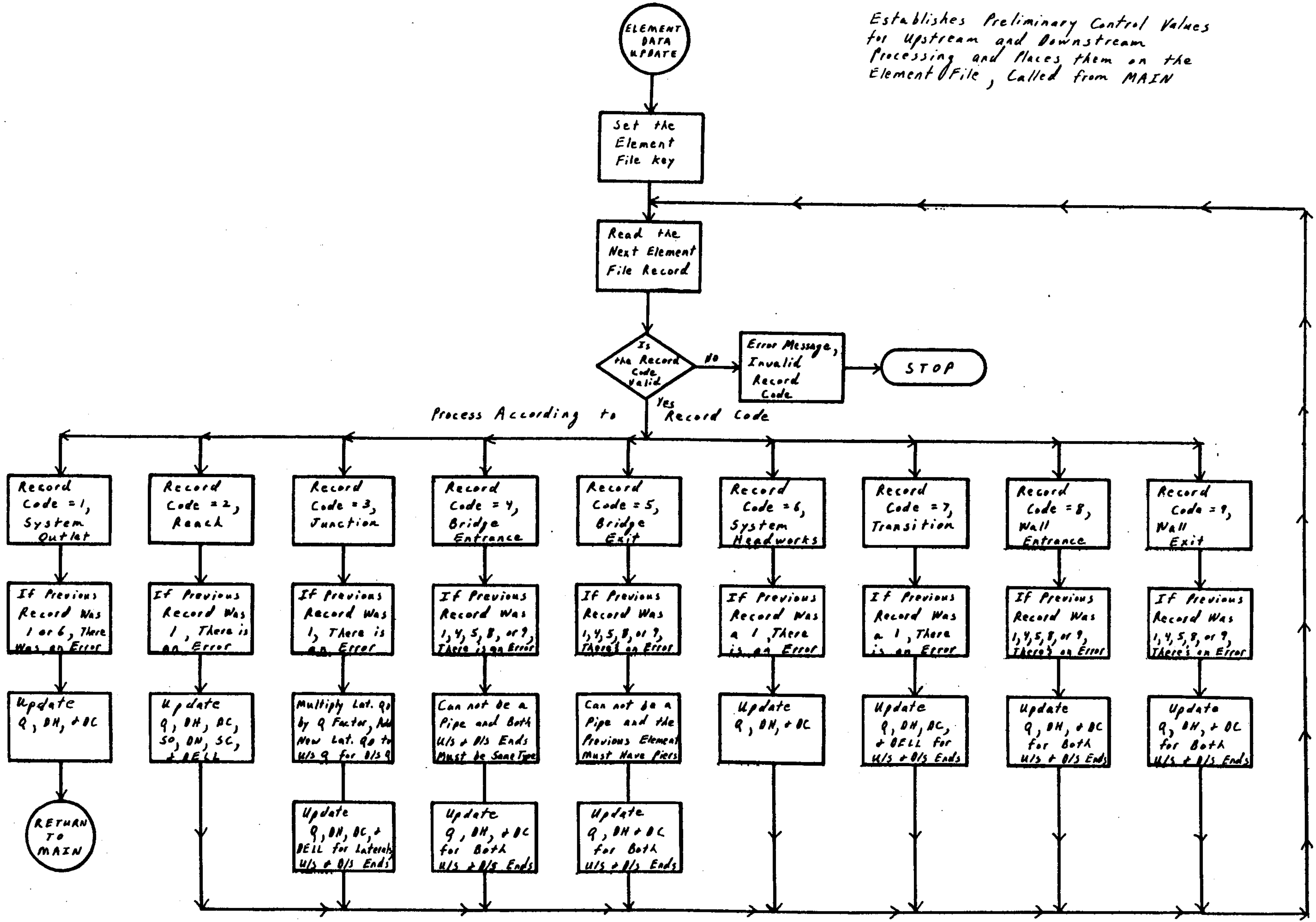
PPMDEP

Locates the Depth at a Hydraulic Jump which occurs in the middle of a Reach, Called from RCHJP.



ELMCHG

Establishes Preliminary Control Values for Upstream and Downstream Processing and Places them on the Element File, Called from MAIN



WRITEN

Prints the Combined Water Surface Profile and Creates the Plot File when Records Are Printed, Called from MAIN.