MATHEMATICAL MODELS FOR POLLUTION CONTROL IN THE USK ESTUARY .

Michael Werner Rogers , B.Sc.(Jt.Hons.)

Volume 2

Thesis submitted to the University of Aston 1977 .

VOLUME 2

CONTENTS

APPENDICES

APPENDIX A

The Sag Severity Index Parameter

THE SAG SEVERITY INDEX

CONTENTS

A.1 Introduction

There is a lack of numerical descriptor indeces for Water Quality. If an investigator attempted to convey the water quality of a body of water, there is a necessity to give complete graphical displays, numerical tables or statistical parameters which would allow a measure of the extremes to be apparent. It is clearly desirable if a single number on an ordinal or cardinal scale could be used to describe the Chemical or Biological state of a water-mass.

This paper outlines a SAG SEVERITY INDEX (= S.S.I). The statistic was introduced to assist in the numerical description of an estuary Dissolved Oxygen profile in order to use statistical methods to analyse overall effects of variations in the ecological, hydrological or pollutant inputs.

The theory applied to this distribution can, with small modifications, be also applied to any other substance whose distribution is to be summarised.

A.2 <u>The One Dimensional Sag Severity Index</u>

This index summarises the degree of severity of a D.O. curve at less than complete saturation. One dimensional graphs result from D.O. profiles at a given time for a water mass or a temporal profile at one point (fig.A1).

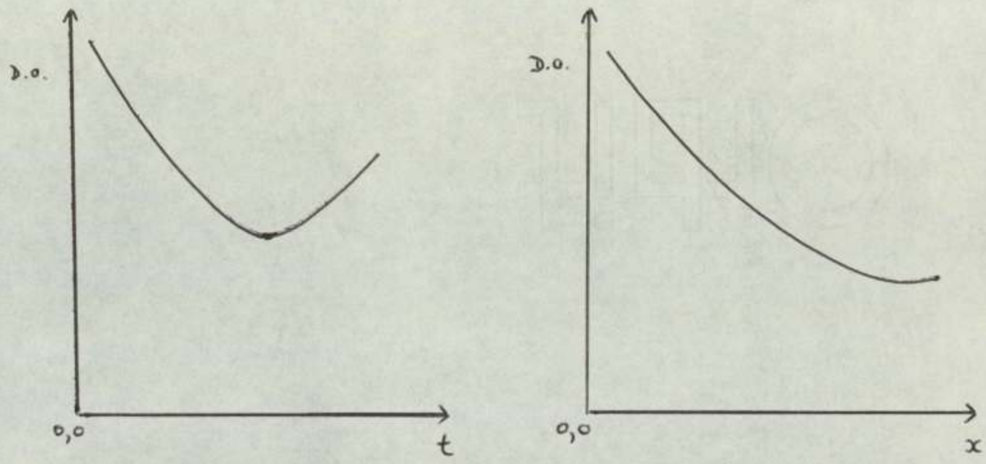


Figure A1

What the index must convey is some 'degree of absence' that is easily interpreted.

A crude measure would be the minimum D.O. during the time or distance of interest, but as fig. 2 shows, this would be most unsatisfactory.

An alternative would be E(x). SD(x) i.e., the mean value multiplied by the standard deviation. This has the drawback of being a measure of the D.O. present and is not relative to the amount that could be present. This definition, however, is influenced by extreme values and the frequency of their occurrence, so it is a more useful index.

The following definition has been found useful for time independent spatial D.O. distribution.

$$SSI = \frac{\int_{x_0}^{x_1} \left\{ DO_{SAT}(x) - DO_{PRE}(x) \right\} . dx}{\int_{x_0}^{x_1} \left\{ DO_{PRE}(x) \right\} . dx}$$

where x_0 and x_1 are the spatial limits of integration in this case. It could be written as

$$SSI = \frac{\text{D.Oxygen absent from } x_0 \text{ to } x_1}{\text{D.Oxygen present from } x_0 \text{ to } x_1}$$

In practice it would be advisable to redefine the above to its inverse. As most studies deal with situation involving oxygen deficiencies, it is more likely to involve cases where D.O. (pres) 0 and so magnify the SSI's numerical values. In the reciprocal definition severe deficiencies tend to a SSI of zero, a more manageable limit. Comparisons between similar deficiencies are then also more plausible.

The basic definition has the advantage of relating two quantities - the amount of oxygen present, and the amount that should be present in normal unpolluted environments. It is also independent of units, so it can be calculated from % distributions or absolute concentration terms.

The working definition of SSI used in the Usk Estuary Model Investigation is then

$$SSI = \int DO_{abs}.dx \;/\; \int DO_{pres}.dx \qquad (3)$$

where, in practice D.O.(abs) = D.O. (sat.) - D.O. (pres)

This was found to be a reasonably sensitive function when differences due to data variations were tested. This was due to the fact that once general parameters for the estuary are fixed, the D.O. (sat.) is relatively constant and so any increase or decrease in the D.O. (pres) is at the expense of D.O. (abs) so a small effect is magnified in relative terms of the index:

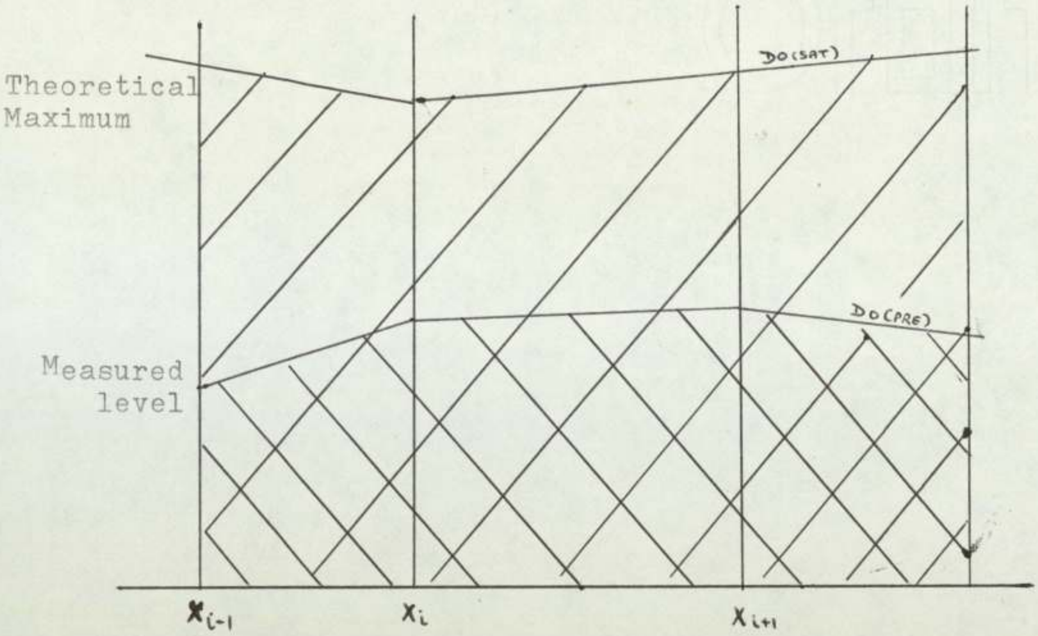
$$SSI = \frac{S - x}{x} \rightarrow \frac{d(SSI)}{dx} = \frac{-S}{x^2}$$

where x = DO (pres), dx is a small increase in DO (pres).

Some examples of the one dimensional SSI follow in the section on worked examples.

Usually data is in the form of a set of geographic locations with an appropriate D.O. measurement or a set of temporal co-ordinates again with measurements. Enough data is needed to be able to establish D.O. sat. if measurements are in absolute terms. If data is in % sat. then 100% is taken as the upper limit.

If there is extensive supersaturation within the data, the appropriate data should be edited as this phenomena masks the true oxygen absence levels.



Part of a discrete DO measurement curve

$$DO_{(pres)} = \sum \frac{1}{2}(DO_i + DO_{i+1})(x_{i+1} - X_i)$$

$$= \frac{1}{2} \sum (\overline{DO})(\Delta X_i) \qquad \text{, as for } DO_{(sat)}$$

$$\therefore \; SSI = \frac{\frac{1}{2} \quad (DOS_{i+1} + DOS_i)(X_{i+1} - X_i) - \frac{1}{2} \quad (DO_i + DO_{i+1})(X_{i+1} - X_i)}{\frac{1}{2} \quad (DO_i + DO_{i+1})(X_{i+1} - X_i)}$$

$$S.S.I. = \frac{\sum (DOS_i - DO_i)\Delta x_i + \sum (DOS_{i+} - DO_{i+})\Delta x_i}{\sum DO_i . \Delta x_i + \sum DO_{i+1} . \Delta x_i} \qquad (5)$$

Where $\Delta X_i = X_{i+1} - X_i$, often a constant distance or time, so assume $\Delta X_i = \Delta X_j, \forall_{i,j} \leqslant n$. (5) can be rewritten as

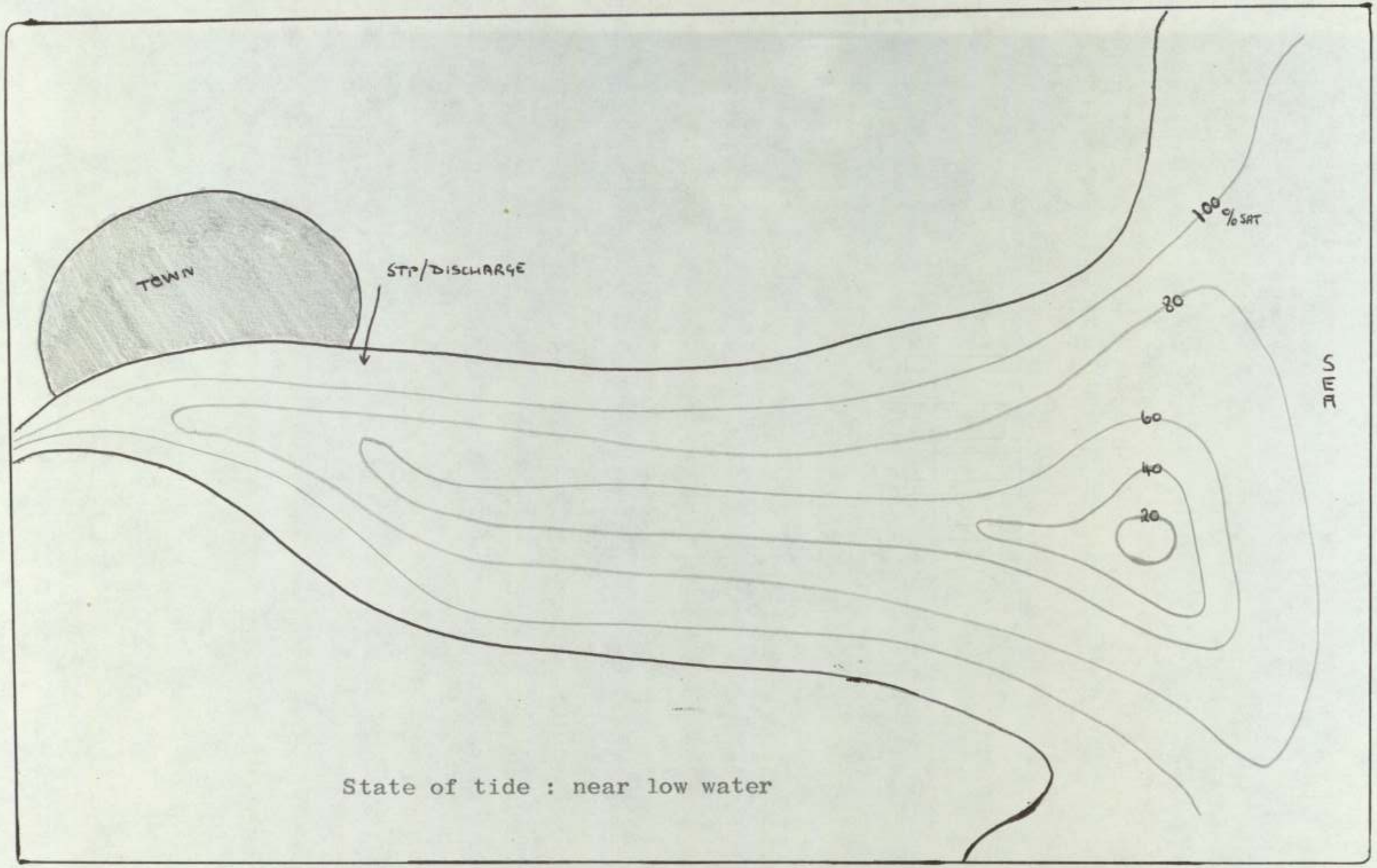
$$SSI = (\sum w_i \; DOS_i - \sum w_i \; DO_i)/(\sum w_i \; DO_i) \qquad (6)$$

where $w_i = 1$ if $i = 1$ or n

$w_i = 2$ if $i \neq 1, \neq n$

This formula only requires two summations and is therefore of practical importance.

A MAP OF A TYPI AL ESTUARY DO PROFILE



State of tide : near low water

Note how the lowest levels have not quite left the confines of the system and will be re-admitted on the next tide.A set of one dimensional SSI's can be obtained by defining sections across the system, or a 2-D for a state of tide.

A.3 Multi-Dimensional Sag Severity Index

The previous discussion dealt with measurements in the temporal or spatial sense. Any realistic situation would require the 4 dimensions of space and time to be considered.

Let the point i be measurable at location x_i, y_i from zero, at a depth of Z_i and at time t_i.

Then the multi-dimensional SSI is defined as

$$SSI = \frac{\iiiint DO_{abs} \quad dx.dy.dz.dt}{\iiiint DO_{pres} \quad dx.dy.dz.dt}$$

where the spatial integration is over points within the estuary or a subset of it, and the time component over the period of interest.

To use numerical integration methods, a four dimensional mesh can be laid over the area and data collected at the intersections. It is improbable that the data is complete so sparse matrices are likely to be a problem. Continuous remote sensing should at least ensure a time-span continuity.

A reasonable alternative definition in the likelihood of very sparse data is

$$SSI = \frac{1}{N} \sum_{i=1}^{N} \{(DOS_i - DO_i)/(DO_i\} \qquad (8)$$

DOS = set of D.O. (Saturated) readings

DO = set of D.O. (Present) readings

This definition is more exactly

$$\overline{SSI} = \frac{1}{VT} \iint_{VT} SSI(x,y,z,t)\ dx.dt$$

In this instance it is not sufficient just to quote the mean without giving some indication of its accuracy. The Standard Error of the Mean is such a statistic:

$$SEM = \frac{\text{Standard Deviation}}{N}$$

where N = number of samples in the summed series
When just a contour map of oxygen levels is available (see opposite)

A series of one dimensional SSI can be obtained by putting a rectangular regularly spaced grid over the estuary. The SSI's are then calculated at

$$x = X_o + n\Delta x \qquad \text{at} \qquad x \leqslant X_e$$
$$y = Y_o + m\Delta y \qquad\qquad\qquad y \leqslant Y_e$$

where X_o is the upstream end of the estuary, Y_o is the lowest part of the estuary (in the catographic sense), X_e and Y_e are the opposite boundaries.

Alternatively, if the data is in all 4 dimensions and the grid method is impractical, a Monte Carlo integration procedure could be used to estimate the overall SSI to a sufficient degree of accuracy. This would require a multiple interpolation routine.

When all these methods are compared, it can be seen that the definition (8) is computationally concise and sufficient for most instances. Exceptions could occur where sampling is clustered in a particular section of the whole area of interest which, when using (8), would skew the overall SSI.

Once the method of calculation of SSI is standardised and consistent, it can be used as an output function for a model. Suppose a change occurs in input parameter d_i. A change is then apparent in the final SSI. Mathematically,

$$\Delta d_i \nrightarrow \Delta SSI$$

If this relationship can be established through a regression equation then the experimenter has a direct estimate of the effect of a change in input on the output oxygen distribution. As the model's adequacy in representing the real physical world should already have been established, the equation can be used directly as a desk top management tool. One of these equations could be established for each major input parameter and as the principle of superposition is applicable, to an extent, in this system, management has an easy cumulative oxygen depletion function available.

A.4 A General Numerical Index

The problem of describing a situation within the entire estuary in a concise form is not only limited to Dissolved Oxygen profiles. Any pollutant is important in that an excess will eventually cause an oxygen depletion but it is possible to limit the number of pollutant parameters to a small number to establish a good picture of the overall Water Quality.

It should be possible to confine water quality as a whole to a single index or number. This section defines a series of mathematical options to establish a water quality index or a numerical summary index.

Each area has its specific problems with regard to particular pollutant so the theory will be non specific.

Define a set of pollutants substances P. The set P has n components (could include D.O.).

e.g. P = (carbonaceous load, nitrogenous load, ammonia, nitrate)

The set S_u contains the upper limits, in terms of concentration, that the estuary can hold without irreversible ecological damage and the set S_e contains lower limits for each pollutant.

The Set F is a series of sensitivity function which gives some measure of the effect of the consent by applicable pollutant concentration on the estuary. A pollutant with a high 'depletion' effect will have a steep sensitivity function. The values in the set S_u and S_e can be translated into 'sensitised' values to give sets S_u^1 and S_e^1

$$S_n^1 = (F).(S_n) \qquad S_e^1 = (F).(S_e)$$

Then the set of actual values recorded, R, is 'sensitised' to give the set R^1. The set of ranges $S_i = S_u - S_e$ is divided into 8 equal intervals. If a sensitised reading falls within the range then it is assigned an integer value 1-8. If the reading is out of limits and unacceptable then a value 9 is assigned. If no reading is taken, value 0 is assigned.

The result is a series of integers 0 to 9 which will allow a numeric water quality figure to be given. Once the order of presentation is agreed then indices become directly comparable.

For example,

P = (carbonaceous loads, ammonia, nitrate, D.O.)

S_u = (5.0, 3.0, 20.0, 10.0) p.p.m.

S_e = (0.0, 0.0, 0.0, 5.0) p.p.m.

F = $(x, x^2, x/5, 5x)$

S_n = (5.0, 9.0, 4.0, 50.0)

S'_e = (0., 0., 0., 25.0)

R = (4.0, 2.0, 17.0, 8.0)

R' = (4.0, 4.0, 3.4, 40.0)

S = (5., 9., 4., 25.)

This gives the following limits for classification

Grade	Carbonaceous Load	Ammonia	Nitrate	D.O.*
1	0.0-0.625	0-1.125	0-0.5	25-28.125
2	0.62-1.25	1.125-2.25	0.5-1.0	28.125-31.25
3	1.25-1.875	2.25-3.375	1.0-1.5	31.25-34.375
4	1.875-2.5	3.375-4.5	1.5-2.0	34.375-37.5
5	2.5-3.125	4.5-5.625	2.0-2.5	37.5-40.625
6	3.125-3.75	5.625-6.75	2.5-3.0	40.625-43.75
7	3.75-4.875	6.75-7.875	3.0-3.5	43.75-46.875
8	4.875-5.0	7.875-9.0	3.5-4.0	46.875-50.0
9	5.0	9.0	4.0	50.0
0	Not measured	Not measured	Not measured	Not measured

*The complement of the index is taken here as it is an inverted distribution in the more present the better the water quality.

Agreeing on a relative importance order of D.O.-Ammonia-Carbonaceous load-Nitrate gives for the example, a water quality index of 4477. Then treating this as a number in the set of integers, it can be stated that any index greater than this represents a water body of lower water quality.

Also, the magnitude of the difference reflects some measure of the degree of difference in water quality of the respective water bodies.

The sensitivity function could be by-passed if a series of numerical weights could be agreed for various pollutants..

In the example, if the D.O. deficiency had a weighting of 10 (it being an overall indicator of pollution), ammonia 7 say, carbonaceous loads 2 and nitrate 0.1. An index evaluated as before but without transformation (say 4317) would be converted to $4 \times 10 + 3 \times 7 + 1 \times 2 + 7 \times 0.1 = 63.7$. Again, the difference between two bodies would reflect some measure of the difference in quality.

Some parameters, however, do not have a linear cause-effect relationship and this causes the above simple theory to be unacceptable.

A second theory based on the above now needs to be developed:

The adjectives used to define water quality are graded and given numerical values in a common range e.g.,

Intolerable	.01
Unacceptable	.1
Poor	1
Unsatisfactory	3
Satisfactory	5
Good	8
Excellent	10

The constituents of pollution are defined and then with the aid of literature and general agreement, concentrations corresponding to the above are defined, (or temperatures etc.) and a function is fitted to the few points established from the above definitions.

For example, for temperatures collected for fish mortality an inverted parabola was the function.

$$f\,(T) = \frac{2.I.T. - T^2}{I^2} = \frac{I^2 - (T-1)^2}{I^2}$$

where I is the ideal condition, which is around 20°C pH values have a similar curve based on an ideal value of 7.0 (i.e. neutral). This approach is covered by Walski and Parker (ASCE June 1974).

The overall index is then calculated from the individual functions and weights. A geometric mean would be a responsive function in terms of extremes.

$$\text{Index} = \left\{ \prod_{i=1}^{N} \{f_i(x_i)\}^{w_i} \right\}^{1/\sum_{i=1}^{N} w_i}$$

where W_i = weight attached to i parameter for a total of N parameters.

There is some loss of information in this index but it does respond well to extreme values for isolated components, which an arithmetic mean would not do.

APPENDIX B

Two Programs for Hydrodynamic Analysis of a General Estuarine System

APPENDIX B - CONTENTS

B.1. Introduction

This is the first phase of the time dependent model with mixed dimensions. Using only basic constant geographic data and a tidal forcing function, water movements are estimated and a temporary file created for later use with a pollutant transport program. There are two programs in this suite, one for a purely one dimensional system and some for a one and two dimensional systems. Many of the subroutines are common to both, but some are specific for the dimension in question.

The one dimensional routines are identified by the file identifier commencing with 'F1', those for the mixed dimensional model by 'F2'.

Both routines have restart facilities and in the most general form the system is:

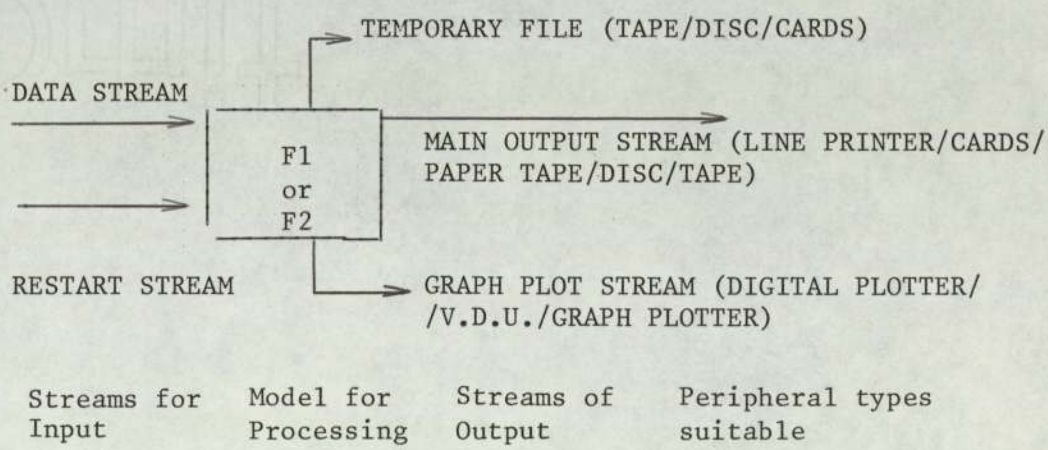


Fig.B.1.A.System Configuration

A complete documentation of each of the twenty two modules specific to the models. In addition a graph plotting package, a data handling package and an inter polation routine have to be supplied. The 'Program Description'/'DATAD' etc, also has to be defined by the user, but it is highly individual to the system configuration being used and, in the current system, the routine DATAD5 is used, having been specially written for the model.

Each module will have annotated to which model it belongs. No specific indication indicates that the module is common to both F1 and F2 models. For compatability, some modules have identical names but are different versions. If all modules are 'lumped' these must be kept apart, otherwise compiler failure will occur in the compose/ consolidate phase due to ambiguous entry points. There is no requirement for same-name modules to occur in any of the versions.

B.2. Model Composition

Model: F1 All modules either not specially allocated or specifically 'for F1', i.e. 11 modules. See table B.2.A.

Model: F2 All modules not itemised as 'for F1', i.e. 18 modules, see Table B.2.A.

Listings may be found under F1 and F2 in the Listings Appendix.

Table B.2.A.

Modules for each model

F1 - Model		F2 - Model	
Block Data	(for F1)	Block Data	(for F2)
CALEF		BAYOUT	(for F2)
CALHBC		BOTANY	(for F2)
CALNOD		CALEF	
DATA		CALHBC	
HYDONE		CALNOD	
HYDROD	(for F1)	DATA	
NODEMK		DIGITS	(for F2)
PRINT1		FIND	(for F2)
QT	(for F1)	FRACT	(for F2)
REST	(for F1)	HYDONE	
UPTIM		HYDROD	(for F2)
INTP*		NODEMK	
DATAD5*		PLOT	(for F2)
		PRINT1	
		PRINT2	(for F2)
		REST	(for F2)
		UPTIM	
		COPYDT*	
		IPLAUS*	
		ADDATE*	
		DATAD5*	

Modules marked * are from other programs developed and documented elsewhere.

Subroutine BAYOUT

(for F2)

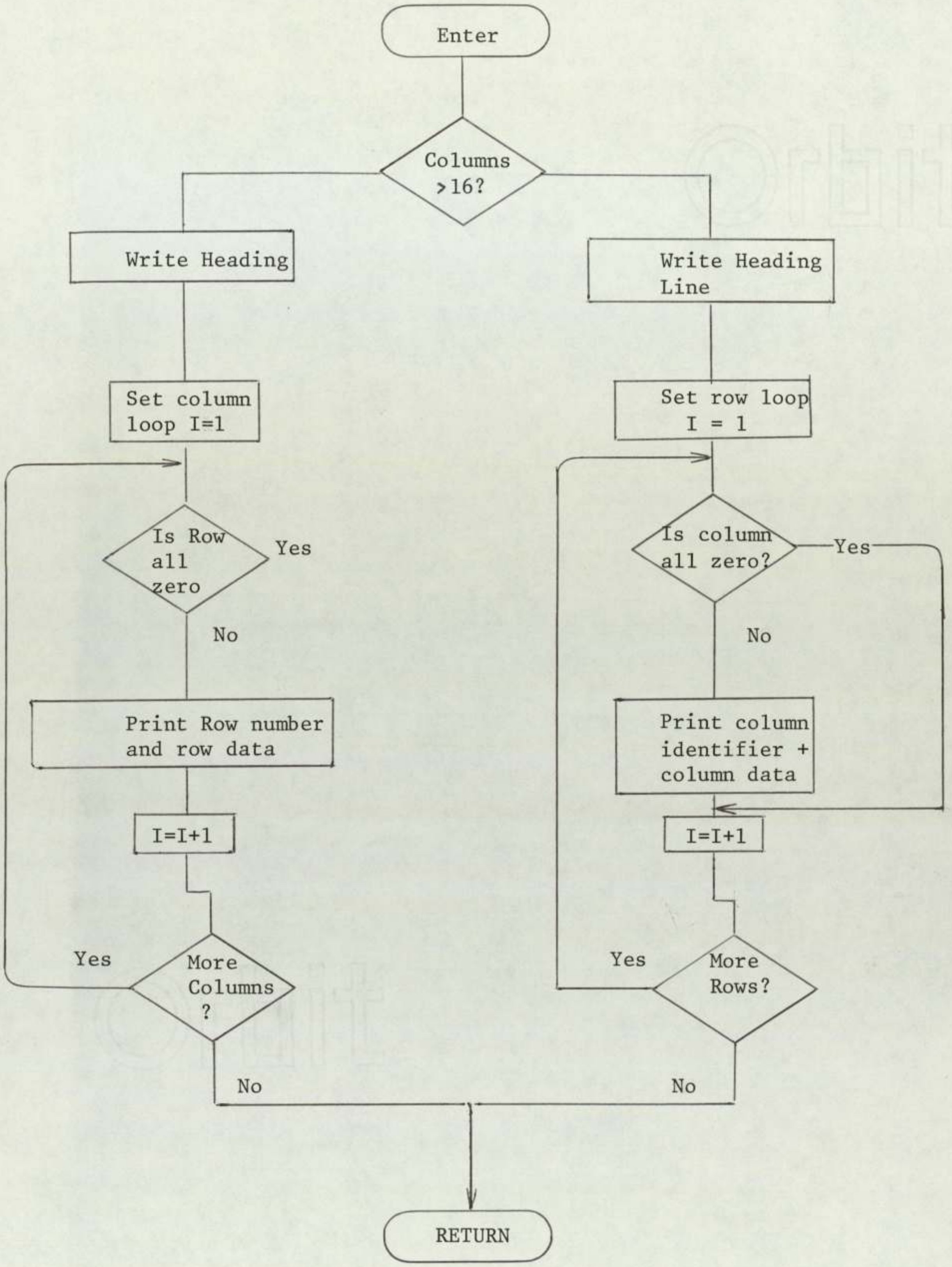
Function	To line print data predicted for bay phase.
System Functions	Read/Write I/O
User Functions	None
Calling Statement	CALL BAYOUT(NMAX,MMAX,D,DT,DESC)
Common Areas	/IO/ , /DATE/
Common Variables Reset	None

Description

The array D(NMAX, MMAX) is to be displayed. There are two options depending on the number of columns to be printed. Rows that are all zero are suppressed and not printed. If there are over 16 columns the array is printed sideways, i.e. each line is a column and rows across a page. Otherwise it can be written properly.

An 8 character identifier (DESC) is printed, also the time of the data (DT) and the current time-date (ICD).

Flow Logic


Enter
Columns >16?
Write Heading
Write Heading Line
Set column loop I=1
Set row loop I = 1
Is Row all zero
Yes
No
Is column all zero?
Yes
No
Print Row number and row data
Print column identifier + column data
I=I+1
I=I+1
Yes
More Columns ?
No
Yes
More Rows?
No
RETURN

B.4. Subroutine BLOCK DATA

(for F1)

Function	To set up common areas and initialise.
Common Areas	/IO/ ,/NODE/ , /LIMITS/
Common Variables Set:	Area NODE is set to zero NCD, IPR (card reader, line printer) in IO are set to 5 and 6 (4-70 MJ convention defaults) LIM1 in 'LIMITS' is set to 1.
Notes	Under ICL 4-50 or 4-70 this module has to appear as the first module, under ICL 1900 series the block common data has to appear as the last module in a program.

B.5. Subroutine BLOCK DATA

(for F2)

Function	Establish and partly initialise common areas of F2 model
System Functions	None
User Functions	None
Calling Statements	BLOCK DATA
Common Areas	/IO/, /NODE/, /ALL/, /LIMITS/, /CTCH22/
Common Variables Reset	NMAX, MMAX in CTCH22; NOUT, NCD, IPR in /IO/, all arrays cleared in /NODE); NMAX2, MMAX2, MMAX1 in /ALL/; NBD, MBD in /CTCH22/; LIM1 in /LIMITS/

Description

Sets some parameters to default values, peripherals data set numbers, and initialises some switches. Only compiler actions.

See notes for F1 version on order of presentation.

B.6. **Subroutine BOTANY**

Function:	Fluid motion in a variable geometry 2-D bay - solution of differential equations.
System Functions:	FLOAT, SQRT, I/O READ/WRITE, SIN, ABS
User Functions:	BAYOUT, FIND, FRACT, DIGITS
Calling Statement:	CALL BOTANY (QIN, YOUT)
Common Area:	/INITDP/, /IO/, /CTCH22/, /TIM/, /SPECS/, /BAYD/
Common Variables Reset:	SEP, VP, UP of /CTCH22/; EST and NEST of /SPECS/ read in

Summary Description:

The routine calculates velocities in x and y plane and heights for a time step. The method employed is a multioperational explicit-implicit scheme as outlined by LEENDERTSE.

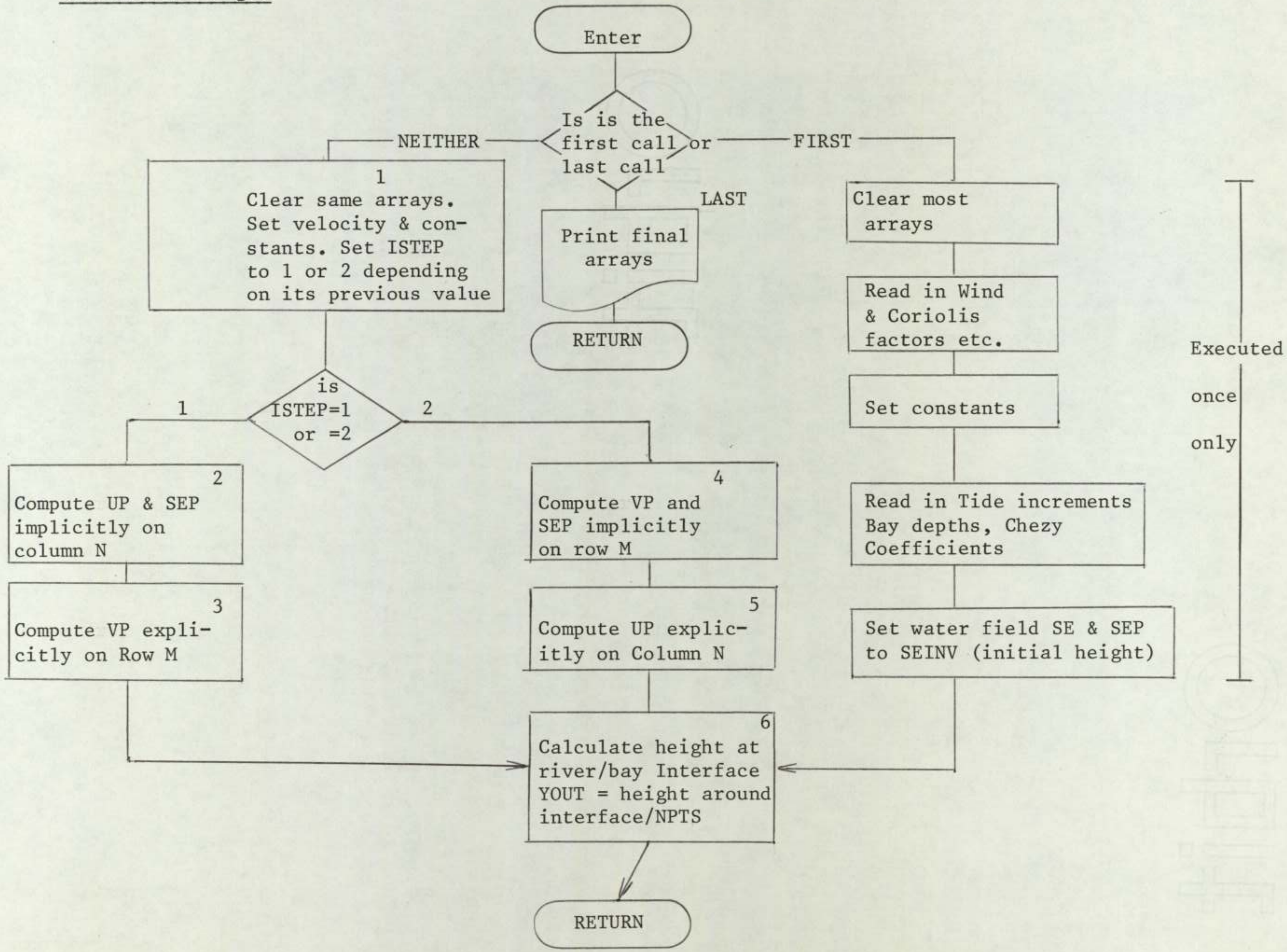
Alternate computations are made explicitly for either UP, SEP or VP, SEP then implicitly for either VP or UP.

On the first call, the depths and Chezy coefficients are read in, also the tide heights forcing function values at the Sea Boundaries are read in and during subsequent calls interpolated for the correct time steps.

Coriolis and wind effects are considered in the Model.

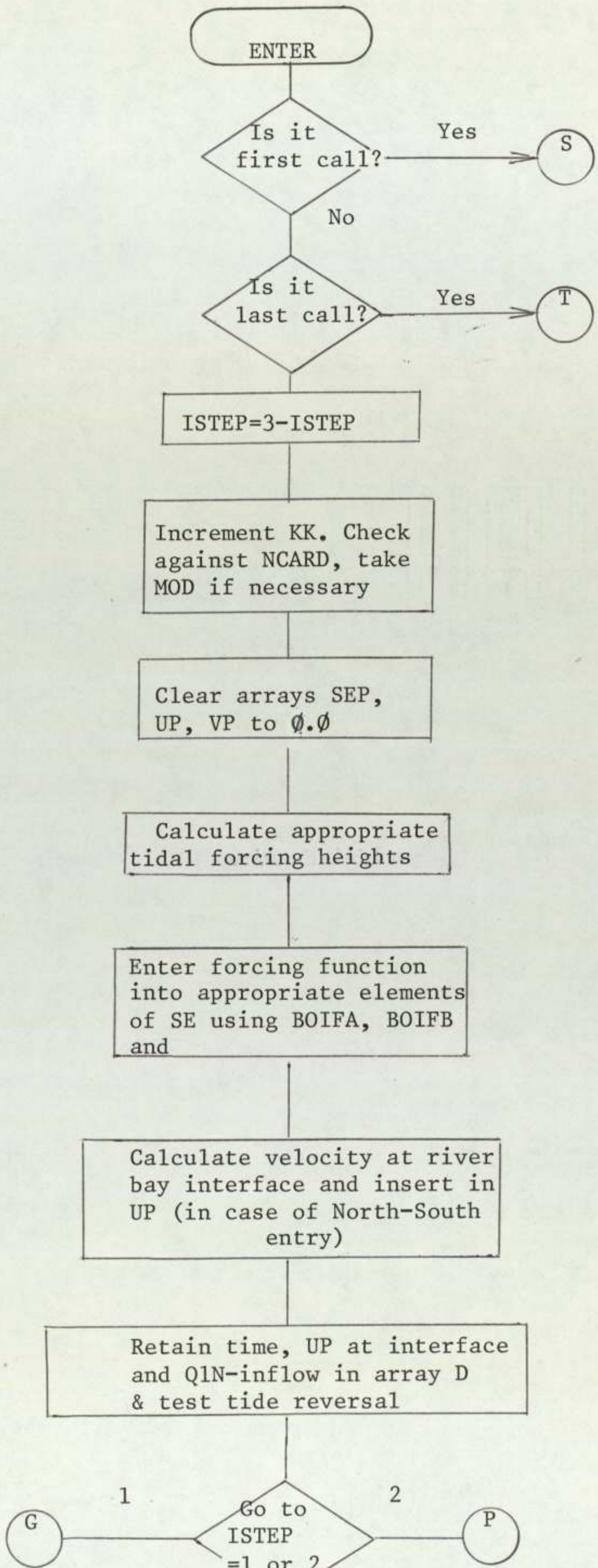
Details of the algorithm are in the theoretical text of the model.

Outline Flow Logic



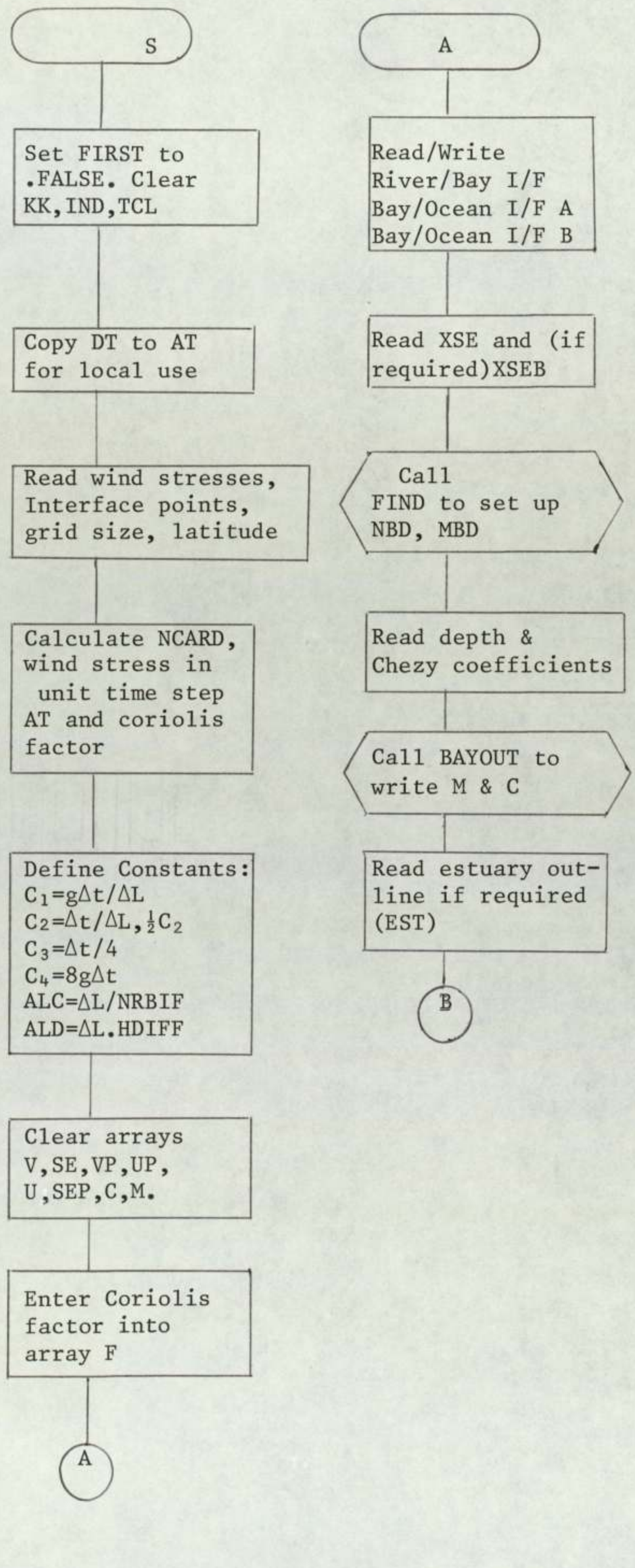
Detailed Flow Logic
Box 1 & Start

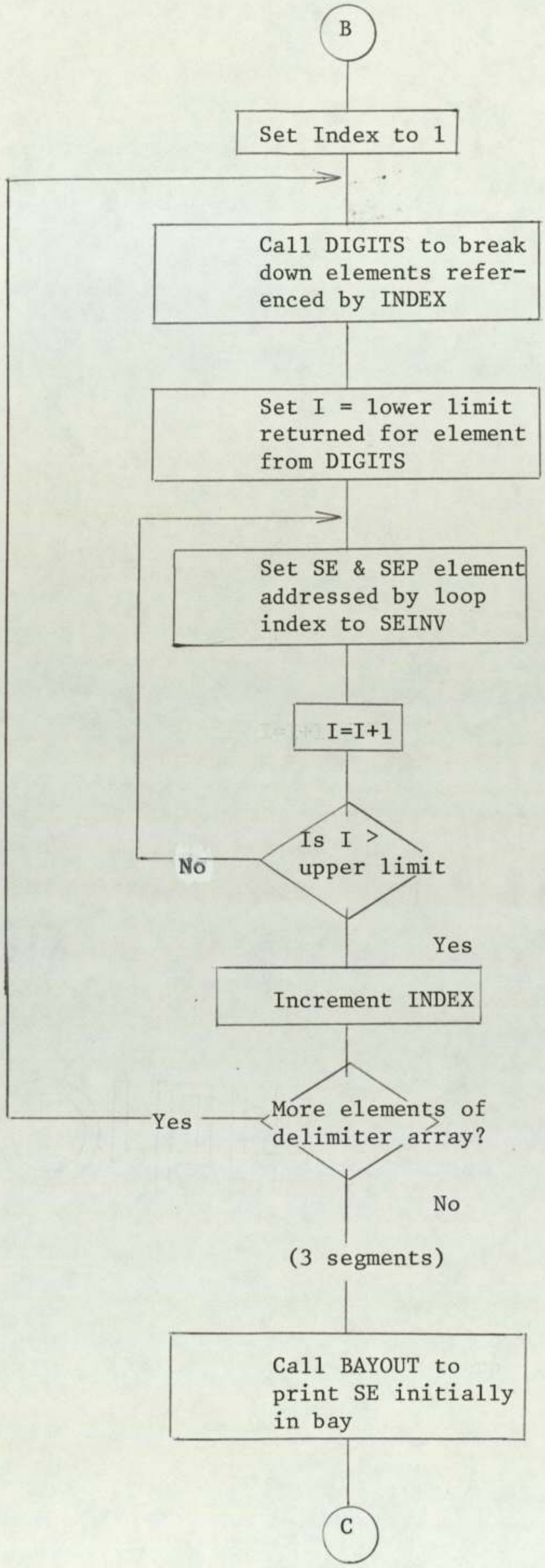
Preparation to a time step computation



Detailed Flow Logic

Section only executed on first call of BOTANY





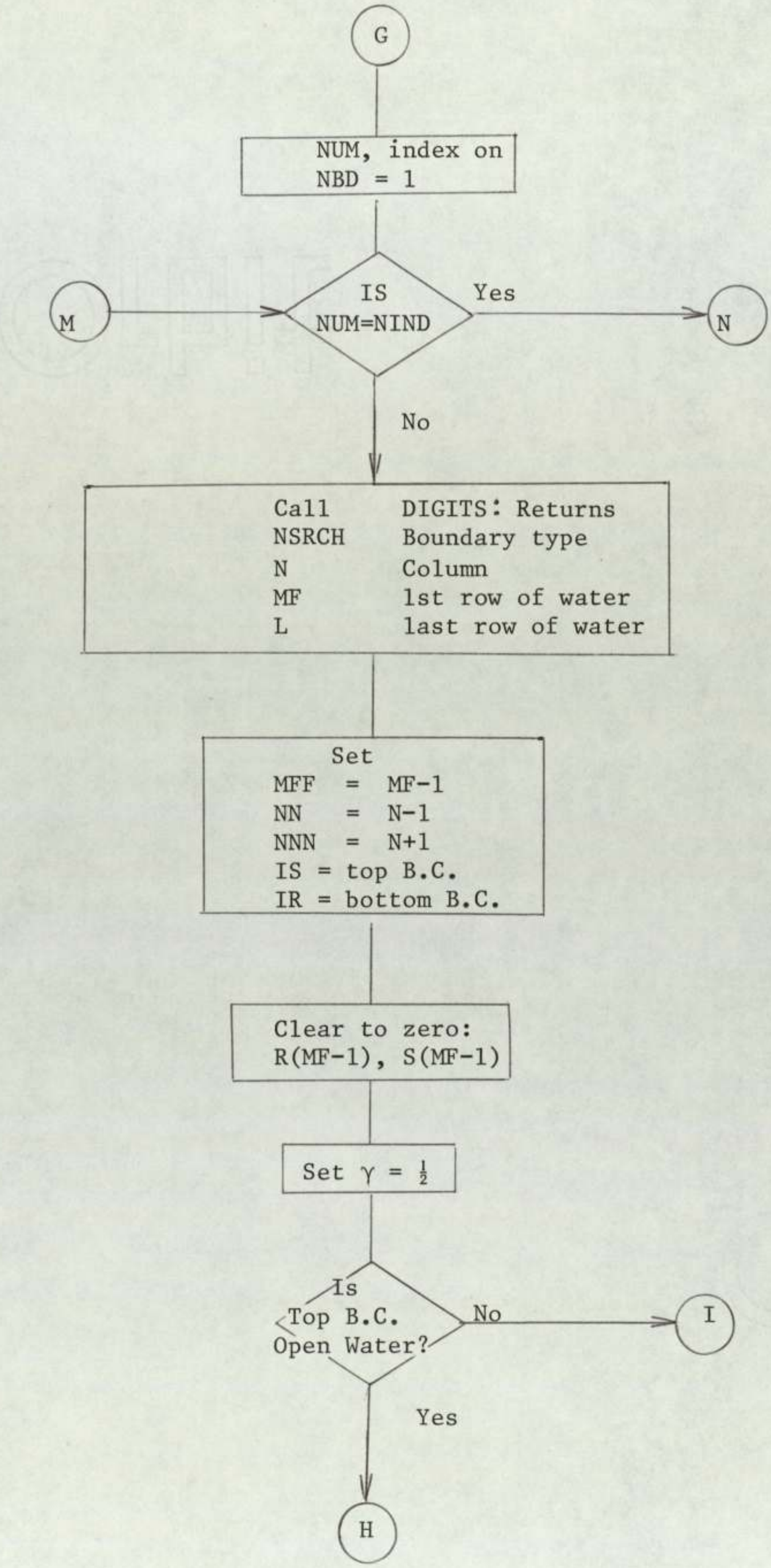
This segment is duplicated 3 times to initialise bay

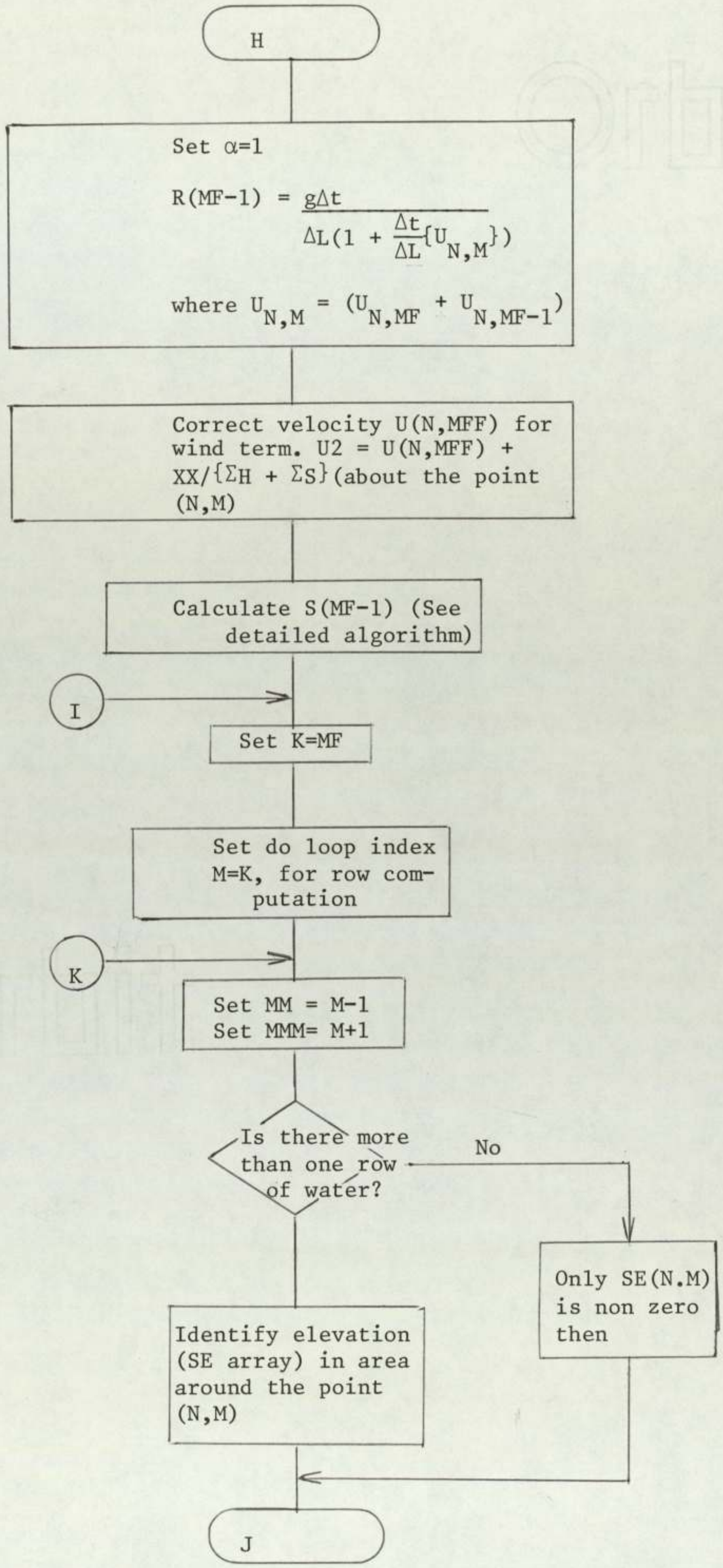
A) INDEX = NUM
Delimiter array NBD

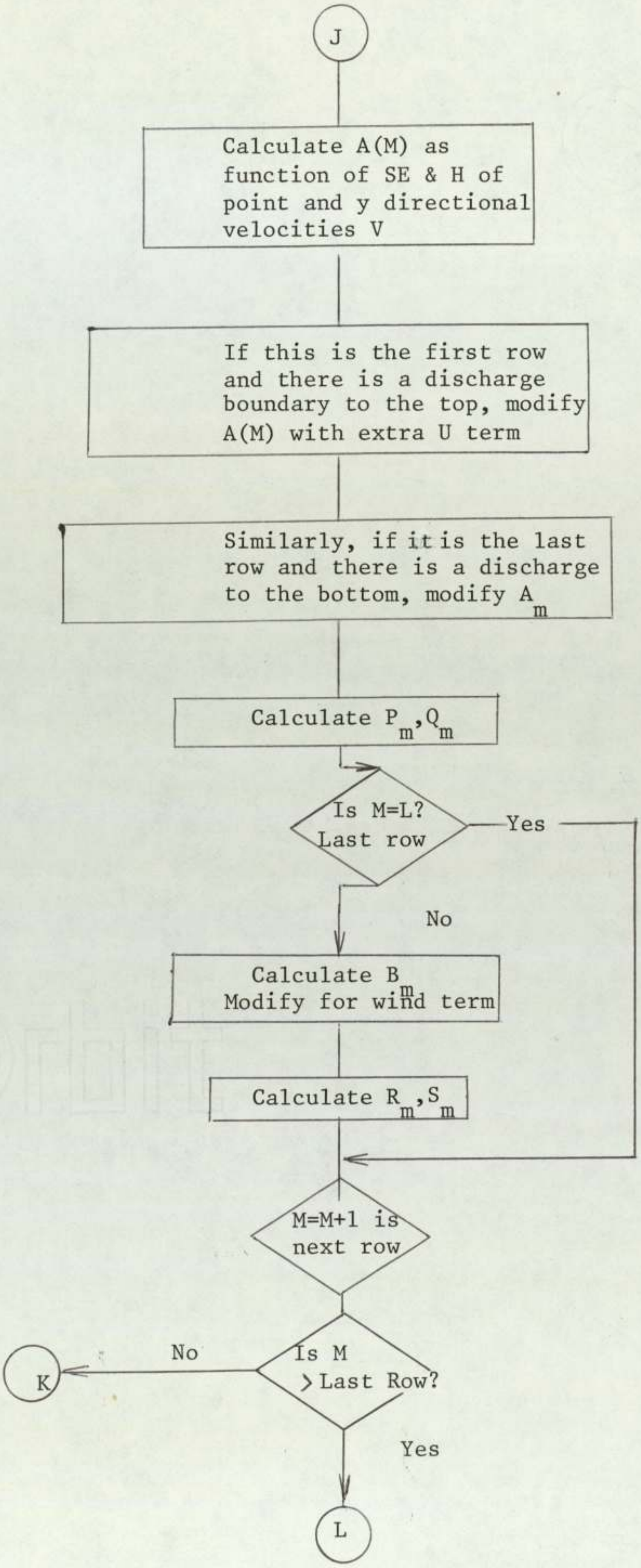
B) INDEX = NA
Delimiter array MOBD

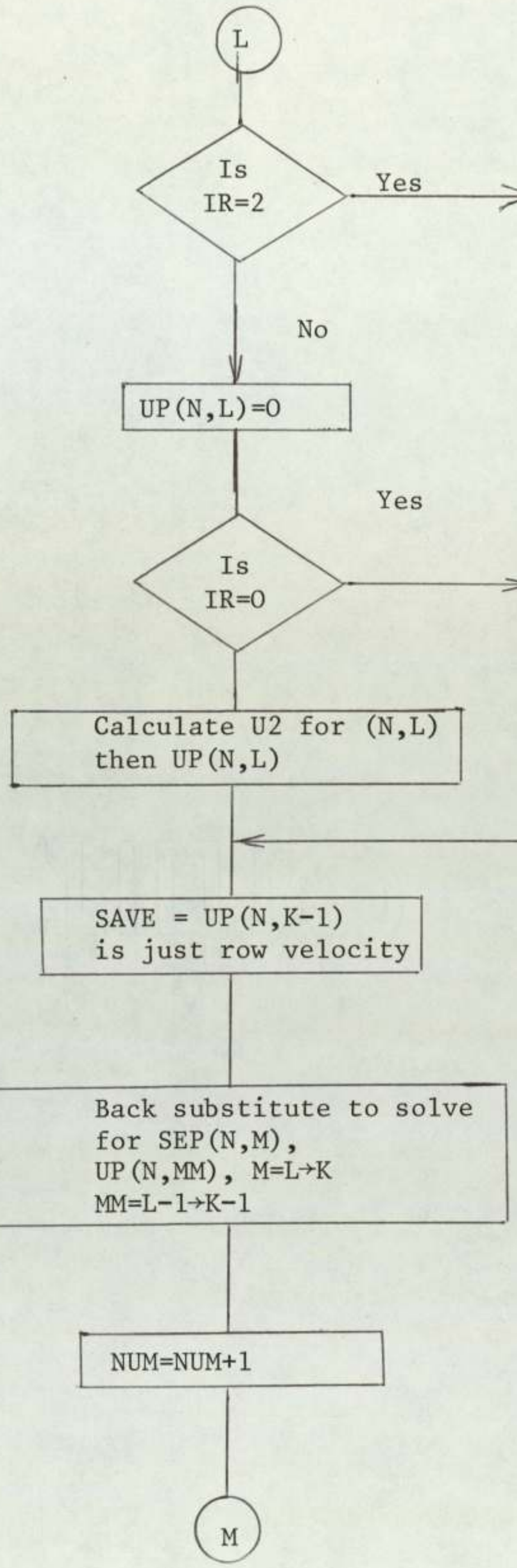
C) INDEX = NA
Delimiter array NOBD

Box 2 Implicit Computation of UP and SEP on Column N

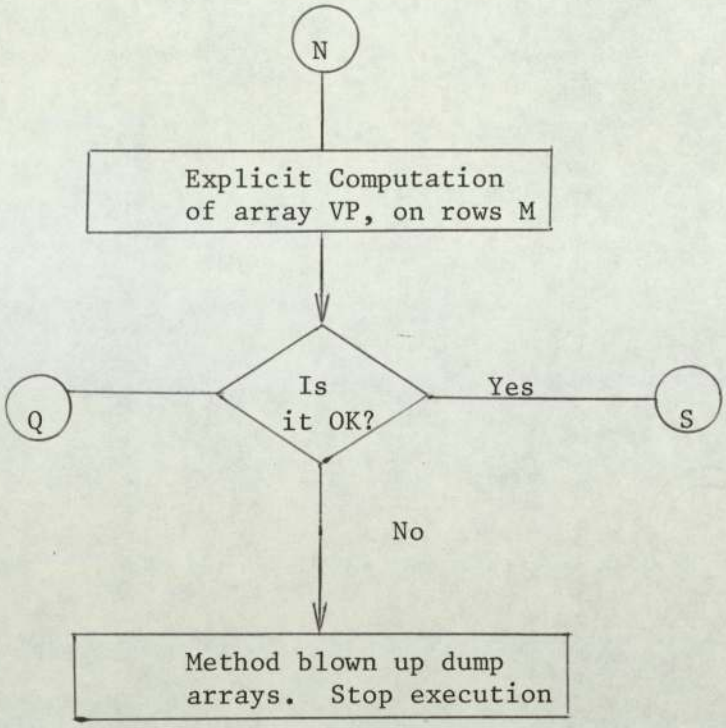



H
Set α=1
R(MF-1) = gΔt / ΔL(1 + Δt/ΔL{U N,M})
where U N,M = (U N,MF + U N,MF-1)
Correct velocity U(N,MFF) for wind term. U2 = U(N,MFF) + XX/{ΣH + ΣS}(about the point (N,M)
Calculate S(MF-1) (See detailed algorithm)
I
Set K=MF
Set do loop index M=K, for row computation
K
Set MM = M-1
Set MMM= M+1
Is there more than one row of water?
No
Only SE(N.M) is non zero then
Identify elevation (SE array) in area around the point (N,M)
J

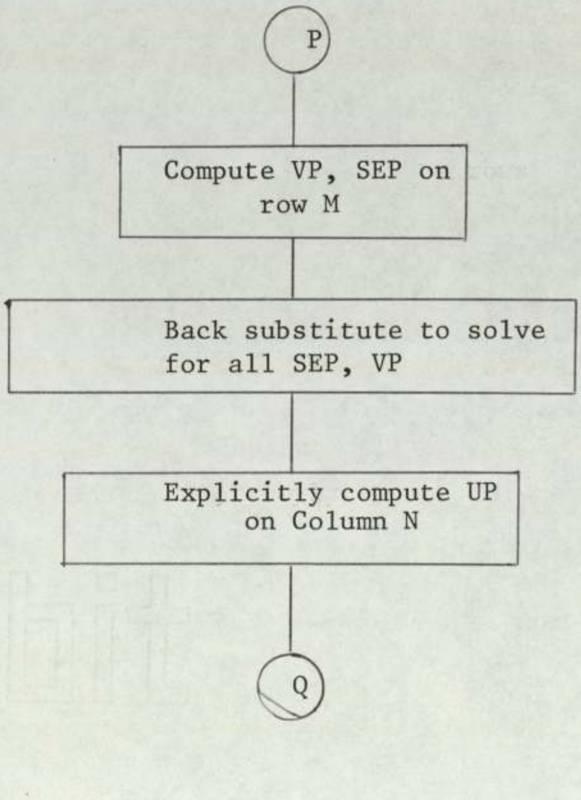

J
Calculate A(M) as function of SE & H of point and y directional velocities V
If this is the first row and there is a discharge boundary to the top, modify A(M) with extra U term
Similarly, if it is the last row and there is a discharge to the bottom, modify A_m
Calculate P_m, Q_m
Is M=L? Last row
Yes
No
Calculate B_m Modify for wind term
Calculate R_m, S_m
M=M+1 is next row
Is M > Last Row?
No
K
Yes
L


L
Is
IR=2
Yes
No
UP(N,L)=0
Yes
Is
IR=0
Calculate U2 for (N,L)
then UP(N,L)
SAVE = UP(N,K-1)
is just row velocity
Back substitute to solve
for SEP(N,M),
UP(N,MM), M=L→K
MM=L-1→K-1
NUM=NUM+1
M

Box 3

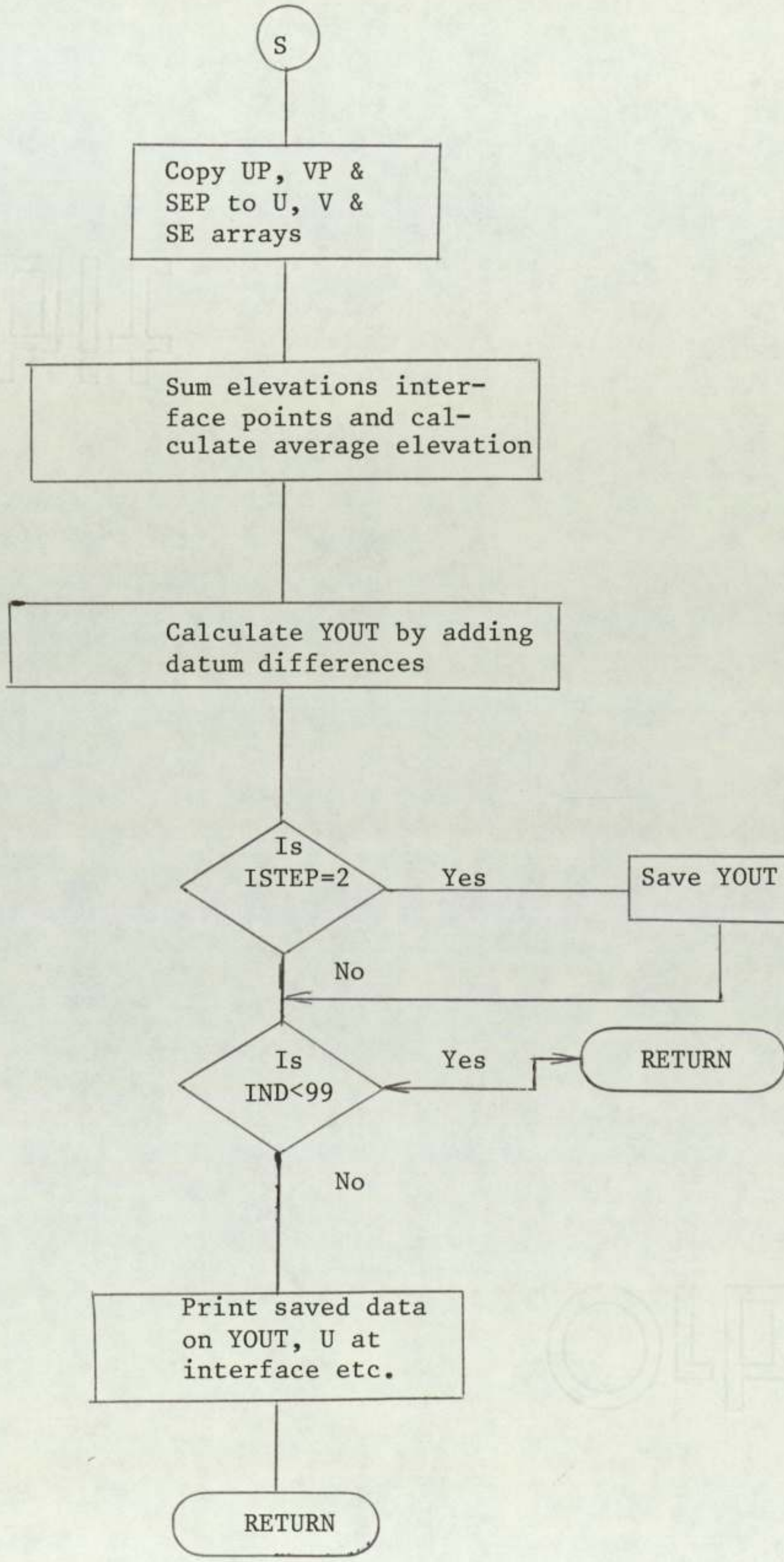


Box 4 & 5 Similar to Box 2-3



Box 6

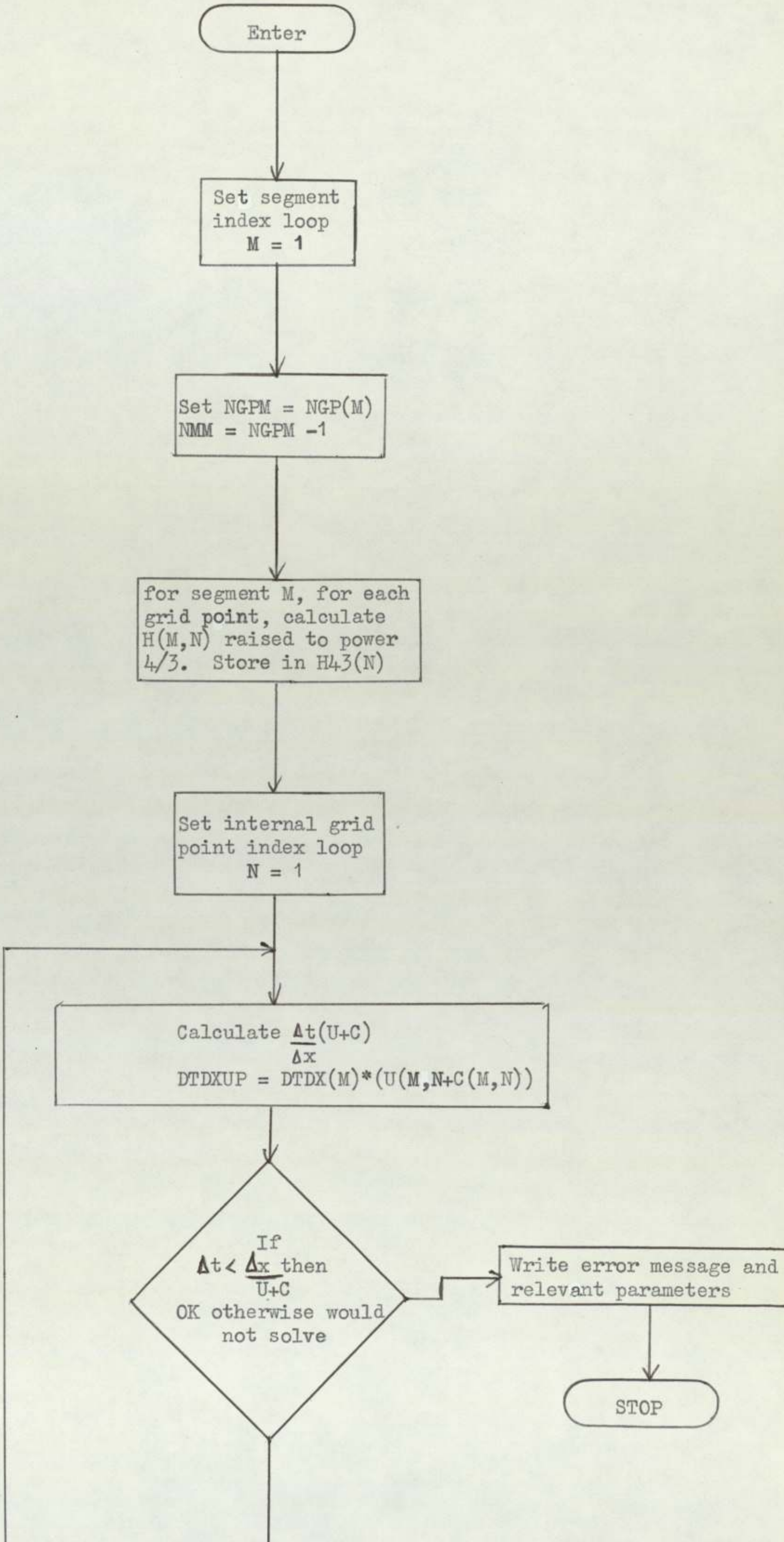
Final Computation of returned values

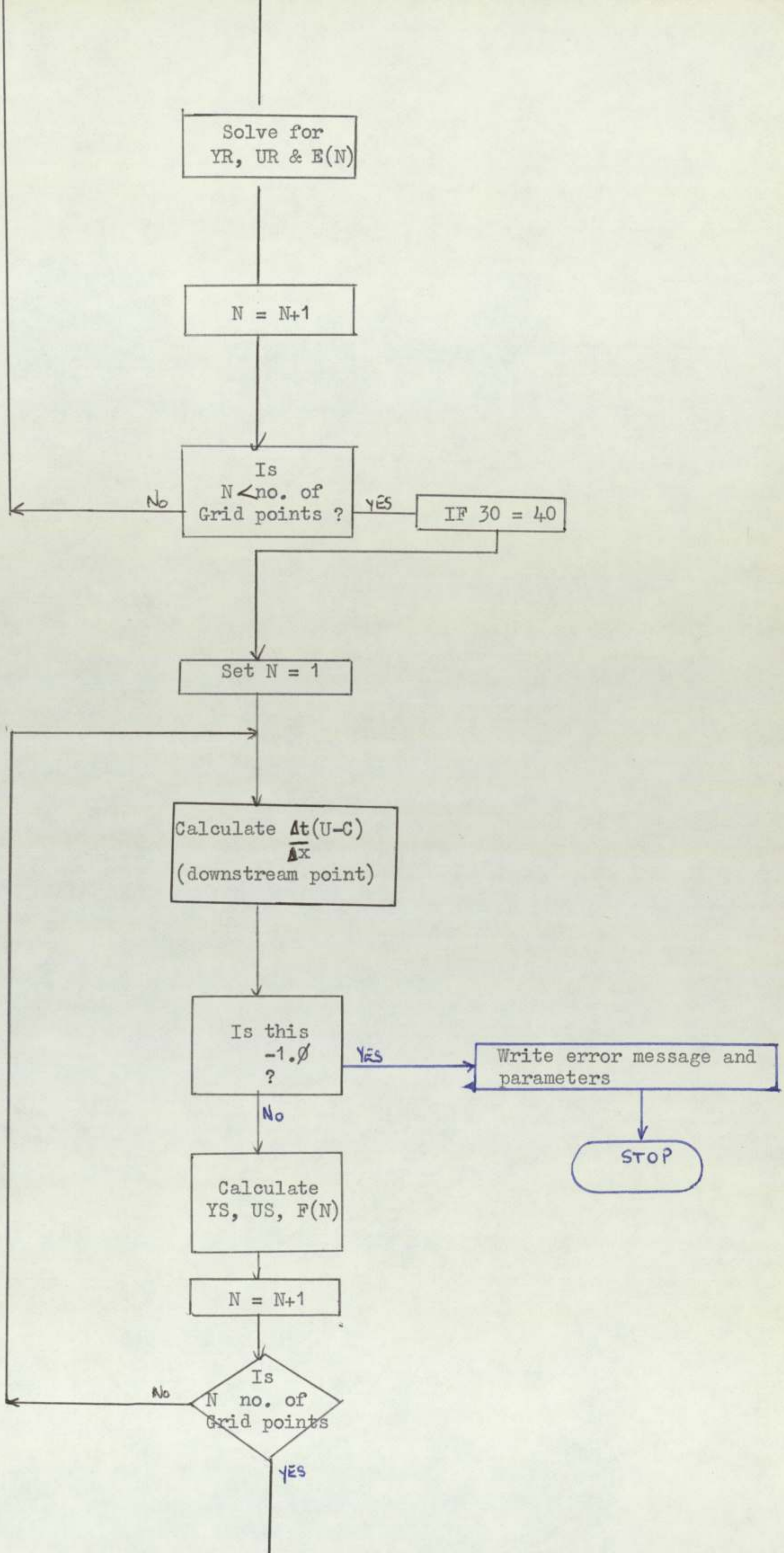


B.7. **Subroutine CALEF**

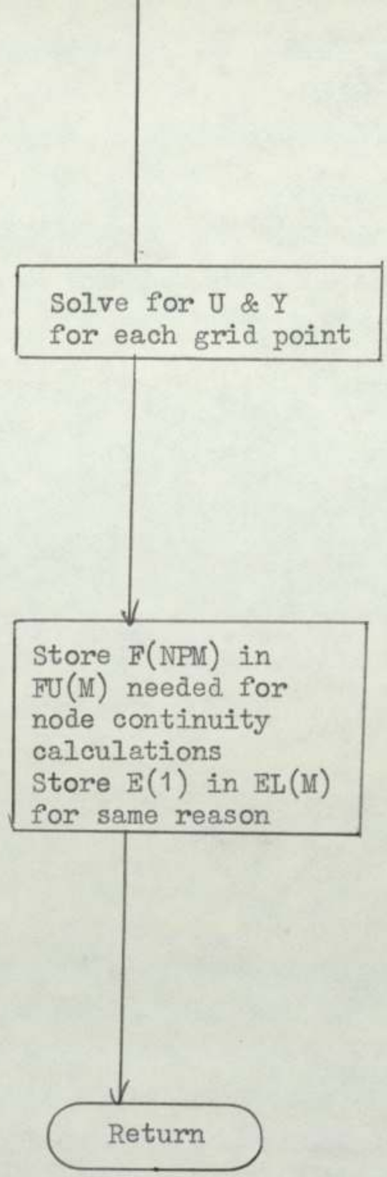
Function:	Calculates upstream and downstream characteristic values and calculates new values of U and Y (velocity and depth in channel) for each grid point.
System Functions:	EXP
User Functions:	None
Calling Statement:	CALL CALEF
Error Messages:	Wave velocity exceeds distance step, segment n
Common Areas:	/IO/, /A/
Common Variables Reset:	Arrays FU, EL, U, Y in area /A/
Description:	The routine solves for new values of U and Y using a distance-time mesh characteristic lines method. The main loop DO 100 is executed for each segment. The value of H for a segment grid point raised to the power 4/3, is stored temporarily in array H43. This array must have a dimension of at least max (NGP(i), i=1, MMAX). The DO 20 loop deals with the upstream characteristic line. The DO 30 loop deals with the downstream characteristic line. The DO 40 loop uses the results from the above to solve simultaneously for a new value of U and Y.

Flow Logic


Enter
Set segment
index loop
M = 1
Set NGPM = NGP(M)
NMM = NGPM -1
for segment M, for each
grid point, calculate
H(M,N) raised to power
4/3. Store in H43(N)
Set internal grid
point index loop
N = 1
Calculate Δt(U+C)/Δx
DTDXUP = DTDX(M)*(U(M,N+C(M,N))
If
Δt < Δx/(U+C) then
OK otherwise would
not solve
Write error message and
relevant parameters
STOP



Solve for YR, UR & E(N)
N = N+1
Is N < no. of Grid points ?
No
YES
IF 30 = 40
Set N = 1
Calculate Δt(U-C)/Δx (downstream point)
Is this -1.Ø ?
YES
Write error message and parameters
STOP
No
Calculate YS, US, F(N)
N = N+1
Is N no. of Grid points
No
YES


Solve for U & Y for each grid point
Store F(NPM) in FU(M) needed for node continuity calculations Store E(1) in EL(M) for same reason
Return

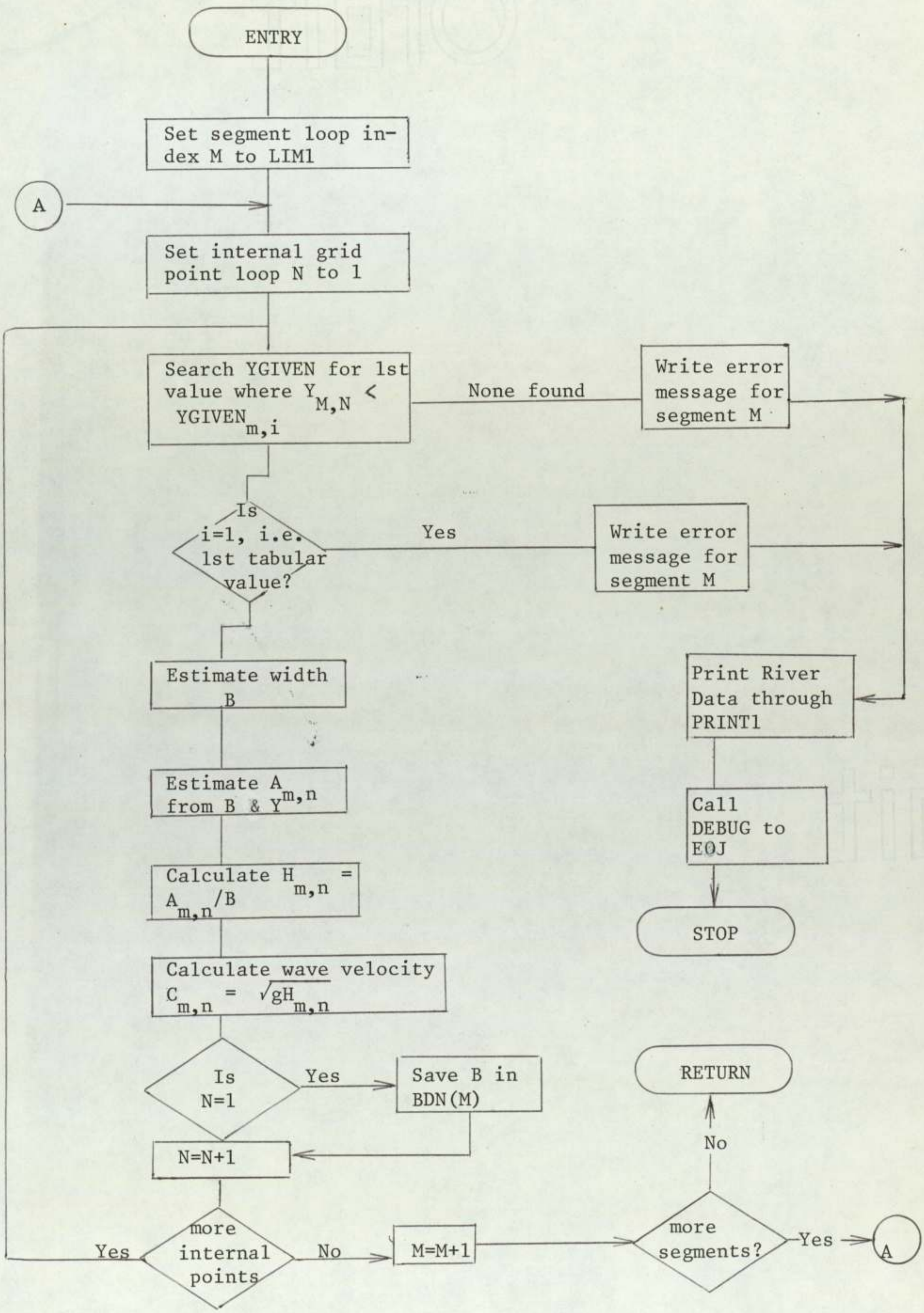
B.8. Subroutine CALHBC

Function:	To estimate area, mean depth and wave velocity at all internal grid points
System Functions:	WRITE, DEBUG, GRAFOR, SORT
User Functions:	PRINT1
Calling Statement:	CALL CALHBC
Error Messages:	A) 'Depth less than minimum given depth, segment m' B) 'Depth exceeds maximum given depth, segment m' C) 'Error in CALHBC'
Common Areas:	/A/, /IO/, /LIMITS/, /NODE/, /C/
Common Variables Reset	A in /C/, H and C in /A/, BDN in /A/

Description

The main loop DO 40 is executed either for all segments or only for the most downstream segment, depending on the value of LIM1. Y(M,H) is the depth of flow predicted for the new time TT. If it is less than 6" it is reset to the minimum allowed. Then this is tested against the tabulated values in YGIVEN to find nearest values, from which a width is interpolated and an area of cross-section estimated. The mean depth is defined as the area/width and the wave velocity is then defined by $\sqrt{gH_{m,n}}$ The downstream width is retained in BDN. For the first call, the calculations are printed to the line printer.

Flow Logic



ENTRY
Set segment loop index M to LIM1
A
Set internal grid point loop N to 1
Search YGIVEN for 1st value where $Y_{M,N}$ < $YGIVEN_{m,i}$
None found
Write error message for segment M
Is i=1, i.e. 1st tabular value?
Yes
Write error message for segment M
Estimate width B
Print River Data through PRINT1
Estimate $A_{m,n}$ from B & $Y_{m,n}$
Call DEBUG to EOJ
Calculate $H_{m,n}$ = $A_{m,n}$/B
STOP
Calculate wave velocity $C_{m,n}$ = $\sqrt{gH_{m,n}}$
Is N=1
Yes
Save B in BDN(M)
RETURN
No
N=N+1
Yes
more internal points
No
M=M+1
more segments?
Yes
A

B.9. Subroutine CALNOD

Function: To solve for continuity at each node and for array Y and U at end grid points of each segment.

System Functions: None

User Functions: None

Calling Statement: CALL CALNOD

Error Messages: Stop 1 - invalid end type for a node (<2 or >5)

Common Areas: /IO/, /A/, /NODE/, /C/

Common Variables Reset: Arrays Y, U in area /A/

Description: Suppose segments i,j,k flow into node n and then out into segment 1. The downstream grid points of segments i,j,k and the upstream grid point of segment 1 are all coincident at the node. The equation of continuity and elevation of water surface equality at each node can be solved.

Equation of Continuity

$$U_i A_i + U_j A_j + U_k A_k = U_1 A_1$$

Elevation characteristics

$$E_{i,j,k} = U_{i,j,k} + gY/C_{i,j,k}$$

$$E_1 = U_1 - gY/C_1$$

where U = velocity, A = area, E = elevation, Y is unknown height at node, C is the wave velocity ($=\sqrt{gH}$)

This gives 5 equations in 5 unknowns

(velocities U and equal height Y) and

so a solution is possible

$U_{i,j,k}=(E_{i,j,k}*C_{i,j,k}-gY)/C_{i,j,k}$ $U_1=(E_1*C_1+gY)/C_1$

Substituting in the equation of continuity

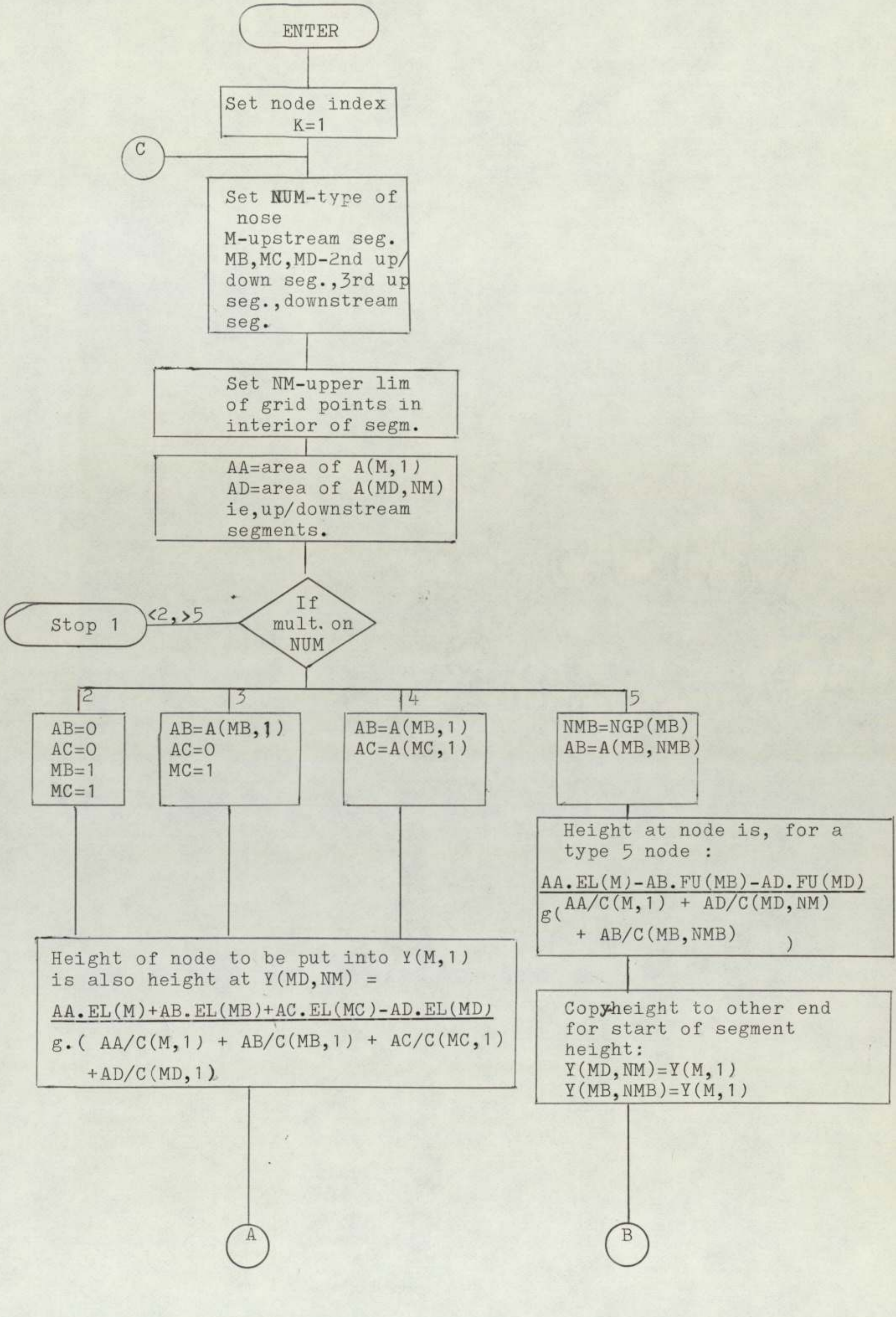
gives

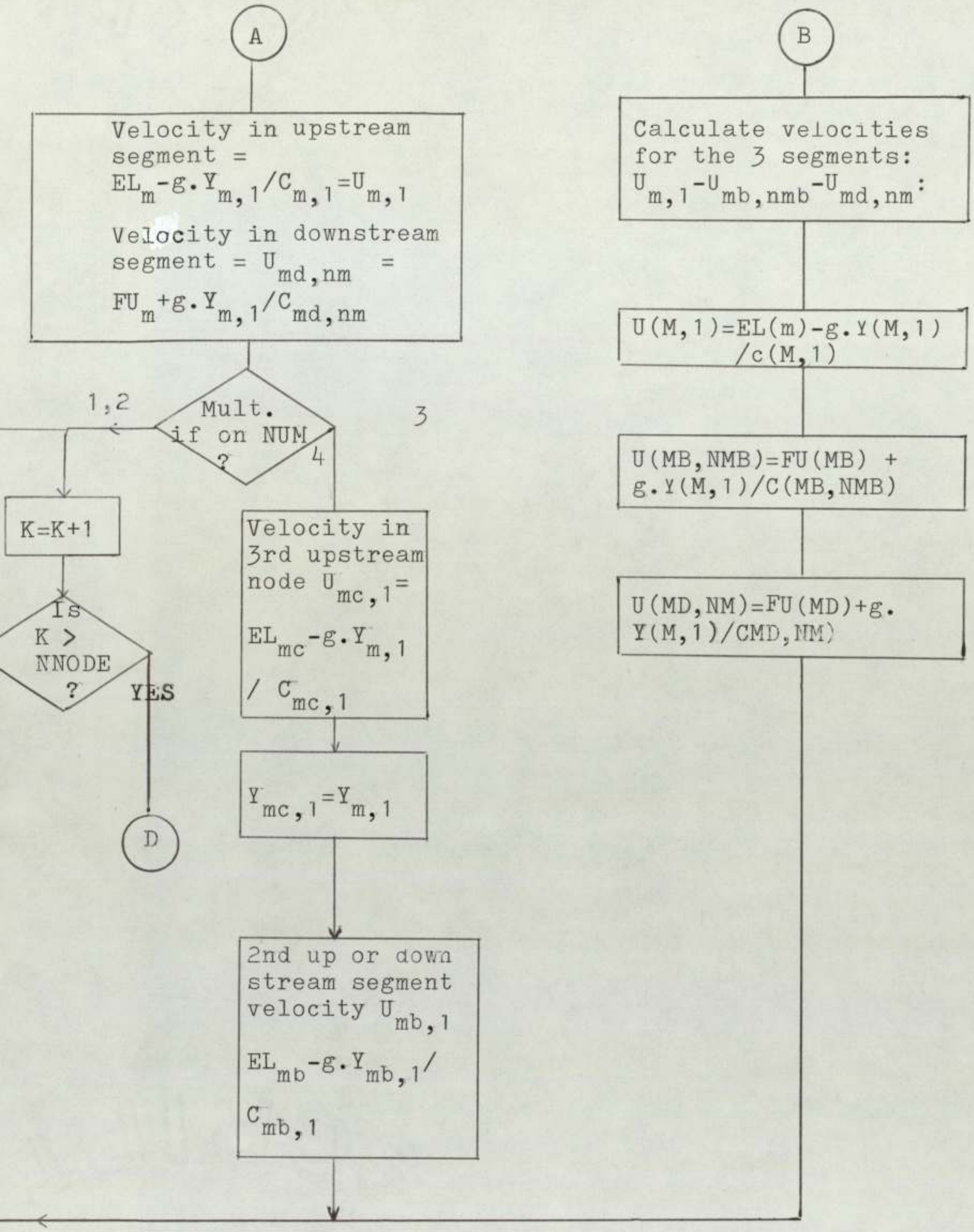
$$Y+(A_iE_i+A_jE_j+A_kE_k-A_1E_1)/g\left(\frac{Ai}{C_i}+\frac{Aj}{C_j}+\frac{Ak}{C_k}+\frac{Al}{C_1}\right)$$

Once having computed Y, the velocities can be calculated from the characteristic elevation equations as above.

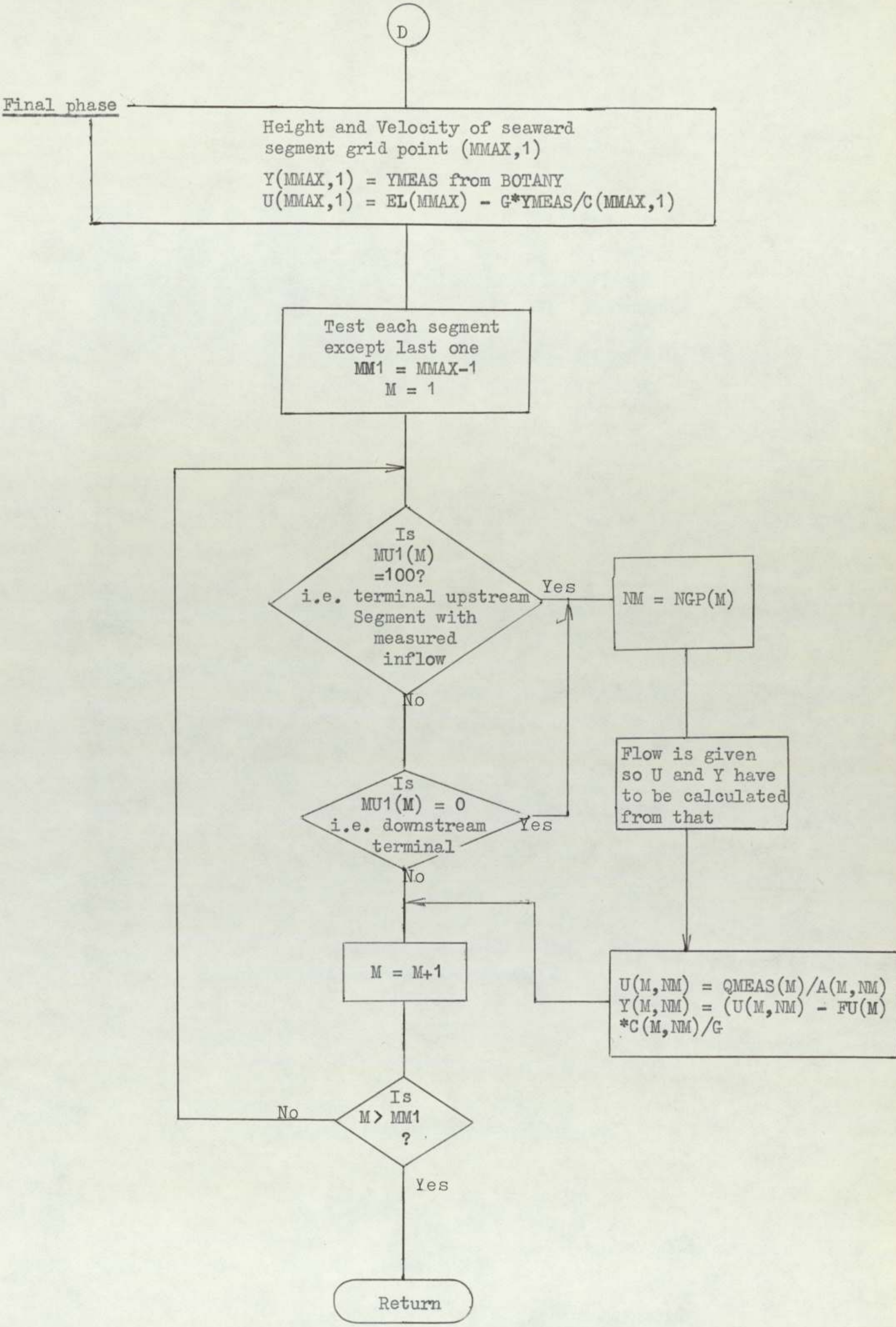
Other combinations of nodes result in directly similar equations (for ITYPND (ref routine NODE) = 2,3,4) with a result for type 5 node being similar except for two terms being subtracted in the dividend in the Y = equation.

The program deals with each case by multiple GoTo's and using zero coefficients in the case where ITYPND is 2, 3 or 4 to save coding and memory.


ENTER
Set node index K=1
C
Set NUM-type of nose M-upstream seg. MB,MC,MD-2nd up/down seg.,3rd up seg.,downstream seg.
Set NM-upper lim of grid points in interior of segm.
AA=area of A(M,1) AD=area of A(MD,NM) ie,up/downstream segments.
If mult. on NUM
<2, >5
Stop 1
2
AB=0 AC=0 MB=1 MC=1
3
AB=A(MB,1) AC=0 MC=1
4
AB=A(MB,1) AC=A(MC,1)
5
NMB=NGP(MB) AB=A(MB,NMB)
Height at node is, for a type 5 node :
AA.EL(M)-AB.FU(MB)-AD.FU(MD)
g(AA/C(M,1) + AD/C(MD,NM) + AB/C(MB,NMB))
Height of node to be put into Y(M,1) is also height at Y(MD,NM) =
AA.EL(M)+AB.EL(MB)+AC.EL(MC)-AD.EL(MD)
g.(AA/C(M,1) + AB/C(MB,1) + AC/C(MC,1) +AD/C(MD,1)
Copy height to other end for start of segment height:
Y(MD,NM)=Y(M,1)
Y(MB,NMB)=Y(M,1)
A
B



Section for back substitution of velocities

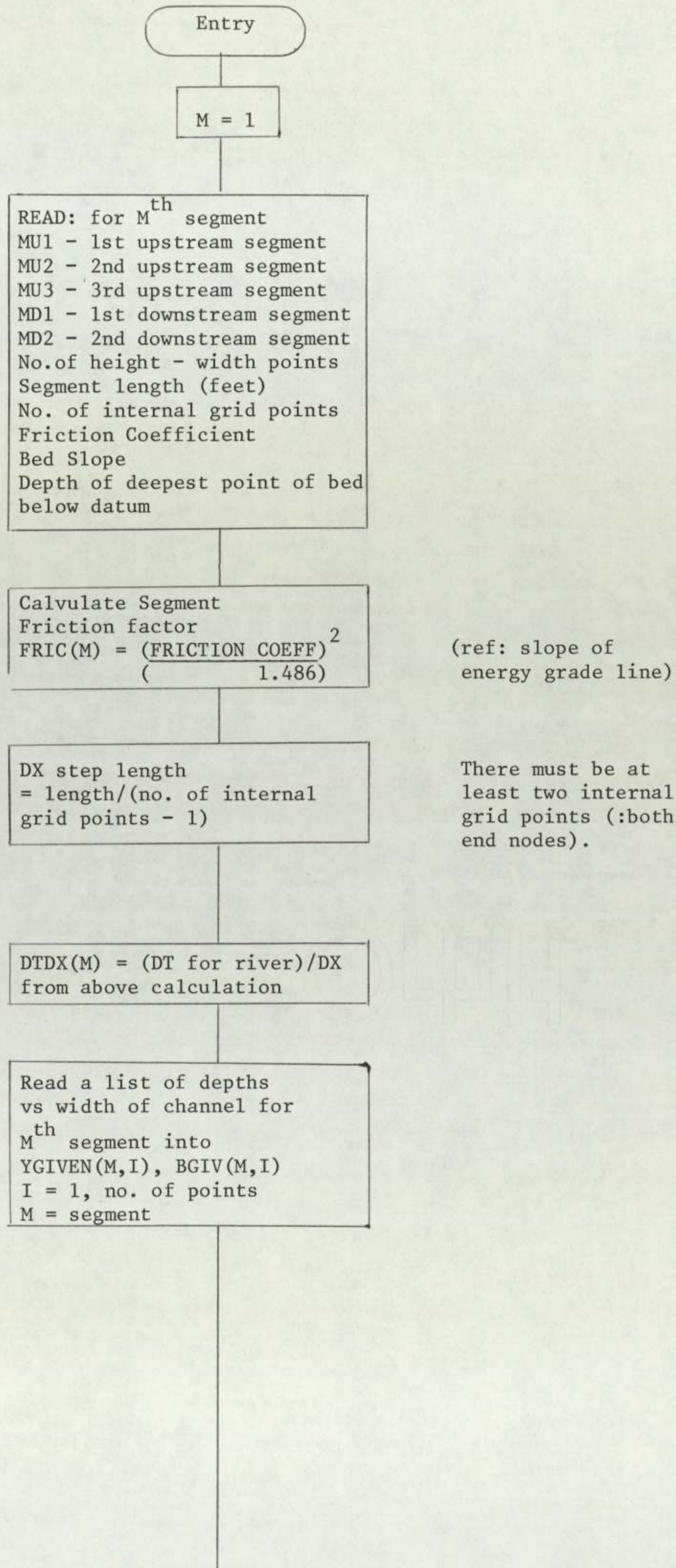

D
Final phase
Height and Velocity of seaward segment grid point (MMAX,1)
Y(MMAX,1) = YMEAS from BOTANY
U(MMAX,1) = EL(MMAX) - G*YMEAS/C(MMAX,1)
Test each segment except last one
MM1 = MMAX-1
M = 1
Is MU1(M) =100? i.e. terminal upstream Segment with measured inflow
Yes
No
NM = NGP(M)
Flow is given so U and Y have to be calculated from that
Is MU1(M) = 0 i.e. downstream terminal
Yes
No
M = M+1
U(M,NM) = QMEAS(M)/A(M,NM)
Y(M,NM) = (U(M,NM) - FU(M) *C(M,NM)/G
Is M > MM1 ?
No
Yes
Return

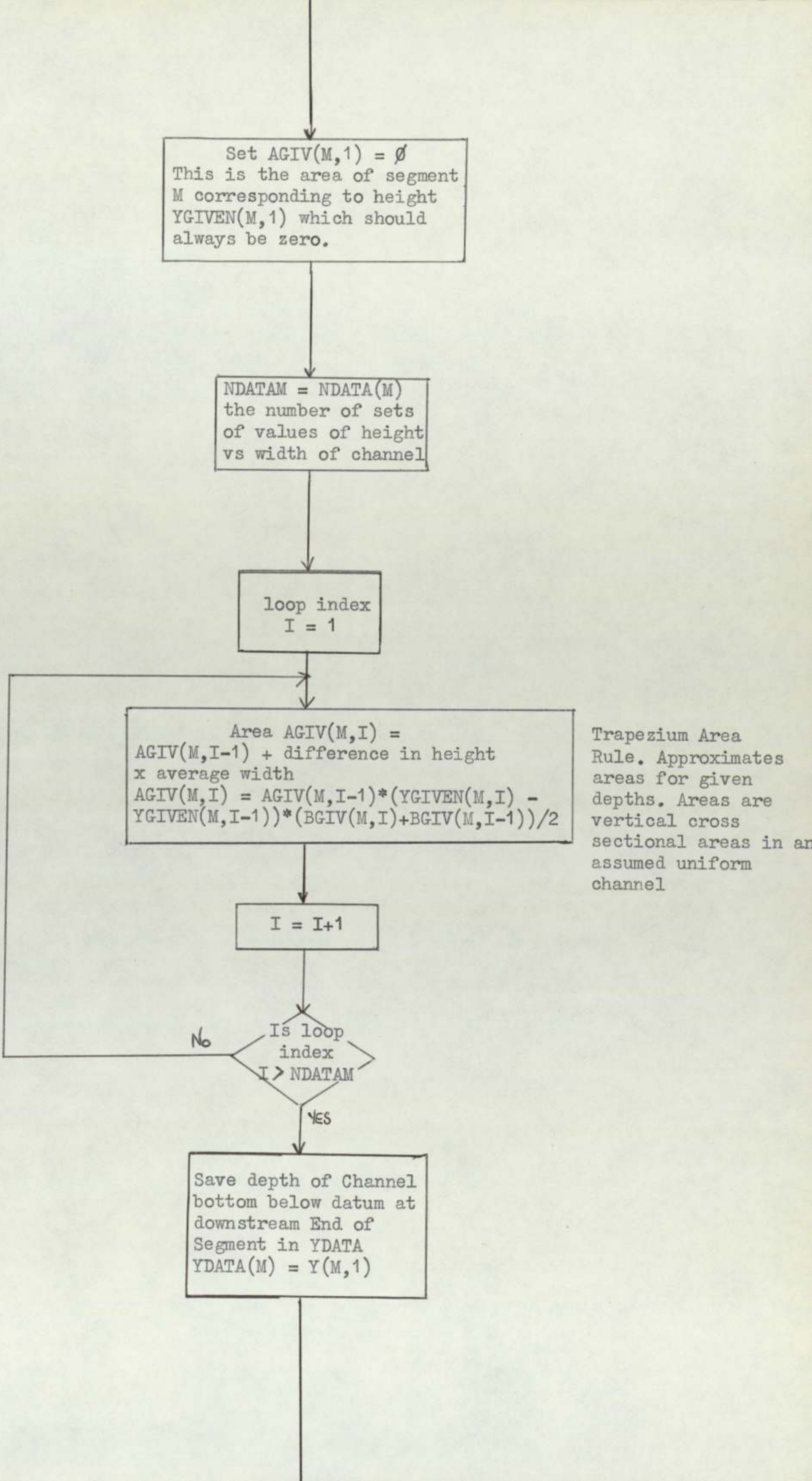
B.10. SUBROUTINE DATA

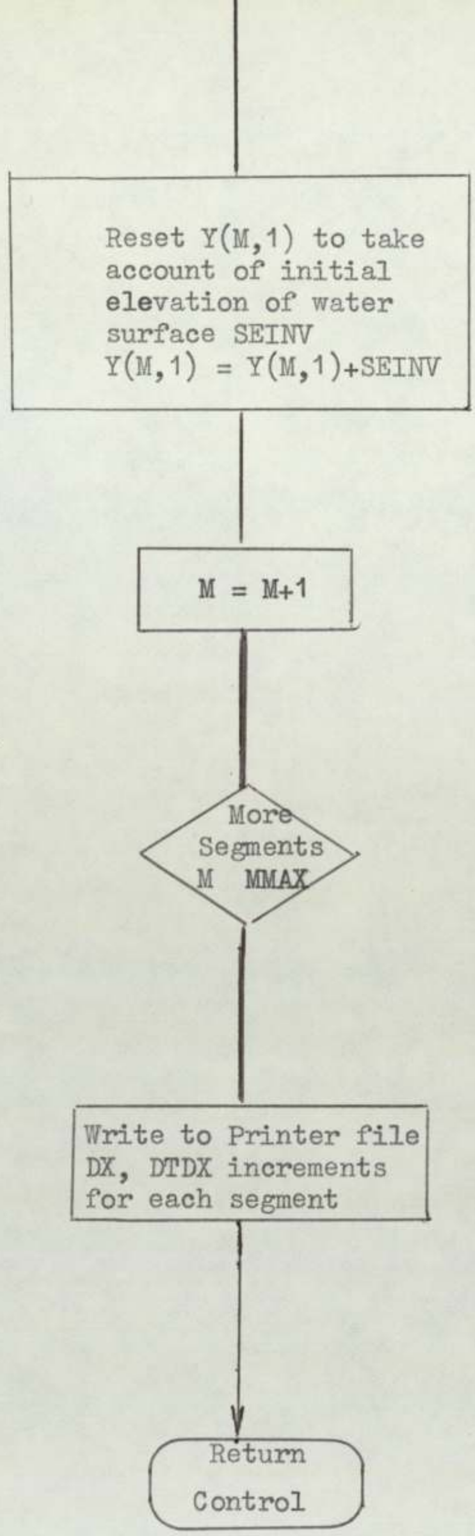
Function:	Read in basic details of one dimensional segments and compute some basic parameters.
Common Areas:	/INITDP/,/IO/, /A/, /NODE/
User Supplied Routines:	None
System Routines:	READ/GET WRITE/PUT
Calling Statement:	CALL DATA
Error Messages:	STOP 2 Number of grid points for a segment less than 2.
Common Variables Altered:	/A/, YGIVEN, Y(M,1), BG1V, FRIC, DTDX, MU1, MU2, MU3, MD1, MD2, NDATA, XLNGTH, SBED, AGIV, YDATA; mostly read in but some pre-computation
Description:	For each segment, certain parameters are required. These are the up and downstream segment numbers to which it is connected, the number of internal grid points required, length and friction coefficient, bed slope and relative heights to other segments. An overall friction coefficient for the segment is then calculated along with length increments and the dt/dx ratio. Further, for each segment, a list of heights above lowest point and channel widths is input. The maximum number of points allowed is dependent on the second dimensions of the arrays YGIVEN - BGIV & AGIV.

From this data an estimate of the area corresponding to a certain height is made.

Once all segment data is read in, the step lengths DX and DT/DX are printed for each segment as they are useful if timestep problems occur.


Entry
M = 1
READ: for Mth segment
MU1 - 1st upstream segment
MU2 - 2nd upstream segment
MU3 - 3rd upstream segment
MD1 - 1st downstream segment
MD2 - 2nd downstream segment
No.of height - width points
Segment length (feet)
No. of internal grid points
Friction Coefficient
Bed Slope
Depth of deepest point of bed below datum
Calvulate Segment Friction factor
FRIC(M) = $\left(\frac{\text{FRICTION COEFF}}{1.486}\right)^2$
(ref: slope of energy grade line)
DX step length = length/(no. of internal grid points - 1)
There must be at least two internal grid points (:both end nodes).
DTDX(M) = (DT for river)/DX from above calculation
Read a list of depths vs width of channel for Mth segment into YGIVEN(M,I), BGIV(M,I)
I = 1, no. of points
M = segment

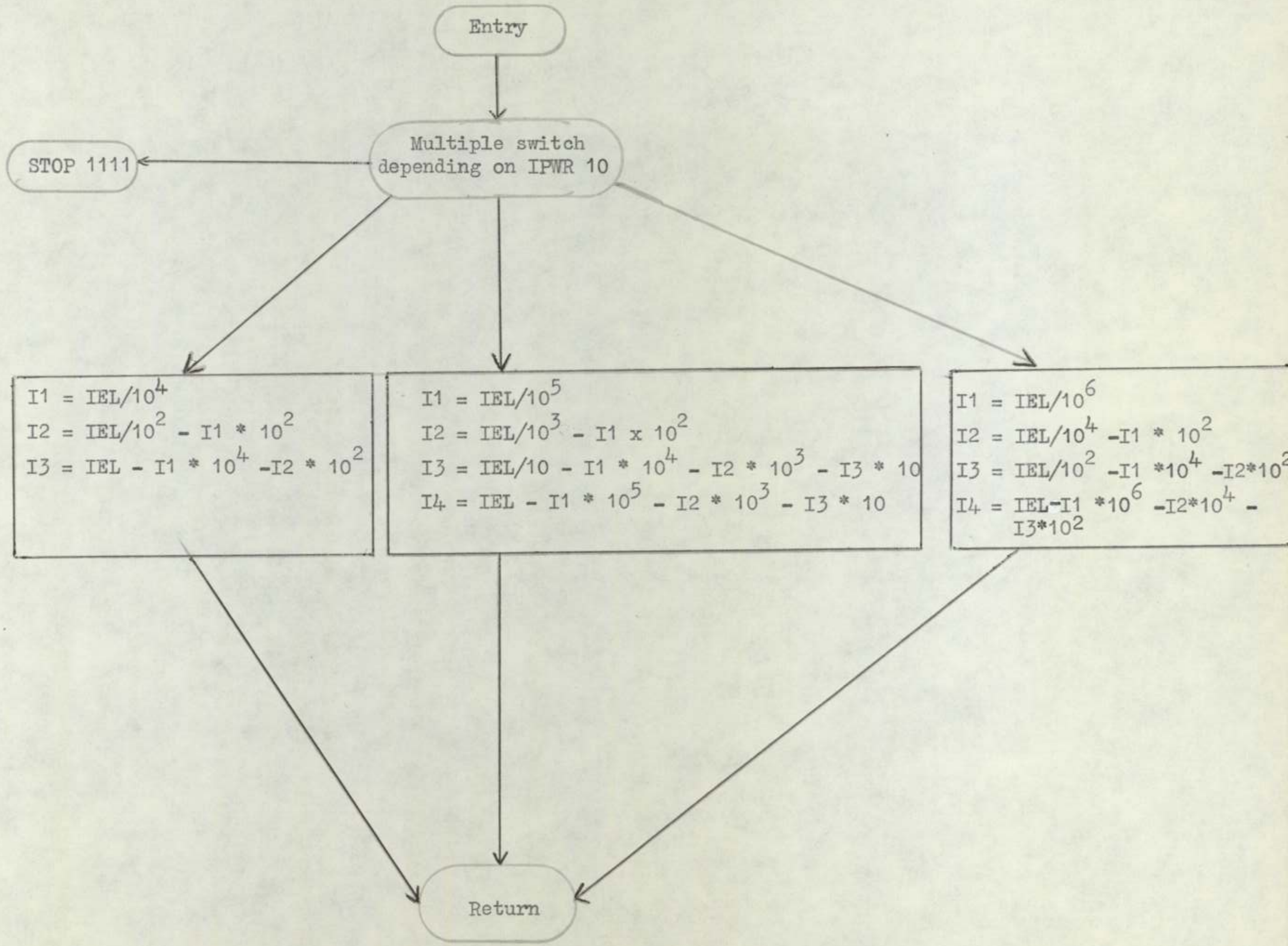

Set AGIV(M,1) = Ø
This is the area of segment M corresponding to height YGIVEN(M,1) which should always be zero.
NDATAM = NDATA(M) the number of sets of values of height vs width of channel
loop index I = 1
Area AGIV(M,I) = AGIV(M,I-1) + difference in height x average width
AGIV(M,I) = AGIV(M,I-1)*(YGIVEN(M,I) - YGIVEN(M,I-1))*(BGIV(M,I)+BGIV(M,I-1))/2
Trapezium Area Rule. Approximates areas for given depths. Areas are vertical cross sectional areas in an assumed uniform channel
I = I+1
Is loop index I > NDATAM
No
YES
Save depth of Channel bottom below datum at downstream End of Segment in YDATA
YDATA(M) = Y(M,1)


Reset Y(M,1) to take account of initial elevation of water surface SEINV
Y(M,1) = Y(M,1)+SEINV
M = M+1
More Segments
M MMAX
Write to Printer file DX, DTDX increments for each segment
Return Control

B.11. Subroutine DIGITS
(for F2)

Function:	To break down a I * 4 digit into specific patterns when one word is used to hold multiple data as in the Fischer Model.
System Functions:	None
User Functions:	None
Calling Statement:	CALL DIGITS (IEL, IPWR10, I1, I2, I3, I4)
Error Messages:	STOP 1111 if IPWR10 is less than 3 or greater than 6.
Common Variables:	None
Description:	Because of storage shortage, a 4 byte 32 bit word can be used more efficiently by storing more than one variable in it. The maximum digit that can be stored in this case is 2^{31}-1. The disadvantage is that once the data is so stored, if it has to be accessed fairly frequently, a process of selective division has to be completed each time. This is time consuming but as opposed to using a segmentation of the program it is a better method. There are 3 basic schemes for dividing out parameters in keeping with the convention used by other parts of the Fischer Model.

Flow Logic



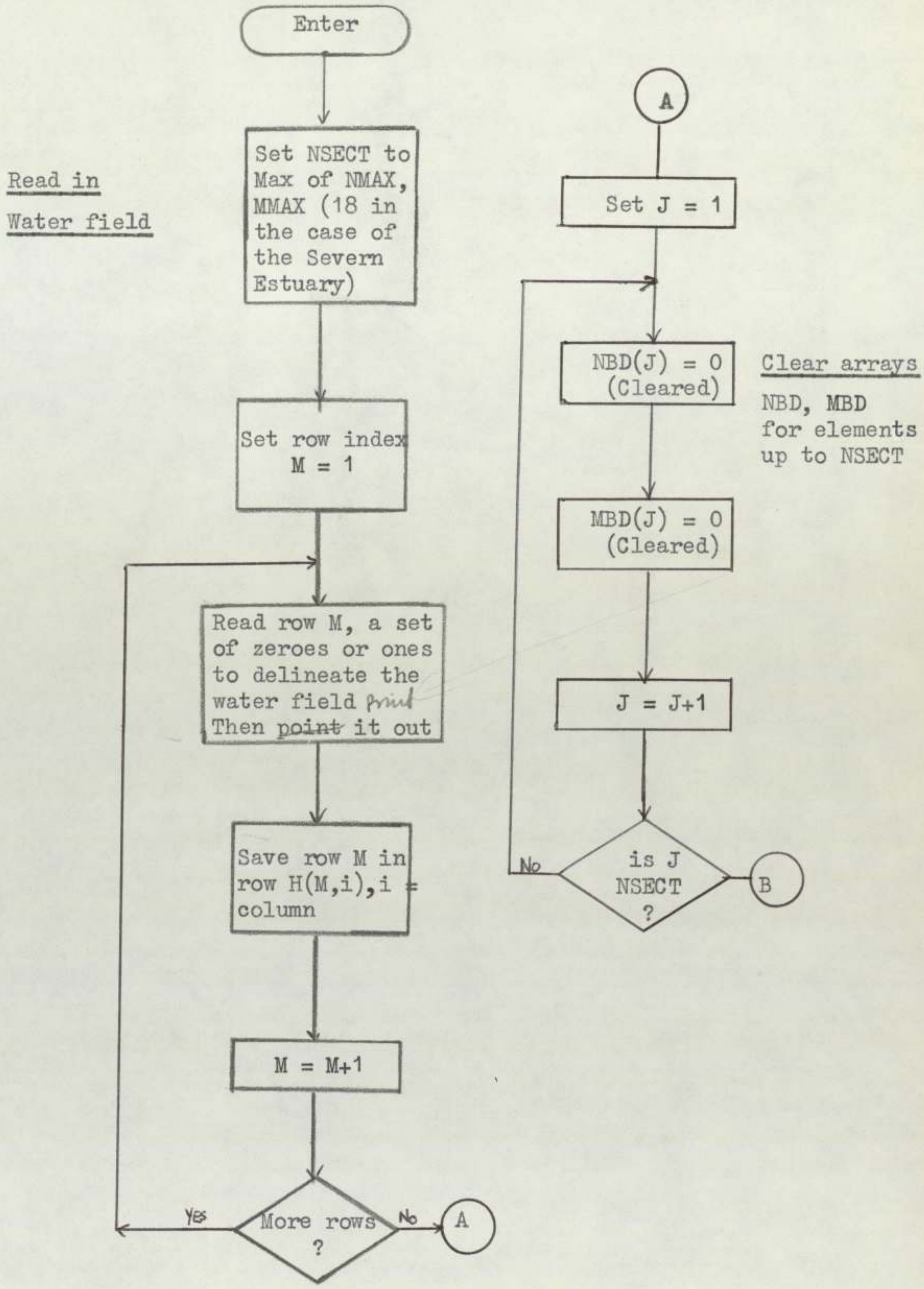
Entry
Multiple switch depending on IPWR 10
STOP 1111
I1 = IEL/10^4
I2 = IEL/10^2 - I1 * 10^2
I3 = IEL - I1 * 10^4 -I2 * 10^2
I1 = IEL/10^5
I2 = IEL/10^3 - I1 x 10^2
I3 = IEL/10 - I1 * 10^4 - I2 * 10^3 - I3 * 10
I4 = IEL - I1 * 10^5 - I2 * 10^3 - I3 * 10
I1 = IEL/10^6
I2 = IEL/10^4 -I1 * 10^2
I3 = IEL/10^2 -I1 *10^4 -I2*10^2
I4 = IEL-I1 *10^6 -I2*10^4 - I3*10^2
Return

B.12. **Subroutine FIND**
(for F2)

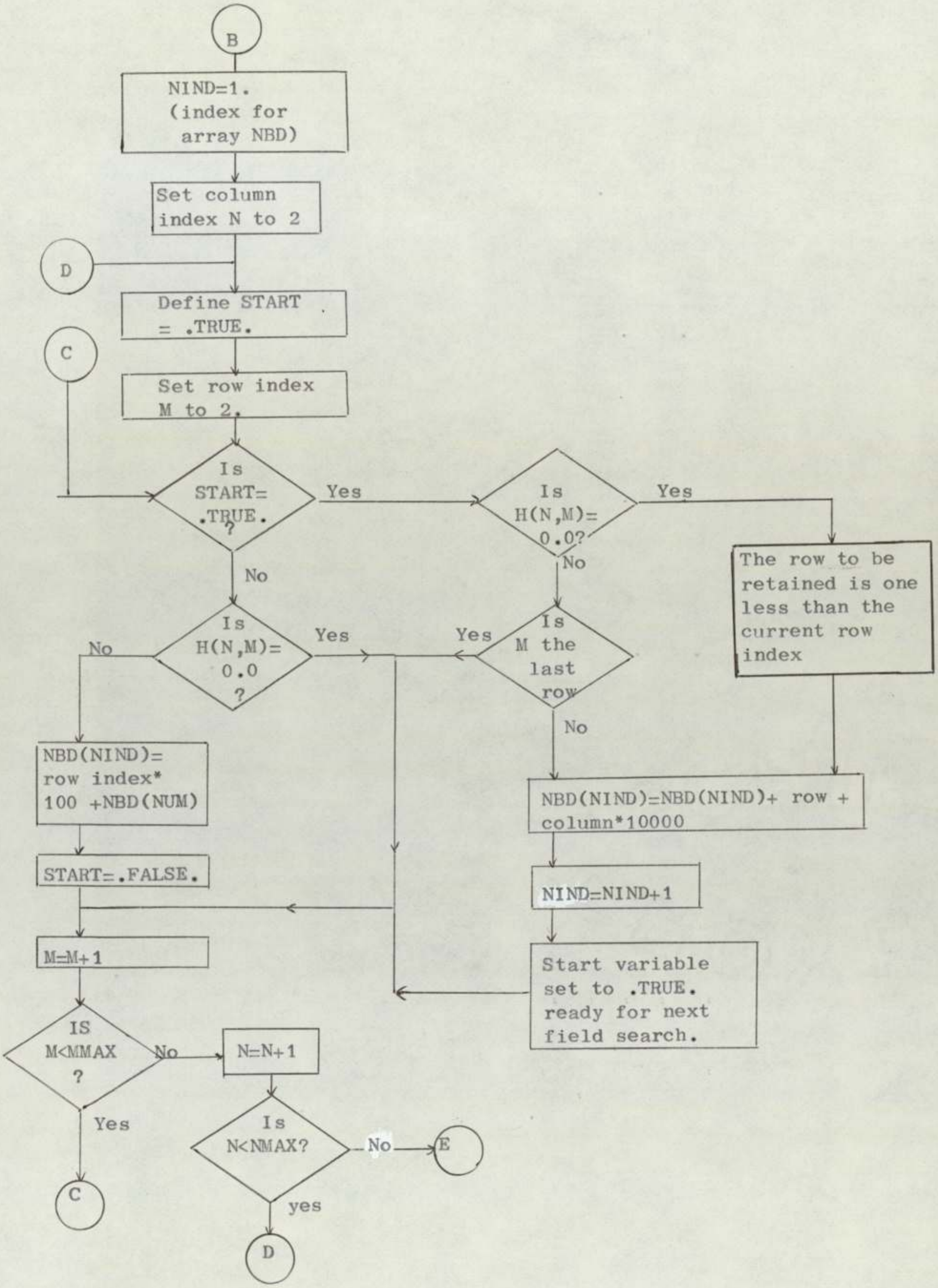
Function:	To set up boundary identifiers in row and column identifiers for the bay program.
System Functions:	MAXO, FLOAT, I/O
User Functions:	DIGITS
Calling Statement:	CALL FIND
Error Messages:	None
Common Areas:	/IO/, /CTCH22/
Common Variables Reset:	NBD, MBD, H (used as work area) of /CTCH22/
Description:	Each row and column of the grid will have specific geometry and so the Bay routine must be able to easily and quickly identify where the water field starts, ends and any other boundary conditions. The first DO loop reads in a set of 0 x 1's to identify whether grid points are in or out of the water field. DO 1 clears NBD & MBD arrays ready for calculations. For each column, NBD has retained start and end row of water field row through the DO 2 loop. The DO 3 loop duplicates the calculations for rows and array MBD.

These values now need to be updated for the column/row water boundaries and discharge arrays NQBD, MQBD, NOBD, MOBD.
For one column, find the relevant NBD number and find the start-row and end row of the water boundary.
For each of these arrays the constituents numbers are split and tested against the split values of NBD & MBD to see if any of these coincide.
Because the need for a row and column of zeroes all around the bay section, the arrays for discharges and water boundaries should have identification one less than the actual row and column (as they disregard the sets of zeroes all around the bay).
The values of NBD and MBD are printed for easy checking.

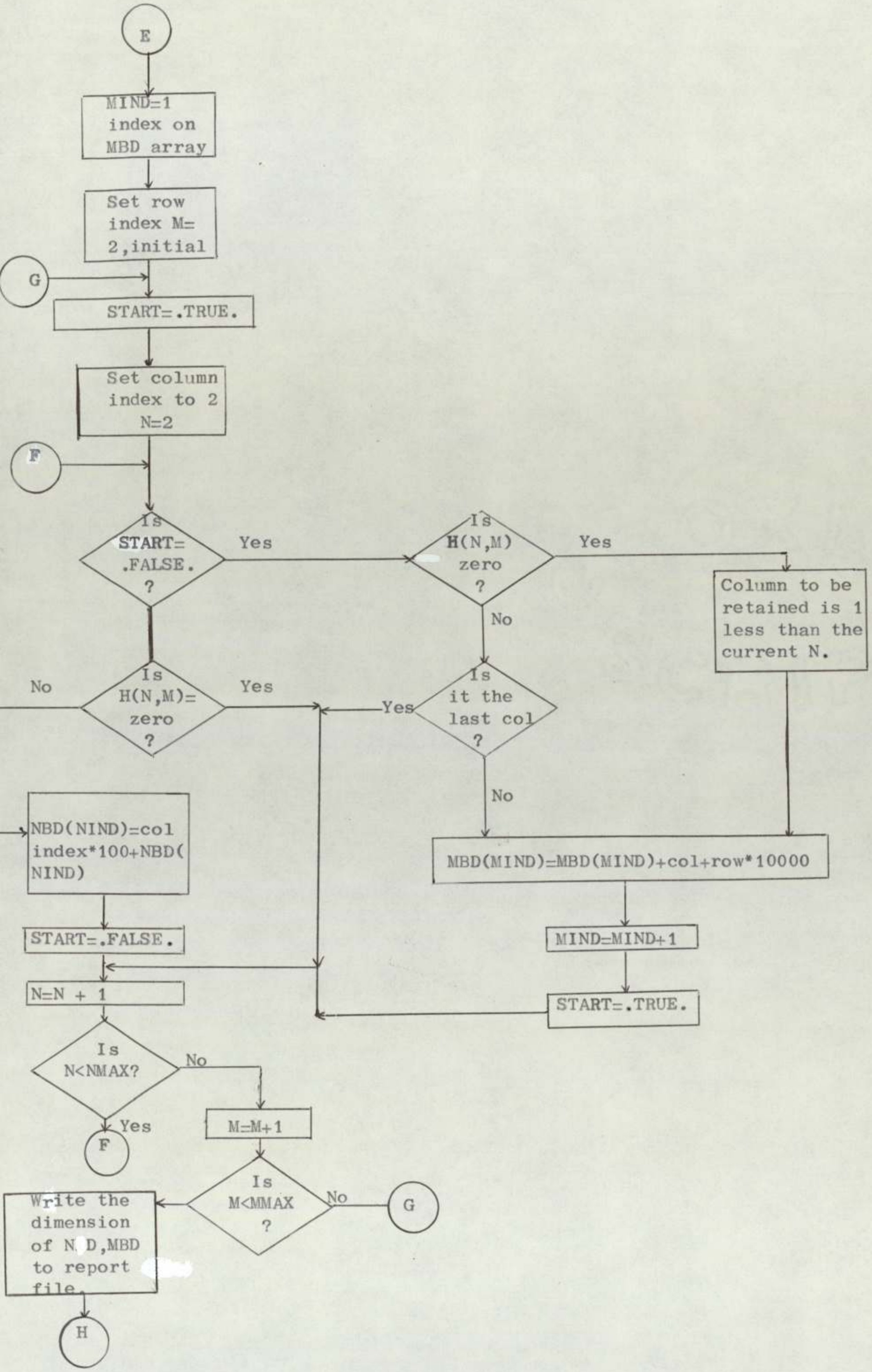
Flow Diagram



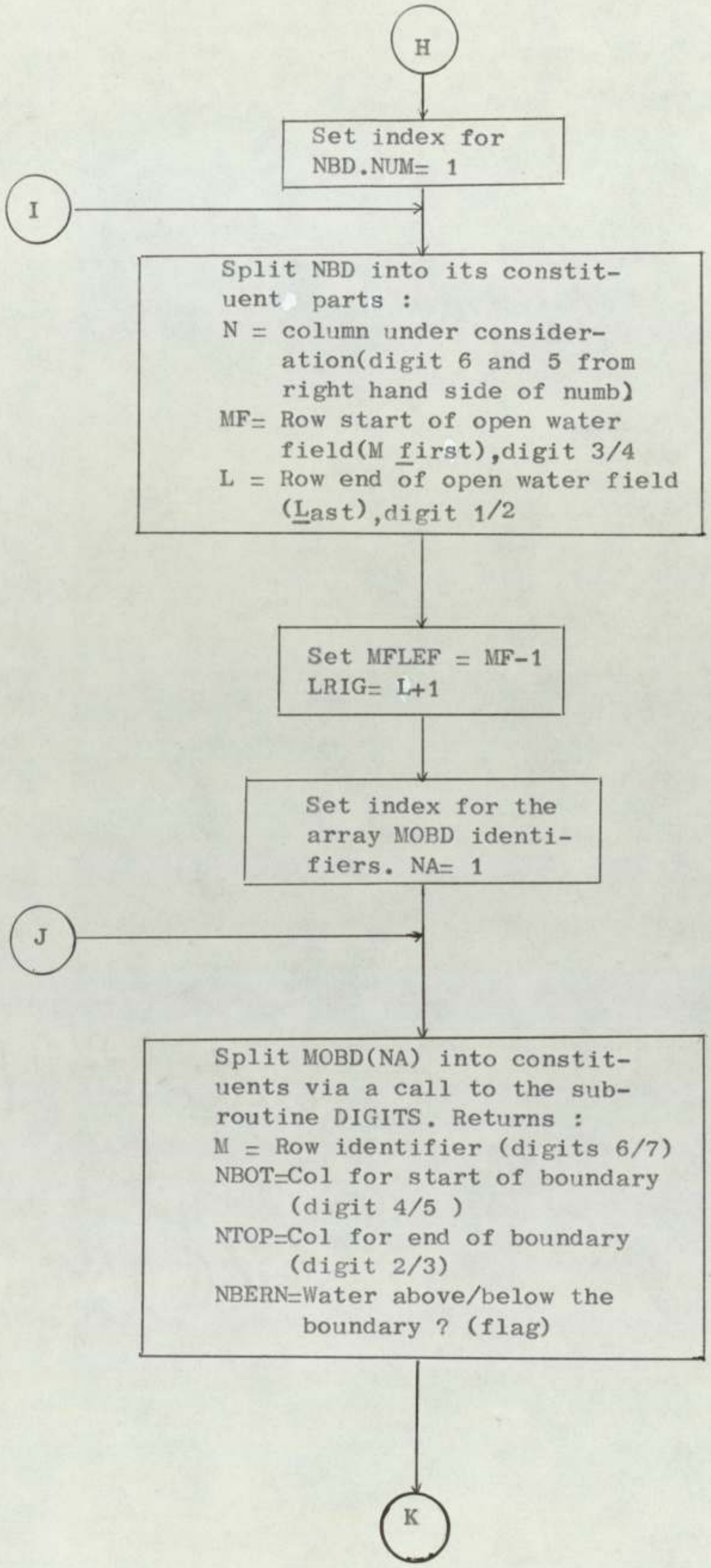
SET UP NBD - Part one.



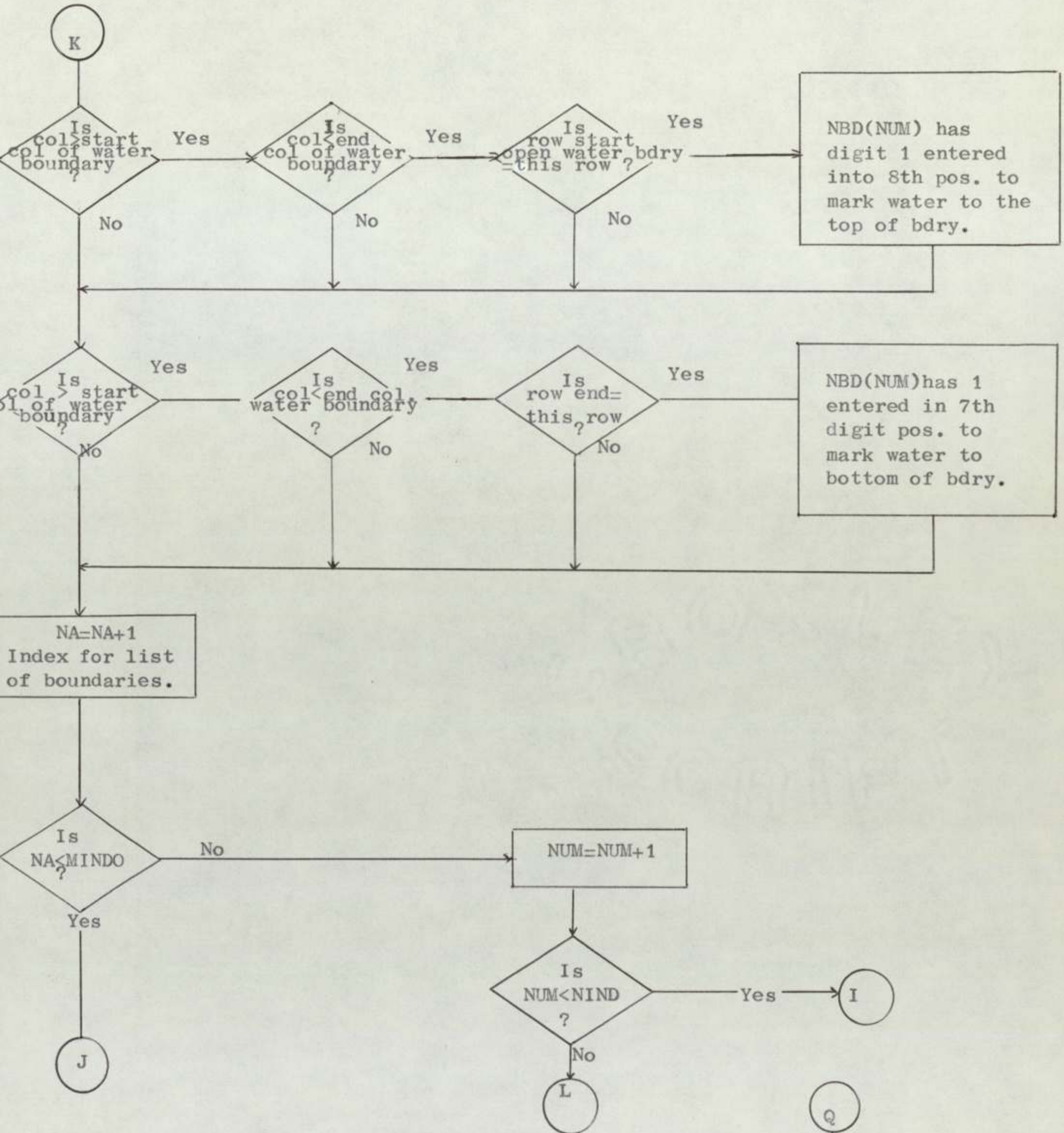
SETTING UP MBD - Part one

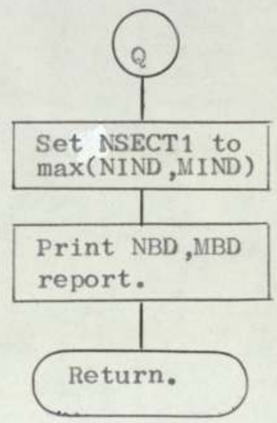


Update NBD and MBD for open water boundaries and discharge boundaries



Prepares to update NBD for Row open water boundaries.





Similar segments are coded for

MBD-NOBD Column open water boundaries L - M

NBD-MQBD Row discharge boundaries M - N

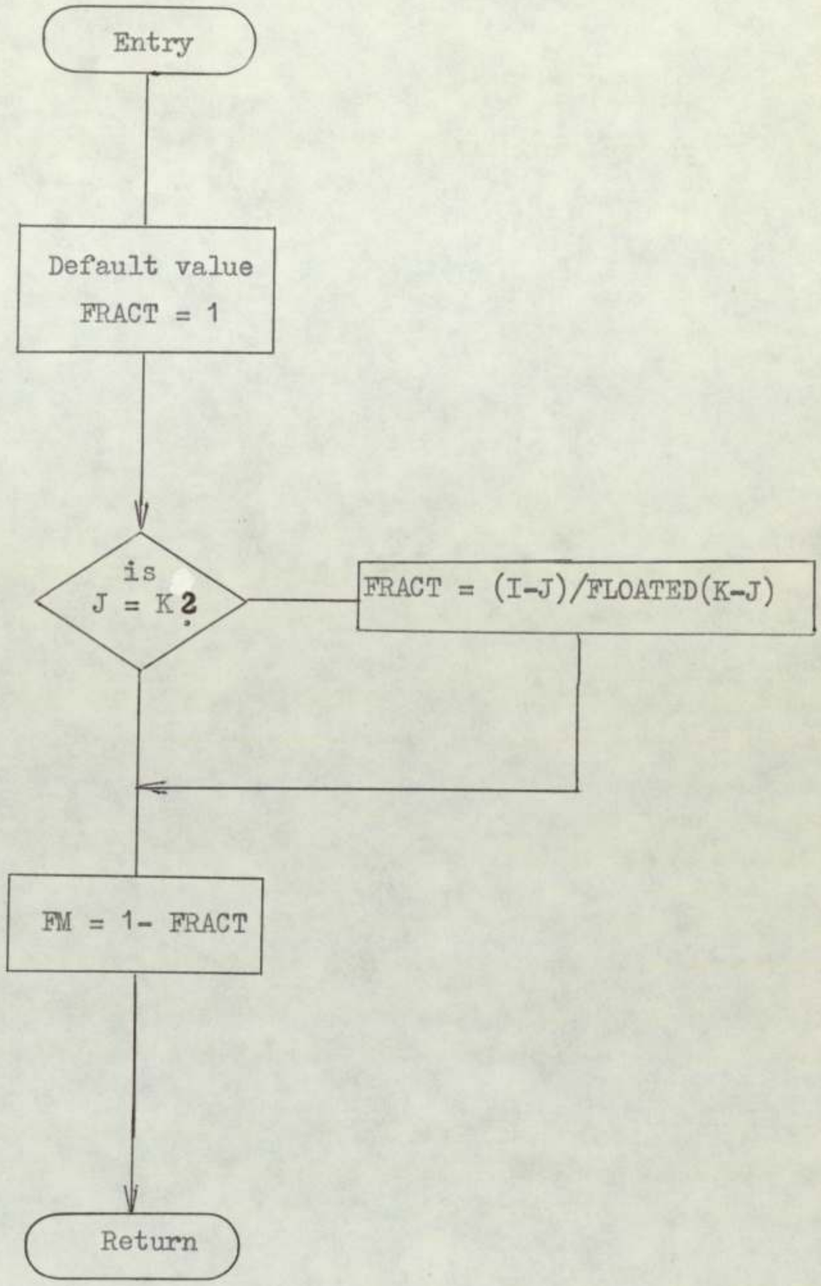
MBD-NQBD Column discharge boundaries N - Q

B.13. <u>Function FRACT</u>

(for F2)

Function:	To evaluate a proportion between two integers.
System Functions:	FLOAT
User Functions:	None
Calling Statement:	X = FRACT (I, J, K, FM)
Error Messages:	None
Common Areas:	None
Common Reset:	None
Description:	The function evaluates (I-J)/(K-J) and as this proportion is used extensively for linear interpolation between time steps it is set as a real function. A lot of CPU time can be gained if this were in machine code. If K = J the ratio is not defined and the function is set to unity. FM is always set to 1-(value of function)

Flow Diagram



B.14. Subroutine HYDONE

Function:	Controlling program for the River one dimensional phase.
System Functions:	READ/WRITE I/O, FLOAT,
User Functions:	DATA, CALHBC, CALEF, CALNOD, CALHBC, NODEMK
Error Messages:	None
Calling Statement:	CALL HYDONE (YIN, QOUT)
Common Areas:	/IO/, /LIMITS/, /INITDP/, /A/, /C/, /NODE/
Common Variables Reset:	GDT,G, G2, Y, U in /A/

Description:

This routine controls the calling sequence to the various programs in the river one dimensional phase.

The major part of this program is executed only once on the first preparatory call from executive program HYDROD.

DATA subroutine reads in tabular data about each segment as well as physical constants. This data is printed and some constants involving g (acceleration due to gravity) are calculated and stored.

An initial set of values of Y are calculated. For each segment, the downstream height is compared to the upstream height and a distance-height proportionality factor (AVQ) is calculated and Y (Seg M, grid point N) = Y (Seg M, grid point 1) + AVQ * (distance grid point N to grid point 1).

If the segment is terminal, a similar calculation is made.

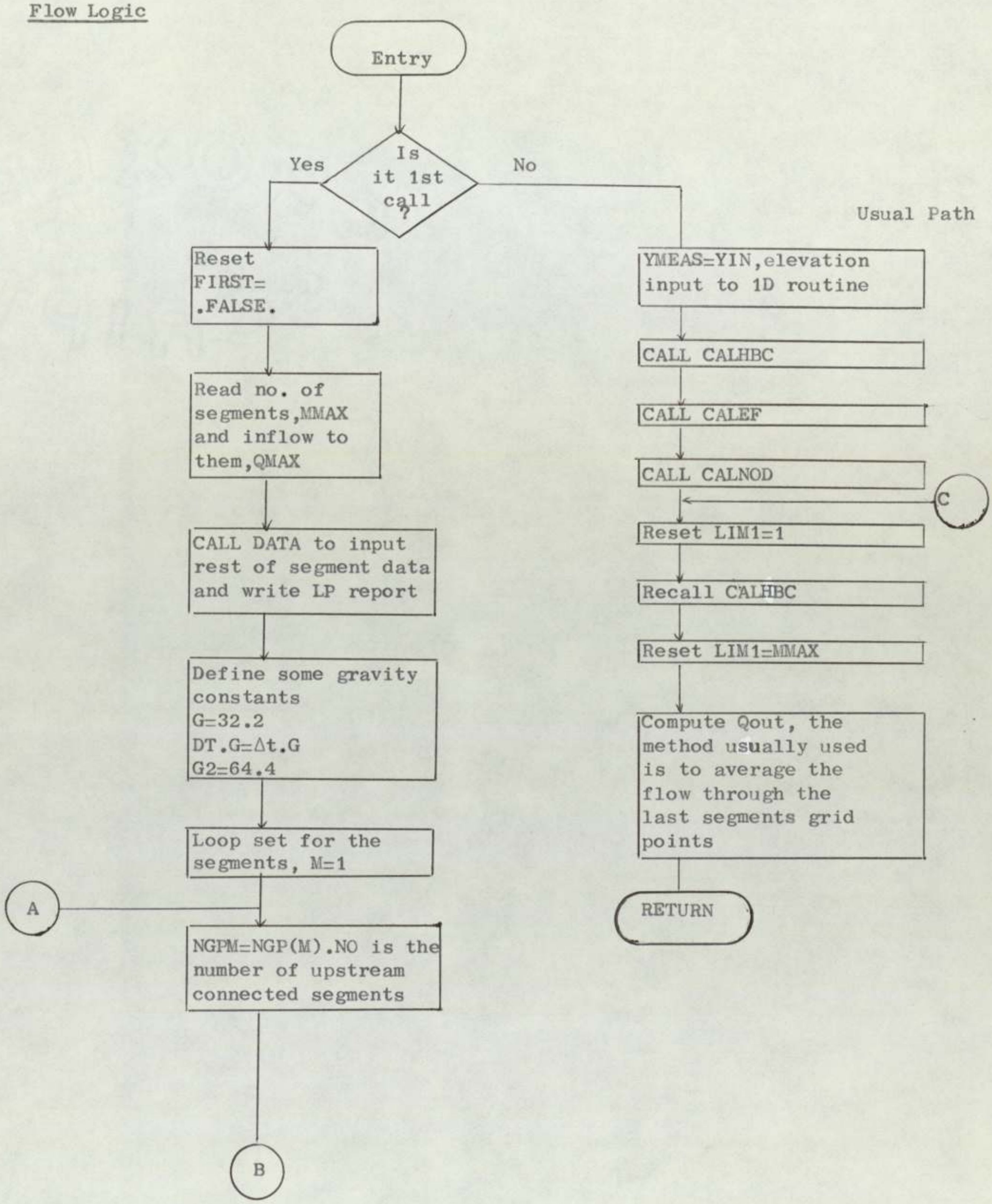
With this data, an initial calculation of H, B & C is made.

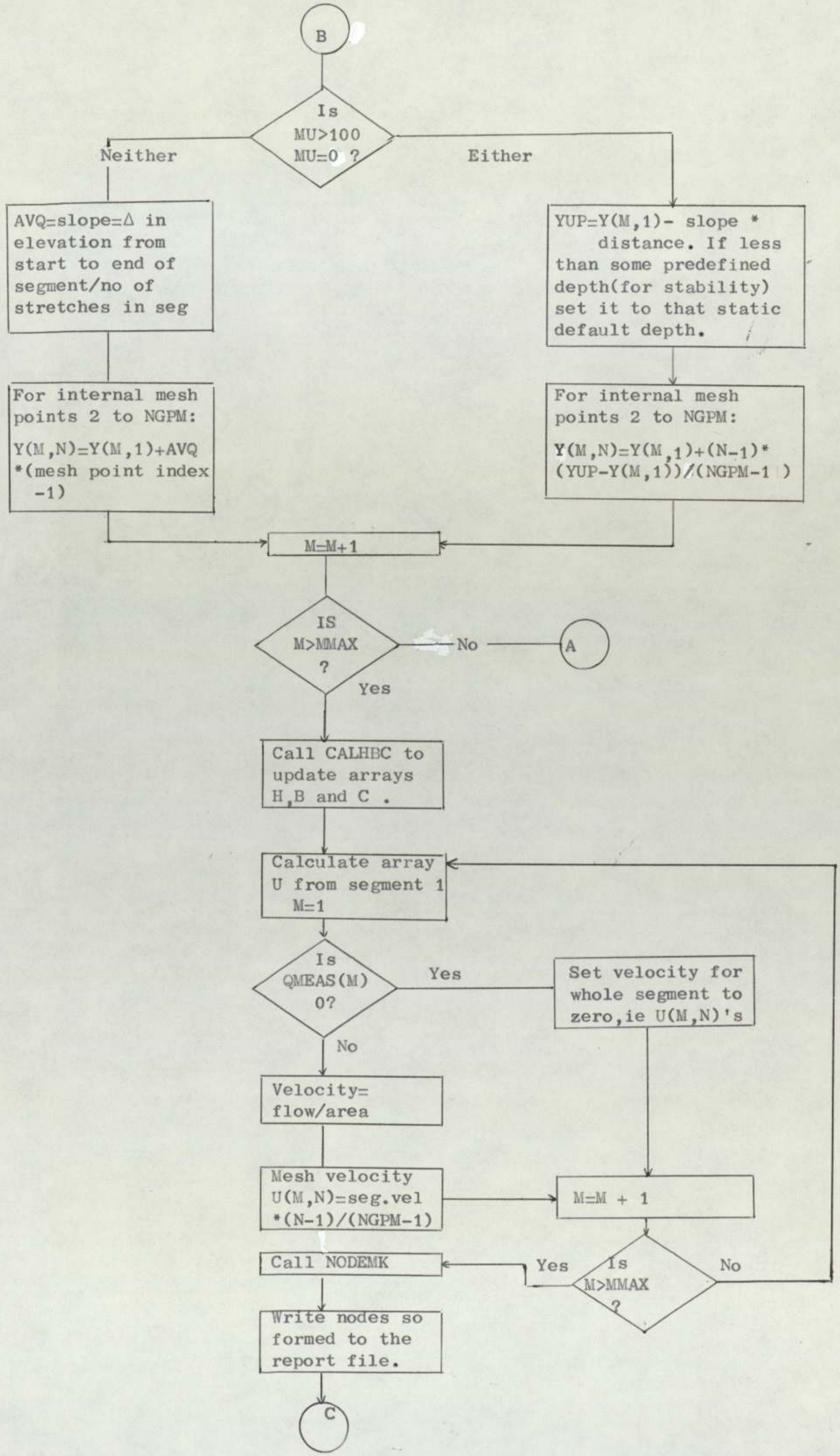
An initial set of values of U are calculated. For each segment, if the upstream discharge is zero (QMEAS) then the velocity is set to zero. If it is non zero, the velocity is approximated by flow/area and then scaled to each grid point.

Routine NODEMK is called to construct node identifiers. Then CALHBC is called only for segment MMAX, the segment with the 1D-2D interface where relevant. This is accomplished by setting LIM1 to MMAX1. After the call, LIM1 is set back to 1.

Statement label 200 is where a normal entry to HYDONE begins YMEAS is set to the input height YIN. Then a time step one dimensional computation is carried out through calling CALHBC, CALEF, CALNOD & CALBC again. QOUT is then calculated from the calls of these programs.

Flow Logic


Entry
Is it 1st call ?
Yes
No
Usual Path
Reset FIRST= .FALSE.
Read no. of segments,MMAX and inflow to them,QMAX
CALL DATA to input rest of segment data and write LP report
Define some gravity constants
G=32.2
DT.G=Δt.G
G2=64.4
Loop set for the segments, M=1
A
NGPM=NGP(M).NO is the number of upstream connected segments
B
YMEAS=YIN,elevation input to 1D routine
CALL CALHBC
CALL CALEF
CALL CALNOD
C
Reset LIM1=1
Recall CALHBC
Reset LIM1=MMAX
Compute Qout, the method usually used is to average the flow through the last segments grid points
RETURN


B
Is
MU>100
MU=0 ?
Neither
Either
AVQ=slope=Δ in elevation from start to end of segment/no of stretches in seg
YUP=Y(M,1)- slope * distance. If less than some predefined depth(for stability) set it to that static default depth.
For internal mesh points 2 to NGPM:
Y(M,N)=Y(M,1)+AVQ *(mesh point index -1)
For internal mesh points 2 to NGPM:
Y(M,N)=Y(M,1)+(N-1)*(YUP-Y(M,1))/(NGPM-1)
M=M+1
IS
M>MMAX
?
No
A
Yes
Call CALHBC to update arrays H,B and C .
Calculate array U from segment 1 M=1
Is
QMEAS(M)
0?
Yes
Set velocity for whole segment to zero,ie U(M,N)'s
No
Velocity= flow/area
Mesh velocity U(M,N)=seg.vel *(N-1)/(NGPM-1)
M=M + 1
Call NODEMK
Yes
Is
M>MMAX
?
No
Write nodes so formed to the report file.
C

B.15. Executive Program HYDROD

(for F1)

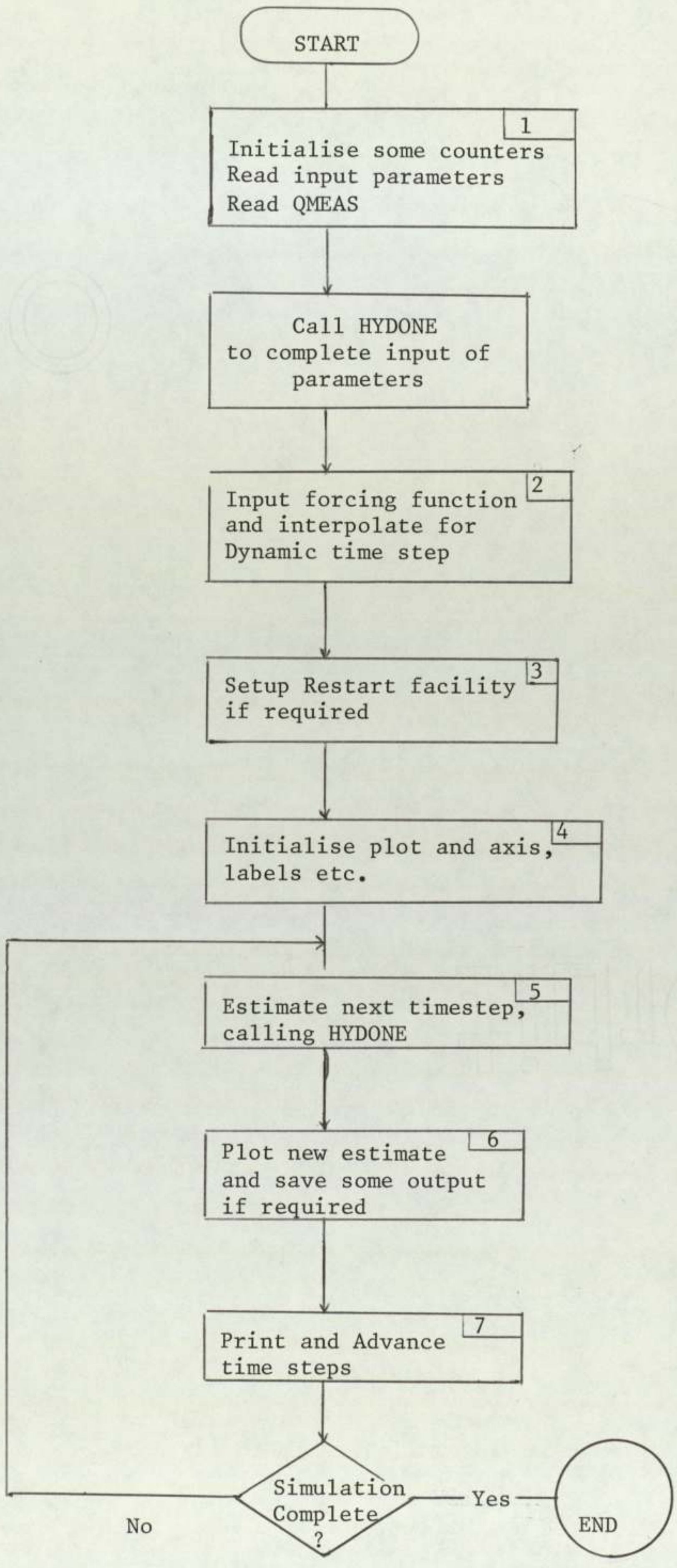
Function:	Main controlling program for 1 dimensional model
System Functions:	I/O, TIME, GRAFOR, MOD
User Functions:	REST (version for F1), INTP, HYDONE, QT, UPTIME, PRINT1
Error Messages:	-
Common Areas	/IO/, /A/, /NODE/, /INITDP/, /COR/, blank
Common Variables Reset:	NOUT in /IO/; DT,MMAX1, QMEAS in /A/ YA1 in /INITDP/ and UA1 in blank area
Description:	As an executive routine it mainly contains controlling parameters, counters and switches. The graphical output is controlled from this routine, also the tidal forcing function is prepared here for use in the routine HYDONE.
	Initially, run parameters are read in: switches set and options and scope of the problem.
	The first subroutine call is to QT if time varying flow input is required.
	Routine HYDONE is called to enable completion of the main input parameters for the one dimensional phase.
	The array XSE is input as the tidal forcing function. The model will loop in time

steps of one minute or so, and input data off charts are usually available in time steps of 15-60 minutes. As a simulation will run for several tidal cycles, the input XSE is interpolated and saved in a large array YSE which is large enough to accommodate a whole tidal cycle for time steps of as small as ½ minute. If RT, the restart time, is non zero, routine REST is called to attempt to locate the restart time.
The appropriate starting place is located in YSE, and if necessary some parameters are written to the output stream. If required, the graph plot is prepared and the forcing function plotted in its entirety.
The main calculation loop starts at statement number 1ØØ. Before the river calculation routine HYDONE, the current water height/velocities are saved in YA1 and UA1 with some optional tests for turning tides.
If flows are variable, they are updated for the current time and then HYDONE calculates new values for water height and velocities. The boundary values are saved. UPTIME is called to see if this

cycle is to be printed and next print time updated.

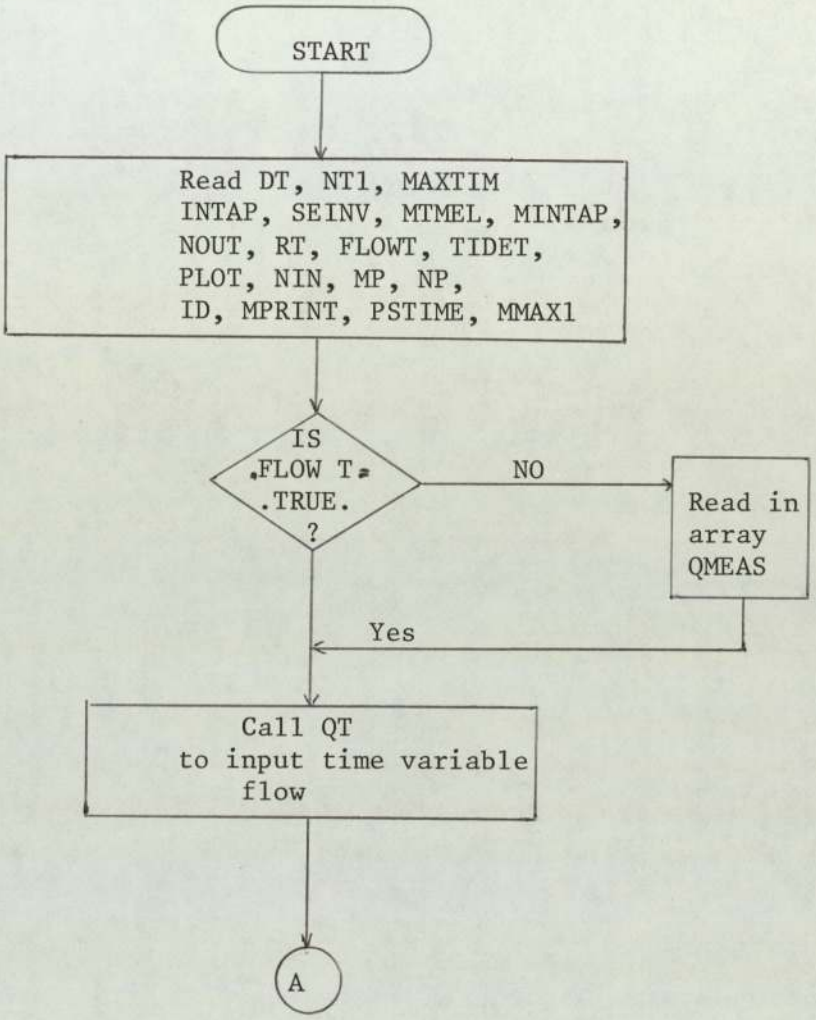
Loop parameters are updated and control returns to the main loop start if the simulation is incomplete.

Outline Flow Logic

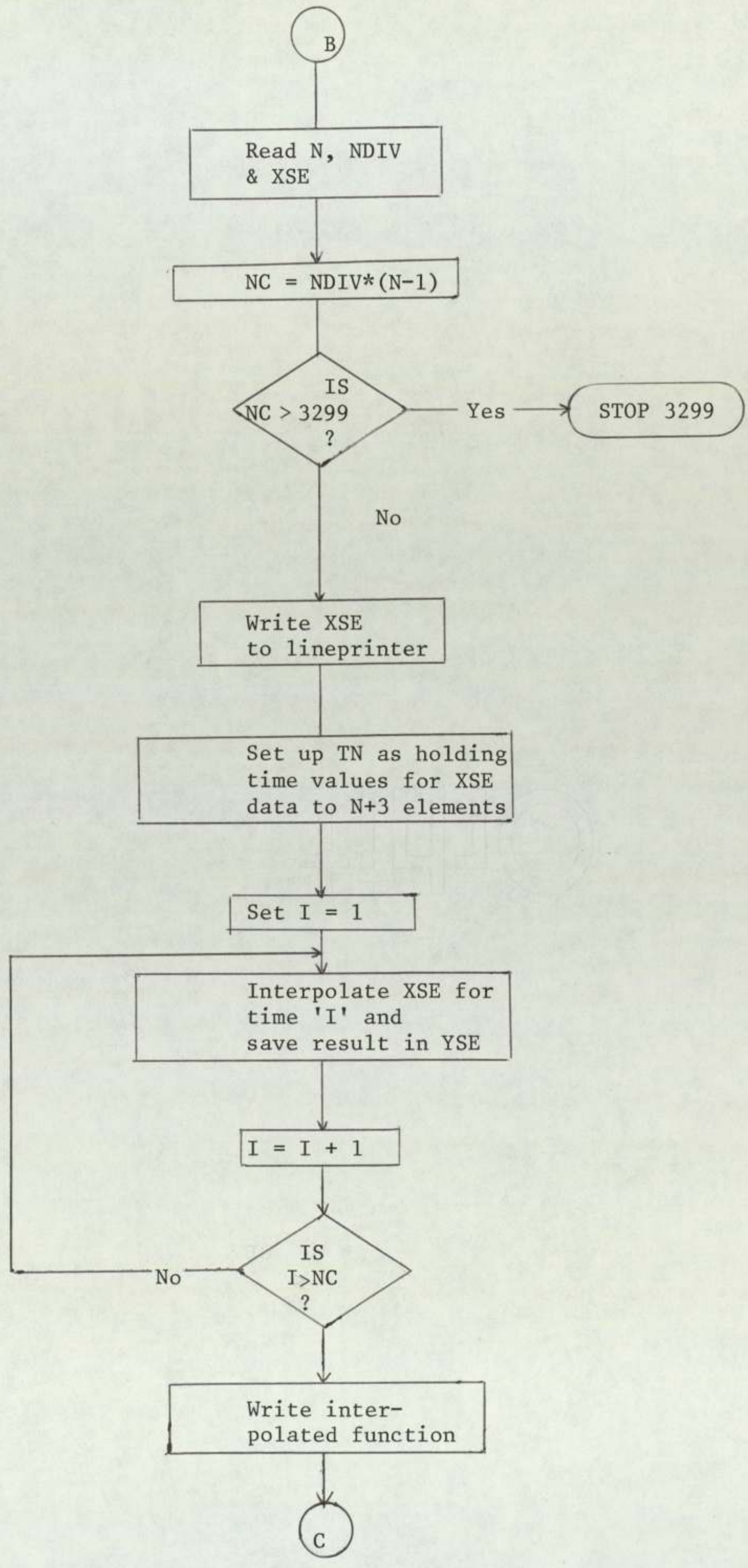


Detailed Flow Logic

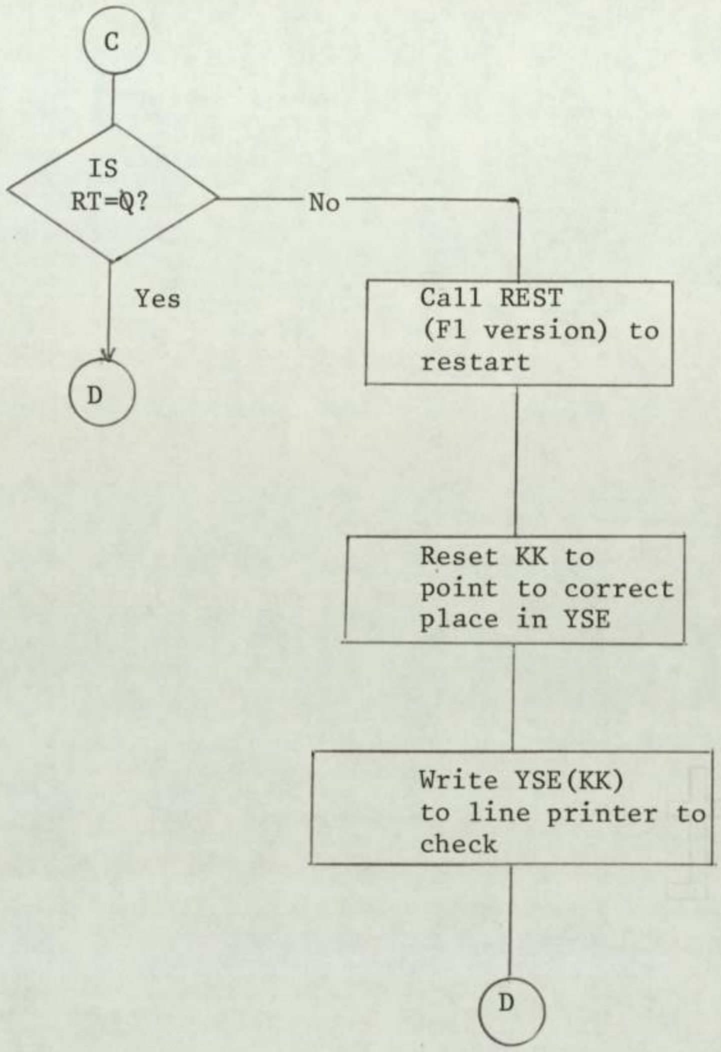
Box 1



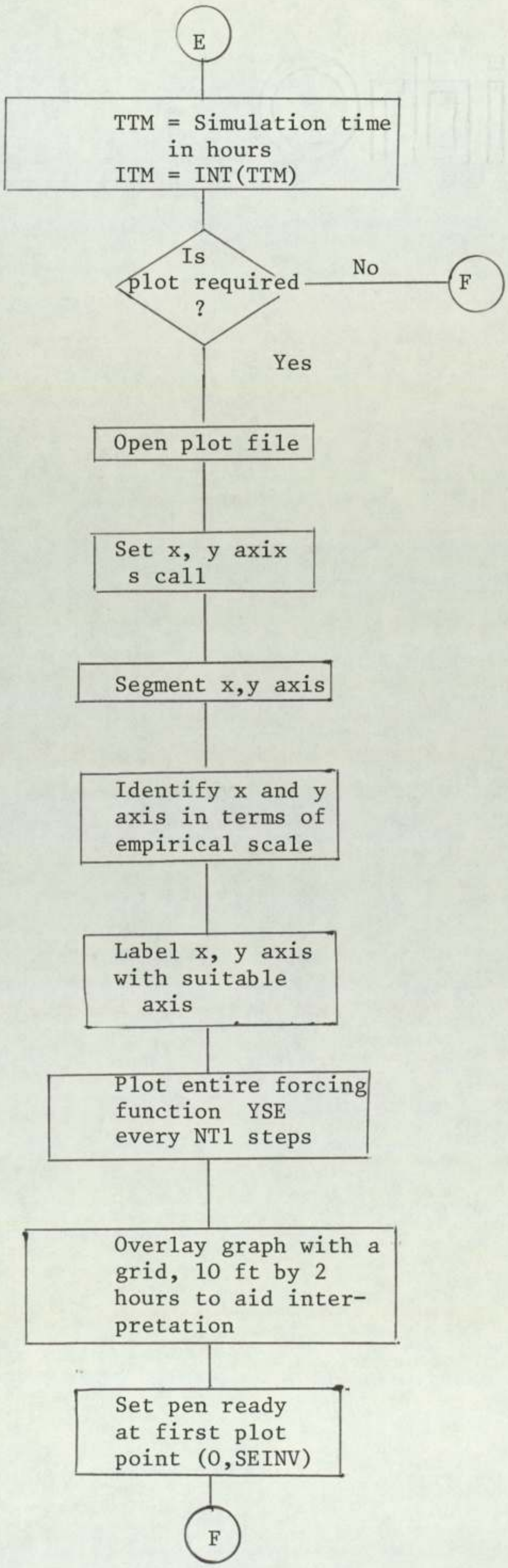
Box 2


B
Read N, NDIV & XSE
NC = NDIV*(N-1)
IS NC > 3299 ?
Yes
STOP 3299
No
Write XSE to lineprinter
Set up TN as holding time values for XSE data to N+3 elements
Set I = 1
Interpolate XSE for time 'I' and save result in YSE
I = I + 1
IS I>NC ?
No
Write interpolated function
C

Box 3

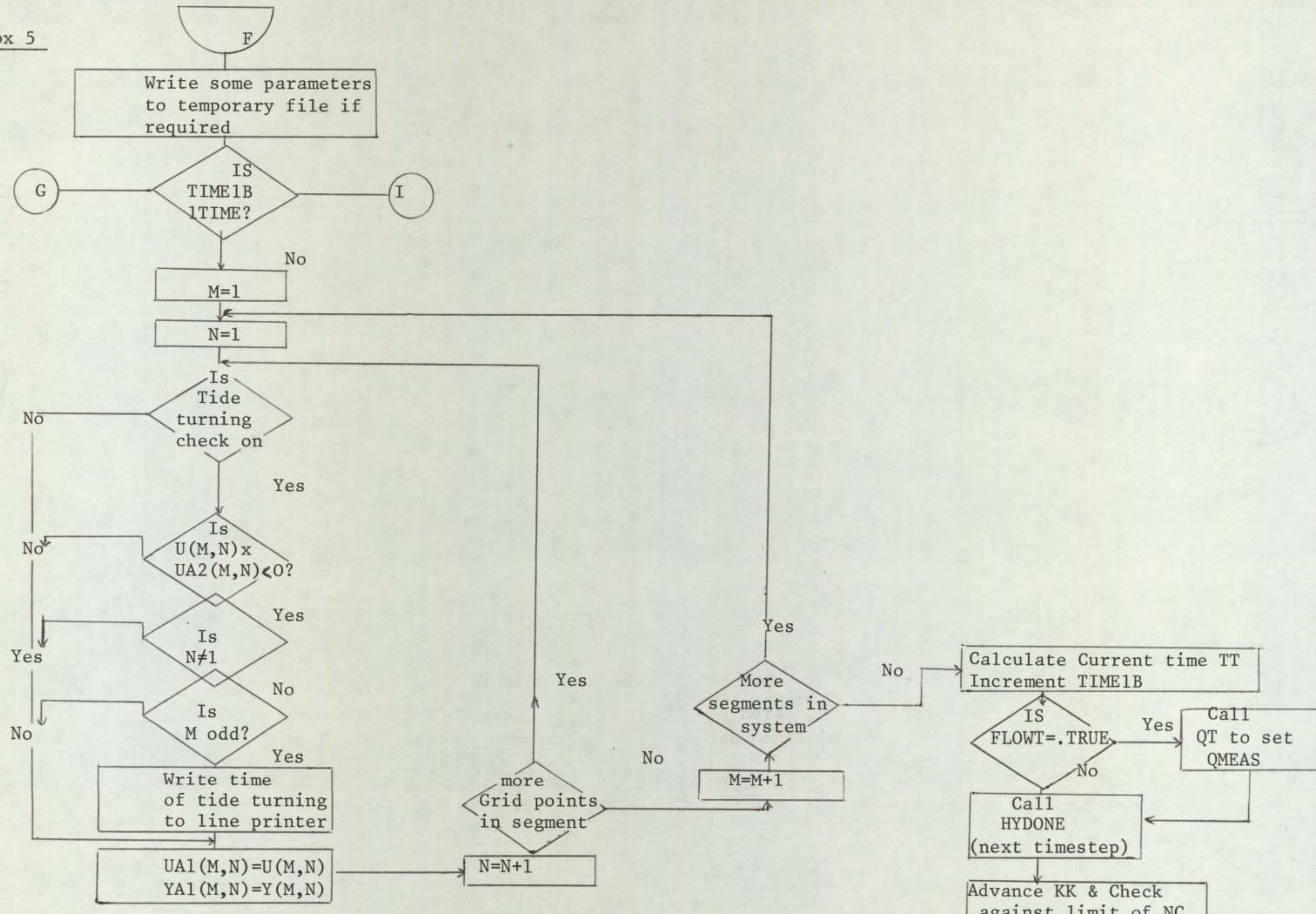

C
IS
RT=Q?
No
Yes
D
Call REST
(Fl version) to
restart
Reset KK to
point to correct
place in YSE
Write YSE(KK)
to line printer to
check
D

Box 4

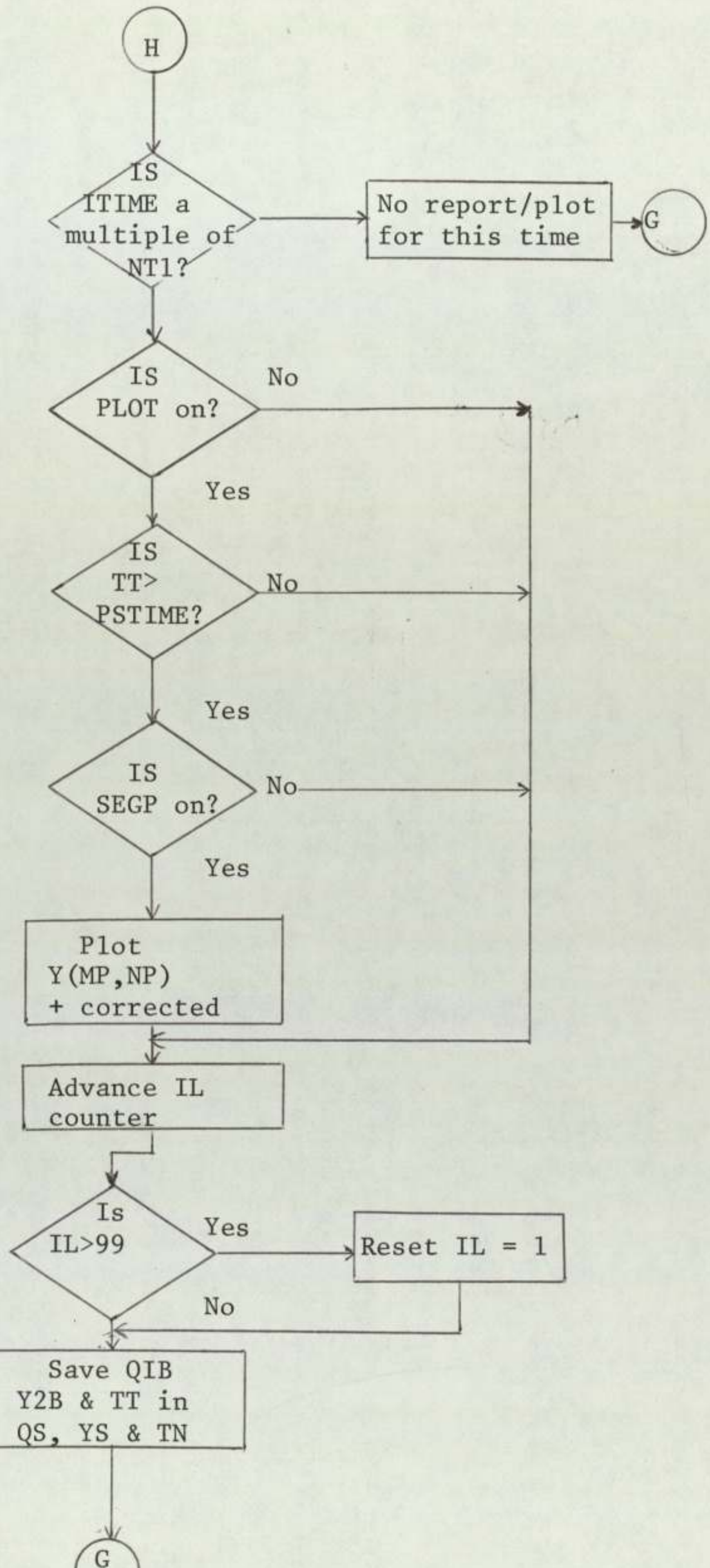


E
TTM = Simulation time in hours
ITM = INT(TTM)
Is plot required ?
No
F
Yes
Open plot file
Set x, y axix s call
Segment x,y axis
Identify x and y axis in terms of empirical scale
Label x, y axis with suitable axis
Plot entire forcing function YSE every NT1 steps
Overlay graph with a grid, 10 ft by 2 hours to aid interpretation
Set pen ready at first plot point (0,SEINV)
F

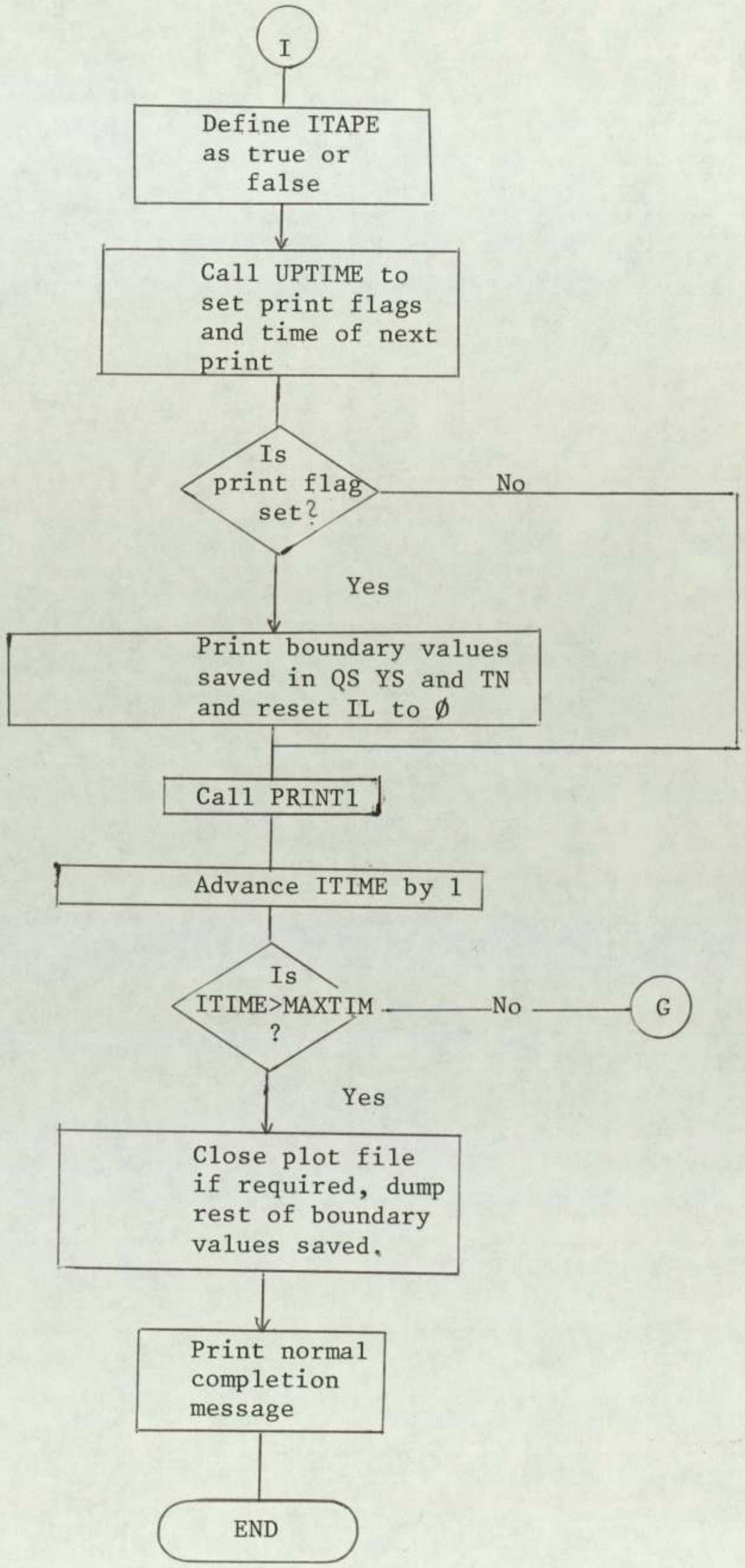
Box 5


F
Write some parameters to temporary file if required
G
IS TIME1B 1TIME?
I
No
M=1
N=1
Is Tide turning check on
No
Yes
Is U(M,N)x UA2(M,N)<0?
No
Yes
Is N≠1
Yes
No
Is M odd?
No
Yes
Write time of tide turning to line printer
UA1(M,N)=U(M,N)
YA1(M,N)=Y(M,N)
N=N+1
more Grid points in segment
Yes
No
M=M+1
More segments in system
Yes
No
Calculate Current time TT
Increment TIME1B
IS FLOWT=.TRUE.
Yes
No
Call QT to set QMEAS
Call HYDONE (next timestep)
Advance KK & Check

Box 6


H
IS ITIME a multiple of NT1?
No report/plot for this time
G
IS PLOT on?
No
Yes
IS TT> PSTIME?
No
Yes
IS SEGP on?
No
Yes
Plot Y(MP,NP) + corrected
Advance IL counter
Is IL>99
Yes
Reset IL = 1
No
Save QIB Y2B & TT in QS, YS & TN
G

Box 7


I
Define ITAPE as true or false
Call UPTIME to set print flags and time of next print
Is print flag set?
No
Yes
Print boundary values saved in QS YS and TN and reset IL to Ø
Call PRINT1
Advance ITIME by 1
Is ITIME>MAXTIM ?
No
G
Yes
Close plot file if required, dump rest of boundary values saved.
Print normal completion message
END

B.16. Executive Program HYDROD

(for F2)

Function:	Main controlling program for multi-dimensional hydrodynamic phase of Model
System Functions:	WRITE/READ I/O, TIME, MINØ
User Functions:	HYDONE, BOTANY, FRACT, UPTIME, PRINT 1, PRINT 2, COPYDT, IPLAUS, ADDATE
Error Messages:	None
Common Areas:	/IO/, /A/, /NODE/, /CTCH 22/, /SPECS/, /INITDP/, /TIM/, /BAYD/, /DATE/, blank common
Common Variables Reset:	Most of problem parameters are read in here. Arrays UA1, YA1, UA, VA of the blank area and Arrays Y, U, SEP, VP, UP of /A/ are reset.
Description:	This is the overall executive program for phase one of the FISCHER Model. Input here are problem parameters, water boundary identifiers and time of print/plot/tape parameters. Up to statement labelled 100 all basic inputs are made with initialising calls to routines HYDONE & BOTANY. From 100 to 260 is a river run, from 150 to 200 is a bay run. Then the options for output are tested. ITAPE is set to true every INTAP cycles, if INTAP is not zero and the time is greater than MINTAP number of time cycles.

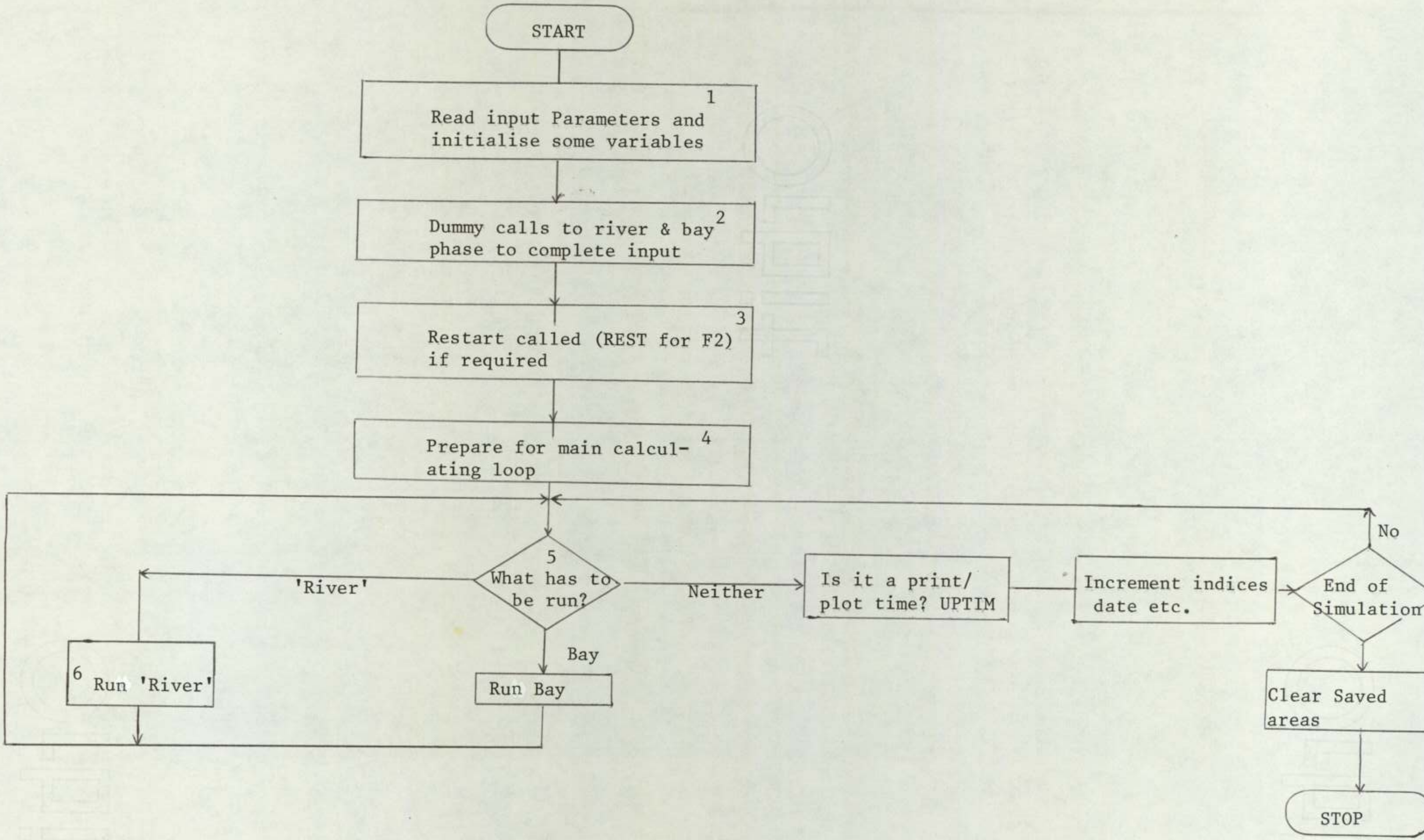
Description Contd: As the print time will possibly not coincide with either the most recent or second most recent run time of river and bay, linear interpolation is carried out between these two values to determine data for the correct time. The data for the most recent run is also pushed down to second most recent data in anticipation of the next cycle. This is programmed once for the river and once for the bay phase as different arrays are involved.

PRINT 1 is called for river data output, PRINT 2 for bay data.

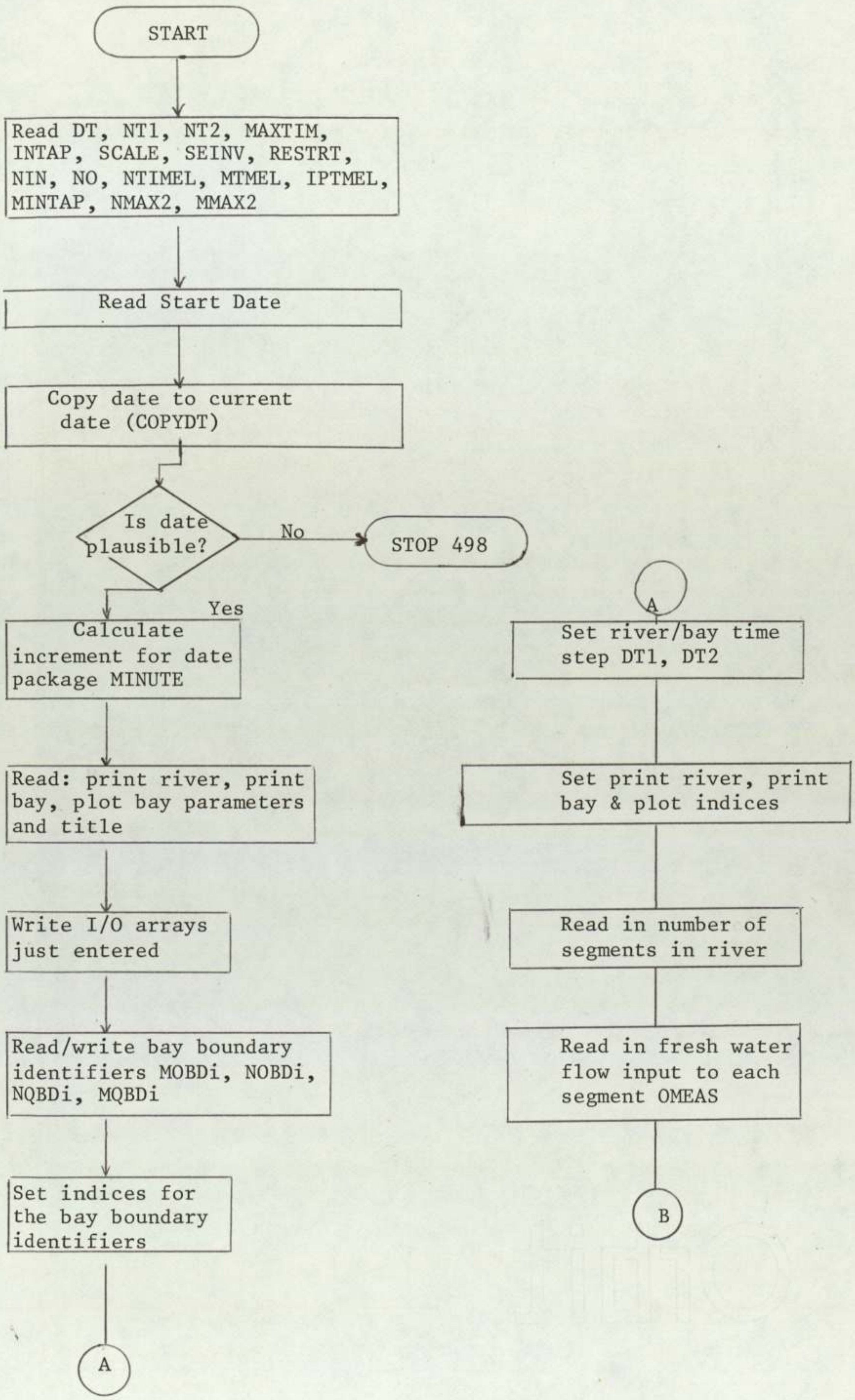
When the I/O section has been executed, the time cycle index is updated and tested against maximum number of cycles required for the run. If less than the upper limit, control jumps to label 100 to test whether bay or river are due for a run.

If EOJ is reached a terminal message is output and program execution ends.

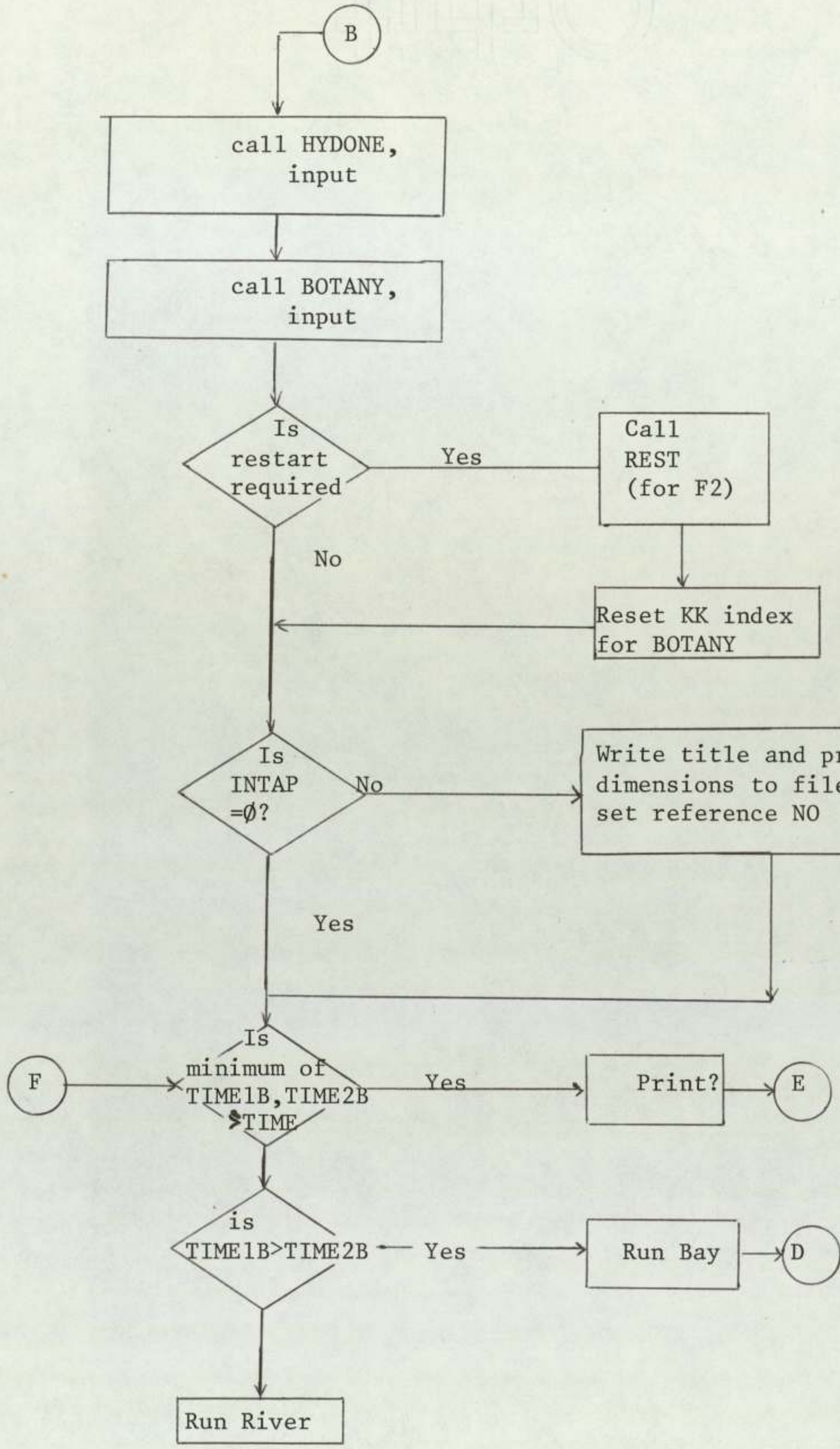
Outline Flow Logic



Box 1


START
Read DT, NT1, NT2, MAXTIM, INTAP, SCALE, SEINV, RESTRT, NIN, NO, NTIMEL, MTMEL, IPTMEL, MINTAP, NMAX2, MMAX2
Read Start Date
Copy date to current date (COPYDT)
Is date plausible?
No
STOP 498
Yes
Calculate increment for date package MINUTE
Read: print river, print bay, plot bay parameters and title
Write I/O arrays just entered
Read/write bay boundary identifiers MOBDi, NOBDi, NQBDi, MQBDi
Set indices for the bay boundary identifiers
A
A
Set river/bay time step DT1, DT2
Set print river, print bay & plot indices
Read in number of segments in river
Read in fresh water flow input to each segment OMEAS
B

Box 2 to 5


B
call HYDONE, input
call BOTANY, input
Is restart required
Yes
Call REST (for F2)
No
Reset KK index for BOTANY
Is INTAP =Ø?
No
Write title and problem dimensions to file data set reference NO
Yes
F
Is minimum of TIME1B,TIME2B ≥TIME
Yes
Print?
E
is TIME1B>TIME2B
Yes
Run Bay
D
Run River

C

Box 6

Run River

Begin River Run
F2 = FRACT(TIME1B, TIME2A, TIME2B, F2C)
i.e.
F2 = { t from most recent 1D run to 2nd most recent 2D phase run / t between 2 two D phase runs }

F2C = 1-F2

YIN (depth of flow above channel at 1D-2D interface) =
(1-F2) Y2A + F2 * Y2B

Copy last values of V & Y to next to last arrays
UA1 ( ) = U ( )
YA1 ( ) = Y ( )

2nd most recent flow at interface = most recent flow
in 1D-2D I/F Q1A = Q1B

Reset: Second most recent time =
most recent time

Most recent time = most recent time +
increments for river time step

Call
HYDONE(Y-1N,Q1B)
(1-D phase)

F

Box 7

Run Bay

Begin Bay Run
F1 = FRACT(TIME2B, TIME1A, TIME1B, F1C)
i.e.
F1 = t from most recent 2D run to 2nd most recent 1D run / between 2 one D phase runs

F1C = 1-F1

QIN (Flow at 1D-2D interface =
(1-F1) * Q1A + F1 * Q1B

Copy last values of SE, U & V to
2nd most recent
SEA ( ) = SEB ( )
UA ( ) = UB ( )
VA ( ) = VB ( )

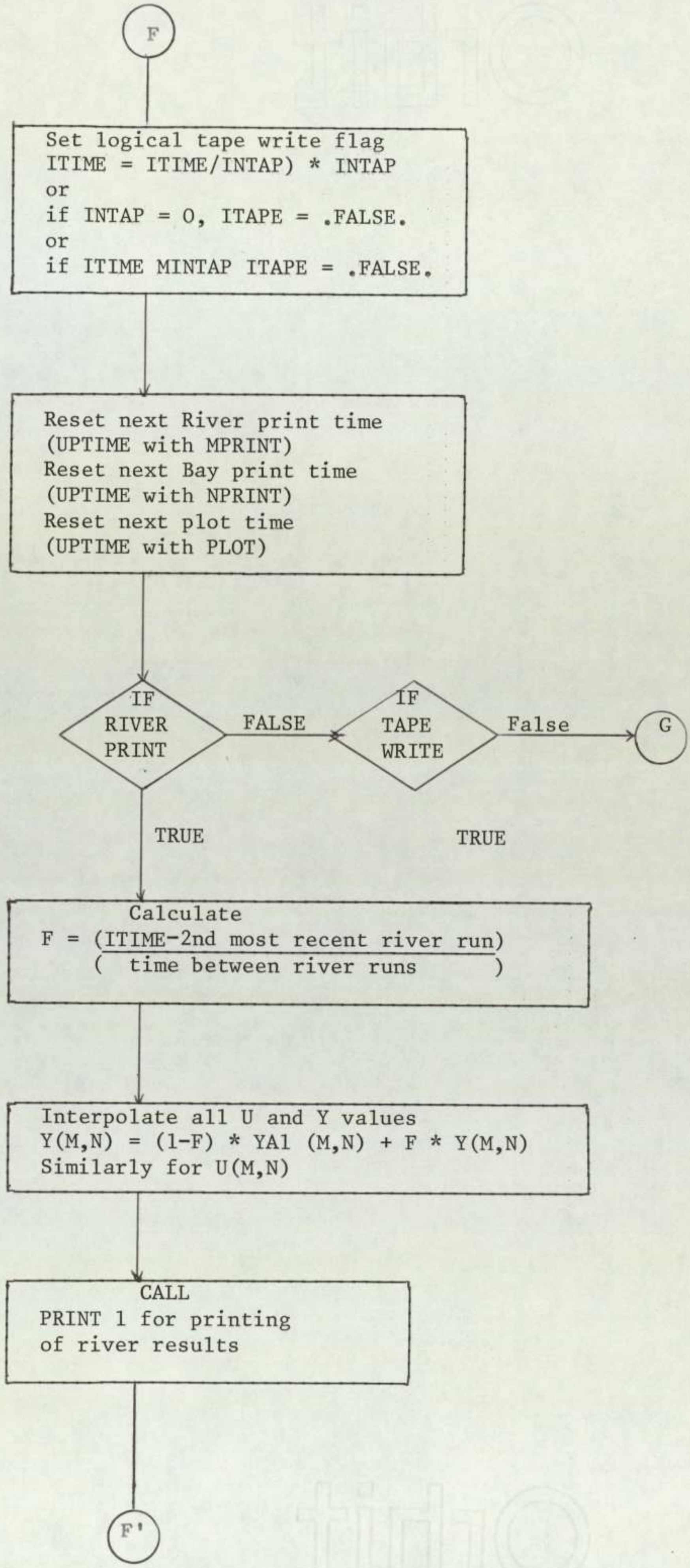
2nd most recent height = most
recent height at interface
YA2 = Y2B

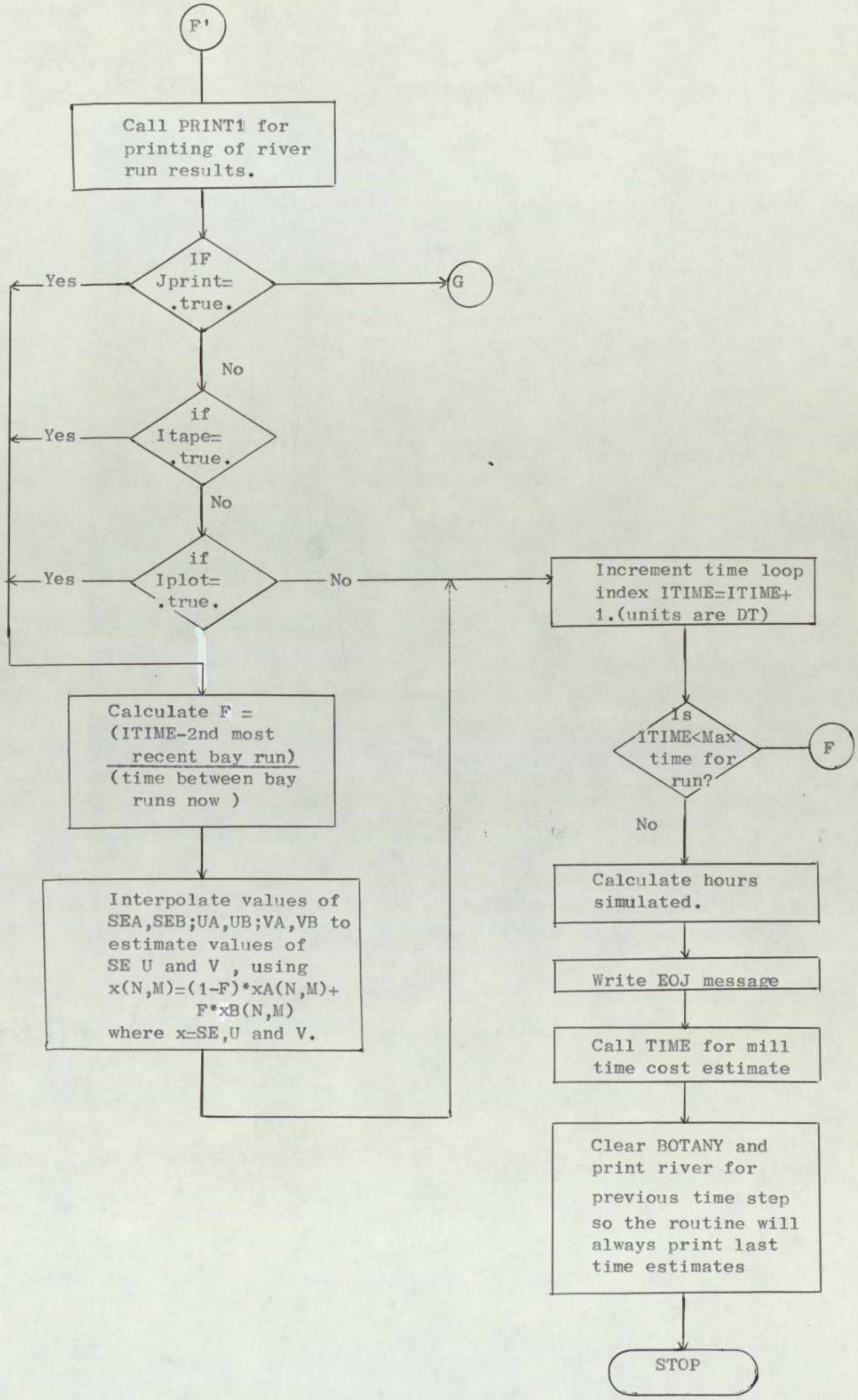
Reset: 2nd most recent 2nd D phase
time = most recent 2-D phase time

Most recent time 2 most recent time +
bay time step increment

Call
BOTANY(QIN,Y2B)
(2D phase)

F


F
Set logical tape write flag
ITIME = ITIME/INTAP) * INTAP
or
if INTAP = 0, ITAPE = .FALSE.
or
if ITIME MINTAP ITAPE = .FALSE.
Reset next River print time
(UPTIME with MPRINT)
Reset next Bay print time
(UPTIME with NPRINT)
Reset next plot time
(UPTIME with PLOT)
IF RIVER PRINT
FALSE
IF TAPE WRITE
False
G
TRUE
TRUE
Calculate
F = (ITIME-2nd most recent river run) / (time between river runs)
Interpolate all U and Y values
Y(M,N) = (1-F) * YA1 (M,N) + F * Y(M,N)
Similarly for U(M,N)
CALL
PRINT 1 for printing
of river results
F'



F'
Call PRINT1 for printing of river run results.
IF Jprint= .true.
Yes
G
No
if Itape= .true.
Yes
No
if Iplot= .true.
Yes
No
Calculate F = (ITIME-2nd most recent bay run) / (time between bay runs now)
Interpolate values of SEA,SEB;UA,UB;VA,VB to estimate values of SE U and V , using x(N,M)=(1-F)*xA(N,M)+ F*xB(N,M) where x=SE,U and V.
Increment time loop index ITIME=ITIME+ 1.(units are DT)
Is ITIME<Max time for run?
F
No
Calculate hours simulated.
Write EOJ message
Call TIME for mill time cost estimate
Clear BOTANY and print river for previous time step so the routine will always print last time estimates
STOP

B.17. Subroutine NODE MK

Function:	To identify cross connections and compute node identifiers for the Fischer Model, river phase
System Functions:	None
User Functions:	None
Calling Statement:	CALL NODEMK
Error Messages:	None
Common Areas:	/IO/, /A/, /NODE/
Common Variables Reset:	Those arrays in /NODE/, NNODE in /A/ is defined on exit as the number of nodes in the system.
Description:	The program constructs a set of identifiers for each node on the segment. There are 5 arrays in common area /A/ with the following meaning

ITYPND	Type of Node
NDWNSEG	Downstream Segment
IIIUPS	3rd upstream segment, if any
IIUPDN	2nd up or downstream segments, if any
MUPSTR	Upstream Segment (premier)

ITYPND carries a further numeric code:

2 → A 2	Segment node with one upstream, one downstream segment
3 → A 3	Segment node with two upstream, one downstream segment
4 → A 4	Segment node with three upstream, one downstream segment
5 → A 3	Segment node with one upstream, two downstream segments.

In the original program, all the above were clustered into a nine integer number which was greater than the 4-70 integer mode range and so they had to be identified as individual arrays.

NNODE is the index referring to the node number currently being constructed. MSCIP array holds segment numbers already considered and INUM is its indexing variable, it is cleared on entry to the routine.

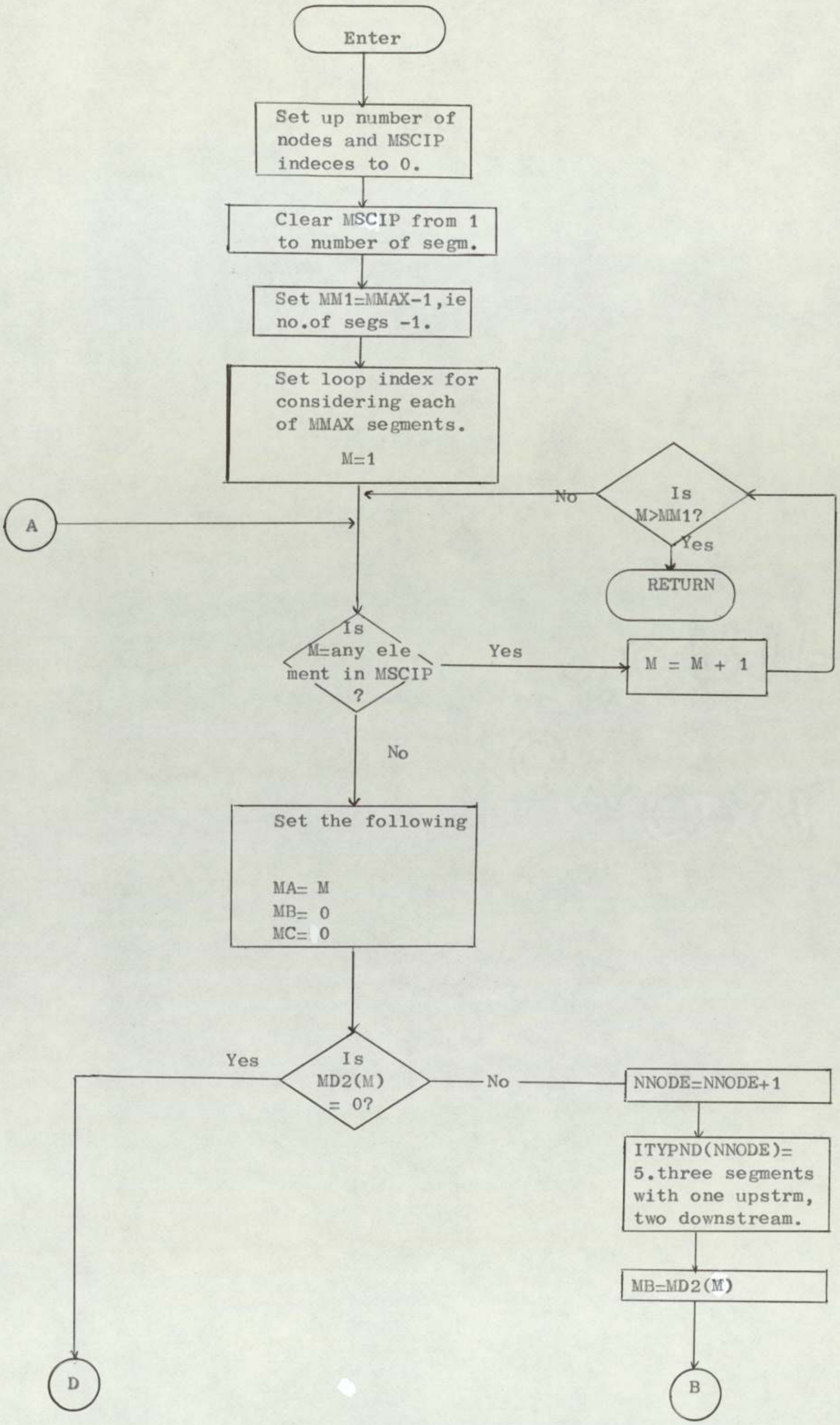
Now each segment is considered in turn. The segments connected to each segment are held in arrays MU1, MU2, MU3, MD1, MD2. The DO 200 loop is the master loop in this routine.

The DO 40 loop test to see whether any segment can be skipped if already dealt with due to specific geometries. If any of the IF tests succeeds, control goes to end of the master loop for the next segment.

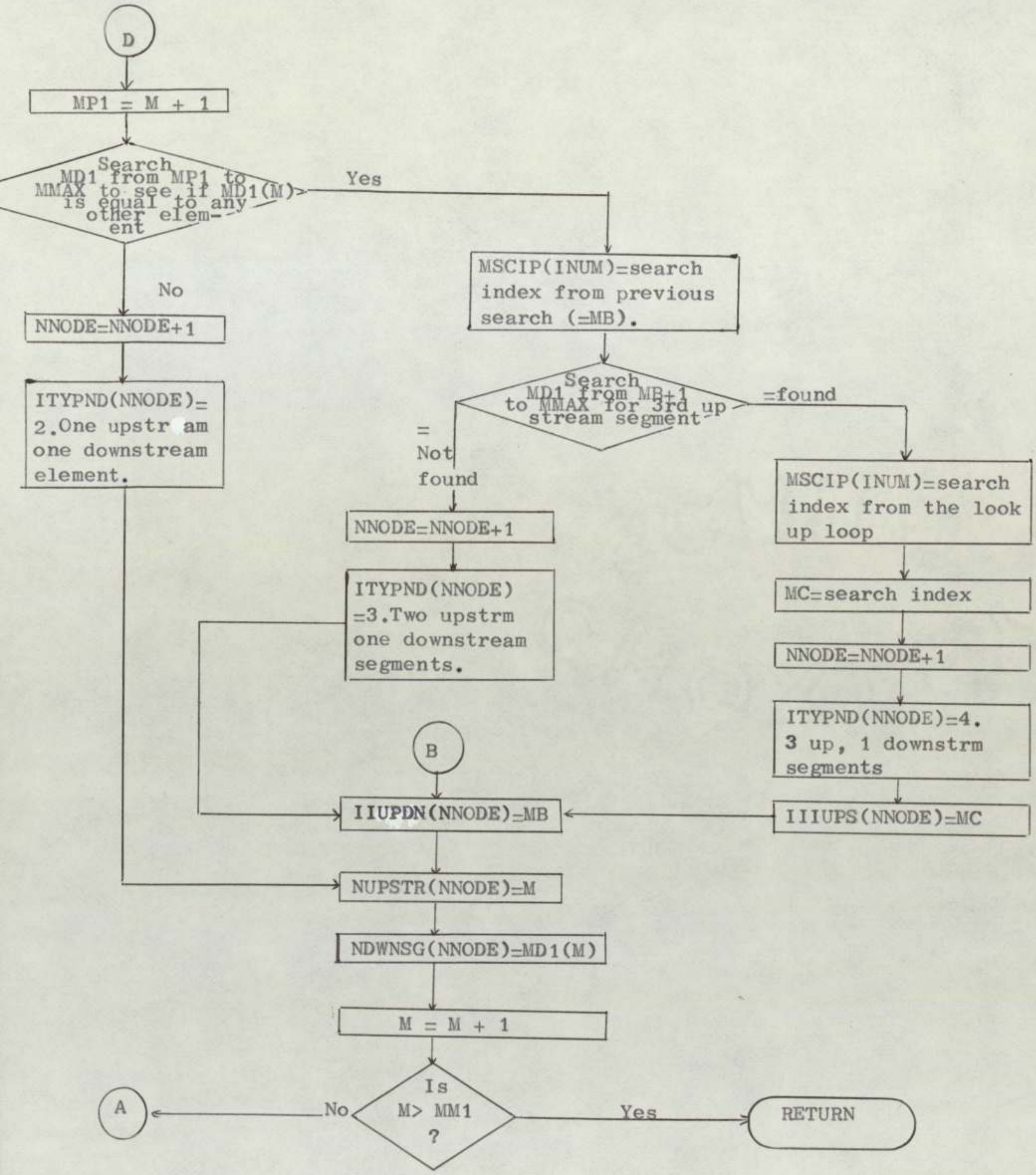
MA, MB, MC are holding locations for the upstream segment, 2nd up or downstream segment, 3rd up-stream segment and they are set now as MA=M (as it is considering the downstream node of segment M), MB and MC are set to zero at this stage. If MD2(M) is non zero, it is a type 5 node (see above) and control jumps to statement 160 where MB can then be defined as MD2(M).

DO 50 looks to see if in the current segment, any other segment has the same value in MD1, i.e. the downstream segment. If not, the node is a type 2 node with one up, one downstream node. The choice remaining now is between a type 3 or 4 node. This is resolved in DO 110 where a similar search to the previous one is carried out.

The calculated values of MA, MB, MC and MD1(M) are then slotted into the various arrays in common /NODE/ as appropriate to the type of node being considered.



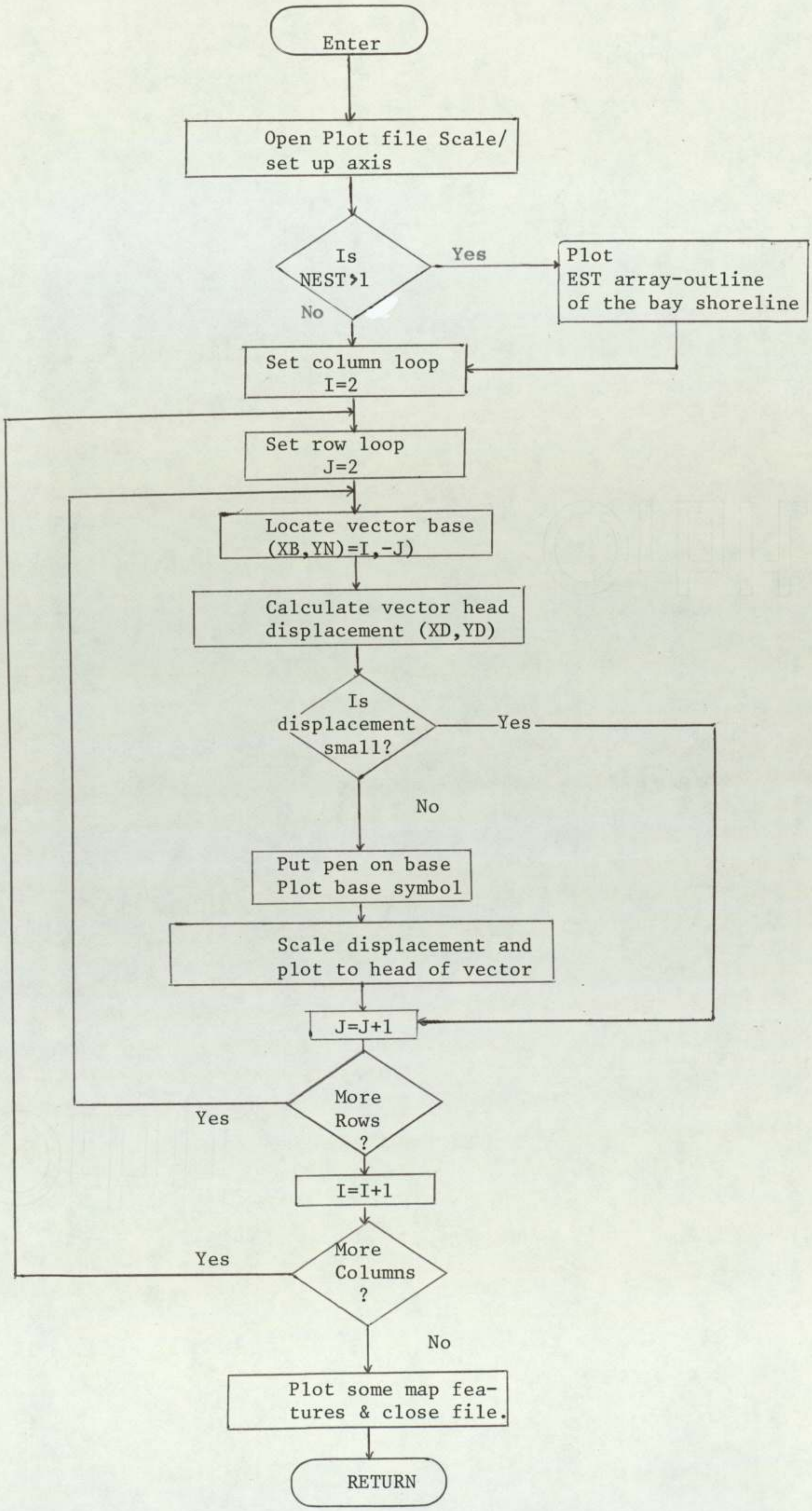
Enter
Set up number of nodes and MSCIP indeces to 0.
Clear MSCIP from 1 to number of segm.
Set MM1=MMAX-1,ie no.of segs -1.
Set loop index for considering each of MMAX segments.
M=1
A
Is M>MM1?
No
Yes
RETURN
Is M=any element in MSCIP ?
Yes
M = M + 1
No
Set the following
MA= M
MB= 0
MC= 0
Is MD2(M) = 0?
Yes
No
NNODE=NNODE+1
ITYPND(NNODE)= 5.three segments with one upstrm, two downstream.
MB=MD2(M)
D
B


D
MP1 = M + 1
Search MD1 from MP1 to MMAX to see if MD1(M) is equal to any other element
Yes
No
NNODE=NNODE+1
ITYPND(NNODE)=2.One upstr am one downstream element.
MSCIP(INUM)=search index from previous search (=MB).
Search MD1 from MB+1 to MMAX for 3rd up stream segment
=found
= Not found
NNODE=NNODE+1
ITYPND(NNODE) =3.Two upstrm one downstream segments.
MSCIP(INUM)=search index from the look up loop
MC=search index
NNODE=NNODE+1
ITYPND(NNODE)=4. 3 up, 1 downstrm segments
B
IIUPDN(NNODE)=MB
IIIUPS(NNODE)=MC
NUPSTR(NNODE)=M
NDWNSG(NNODE)=MD1(M)
M = M + 1
Is M> MM1 ?
No
A
Yes
RETURN

B.18. **Subroutine PLOT**

(for F2)

Function:	To plot the velocity distribution in the Bay for the Fischer Model on a CALCOMP plotter.
System Functions:	ABS, GRAFOR, ATAN, SIN, COS
User Functions:	None
Calling Statement:	CALL PLOT (NMAX, MMAX, UP, VP, SCALE)
Error Messages:	None
Common Areas:	/SPECS/
Common Variables Reset:	None
Description:	Each call to this routine produces one plot file. The calculated velocities UP and VP are vector added to produce a vector map of the bay. Calls 1, 3 & 4 to Grafor are to set the axis and other parameters. Then the DO 100 loop considers each point in turn, calculates the base and head of vector and then plots a small of vector base symbol. If NEST>1 a bay outline is overlayed on the plot. The final set of calls to Grafor writes some geographical features on the plot. When all points are plotted, the file is closed and automatically queued for plotting (under MJ,ICL4-70 system).


Enter
Open Plot file Scale/ set up axis
Is NEST>1
Yes
No
Plot EST array-outline of the bay shoreline
Set column loop I=2
Set row loop J=2
Locate vector base (XB,YN)=I,-J)
Calculate vector head displacement (XD,YD)
Is displacement small?
Yes
No
Put pen on base Plot base symbol
Scale displacement and plot to head of vector
J=J+1
More Rows ?
Yes
I=I+1
More Columns ?
Yes
No
Plot some map features & close file.
RETURN

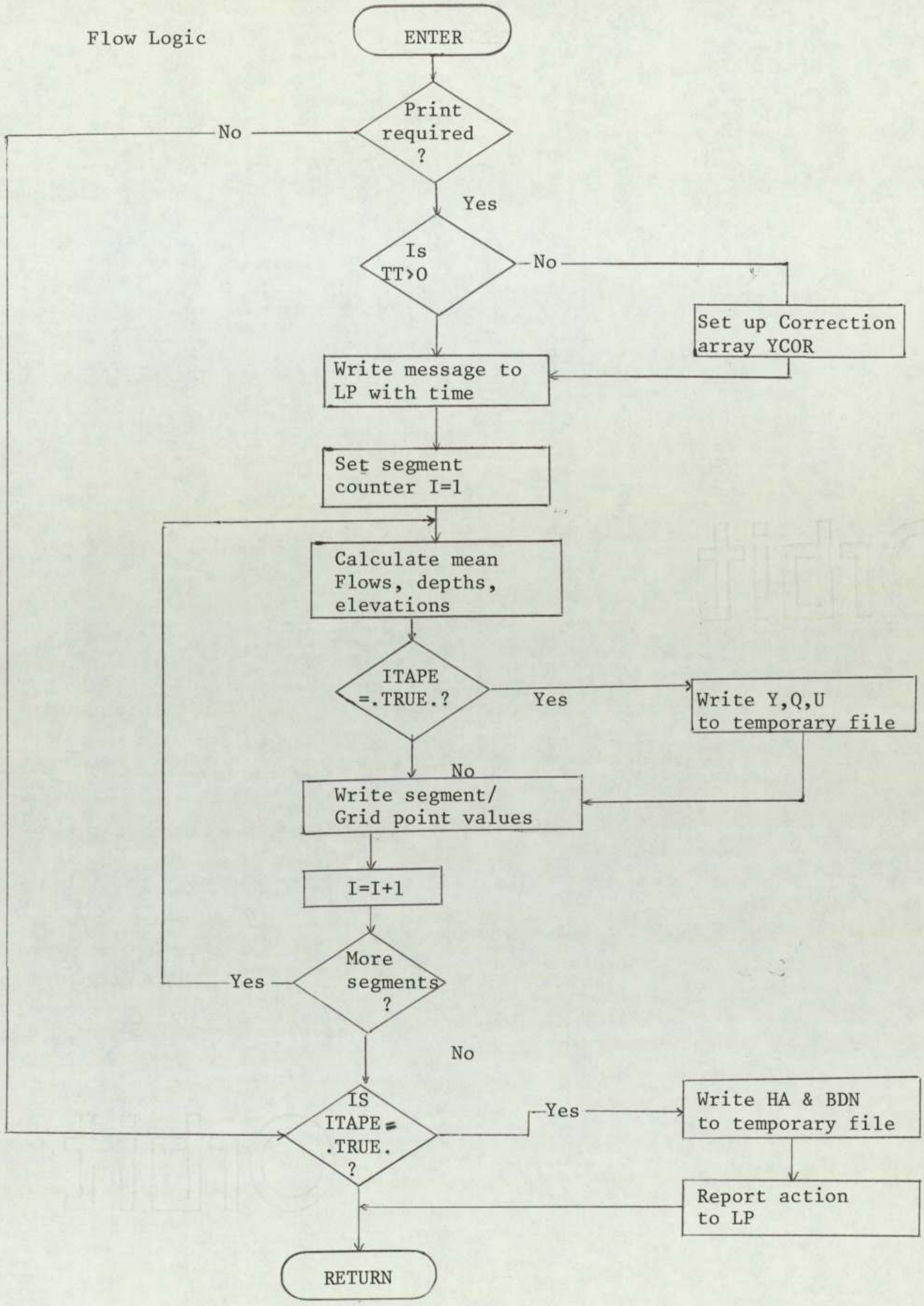
B.19. Subroutine PRINT 1

Function:	To generate report of one dimensional river phase to line printer and temporary file if required.
System Functions:	Read/Write I/O, FLOAT
User Functions:	None
Calling Statement:	Call PRINT1(TT,IPRINT,ITAPE)
Common Areas:	/IO/,/INITDP/,/A/,/C/,/COR/
Common Variables Reset:	YCOR in /COR/ on first call only, C in /A/ indirectly, H in /A/ indirectly, BDN in /A/
Description:	A dummy call is made from HYDROD with a negative calling time TT, and IPRINT set to .TRUE. to flag a river print required. This call sets up an elevation correction array which is used throughout the simulation and made the module 23% faster. Then the river is printed for initial conditions set through SEINV.

The main printing loop is DO 80. Flows and mean depths are estimated for each segment and if required (set ITAPE), sufficient data written to the temporary file to enable later restart. The results for each segment are tabulated and some internal grid point predictions are also written out (velocity/flow).

If required, further data is written to temporary file for input to second program . A message indicating time and block number of temporary file write is sent to the line printer.

Flow Logic



ENTER
Print required ?
No
Yes
Is TT>0
No
Set up Correction array YCOR
Write message to LP with time
Set segment counter I=1
Calculate mean Flows, depths, elevations
ITAPE =.TRUE.?
Yes
Write Y,Q,U to temporary file
No
Write segment/ Grid point values
I=I+1
More segments ?
Yes
No
IS ITAPE = .TRUE. ?
Yes
Write HA & BDN to temporary file
Report action to LP
RETURN

B20

Subroutine PRINT2
(for F2)

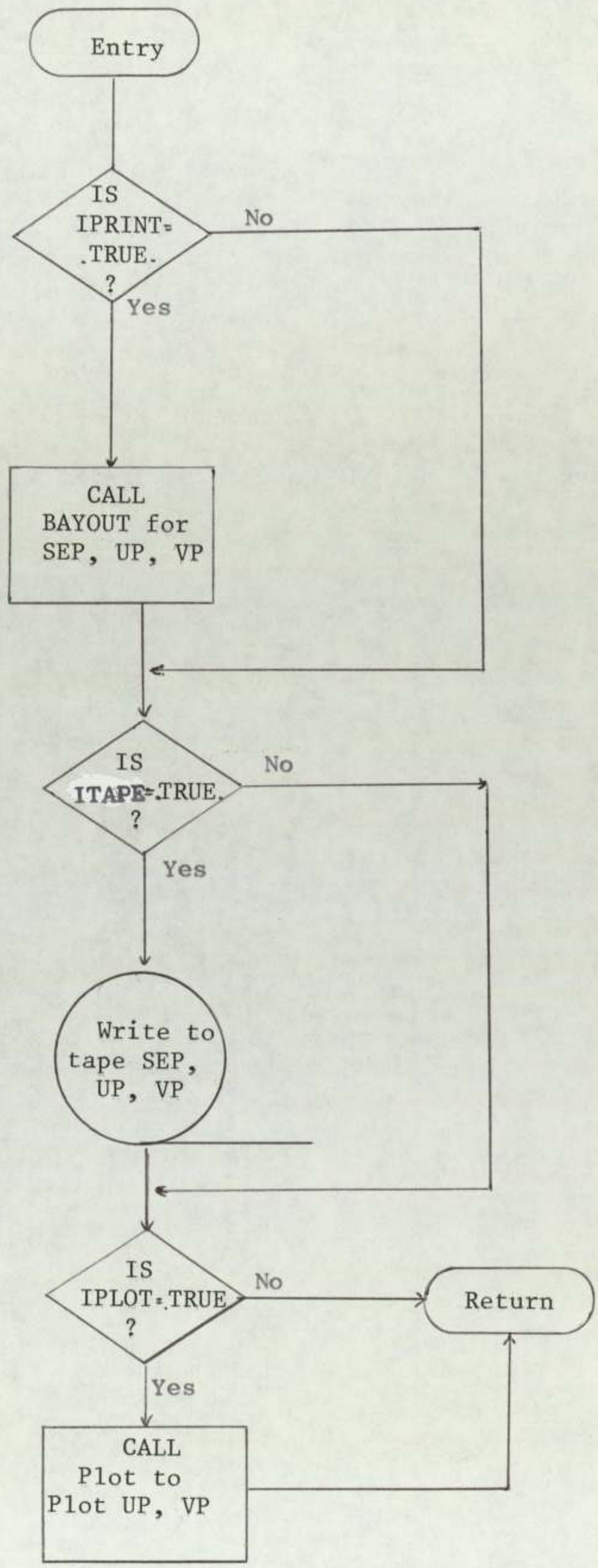
Function:	To print output for the Bay (through BAYOUT), write relevant data to temporary file if required and also test for plotting of Bay data.
System Functions:	WRITE
User Functions:	BAYOUT, PLOT
Calling Statement:	CALL PRINT2 (T, IPRINT, ITAPE, IPLOT)
Error Messages:	None
Common Areas:	/IO/, /CTCH22/, /SPECS/
Common Variables Reset:	None
Description:	T is the current time; IPRINT, ITAPE and IPLOT are logical fullword variables set for printing, writing to tape and plotting of Bay data. A value of .TRUE. will cause the I/O operations, .FALSE. inhibits.

The line printer dump is considered first. If it is required the arrays SEP, UP & VP (Elevation, velocity in X, velocity in Y direction) are printed through routine BAYOUT.

ITAPE is tested to see if a temporary file I/O is required. If it is arrays SEP & UP & VP are written to the temporary file.

IPLOT is tested for graph plot of UP & VP vectors on the CALCOMP via routine PLOT.

Flow Logic


Entry
IS IPRINT=.TRUE. ?
No
Yes
CALL BAYOUT for SEP, UP, VP
IS ITAPE=.TRUE. ?
No
Yes
Write to tape SEP, UP, VP
IS IPLOT=.TRUE ?
No
Return
Yes
CALL Plot to Plot UP, VP

B.21. **Subroutine QT**

(for F1)

Function: To permit true varying flow data at system upstream limits.

System Functions: I/O with END option.

User Functions: None

Calling Statement: CALL QT(T,QMEAS, MMAX)

Error Messages STOP 219 - End of data reached prematurely

Common Areas: None

Description: There are two main segments to the routine, one sets QMEAS values depending on time of simulation when called, and one reads in the time variant flow data on the first call only.

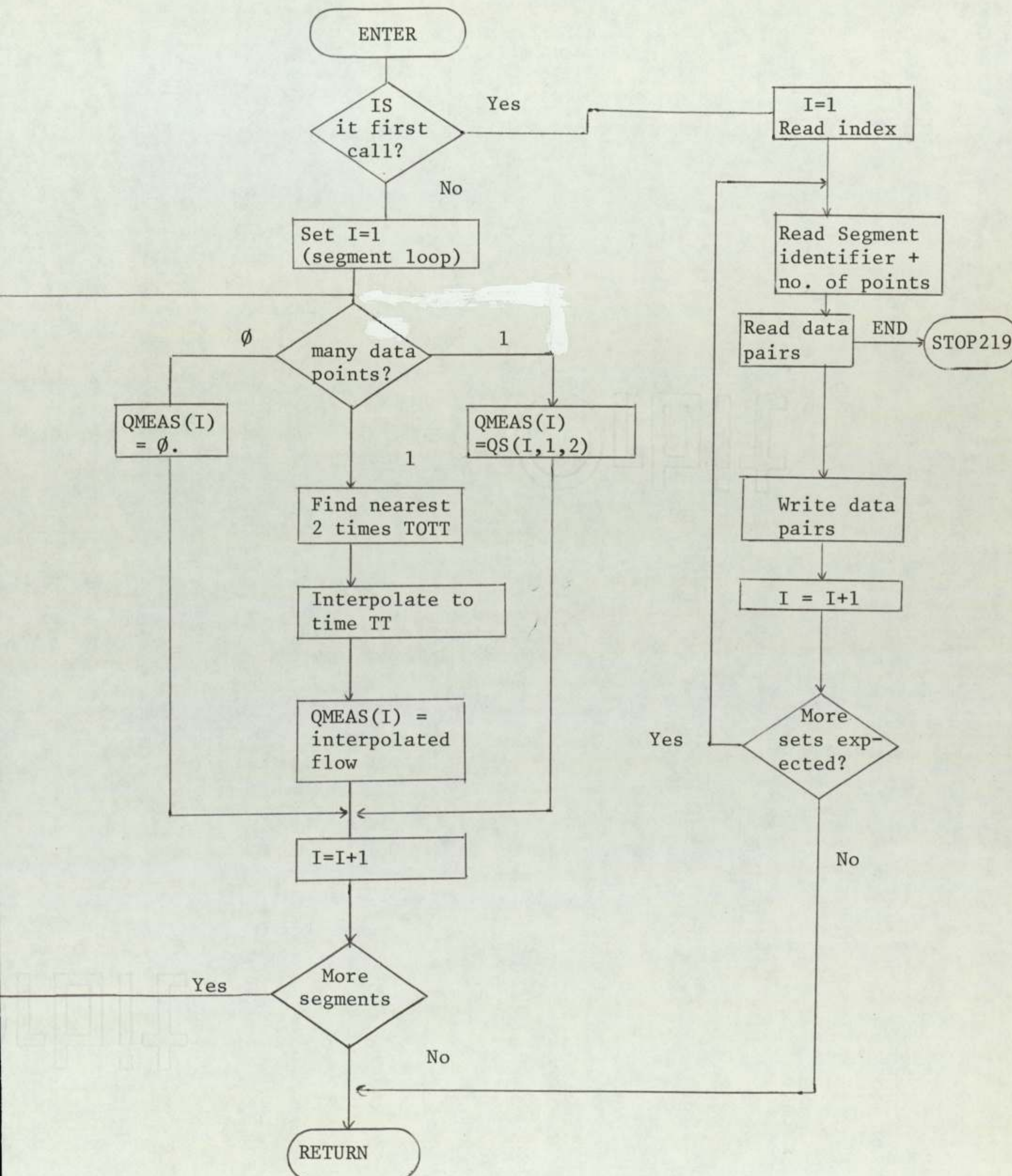
The DO 10 loop is the main calculating loop. NQ_i element holds the number of flow data points for segment i. If this is Ø no fresh water flow into segment is assumed, if it is one then a constant inflow is assumed. If data is time variable, then a search is made of the two times bracketing TT (ie tests against QS(I,J,1) for segment I, data point J, J=1,NQ(I)). An estimate of $QMEAS_i$ is made by interpolating the two flows corresponding to the two nearest times located, when each segment has been allocated a flow input through QMEAS control returns. The option should be used for flow variation in time intervals several orders of magnitudes higher than the simulation time step.

The read in section reads in, for each of MMAX segments, at least two card images. The first contains two parameters:

NSEG The segment for which flow data is to follow
NPTS Number of data pairs available.

The second card (and subsequently as required) contains sets of values of time (in hours from zero of simulation) and flow rate (in cubic feet per second). At least one pair of values should appear, even if only Ø & Ø. are entered.
The input data is printed to the line printer for visual checking.

Flow Logic



ENTER
IS it first call?
Yes
No
Set I=1 (segment loop)
many data points?
Ø
1
1
QMEAS(I) = Ø.
QMEAS(I) =QS(I,1,2)
Find nearest 2 times TOTT
Interpolate to time TT
QMEAS(I) = interpolated flow
I=I+1
More segments
Yes
No
RETURN
I=1 Read index
Read Segment identifier + no. of points
Read data pairs
END
STOP219
Write data pairs
I = I+1
More sets expected?
Yes
No

B.22. **Subroutine REST**
(for F1)

Function:	To provide a restart facility for the F1 model
System Functions:	REWIND, READ/WRITE I/O, ABS
User Functions:	None
Calling Statement:	CALL REST (TIME, NIN)
Error Messages:	STOP55, STOP56
Common Areas:	/A/, /IO/
Common Variables Reset:	Y, U in /A/

Description:

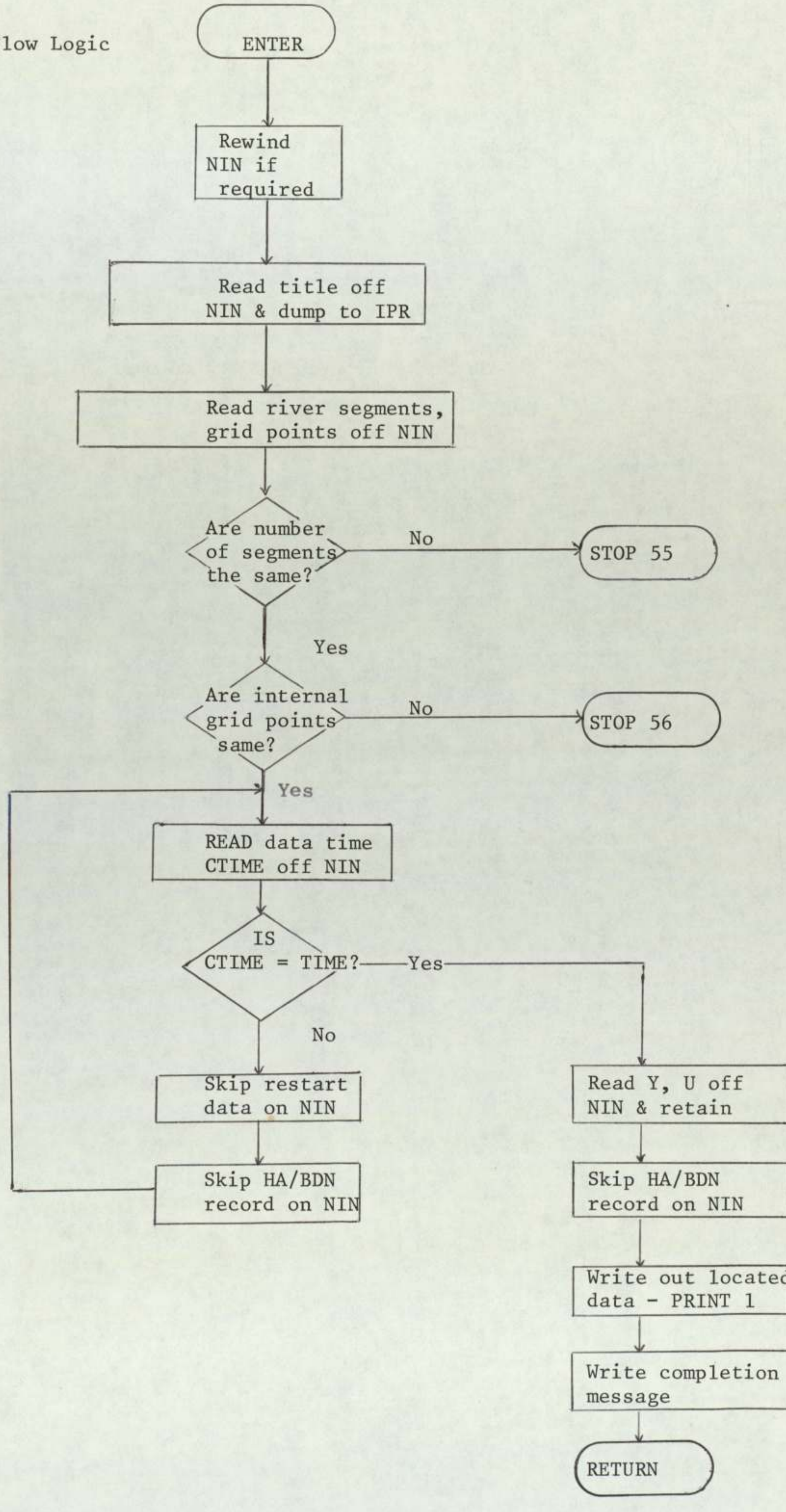
The data for restart is in file data set reference NIN. The first record in that file is the title of the generating run (see array ID in HYDROD). This title is read in and line printed for confirmation that the data sets match. The second record contains the number of segments in the generating system, and also the number of grid points per segment. These are all checked against the current run parameters, if they do not match then either

STOP 55 or STOP 56

stops execution of the restart facility and the file inputs ought to be checked.

As each data block starts with the time of the data as a simple item on the first record, this is read off singly. This is tested against the restart time and if the restart time is not reached, the whole record is skipped. If it is near the restart time, the rest of the record is kept and Y and U set ready to restart. Appropriate completion messages are output.

Flow Logic


ENTER
Rewind NIN if required
Read title off NIN & dump to IPR
Read river segments, grid points off NIN
Are number of segments the same?
No
STOP 55
Yes
Are internal grid points same?
No
STOP 56
Yes
READ data time CTIME off NIN
IS CTIME = TIME?
Yes
No
Skip restart data on NIN
Skip HA/BDN record on NIN
Read Y, U off NIN & retain
Skip HA/BDN record on NIN
Write out located data - PRINT 1
Write completion message
RETURN

B.23. **Subroutine REST**
(for F2)

Function: To allow a restart facility for the F2 model.

System Functions: READ/WRITE I/O, REWIND, ABS

User Functions: PRINT1, BAYOUT

Calling Statement: CALL REST (TIME, NIN)

Error Messages: STOP 118, STOP 1212.

Common Areas: /A/, /CTCH22/

Common Variables Reset: Y, U in /A/; SE, U, V in /CTCH22/

Description: Identical to REST for F1 except that addition bay data has to be skipped or read to retain. These are the statements READ(NIN,300). Also the final report to the line printer has additional calls to BAYOUT to allow a visible check on bay data to be made. The flow logic is identical also, with the above expansions.
STOP118 is reached when problem dimensions do not match input data.
STOP1212 is reached when internal grid point structure does not match that of input data.

B.24. Subroutine UPTIME

Function: To update next time of printout and to set print flag for current time step.

System Functions: None

User Functions: None

Calling Statement: CALL UPTIME (IT, IDOARA, INDEX, IFLAG, ITIME, IL)

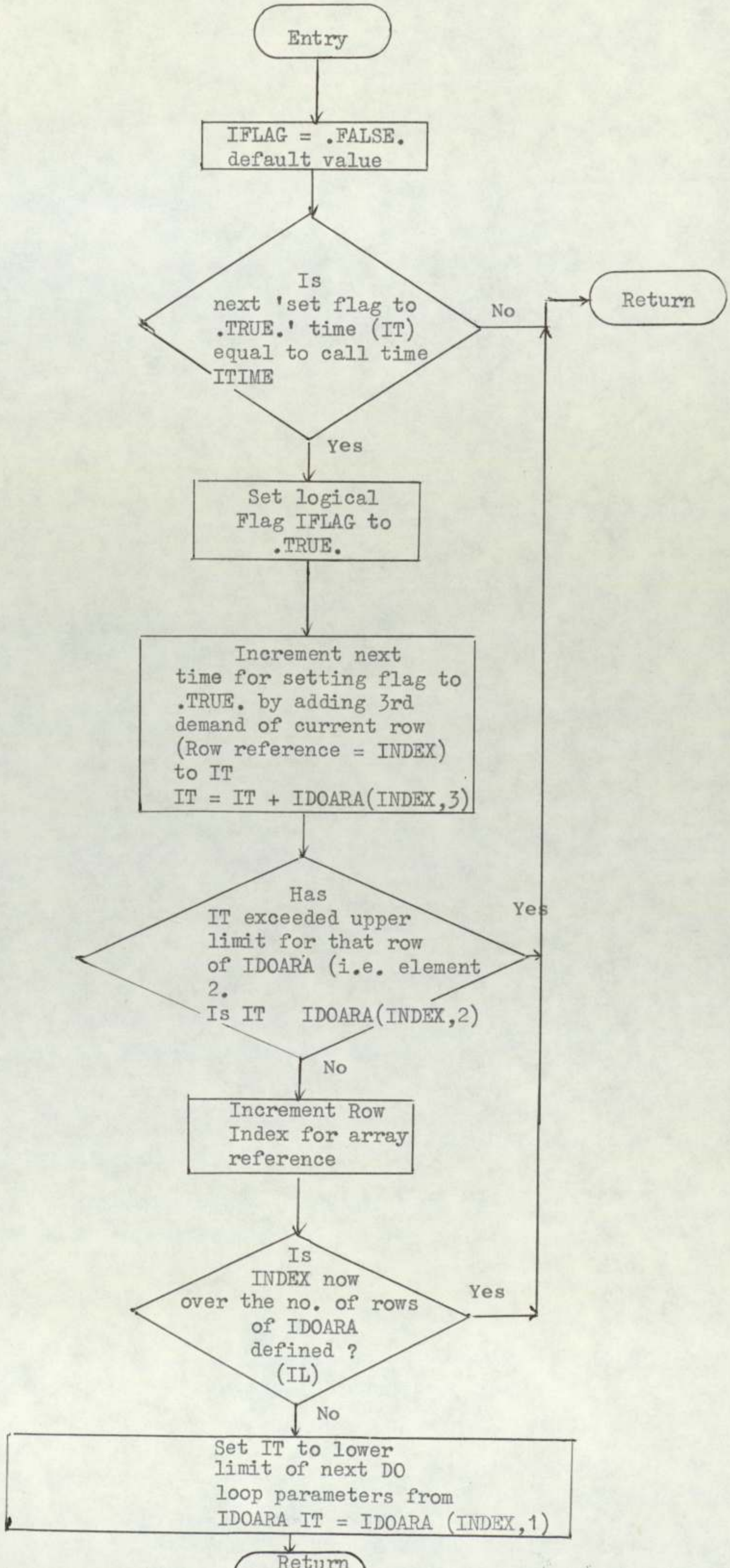
Error Messages: None

Common Variables: None

Description: For use in the Fischer Model, it sets a logical variable IFLAG to .TRUE. or .FALSE. depending on whether the current time ITIME is equal to the next time for printout 'IT'. The reason for the routine is a special way of storing the desired printout time. The array IDOARA is a two dimensional array of size (15,3). The second dimension must be 3 but the first dimension is more flexible. Each row of this array carries a set of 3 integers from a DO statement definition, say I,J,K. This is a standard form of asking for printout at time = I, I + K, I + 2K, I + 3K ...... until I + nJ>K. Then another set can take over.

If the first row of the array were to hold the elements 7, 15, 3 then printing would occur at times 7, 10, 13 and for other times a .FALSE. flag would be set.

Logical Flow



B.25.

Intermediate File Generated by Models F1 and F2.

The file must contain sufficient data to enable

A) A restart to be possible from any record-time
B) Abstraction of sufficient data to allow the pollutant models to be run.

Files generated by F1 is a subset of the file generated by F2. The F2 temporary file structure is:

ORDER	RECORD TYPE	FORM AT	CARD IMAGES	DESCRIPTION OF CONTENTS
1	1	20A4	1	Alphanumeric problem title - ID
2	2	20I4	1	Number of segments - MMAXI Number of columns in bay - NMAX Number of rows in bay - MMAX Number of grid points in each segment - NGP
→3	3	E10.3	1	Time, within simulation of data - CTIME
↓ 4	4	8E10.3	INT(3n/8)+1 for each segment	Values of Y, Q, U at gridpoint i, i=1, n for each segment. A total of at least MMAX1 card images.
↓ –5	5	10F8.3	3x{INT(NMAXx NMAX1/8 + 1}	The following in order of appearance: SE(Col, row) col=1,NMAX,row=1,MMAX U (" ") same V (" ") same

F1 Variant:

Items in list for record type 2 set to zero are NMAX & MMAX

Record type 5 does not appear.

Order of appearance is cyclical for record types 3 - 5 after initial parameters are read in or written.

B.26. TIMINGS

In practice the limiting factor has been found to be the amount of information generated in the line printer and intermediate file.

Average times for a 4-70 were:

4000 etus $\simeq$ 1000 secs $\simeq$ 17 minutes mill time for an 8 day simulation on a 9 segment river,

a 16 x 11 bay and time steps of 1 and 3 minutes.

Enough data was generated within this time to fulfil most purposes (river every $\frac{1}{2}$ hour, bay every hour).

At current costs, this is in the order of £100 to include consumables like line printer paper, graph plotting roll etc, and system configuration to set up a run against a data file.

B.27. <u>Data Input Structures for F1 and F2</u>

Both inputs follow similar patterns, with F2 requiring more. Each data item will be noted whether it is for F1 or F2. All items for F1 also are required for F2. Input that is optional depending on other switches set elsewhere or read in are bracketed.

Some options are mutually exclusive, others overlap. A starred item is only for the model indicated.

The type column consists of identifiers of the form

ABn

A = <u>R</u>eal / <u>I</u>nteger / <u>A</u>lphanumeric / <u>L</u>ogical

B = <u>A</u>rray / blank

n = minimum elements required to allow program to run successfully. If * then variable number depending on dimensions of the problem.

Itemised Data Records Layout

Model	Record/ Card No.	Col from (Y=free	Col to format)	Variable Name	Type	Description of Variable
*F1	1	Y		DT	R	Step time in seconds
				NT1	I	Plot/report interval for boundary values
				MAXTIM	I	Simulation Steps of DT to be run
				INTAP	I	Frequency to temporary file report
				SEINV	R	Initial Elevation
				MTMEL	I	Number of do loops for river printing
				MINTAP	I	Minimum cycles before temporary file dumps
				NOVT	I	Temporary file data set reference number
				RT	R	Restart time
				FLOWT	L	Time variable fresh water flow?
				TIDET	L	Test for turning tides?
				PLOT	L	Plot input vs some predicted curves?
				NIN	I	Restart file data set reference
				MP	I	Segment number to plot
				NP	I	Grid point within segment to plot
*F2	2	Y		DT	R	As * F1 Rec 1
				NT1	I	Multiple of DT for river time increment
				NT2	I	Multiple of DT for bay time increment
				MAXIM	I	As * F1 Rec 1
				INTAP	I	As * F1 Rec 1
				SCALE	R	Scale of final velocity vectors on graph
				SEINV	R	As *F1 Rec 1
				RESTRT	R	As *F1 Rec 1 (RT)

Contd...

Model	Record/ Card No.	Col from (Y=free	Col to format)	Variable Name	Type	Description of Variable
				NIN	I	As * F1 Rec 1
				NO	I	As * F1 Rec 1 (NOUT)
				NTIMEL	I	Number of do loops for bay output
				NTIMEL	I	As * F1 Rec 1
				IPTMEL	I	Number of do loops for plotting vector output
				MINTAP	I	As * F1 Rec 1
				NMAX2	I	Columns in the bay
				MMAX2	I	Rows in the bay
*F2	3	Y		ISD	IA5	5 element date of start of simulation
	4	1	80	ID	AA80	80 character title (20 groups @ A4)
*F2	5	Y		NPRINT	IA3	Do loop parameters for bay printing
	6	Y		MPRINT	IA3	Do loop parameters for river printing.
*F2	7	Y		NPLOT	IA3	Do loop parameters for bay velocity plots
*F1	8	Y		PSTIME	R	Start time of plotting
*F2	9	Y		MINDO	I	Number of row open water boundaries
				MOBD	IA1	Open water boundaries
*F2	10	Y		MQBDO	I	Number of row discharge boundaries
				MQBD	IA1	Row discharge boundaries
*F2	11	Y		NINDO	I	Number of column open water boundaries
				NOBD	IA1	Column open water boundaries
*F2	12	Y		NQBDO	I	Number of column discharge boundaries
				NQBD	IA1	Column discharge boundaries
	13	Y		MMAX1	I	Number of segments in river phase
*F2	14	Y		QMEAS	IA	MMAX1 values of fresh water flows to segments

Contd...

Model	Record/ Card No.	Col from (Y=free	Col to Format)	Variable Name	Type	Description of Variable
(F1)A	15	Y		QMEAS	IA	MMAX1 values of freshwater flows to segments
(F1)B	16	Y		NSEG	I	Segment number
				NPTS	I	Data points available for flow data
(F1)B	17	Y		QS	RA	Sets of time vs flow data for segment NSEG.
(F1)B	*	*	NOTE			Rec 16-17 once for each of MMAX 1 segments
Subroutine data						
	18	21	24	MU1(M)	IA1	1st upstream segment connected to M
		25	28	MU2(M)	IA1	2nd upstream segment connected to M
		29	32	MU3(M)	IA1	3rd upstream segment connected to M
		33	36	MD1(M)	IA1	1st downstream " " " "
		37	40	MD2(M)	IA1	2nd downstream " " " "
		41	44	NDATA(M)	IA1	Data point - depth vs width
		45	55	XLNGTH(M)	RA1	Length of segment in feet
		56	60	NGP(M)	IA1	Internal grid points
		61	65	FRICO	R	Manning Friction coefficient for segment
		66	75	SBED(M)	RA1	Bed slope
		76	80	Y(M,1)	RA1	Downstream elevation in feet
	19	1		YGIVEN(M,1)	RA1	(Recorded depths (YGIVEN(M,i))
		11	20	BGIV(M,1)	RA1	(against breadth (BGIV(M,i)),i=
		21	30	YGIVEN(M,2)	RA1	(1, NDATA(M)), total of
		31	40	BGIV(M,2)	RA1	(INT/NDATA(M)/2+1 cards
(ENDDATA)	*	*	NOTE			Rec.18-19 repeated in ascending order for each of MMAX 1 segments

Contd...

Model	Record/ Card No.	Col from (Y=free	Col to format)	Variable Name	Type	Description of Variable
*F2	20	Y		SEINV	R	Initial Elevation
				XSTRSS	R	Wind stress, X direction
				YSTRSS	R	Wind stress, Y direction
				NTSTPS	I	Steps of tidal input data
				ANGLAT	R	Angle of Latitude
				AL	R	Length of grid spacings
				NSEAB	I	Boundaries switch
				NDIV	I	Number of tidal function measurements
				HDIFF	R	Height difference, Bay/River phase
				NRBIF	I	Number of Bay/River interface points
				NBOIFA	I	Number of points on bay/ocean interface A
				NBOIFB	I	Number of points on bay/ocean interface B
*F2	21	Y		RBIF	IA2	River Bay interface points
*F2	22	Y		BOIFA	IA2	Bay/Ocean interface points, A
*F2	23	Y		BOIFB	IA2	" " " " B
*F2	24	Y		XSE	RA2	Tidal input function, interface A
*(F2)	25	Y		XSEB	RA2	" " " " B
*F2	26	20	F4.0	H	RA*	Depths of bay at grid points, read in column wise, row wise.

Contd..

Model	Record/ Card No.	Col from (Y = free	Col to format)	Variable Name	Type	Description of variable
*F2	27	20	F4.0	C	RA*	Chezy coefficient at grid points, read in by columns, then rows, as for H.
*(F2)	28	Y		NEST	I	Number of points in estuary outline
				EST	RA1	Estuary outline, x-y points
*F1	29	Y		N	I	Points in tidal function for boundary
		Y		NDIV	I	Time steps between tidal date
		Y		XSE	RA2	Tidal Function at boundary

B.29. Data Records Sequence for Models F1 and F2

Model F1		Model F2	
1*		2	
4		3	
6		4	
8 → 15 or 16		5	
15 / 16	Flow Data Option	6	
17 (after 16)		7	
15, 17 → 18		9	
18	Repeated for	10	
19 ⟲	each segment	11	
29		12	
		13	
* Numeric record identifiers refer to records itemised in B.28		14	
		18 ⟲	Repeated for each segment
		19	
		20	
		21	
		22	
		23	
		24 → 25 or 26	2nd interface
		25	option
		26	
		27	
		28	Can be omitted.

B.30. Structure of NOBD, MOBD, NQBD, MBD, NBD

All are compressed integers arrays to relay boundary data to the F2 model:

NOBD - Column open water boundaries

MOBD - Row " " "

NQBD - Column discharge "

MQBD - Row " "

NBD - Column water boundaries

MBD - Row " "

NOBD/MOBD

General Structure: IJKL, where IJK are two digits,

L is a one digit number.

KEYWORD

	NOBD	MOBD	Read as description
I	Column	Row	Under consideration
J	Row	Column	for start of open water boundary
K	Row	Column	for end of open water boundary
L	1=LEFT	1=ABOVE	water is - the boundary
	0=RIGHT	0=BELOW	outlined

NQBD/MQBD

Same as NOBD/MOBD except a discharge boundary is defined and velocities provided for those boundaries.

NBD/MBD

Calculated internally in 'FIND', they summarise all boundary information. The General Structure is

IJKLM, where KLM are 2 digits; I, J are single digits.

KEYWORD

Digits	NBD	MBD	Read on Description
I	2=ABOVE	2=LEFT	Discharge boundary to __ of open water field
	1=ABOVE	1=LEFT	Open water boundary to __ of open water field
	Ø=ABOVE	Ø=LEFT	Land boundary to __ of open water field.
J	2=BELOW	2=RIGHT	Discharge b oundary to __ of open water field
	1=BELOW	1=RIGHT	Open water boundary to __ of open water field
	Ø=BELOW	Ø=RIGHT	Land boundary to __ of open water field
K	COLUMN	ROW	Under consideration
L	ROW	COLUMN	Start of respective boundary
M	ROW	COLUMN	End of respective boundary

Note: in certain areas the largest number reached in these two arrays is $>2^{23}$ and ∴ unsuitable for a 24 bit machine without double length integer facilities (1900 users).

B.31. The Variable Names Index to Models F1 and F2

Layout

Each entry has the following format

NAME	(Type/Software location)	Total word size/type* word size
	description	

NAME	The name of the variable being sought, up to 6 characters with FORTRAN IV conventions applying
(Type)Software location:	Either 'A Common area name', or the common area in which it appears, or F1 - F2 depending on the model in which it occurs followed by the subroutine name.
Total word size:	Limits of an array if relevant, otherwise single word length assumed.
Type:	'A' for alphanumeric characters 'I' for integers 'L' for Logical 'R' for Real
Word Size:	Usually '4' for ICL 4-70 but can also be '8' (double precision), '2' (half word), or '1' (logical flags).
Description:	Brief outline of function of variable.

The index is not exhaustive but contains all main variables and will require updating for other systems and when programs are developed further.

A Common Area

Area holding most of the arrays with the physical parameters of the system (river section)

A Common /C/ 9*6R*4

Common array holding cross-sectional area of the river phase segments

A F2-BOTANY 99*R*4

Work array in the implicit phase of the two dimensional computation.

AA CALNOD R*4

Cross-sectional area of primary upstream segment to node K

AB CALNOD R*4

Cross-sectional area of 2nd upstream or downstream segment to the node K

AC CALNOD R*4

Cross-sectional area of 3rd upstream segment to node K.

AD CALNOD R*4

Cross-sectional area of 1st grid point in the primary downstream segment

AG F2-BOTANY R*4

Acceleration due to gravity,32.2 ft/sec/sec (see G).

AGIV Common /A/ 9*11R*4

Cross-sectional areas input. The i,j th element is the cross -sectional area of a one dimensional river segment corresponding to a given depth,YGIVEN(I,J)

AL F2-BOTANY R*4

Interior grid spacing , in feet. The grid fixed must have square sides. 8000 feet used in the system being studied.

ALC F2-BOTANY R*4

Constant defined on the first call to the routine = (Δl)/(number of river-bay interface points)

ALD F2-BOTANY R*4

Step cross-sectional area, river to bay interface =Hdiff*Δl
Allows use of different datums for different phases.

ALL Common Area

Area holding limits of bay,switches and array for the interface concentrations ,CPASS.Not used in this version but can be used to run both models via a buffer to direct usage of the pollutant transport program.

ALPHA F2-BOTANY R*4

Constant Factor in Computations in bay. Set to 0,.5 or 1.0 in various parts of BOTANY

ANGLAT F2-BOTANY R*4

The angle of latitude in the northern hemisphere in degrees. Should be set to negative if system is in southern hemisphere. Used in the calculation of the Coriolis Force.

AT F2-BOTANY R*4

Time step for two dimensional bay phase calculations

AVG PRINT1 R*4

Sum of elevations for all internal grid points for a segment.

AVGH PRINT1 R*4

Sum of mean depths of flow for a segment in the river phase.

AVQ HYDONE R*4

The discharge from the last segment to the sea (or into the bay phase) over one time step as an average over the mesh points of the last segment.

B CALHBC R*4

Calculated breadth(width) of the channel for a depth of Y(M,N)

B F2-HYDROD 99R*4

Work array in implicit computations.

BDN Common /A/ 9R*4

Width of one dimensional segment at water level at downstream end of a segment.Periodically written to save file.

BETA F2-BOTANY R*4

'Constant' in bay computations, set to 0.,.5 or 1.

BGIV Common /A/ 9*11R*4

Array of data input in DATA,holding widths of segments corresponding to depths held in YGIVEN.

BOIFA	F2-BOTANY	20*2I*4	

Array holding bay/ocean interface points for interface A.

BOIFB   F2-BOTANY   20*2I*4
Array holding bay/ocean interface points for interface B.

BUP   Common /A/   9R*4
Width of one segment at water level at upstream end of that segment.

C   Common /A/   9*6R*4
An array whose i,j th element is the wave velocity of seg. i ,mesh point j = g.H(I,J)

C   Common Area
Area holds cross-sectional area data.

C   Common /CTCH22/   21*17R*4
Holds the value of the Chezy Coefficient for each grid point in the bay.

C1   F2-BOTANY   R*4
Constant defined as $g.\Delta t/\Delta l$

C2   Common /CTCH22/   21*17R*4
As C in this common area

C2   F2-BOTANY   R*4
Constant , set to $\Delta t/\Delta l = g.C1$

C3   F2-BOTANY   R*4
Constant , set to $\Delta t/4$

C4   F2-BOTANY   R*4
Constant , set to $8.g.\Delta t$

COR   Common area   9*4R*4
Area holding correction array for use in PRINT1.

CPASS   Common /ALL/   9*4R*4
For use by pollutant transport program, to hold the interface concentrations.

CTCH22   Common area
Area holding current and previous estimates of U,V and SE and geometric information on the bay boundaries.

CTIME   F1-REST   R*4
Time read off temporary file, data set reference (channel) NIN

D Common /BAYD/ 100*4R*4
Array for retaining interface values calculated in BOTANY

DELTA F2-BOTANY R*4
'Constant' in bay computations ,usually set to .5

DESC F2-BAYOUT A8
Eight character identifier of the array being listed.

DT F2-BAYOUT R*4
Current time in hours from start of simulation run.

DT Common /A/ R*4
The basic time step for the simluations.In model F1 this is the time step, in model F2 the step can be any multiple of this basic unit.DT is defined in seconds.

DT Common /CTCH22/ R*4
Time step,in seconds, for running the bay prediction routine.

DT F2-HYDROD R*4
The basic time step for the mixed dimension model.

DT1 Common /A/ R*4
See DT, Common /A/

DT1 F2-HYDROD R*4
Time step in seconds for the river phase, being the product of DT and NT1.

DT2 F2-HYDROD R*4
Time step in seconds for the bay phase, being the product of DT and NT2

DTDX Common /A/ 9R*4
Array holding $\Delta t/\Delta x$ for each segment.Units are secs.per cm.

DX Common /A/ 9R*4
Array holding length of internal mesh step for each segment.

E CALEF 9R*4
Array holding data for one characteristic line

E PRINT1 9*6R*4
Elevation in segment i , grid point j .For all grid points within each segment.

EL Common /A/ 9R*4
Array of values of river elevations at downstream end of the segments in the river phase.

EST Common /SPECS/ 52R*4
Array holding outline of bay being simulated .Up to 26 nodes for drawing can be specified.

F CALEF 9R*4
Array holding data for the second characteristic line.

F F2-BOTANY 21R*4
Array holding Coriolis term for each row.Strictly each row is on a different latitude and thus have a different Coriolis term.In this study the effect is two orders of magnitude below the average effect and thus negligable.

FE F2-BOTANY R*4
Coriolis Force term constant,later saved in array F.

FIRST CALHBC L*4
Logical switch to enable printing of calculations in first call to be cut off for subsequent calls.

FIRST QT L*1
Logical switch to label 1st call to QT to trigger inputs.

FLOWT F1-HYDROD L*4
Logical variable for the type of flow data input. If .TRUE. then QT is called to input flow data(time variant).If set to .FALSE. , time invariant flow data is expected.

FF HYDONE R*4
Reciprocal of NN

FRIC Common /A/ 9R*4
Array holding friction coefficients for each segment.

FRICM CALEF R*4
Temporary location of friction coefficients

FU Common /A/ 9R*4
Array of values calculated in CALEF for characteristics.

ICYCLE Common /ALL/ I*4

For use by the pollutant transport program-the number of the current overall cycle.

ID F1-HYDROD 20A*4

Array used to hold an 80 character problem title.

II F1-Block Data-Common /NODE/ 45I*4

Array to blanket arrays ITYPND,NDWNSG,IIIUPS,IIUPDN,NUPSTR for initialisation.

IIUPDN Common /NODE/ 9I*4

Array holding the second up or downstream segment depending on code in ITYPND.

IIIUPS Common /NODE/ 9I*4

Array holding the number of the third upstream segment if there is one.

IFLAG UPTIME L*4

Logical flag holding decision to print/plot to temporary file for current call of UPTIME .

IL F1-HYDROD I*4

Index for saving boundary values in arrays QS,YS.

IND Common /BAYD/ I*4

Index to control storage of interface data in array $D_{i,j}$.

INITDP Common Area

Area holding height data

INTAP F1-HYDROD I*4

The interval in terms of number of loops between reports to an intermediate file.This is usually set to report at hourly intervals.

INTAP F2-HYDROD I*4

Frequency between reports to intermediate file, in terms of number of steps of DT seconds.

INUM NODEMK I*4

The next vacant element in MSCIP.

IO Common Area

Holds data set reference numbers of input/output channels.

IPLOT F2-HYDROD L*4

Logical Variables set to true if plot output is required, for bay velocity vectors.

IPR Common /IO/ I*4

Peripheral number of output line printer.

G	Common /A/	R*4	Acceleration of gravity, 32.2 ft/sec/sec
G2	Common /A/	R*4	Defined as 2g = 64.4 ft/sec/sec
GAMMA	F2-BOTANY	R*4	Constant in the computational scheme. Should be .5
GDT	Common /A/	R*4	Product of timestep DT and g.
GG2	CALEF	R*4	Reciprocal of G.
H	Common /A/	9*6R*4	Array whose i,j th element is the mean depth of flow at the point i,j.
H	Common /CTCH22/	21*17R*4	Depth ,in feet, of bottom of bay below grid point as defined by the indeces.
H2	Common /CTCH22/	21*17R*4	Refers to previous array
H43	CALEF	9R*4	Array holding reciprocal values of H(I,J) raised to the power 4/3
HA	PRINT1	9R*4	Average 'mean depth of flow' per segment.
HC2	F2-BOTANY	R*4	Constant = .5 * (Δt/Δ1)= .5 *C2
HDIFF	F2-BOTANY	R*4	The difference between zeroes of datum for river and bay phases.
IBL	PRINT1	I*4	Block counter to record the number of reports to the dumping temporary file that occur through PRINT1
ICD	Common /DATE/	5I*4	The current date reached from the start date of simulation,to nearest minute.Only acts when dt>59.

IPRINT F1-HYDROD L*4
Logical variable set to .TRUE. if a line printer request for a report print is to be issued to the appropiate print module for the current time.

IPRINT F2-HYDROD L*4
Logical variable set to .TRUE. if river phase data to be written to an output device.

IPTMEL F2-HYDROD I*4
Number of implicit do-loops input for plot of a bay velocity map.

IR Common /ALL/ I*4
For use by pollutant transport routine,hours in tide cycle phase

IR F2-BOTANY I*4
Type of boundary on right of,or below, a row or column in bay

IS F2-BOTANY I*4
Type of boundary on left of , or top of , row or column in bay

ISD Common /DATE/ 5I*4
The start date for simulations.

ISTEP F2-BOTANY I*4
Switch to determine order of computation.

ISWTCH Common /ALL/ I*4
For use by pollutant transport program.The type of tidal phase ie flood or ebb phase.

ITAPE F1/F2-HYDROD L*4
Logical variable set to .TRUE. if a report is to be sent to the intermediate file in the current time step.

ITIME Common/TIM/ I*4
Current time from start of simulation in terms of cycles each of the basic time interval dt.

ITIME F1-HYDROD I*4
The cycle counter,each of step dt , up to max of MAXTIM

ITM F1-HYDROD I*4
Integer,truncated, value of variable TTM

ITYPND Common /NODE/ 9I*4
Array holding the node type in a numeric code, 1-5 .

JPRINT F2-HYDROD L*4

Logical variable set to .TRUE. if bay data is to be written to the output channel

K CALNOD I*4

Index of the node being modified.

KK F1-HYDROD I*4

The index for array YSE to locate elements for the tidal forcing function.The index is cyclical up to NC .

KK F2-HYDROD I*4

Refer to KKBOT in Common /TIM/

KKBOT Common /TIM/ I*4

Restart value of KK to be transmitted to routine BOTANY.

L F2-BOTANY I*4

Last row or column of water field in bay.

LIMITS Common area

Area holding start limit for segments for main loop of routine HYDONE.

LIM1 Common /LIMITS/ I*4

The start segment for computations in HYDONE.Alternates between most upstream segment (1) and notional most downstream segment (MMAX1).

LL F2-BOTANY I*4

Last but one row or column in the water field, ie 1-1

LLL F2-BOTANY I*4

Next row or column in water field, ie L+1.

LPRINT Common /A/ I*4

Now redundant, used while debugging to send trace levels through to lower order subroutines.

M General Index I*4

Usually used to access array element for segment data.

M CALNOD I*4

Principal upstream segment to node K.

MAJOR Common /ALL/ I*4

For use by pollutant transport program, the subcycle of a main tidal cycle (ICYCLE).

MAXTIM F1-HYDROD I*4

Number of loops of duration DT for which the simulation is to run.

MAXTIM F2-HYDROD I*4

Number of steps of duration DT in F2 for which the mixed dimension model is to run.

MB CALNOD I*4

Second up/down stream segment to node K if it exists.

MBD Common /CTCH22/ 40I*4

Array holding water field identifiers for bay,constructed by subroutine FIND.

MC CALNOD I*4

3rd upstream segment to node K if it exists.

MD CALNOD I*4

Downstream segment to node K.

MD1 Common /A/ 9R*4

Primary downstream segment number connected to segment i is the i th element of MD1.

MD2 Common /A/ 9R*4

Secondary downstream segment(if it exists) number connected to segment i is the i th element of MD2.

MF F2-BOTANY I*4

First row of water field in a column N in the bay phase.

MFF F2-BOTANY I*4

Last land row before water starts in column N of the bay.

MIND Common /CTCH22/ I*4

Number of elements defined in array MBD through the bay geometry.

MINDO Common /CTCH22/ I*4

Number of elements of MOBD defined.

MINTAP F1/F2-HYDROD I*4

The minimum number of cycles of DT duration that the simulation must run before output to the intermediatory file starts.This is to safeguard runs of the pollutant transport program with transient data.

MLEF F2-BOTANY I*4

Left most row for which there is a column open water boundary.

MM F2-BOTANY I*4
Previous row in bay, ie M-1

MM1 F1-REST I*4
Number of segments for data on 'NIN' , should match with MMAX1.

MM2 F1-HYDROD I*4
Set to zero to retain compatability with t he full model.

MMAX Common /CTCH22/ I*4
Rows in the bay field.

MMAX1 Common /A/ I*4
Number of segments in the 1-dimensional phase.

MMAX1 Common /ALL/ I*4
Number of segments in river segment.

MMAX2 Common /ALL/ I*4
Number of rows in the bay

MMM F2-BOTANY I*4
Next row in bay, ie M+1

MOBD Common /CTCH22/ 9I*4
Array holding row open boundary identifier in the bay.

MP F1-HYDROD I*4
Segment number of height to be plotted on a graph.

MPRINT F1/F2-HYDROD 15*3I*4
Array holding do-loop parameters for printing of one dimensional predictions.

MRIG F2-BOTANY I*4
Right most row for whch there is a column open water boundary in the bay.

MQBD Common /CTCH22/ 9I*4
Array for row discharge boundary in the bay.

MQBDO Common /CTCH22/ i*4
Number of elements of MQBD defined by input.

MSCIP NODEMK 9I*4
Array to hold the segments already considered in node construction

MTME HYDROD I*4
The next cycle for which printing of river phase is required.

MTMEL F1/F2-HYDROD I*4
Number of do-loops to be read in to define line printer report frequencies. Up to 15 loops permitted currently.

MU Common /A/ 9R*4

A location to hold successive values of MU1.

MU1 Common /A/ 9R*4

The major upstream segment connected to segment i is the i th element of MU1 . If it is an upstream terminal boundary then should be set to 100 .

MU2 Common /A/ 9R*4

If it exists, the segment number of the second upstream segment flowing into segment i is in the i th element of MU2.

MU3 Common /A/ 9R*4

If it exists, the segments number of the 3rd segment flowing into segment i at its upstream end is in the i th element of MU3.

N F1-HYDROD I*4

The number of readings of the tidal elevation forcing function to be entered into array XSE. Up to 99 elements permitted.

N General index F1

Do-loop or indeces refers usually to n th interior grid point of a segment.

N General Index F2

Usually references a column N of the bay field.

NA F2-BOTANY I*4

Index for arrays MOBD,NOBD

NBD Common /CTCH22/ 40I*4

Array holding NIND column water identifiers for the bay as established in routine FIND.

NBOIFA F2-BOTANY I*4

Number of points in bay/ocean interface A.

NBOIFB F2-BOTANY I*4

Number of points in bay ocean interface B.

NBOT F2-BOTANY I*4

Lowest column for which there is a row open water boundary in bay.

NC F1-HYDROD I*4

Number of time steps in time units of DT to define a whole number of tidal cycles. Range- up to 3299 elements.

NCARD F2-BOTANY I*4

Steps , in terms of DT (F2-HYDROD) that span input tidal data in arrays XSE,XSEB.

NCD Common /IO/ I*4

Peripheral number of the card reader. Can also be used to input from disc or tape file,depending on operating system.

NDATA Common /A/ 9I*4

Array holding the number of depth-width data points available for each segment on input.

NDATAM HYDONE I*4

Temporary location for number of tabulation points of width of channel vs. depth of channel for each segment.

NDIV F1-HYDROD I*4

Time interval in multiples of DT between readings in XSE.

NDIV F2-BOTANY I*4

The number of data points in XSE,XSEB expected on input.

NDWNSG Common /NODE/ 9I*4

Array holding the number of the next downstream segment.

NEST Common /SPECS/ I*4

Total number of elements of EST array used to define the outline of the bay.Should always be an even number as there are x-y pairs to read in.

NGP Common /A/ 9I*4

Array holding the number of internal grid points per segment.

NGPM Common /A/ I*4

Number of internal grid points in segment M.

NIN F1-HYDROD I*4

Peripheral number of input to the restart facility, usually equal to NOUT so that the new run output is appended to the previous output.Note that on some operating systems this causes a destructive overwrite mode output unless specific precautions are taken.

NIN F2-HYDROD I*4

Data set reference number of the file holding restart data.

NIND COMMON /CTCH22/ I*4

Number of elements defined in array NBD.

NINDO Common /CTCH22/ I*4

Number of elements of NOBD used on input.

NM CALNOD I*4

Number of grid points in segment MD to node K.

NM2 F1-HYDROD I*4

Set to zero to retain compatability with the full mixed model.

NMAX Common /CTCH22/ I*4
Number of columns in the bay.

NMAX2 Common /ALL/ I*4
Columns in the bay

NMB CALNOD I*4
Number of grid points in second downstream segment to node K if it exists.

NN F2-Botany I*4
References the previous column of the bay field.

NN HYDONE I*4
Number of grid points of the most downstream (ie seaward) segment.

NNN F2-BOTANY I*4
References the next column to column N ,ie N+1.

NNODE Common /A/ I*4
The number of nodes in the one dimensional system .

NO F2-HYDROD I*4
Data set reference number of new generated intermediate file.

NOBD Common /CTCH22/ 9I*4
Array for a column open water boundary in the bay field.

NODE Common area
Array holding description of nodes of river segments.

NOUT Common /IO/ I*4
Peripheral number of reporting output stream for later input to the pollutant transport program.

NP F1-HYDROD I*4
Grid point number of segment MP for which heights are to be plotted on the CALCOMP device.

NPLOT F2-HYDROD 15*3I*4
Array holding do-loop parameters for plotting bay velocity vectors.

NPRINT F2-HYDROD 15*3I*4
Array holding do-loop parameters for bay data printing.

NQ QT 9I*2
Half word array holding extent of variable flow data for each segment in the river phase.

NQBD Common /CTCH22/ 9I*4
A column discharge boundary identifier in the bay.

NQBDO Common /CTCH22/ I*4
The number of elements of NQBD defined and entered

NRBIF F2-BOTANY I*4
Number of points in the bay/river interface.

NSEAB F2-BOTANY I*4
Zero if only one bay/ocean interface, 1 if two interfaces are present in the system.

NT1 F1-HYDROD I*4
Frequency of plotting to graph plotter, and of saving the boundary data in QS,YS and TN arrays. The units of NT1 are in steps of DT seconds.

NT1 F2-HYDROD I*4
Multiplier of the basic time step for which the river phase is to be run.

NT2 F2-HYDROD I*4
Multiplier of the basic time step for which the bay phase is to be run.

NTIMEL F2-HYDROD I*4
Number of implicit print do-loops input for the printing of bay predictions.

NTOP F2-BOTANY I*4
Highest column number for which there is a row open water boundary in the bay.

NTSTPS F2-BOTANY I*4
Number of time steps, in units of DT2 , between tidal input data points in XSE and XSEB.

NUM CALNOD I*4
Type of node integer key.

NUM F2-BOTANY I*4
Index for arrays NBD,MBD during computation of predictions.

NUPSTR Common /NODE/ 9I*4
Array holding number of upstream segment.

P F2-BOTANY 99R*4
Work array.

PLOT F1-HYDROD L*4
Logical variable, if .TRUE. , graphical output is stored during computation and a graph plot prepared to be plotted off line after run terminates.

PRT Common /ALL/ L*4
Flag to print or not print in pollutant program.

PSTIME	F1-HYDROD	R*4

Start to plot time, used if the restart facility is employed.

Q	F2-BOTANY	99R*4

Work array in 2D field solution.

Q	PRINT1	9*6R*4

Average over one timestep of length DT of flow in each segment i,grid or mesh point j.

Q1A	F2-HYDROD	R*4

Last inflow to bay from river phase over the interface points.

Q1B	F2-HYDROD	R*4

Most recent inflow to bay from river phase over interface.

Q2A	F2-HYDROD	R*4

Last inflow to river phase from bay over interface.

Q2B	F2-HYDROD	R*4

Most recent inflow to river from bay across interface.

QIN	F2-HYDROD	R*4

Actual flow input to bay from river phase across interface.

QMEAS	Common /A/	9R*4

Array holding fresh water flow into each segment at the current simulation time, in cubic feet per second.

QMN	F1-REST	R*4

Dump location for flow data in off NIN but not required for the restart.

QOUT	HYDONE	R*4

The averaged value of AVQ. Flow passed as output from river phase to input for bay phase.

QS	F1-HYDROD	99R*4

Array used to retain the computed out/inflow at the downstream end of the segment opening to the bay phase.This is for later reporting.This array receives data every NT1 time steps (as do TS and YS).

QS	QT	9*52*2R*4

Array to hold time varying flow data. Element (i,j,1) is flow in segment i, mesh point j at time (i,j,2) in cubic feet per second

R	F2-BOTANY	99R*4

Work array

RBIF	F2-BOTANY	5*2I*4

Array holding river bay interface points.

RESTRT F2-HYDROD R*4

Time ,in hours, at which the restart facility is to act.

RG CALNOD R*4

Reciprocal of G.

RT F1-HYDROD R*4

Restart time if the facility is to be invoked, if not should be set to zero.

S F2-BOTANY 99R*4

Work array

SBED Common /A/ 9R*4

Array holding the slope of the bed of each segment.

SBEDM CALEF R*4

Temporary location for bed slopes of segments.

SCALE Common /SPECS/ R*4

Scale of final vector plot in routine PLOT

SE Common /CTCH22/ 21*17R*4

Array holding last but one elevation from mean water height in the bay predictions.

SEA F2-HYDROD 21*17R*4

Array to retain last prediction of SE.

SEB F2-HYDROD 21*17R*4

See SE in /CTCH22/.

SEGP F1-HYDROD L*4

Logical variable set to .FALSE. if (MP,NP) is not to be plotted.

SEINV Common /INITDP/ R*4

The initial water surface level in the river phase and ,in subroutine BOTANY, the intial level in the bay phase.

SEP Common /CTCH22/ 21*17R*4

Array holding latest predictions of water surface elevation deviation from the mean level.

T F2-BAYOUT 18A*1

Array holding single character identifiers for columns in bay, labelled 0 to 9,A,B,C,D,E,F,G,H. to use on output.

TC	F2-BOTANY	R*4

Time of most recent tide reversal at the river bay interface.

TCL	F2-BOTANY	R*4

Time of previous tide reversal at the river bay interface.

TD	F2-BOTANY	R*4

Duration of last tidal phase in hours (TC-TCL)

TEMP	F2-HYDROD	R*4

Temporary shift register

TEMP1	F2-BOTANY	R*4

Temporary locations usually occupied by elevations.

TEMP10	F2-BOTANY	R*4

Temorary location,usually holding a velocity.

TEMP11	F2-BOTANY	R*4

Temporary location,usually holding a velocity.

TEMP2	F2-BOTANY	R*4

Temporary location for elevations.

TEMP3	F2-BOTANY	R*4

Temporary location for elevations.

TEMP4	F2-BOTANY	R*4

Temporary location for elevations.

TEMP9	F2-BOTANY	R*4

Temporary location for elevations.

TERM	F2-BOTANY	R*4

Temporary location ,usually sum of depths about a point (N,M) .

TIDET	F1-HYDROD	L*4

Logical variable , if .TRUE. , tests are made to determine predicted times of tide turning in odd numbered segments.

TIME	F1-REST	R*4

Real time in hours at which the restart facility is to apply.

TIME1A	F1/F2-HYDROD	I*4

Time of last run of river phase in units of the basic time step.

TIME1B	F1/F2-HYDROD	I*4

Time of next run of river phase.

TIME2A	F2-HYDROD	I*4

Time of previous run of bay phase in units of the basic time step.

TIME2B	F2-HYDROD	I*4

Time of next run of bay phase.

TN F1-HYDROD 99R*4
Array to retain the real time of data saved in QS and YS.

TS F1-HYDROD R*4
Time step of simulation in hours.

TT Common /TT/ R*4
Current time from start of simulation, in hours.

TT F1-HYDROD R*4
Current time in hours from start of river simulation.

TTM F1-HYDROD R*4
Simulation length in hours of real time.

U Common /A/ 9*6R*4
Array whose i,j th element is the current velocity in segment i, mesh or grid point j , in feet per sec.

U Common /CTCH22/ 21*17R*4
Last but one estimate of the velocity in x direction in the bay(East)

U2 Common /CTCH22/ 21*17R*4
See U in common /CTCH22/.

U2 F2-BOTANY R*4
U velocity adjusted for a wind effect.

UA F2-HYDROD 21*17R*4
Array to retain last but one prediction of U in the bay.

UA1 F1-HYDROD 9*6R*4
Array whose i,j th element is the previous times velocity in seg. i,mesh point j , in feet per sec.

UA1 F2-HYDROD 9*6R*4
Array to retain last but one prediction of U in river phase.

UB F2-HYDROD 21*17R*4
Last but one prediction of U in bay.Equivalenced to U2 of /CTCH22/.

UMEAS HYDONE R*4
Calculated velocity of fresh water from flow input/areas.

UP Common /CTCH22/ 21*17R*4
Most recent estimate of East direction velocity component (positive x-direction).

USAV1 F2-HYDROD 9*6R*4
Temporary array to save U from river phase.Partially overlaid by UP in /CTCH22/.

V Common /CTCH22/ 21*17R*4
Last but one estimate of the Y-component velocity (North).

V2 F2-BOTANY R*4

V-velocity adjusted for wind term.

VA F2-HYDROD 21*17R*4

Array to retain last but one prediction of V in the bay.

VB F2-HYDROD 21*17R*4

Last but one prediction of V in bay.Equivalenced to V in /CTCH22/.

VP Common /CTCH22/ 21*17R*4

Most recent estimate of the Y-direction velocity component.

W F1-REST R*4

See WJ, dump location for BDN data.

WJ F1-REST R*4

Dump location for read in data not saved for restart.

W1J F1-REST R*4

Dump location for HA data.

XB F2-PLOT R*4

X co-ordinate of base of velocity vector in bay field.

XD F2-PLOT R*4

X displacement of velocity vector.

XIA F2-BOTANY R*4

Current height of tidal input function at interface A.

XIB F2-BOTANY R*4

Current height of tidal input function at interface B.

XLNGTH Common /A/ 9R*4

Array holding length of each segment, in feet.

XP F1-HYDROD R*4

The x-co-ordinate of a plot pair (x,y).

XSE F1-HYDROD 99R*4

Array to hold input forcing function. Tidal data is in feet at downstream end of segment MMAX1.

XSE F2-BOTANY 88R*4

Tidal forcing function at interface A.

XSEB F2-BOTANY 88R*4
Tidal input function for interface B (bay/ocean).

XSEM F1-HYDROD R*4
Maximum value of the tidal forcing function over one cycle.

XSTRSS F2-BOTANY R*4
Wind stress coefficient in X (east) direction.

XX F2-BOTANY R*4
X direction wind stress in a time step

Y Common /A/ 9*6R*4
Array of depths in segment i,grid point j, of water above the channel bottom in the river segments.

Y1A F2-HYDROD R*4
Last height input to bay from river interface .

Y1B F2-HYDROD R*4
Next height input to bay from river.

Y2A F2-HYDROD R*4
Last height input to river phase at interface.

Y2B F1-HYDROD R*4
Height of tidal forcing function at current time step.Passes to HYDONE.

Y2B F2-HYDONE R*4
Next height input to river phase at interface.

YA Print1 9R*4
Average elevation per segment.

YA1 Common /INITDP/ 9*6R*4
Heights in one dimensional river segments for previous time step.

YA1 F2-HYDROD 9*6R*4
Array to retain last but one prediction of Y in river phase.

YB F2-PLOT R*4
Y-co-ordinate of base of velocity vector.

YCOR Common /COR/ 9*6R*4
Array to correct predictions for relative elevations.

YD F2-PLOT R*4
Y displacement of velocity vector head from base.

YDATA	Common /INITDP/	9R*4	Array holding depth of channel bottom below datum at the downstream end of segments.
YGIVEN	Common /A/	9*11R*4	Array of data input through DATA,tabulating depths of channels corresponding to widths in BGIV.
YIN	F2-HYDROD	R*4	Actual water level sent from bay to river across interface.
YMEAS	Common /A/	R*4	Alternative storage location for input forcing function YIN.
YMH	CALHBC	R*4	Location to hold depth of water for segment M, grid point N.
YOUT	F2-BOTANY	R*4	Predicted height of water across interface from bay to river.
YP	F1-HYDROD	R*4	The Y-co-ordinate of a plot pair (x,y).
YRATIO	CALHBC	R*4	Ratio of (actual water depth-nearest tabulated value in YGIVEN) and (depth difference between adjacent points).
YS	F1-HYDROD	99R*4	Array used for height at downstream end of segment, saved at steps of NT1 loops.
YSAV1	F2-HYDROD	9*6R*4	Temporary array to save Y from river phase.
YSE	F1-HYDROD	3299R*4	Array holding tidal forcing function in steps of DT from 0.
YSTRSS	F2-BOTANY	R*4	Wind stress coefficient in Y-direction (North).
YT	CALHBC	R*4	Given values of depth for segment M,this is table value I.
YUP	HYDONE	R*4	The upstream end height of water in a segment. Should have a lower limit on it.
YY	F2-BOTANY	R*4	The Y-directional wind stress in unit time.

APPENDIX C

Two programs for Pollutant Transport in a General Estuary System - Second Phase to the F1/F2 Models of Appendix B

APPENDIX C - CONTENTS

C.1 Introduction

There are two models, PT1 and PT. PT1 is the purely one dimensional model and natural sequel to F1, whereas PT is the full mixed dimension model and the second phase of F2. Also, model PT1 can be run from the file generated by F2 using an interface to sift out superfluous data. PT1 has some additional options to PT as core store room was available. The methodology is very simple, using previously computed velocities, and blocks of water are moved in accordance. A simple kinetic B.O.D./D.O. mechanism is superimposed to estimate water quality at extremes of tidal cycle (high/low water). To make the model more time dependant and estimate quality every few minutes multiplies the run time by a factor of 2-3 so making it very expensive if mill time is a capital expenditure item.

C.2 Model Composition

PT1 = those modules marked 'for PT1' or not specifically 'for PT' see table C.2.A.

PT = all those modules not marked 'for PT1' see table C.2.A.

TABLE C.2.A. Models PT - PT1 Module Composition

BLOCK DATA	for PT1	BLOCK DATA	for PT
POLTRA	for PT1	POLTRA	for PT
RIVQAL	for PT1	RIVQAL	for PT
BAYQUA	for PT1	BAYQUAL	for PT
TAPECH	for PT1	TAPECH	for PT
BRANCH		BRANCH	
DIFUSE		DIFUSE	
MAP		MAP	
XCPRNT		XCPRNT	
PLTRIV		PLTRIV	
		PLTBAY	for PT
		PTSUBY	for PT
		TRNSPT	for PT

C.3 Subroutine Block Data (for PT1)

Function	To initialise some common areas for Model PT1.
Common Areas	/ALL/, /IO/, /BAY1/, /RIV1/, /D/, /DISCHG/
Common Variables set	All in /DISCHG/, /RIV1/, /D/, /BAY1/ Most in /IO/, /ALL/
Description	Most of the variables and arrays are set to default values.

C.4 Subroutine Block Data (for PT)

As for PT1 but additional common areas of /XC/, /UVCP/, extended /BAY1/

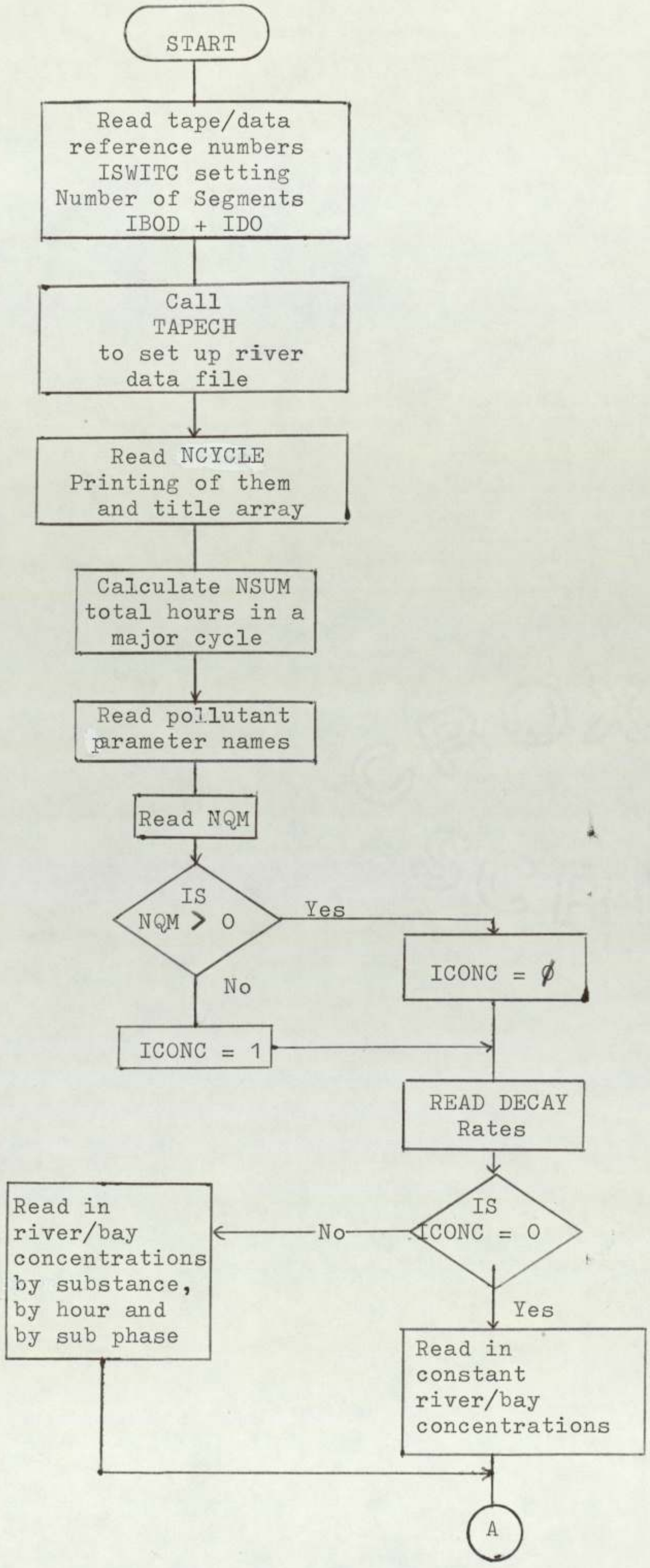
C.5 Executive Routine Poltra (for PT1)

Function	To act as a control program to run the model PT1 against input data from a data stream.
System Functions	READ/WRITE I/O, REWIND, TIME
User Functions	TAPECH (for PT1), RIVQAL (for PT1) BAYQUA (for PT1)
Common Areas	/IO/, /ALL/, /BAY1/, /CONSTI/, /D/, /WORK/
Common Variables Reset	Most of /IO/, /ALL/, all of /BAY1/, /WORK/, /CONSTI/
Error Messages	None

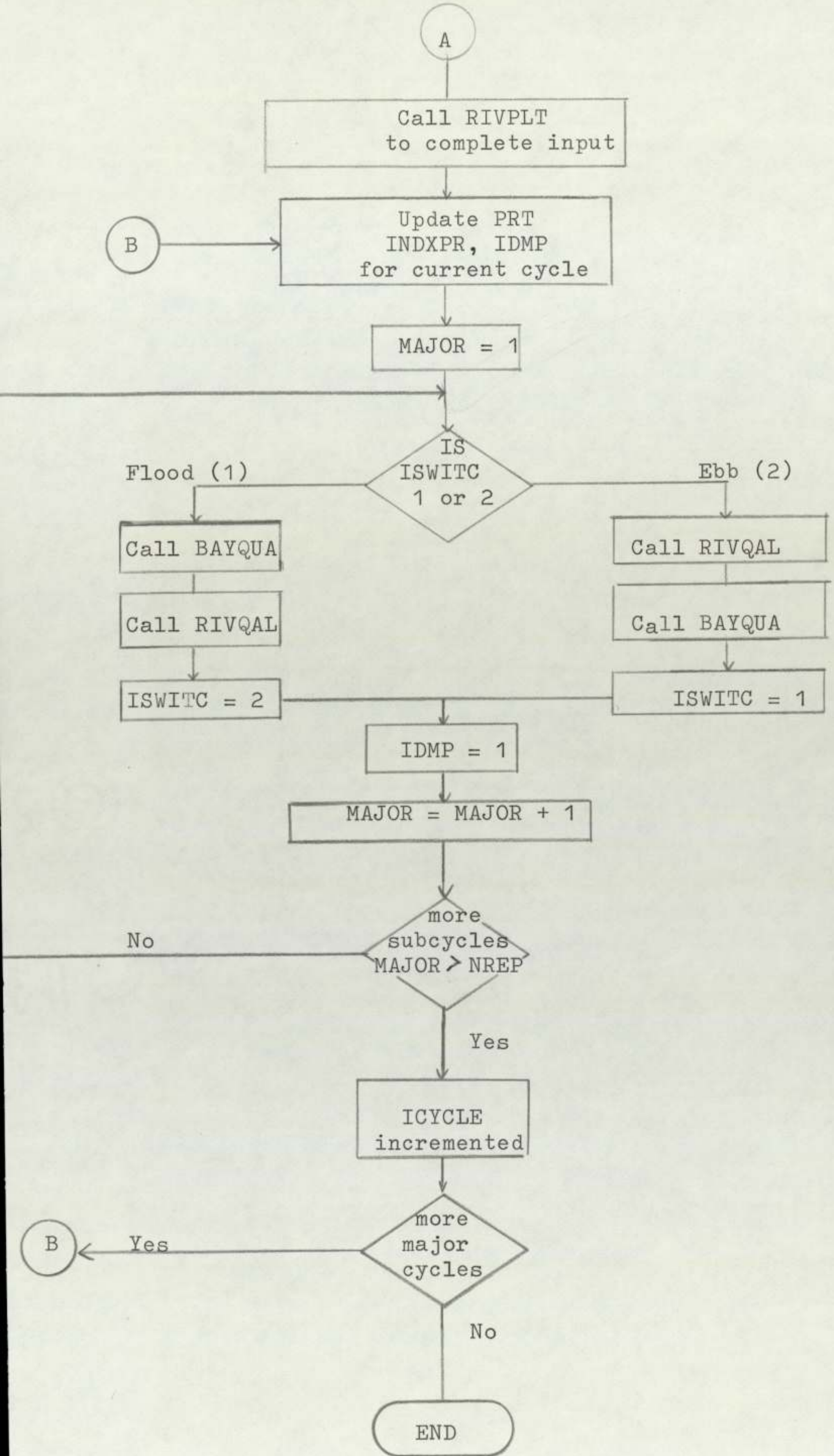
Description

Some run parameters are read in, especially the data reference to the temporary file produced by F1 or F2. TAPECH is called to set up an unformatted file of relevant river data. The remaining executive data is read in, and a dummy call is made of river system and discharges. The loop with index MAJOR is the sub cycle loop, the main loop being controlled by ICYCLE. When NCYCLE cycles have been run the adjusted transfer coefficients are printed and execution terminates.

Flow Logic



START
Read tape/data reference numbers ISWITC setting Number of Segments IBOD + IDO
Call TAPECH to set up river data file
Read NCYCLE Printing of them and title array
Calculate NSUM total hours in a major cycle
Read pollutant parameter names
Read NQM
IS NQM > 0
Yes
No
ICONC = Ø
ICONC = 1
READ DECAY Rates
IS ICONC = 0
No
Yes
Read in river/bay concentrations by substance, by hour and by sub phase
Read in constant river/bay concentrations
A


A
Call RIVPLT
to complete input
Update PRT
INDXPR, IDMP
for current cycle
B
MAJOR = 1
IS
ISWITC
1 or 2
Flood (1)
Ebb (2)
Call BAYQUA
Call RIVQAL
ISWITC = 2
Call RIVQAL
Call BAYQUA
ISWITC = 1
IDMP = 1
MAJOR = MAJOR + 1
more
subcycles
MAJOR > NREP
No
Yes
ICYCLE
incremented
more
major
cycles
B
Yes
No
END

C.6 Executive Routine POLTRA (for PT)

Function	To provide an executive routine for the model PT
System Functions	READ/WRITE I/O, REWIND, TIME
User Functions	TAPECH (for PT), RIVQAL (for PT), BAYQUA (for PT)
Common Areas	/ALL/, /IO/, /BAY1/ (for PT), /CONSTI/, /D/
Common Variables Reset	Most of /IO/, /ALL/, all of /BAY1/, /CONSTI/
Error Messages	Problem identifiers don't match. Title from F2 output and from data set doesn't match. Warning only.

Description

As POLTRA for PT1, but additional executive information for bay is required for the extent of the water field. A dummy call to BAYQUA is made to read in bay data. The main iterative section is as for the PT1 model except the PT versions of each routine is called.

Flow Logic

As for POLTRA, for PT1, without ICONC option as ocean concentrations are regarded as constant in time.

C.7 <u>Subroutine RIVQAL</u> (for PT1)

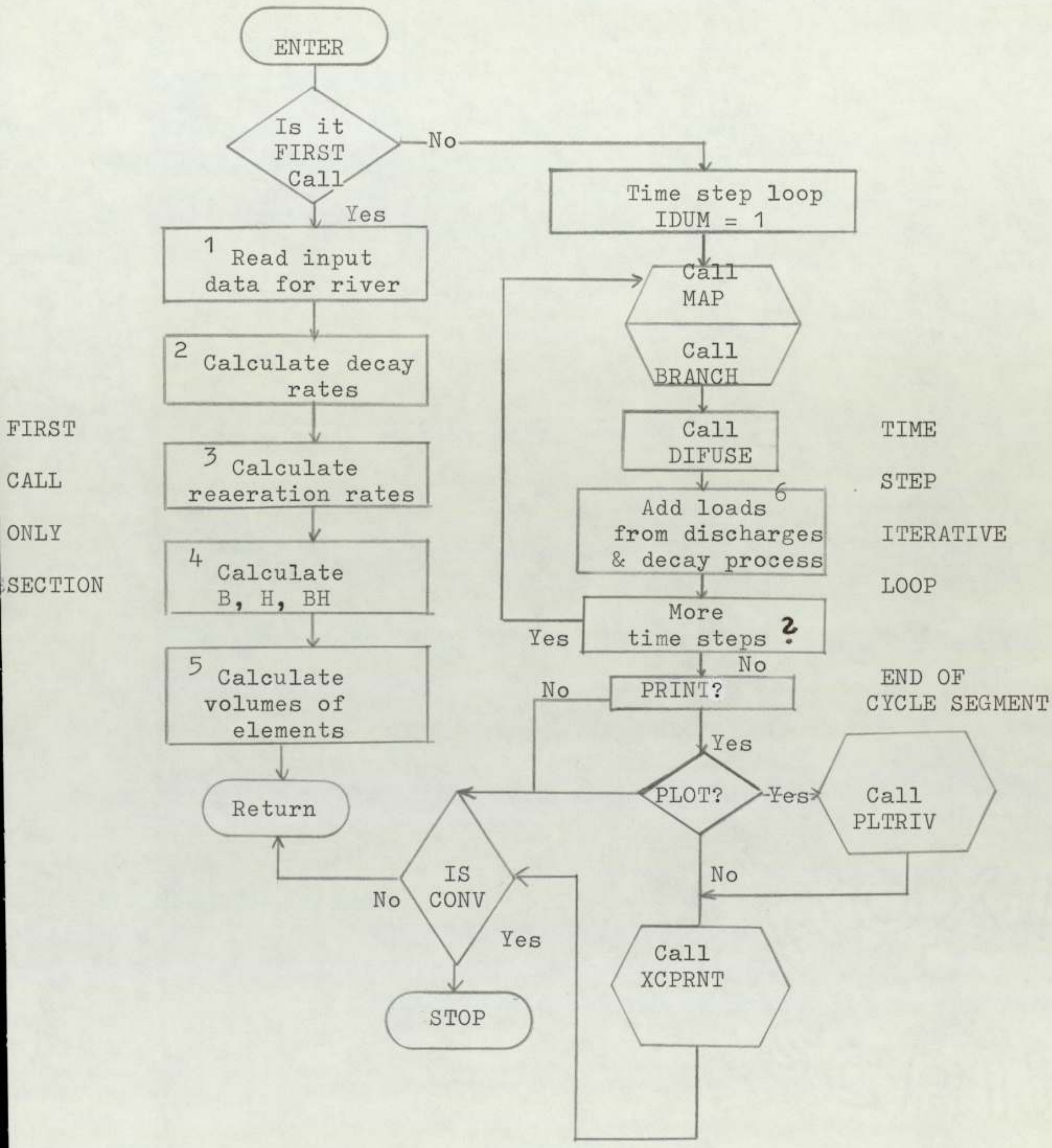
Function	To execute a timestep for the river phase of the model, and to set up certain arrays on the first call.
System Functions	READ/WRITE I/O, FLOAT, EXP, ABS, TIME
User Functions	BRANCH, MAP, DIFUSE, XCPRNT, PLTRIV
Error Message	STOP 201 - Arrays B, H & BH not large enough STOP 1155 - System converged before NCYCLE cycles had been simulated
Common Areas	/IO/, /WORK/, /ALL/, /RIV21/, CONSTI/, /CHABDN/, /DISCHG/, /RIV1/
Common Variables Reset	Most of /WORK/; all of /RIV1, /RIV2/, /DISCHG/, /CHABDN/

<u>Description</u>

A large section is performed only on the first call of the routine, usually a dummy call from the executive routine. This section commences with input of river and discharge parameters. Then the water levels are read back from tape ITP3 and arrays B, H and BH calculated for each segment for each time step. Reaeration rate is calculated from the decay rate and mean water depths for each hour of each subcycle for each segment. Initial volumes of each segment are also calculated.

The main repetition cycle begins at statement label 200. The DO 500 loop is the simulation loop for each time increment. The bay conditions provide the input to CNODE for the last node (label MMAX^{P}1). MAP is called for each segment to move elements, and BRANCH to move elements between segments . DIFUSE for each segment diffuses the results of the previous water movements. The rest of the time step loop adds the discharge inputs to respective elements. Decay is calculated and net concentrations adjusted. When a whole sub cycle has been simulated, optional print or plats are obtained. The PT1 version also tests for convergence of consecutive similar phases. Finally the concentrations passed to the bay are printed.

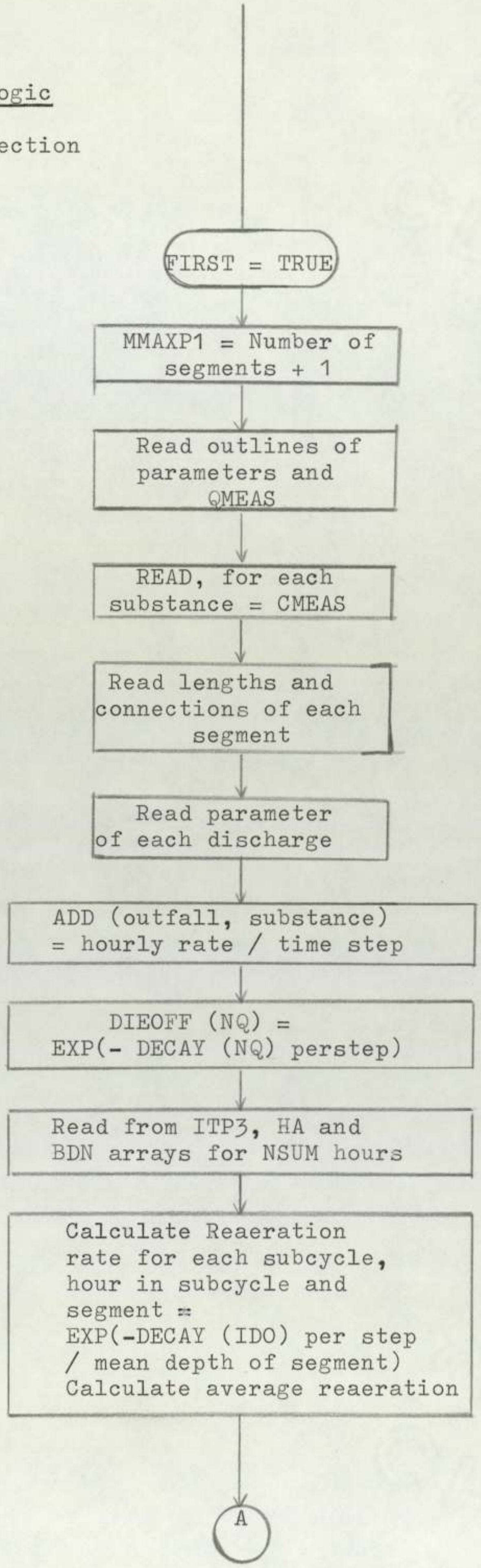
Outline Flow Logic



Detailed Flow Logic

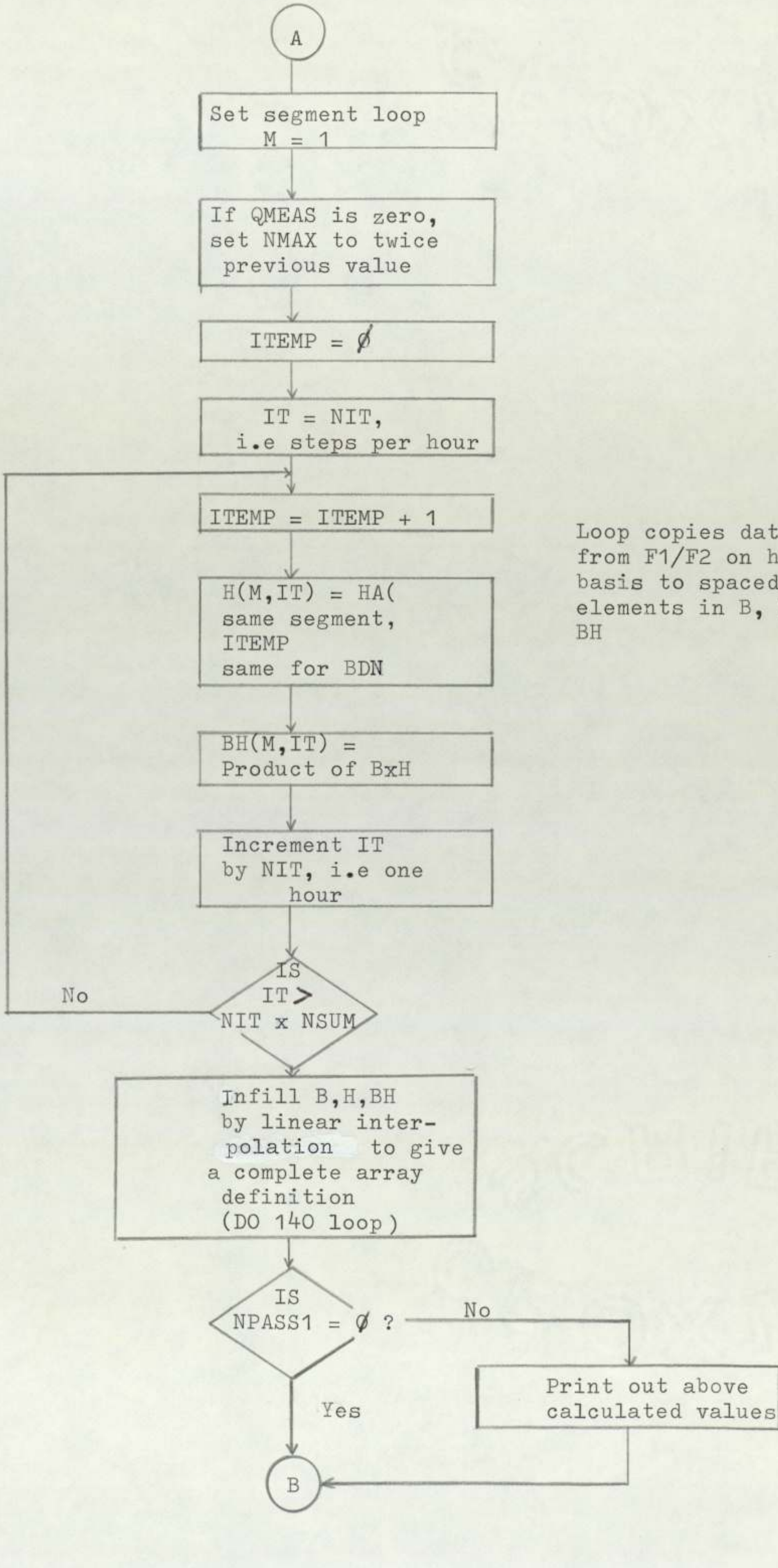
1st call only section

Box 1 - 2 - 3



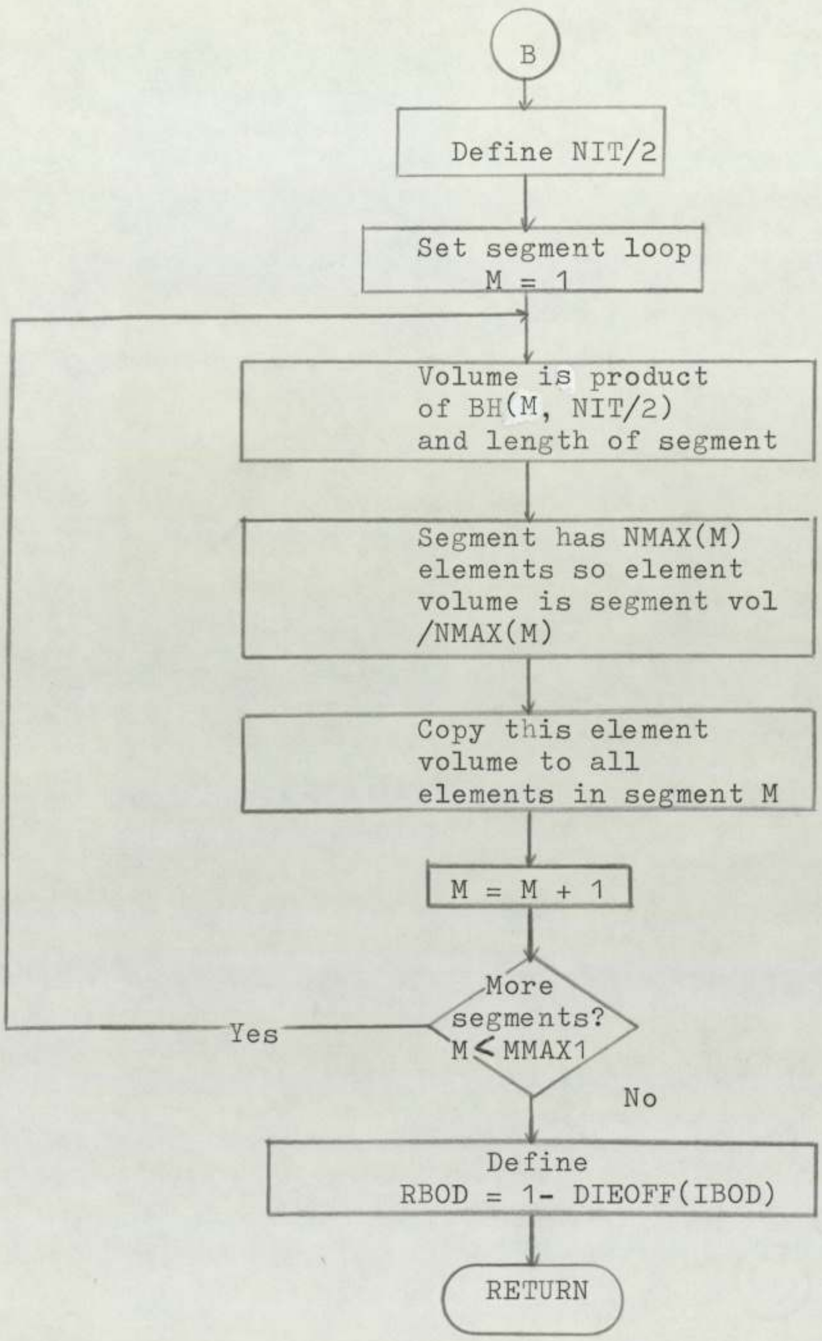
Input of parameters for river / discharge section

1st Call only section
Box 4

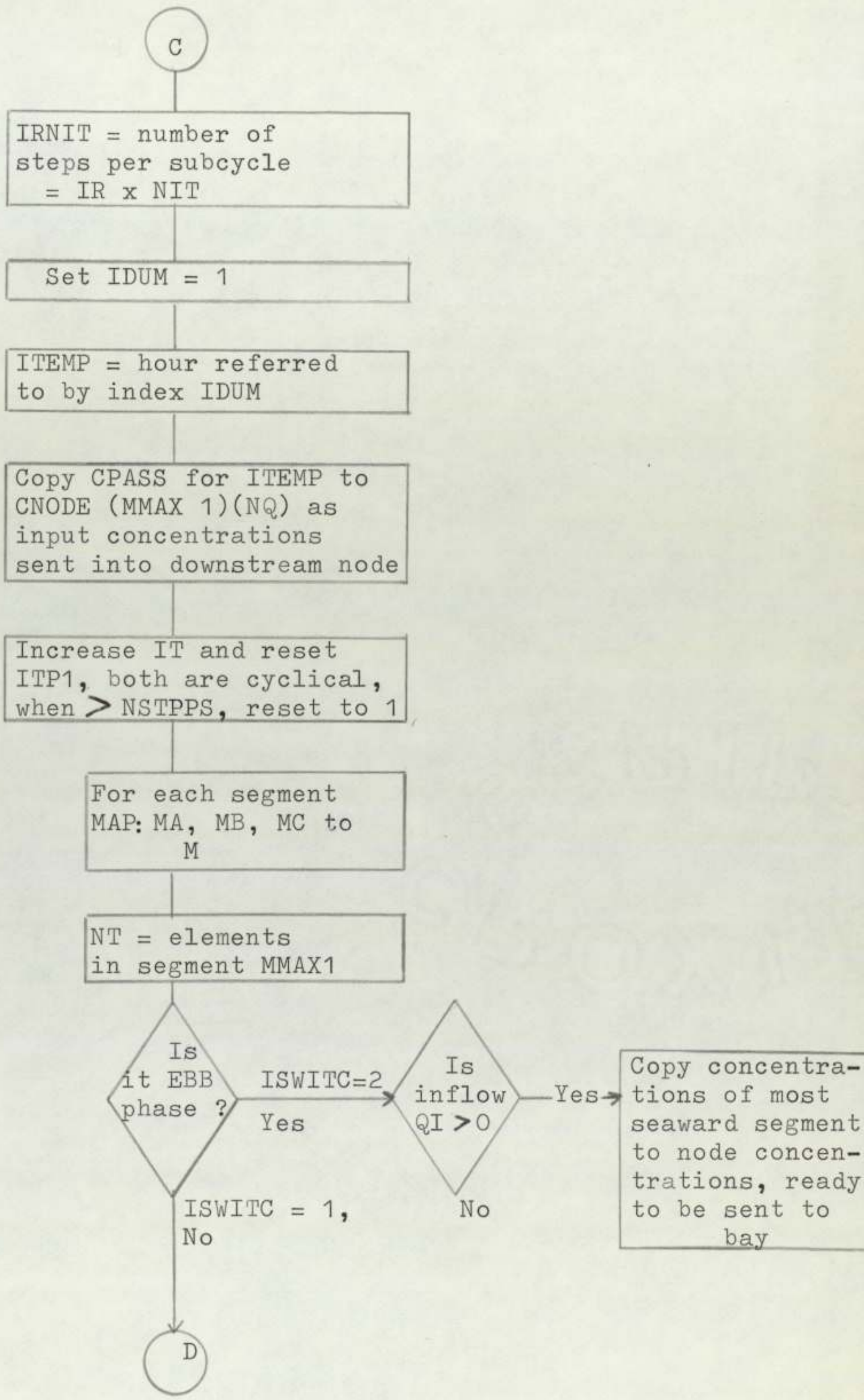


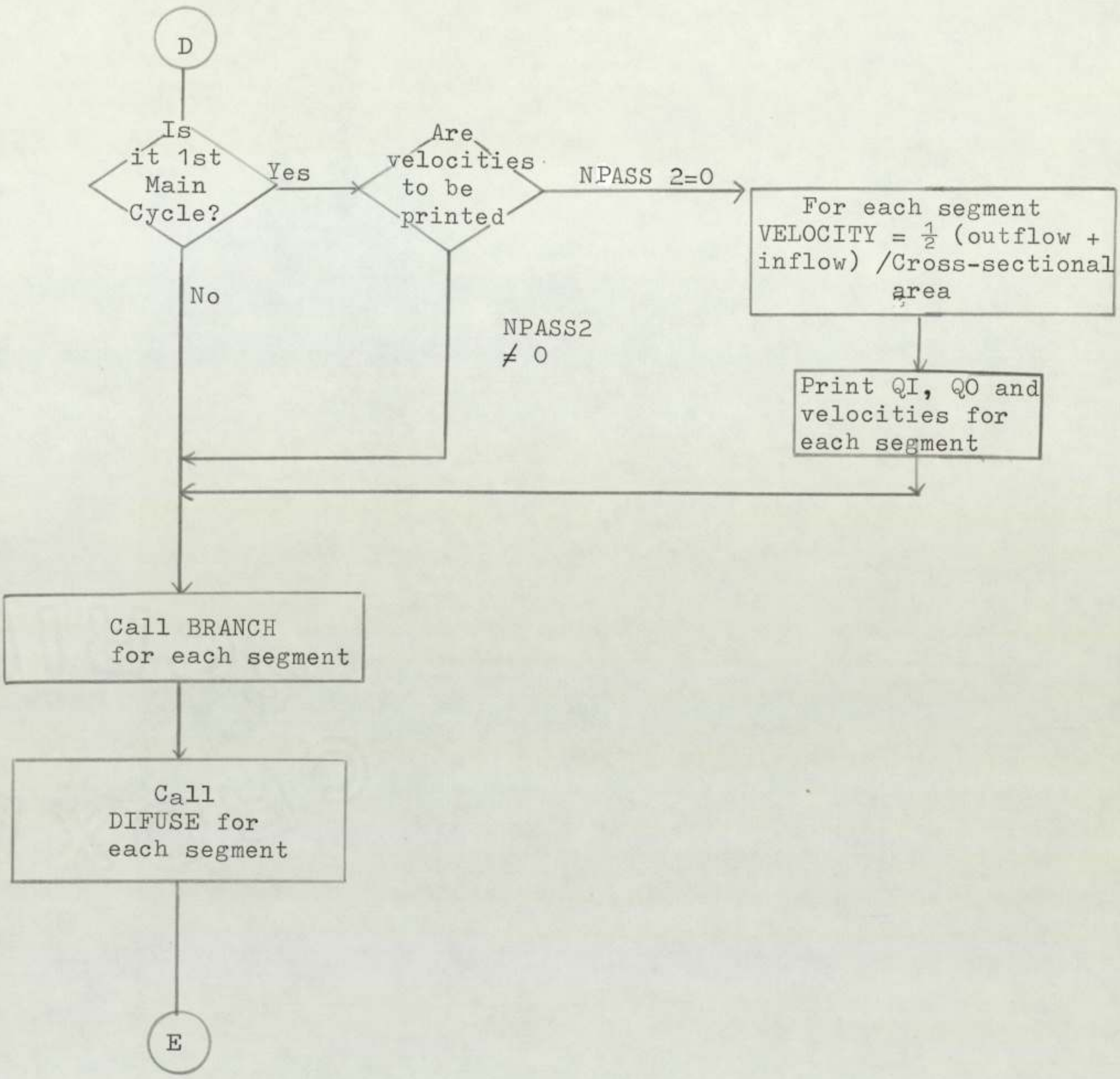
Loop copies data from F1/F2 on hourly basis to spaced elements in B, H and BH

1st call only section
Box 5

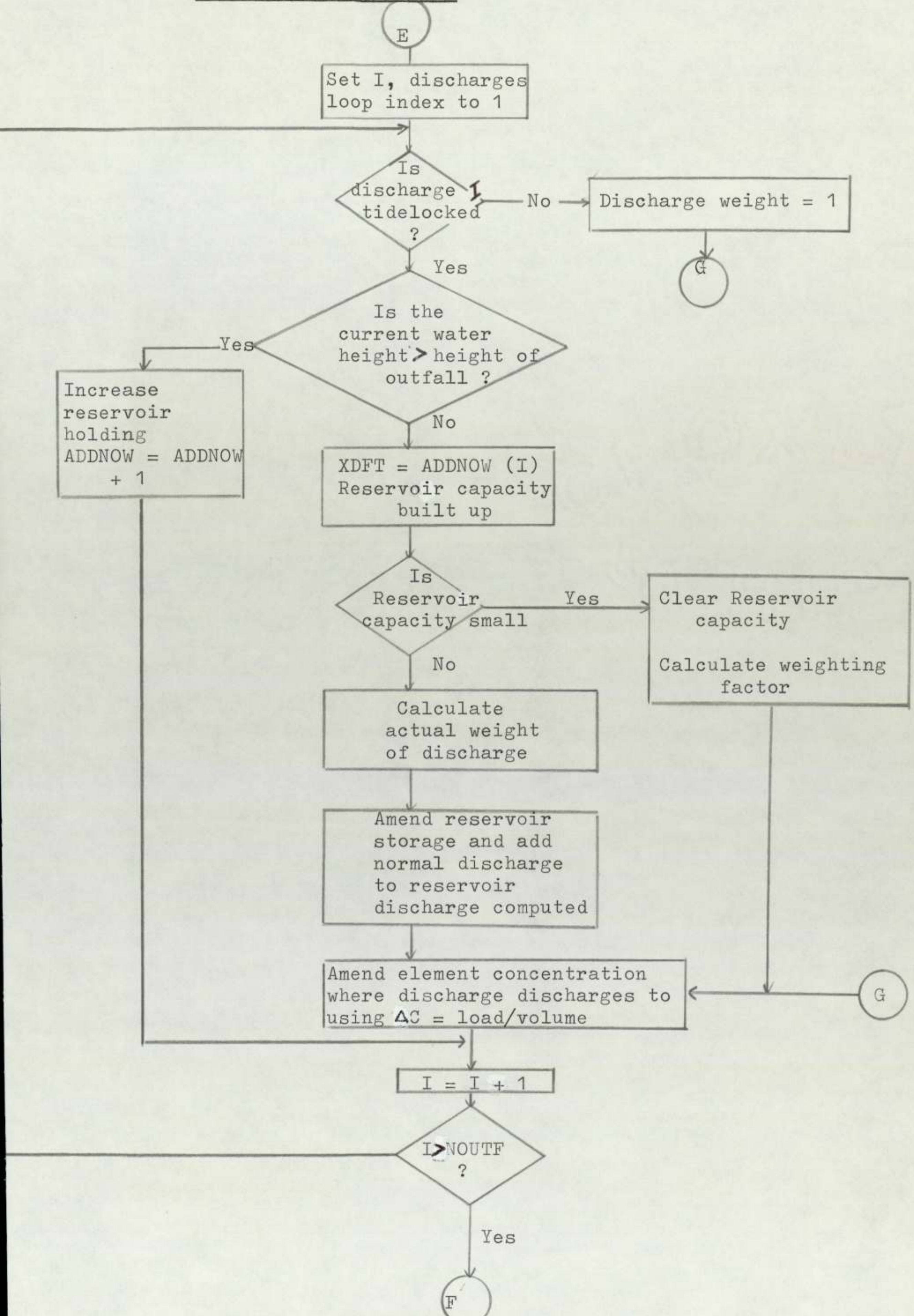


Time step iterative loop

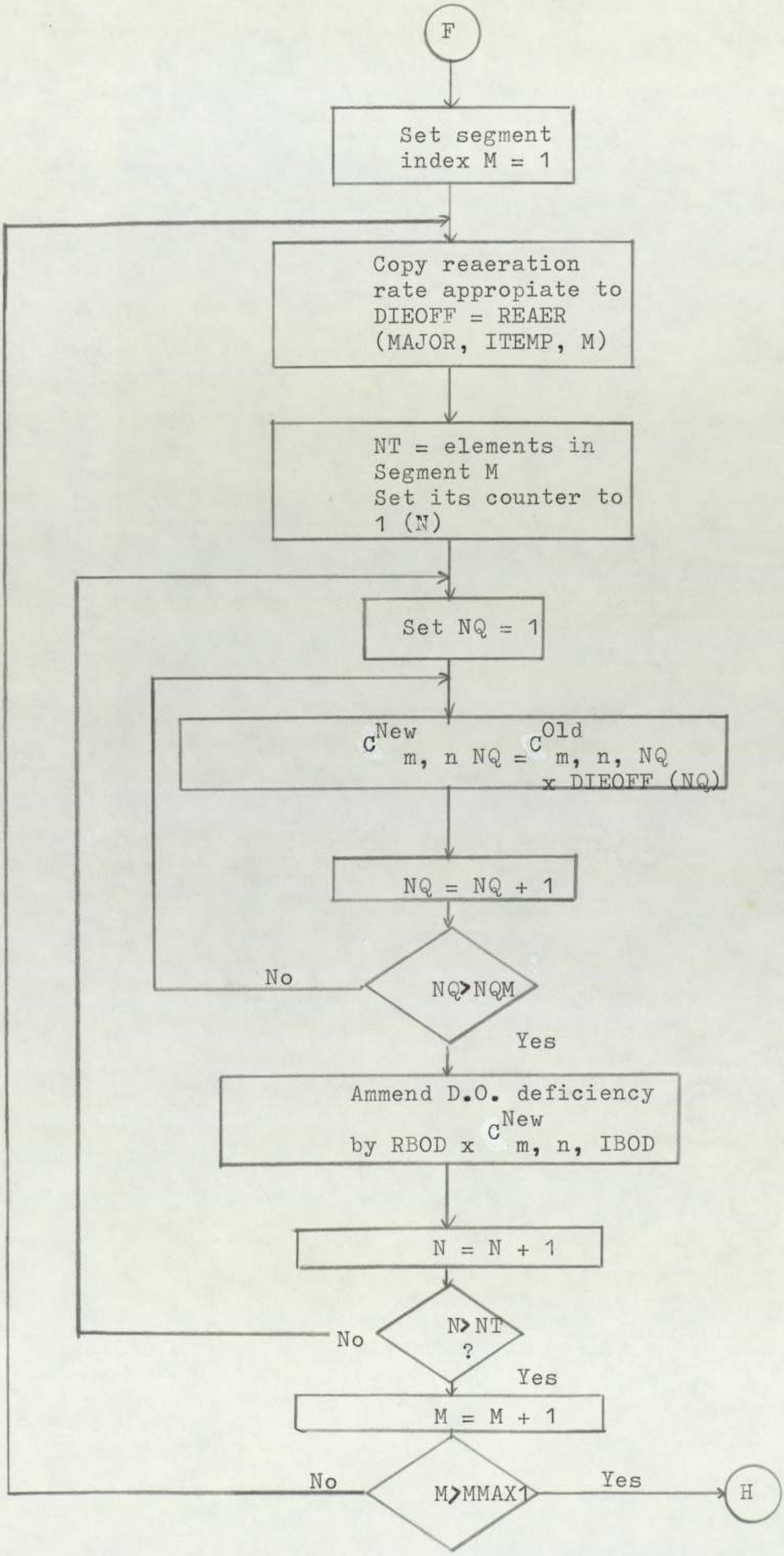



D
Is it 1st Main Cycle?
Yes
No
Are velocities to be printed
NPASS 2=0
NPASS2 ≠ 0
For each segment VELOCITY = ½ (outflow + inflow) /Cross-sectional area
Print QI, QO and velocities for each segment
Call BRANCH for each segment
Call DIFUSE for each segment
E

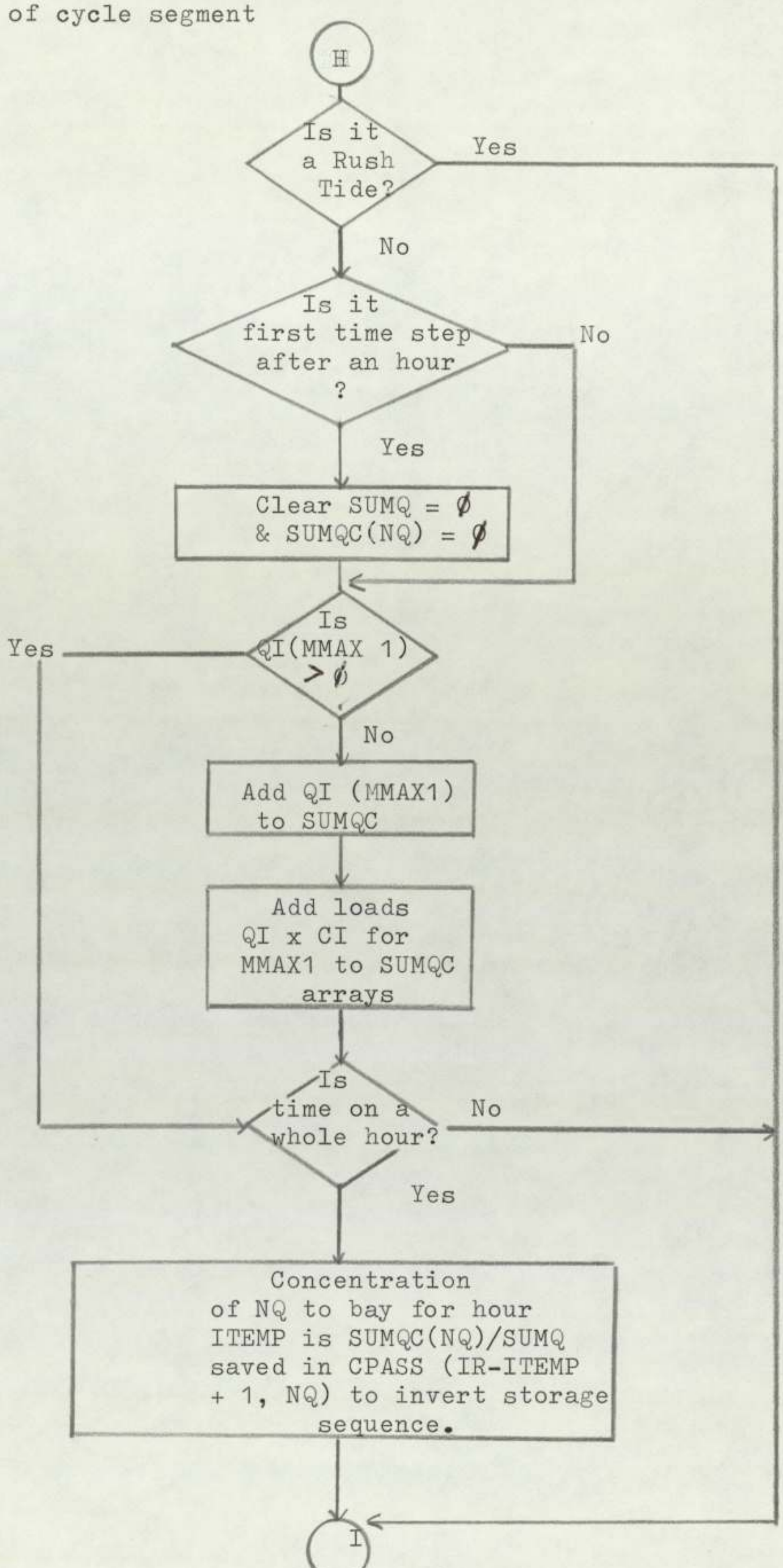
Box 6 Adding discharge loads

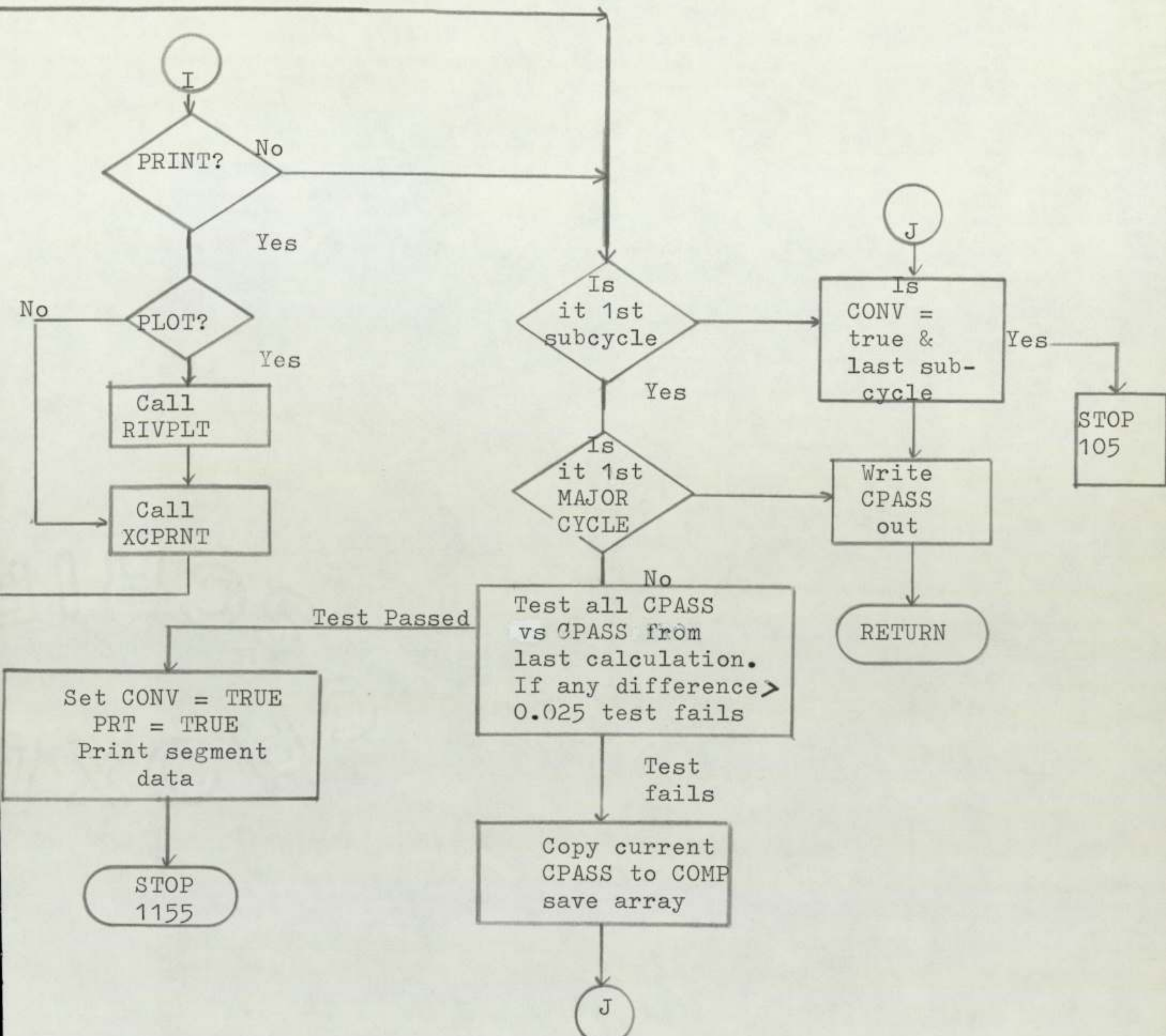


Box 6 'Decay' Process



End of cycle segment





I
PRINT?
No
Yes
PLOT?
No
Yes
Call RIVPLT
Call XCPRNT
Is it 1st subcycle
Yes
Is it 1st MAJOR CYCLE
No
Test all CPASS vs CPASS from last calculation. If any difference > 0.025 test fails
Test Passed
Set CONV = TRUE PRT = TRUE Print segment data
STOP 1155
Test fails
Copy current CPASS to COMP save array
J
J
Is CONV = true & last sub-cycle
Yes
STOP 105
Write CPASS out
RETURN

C.8 Subroutine RIVQAL (for PT)

Function	To execute a timestep for the river phase of the model PT and set up an array for input to the bay phase on alternate subcycles.
System Functions	READ/WRITE I/O, FLOAT, EXP
User Functions	BRANCH, MAP, DIFUSE, XCPRNT, PLTRIV
Error Messages	STOP 201 . Arrays B, H or B H out of limits STOP 200 . Inter polation out of bounds
Common Areas	/IO/, /ALL/, /RIV1/, /RIV2/, /CONSTI/, /CHABDN/, /DISCHG/
Common Variables Reset	All of /RIV1/, /RIV2/, /DISCHG/, /CHABDN/
Description	Identical to RIVQAL for PT1 except that two options are not provided : 1. Time /space variant reaeration rates. 2. Convergence of river interface concentrations.
Flow Logic	The flow logic is identical to that of RIVQAL for PT1 except for the sections dealing with the two exceptions listed above.

C.9 Subroutine BAYQUA (for PT1)

Function	To act as boundary concentration feeder for routine RIVQAL for PT1
System Functions	WRITE
User Functions	None
Common Areas	/ALL/, /BAY1/, /CONSTI/
Common Variables Reset	CPASS in /ALL/

Description

A whole RUSH tide phase's interface concentrations are set up in a short module to retain compatability with PT. Array CONCBY (t, major, nq) in the boundary concentration of pollutant NQ during sub cycle MAJOR at time IT (hour in sub cycle) and selective items passed to CPASS (t, nq) as appropriate by the phase being run.

C.10 Subroutine BAYQUA (for PT)

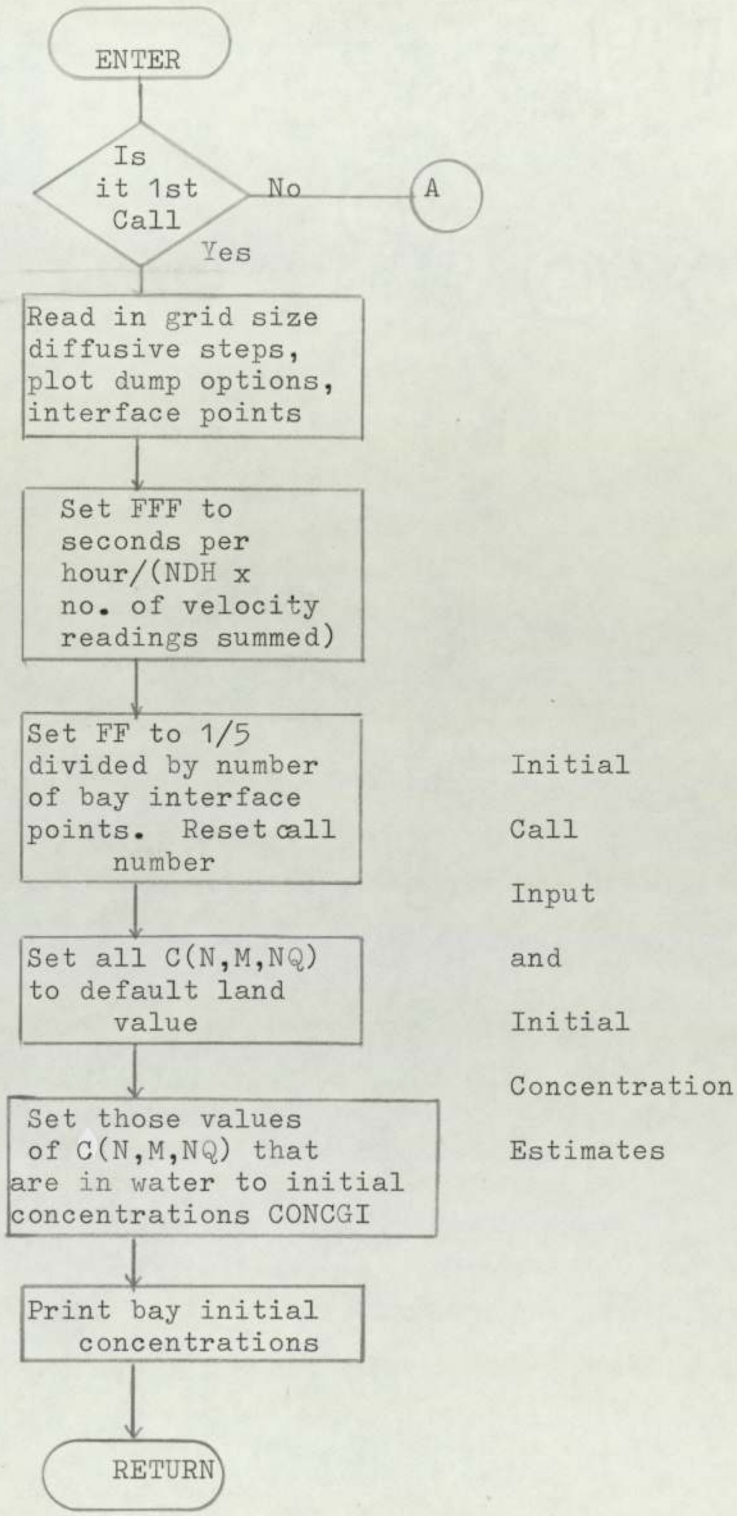
Function	To provide a model for pollutant transport in a two dimensional grid
System Functions	READ/WRITE I/O, FLOAT, EXP,
User Functions	TRNSPT, PLTBAY
Common Areas	/IF/, /IO/, /ALL/, /BAY1/,/CONSTI/, /UVCP/, /TRANS/, /BAYC/
Common Variables Reset	All of /IF/, /TRANS/, /BAYC/, /BAYC/, /BAY1/, /UVCP/, some of /ALL/ IDMPB of /IO/

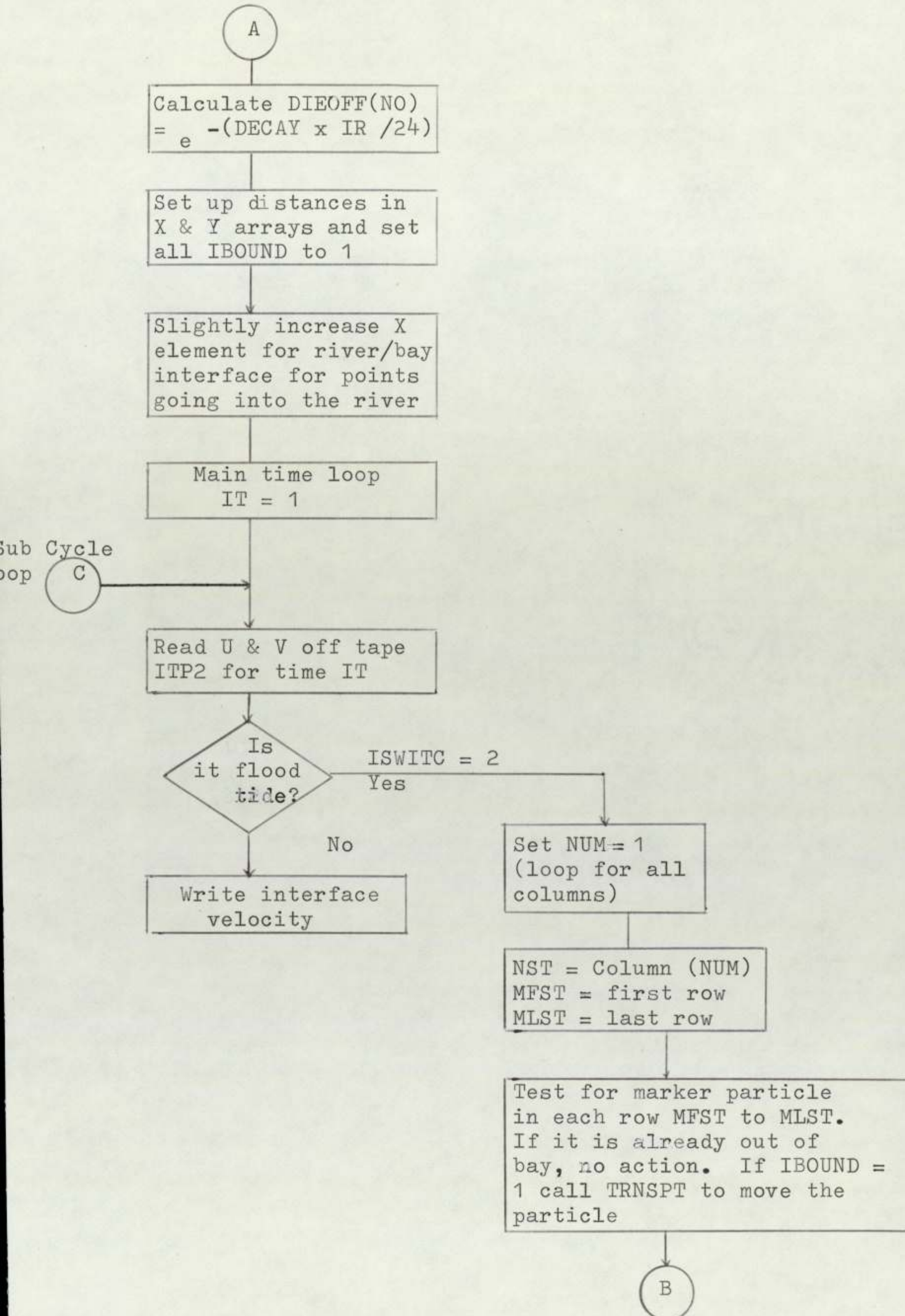
Description

Again, a major part of the program is executed only for completion of input and setting some initial values. An intial estimate of C (N, M, NQ) is made and land points set to 999. Decay factors are set up in DIEOFF, simularly to RIVQUA. Arrays Y and X are set to initial positions in the bay from (0,0). The iterative section starts at statement DO 300, looped once for each simulation hour. The U/V velocities are read off temporary file reference ITP2 then the locations of points terminating in the bay are moved through routine TRNSPT. If this is a flood cycle, clear CPASS and move interface points and establish initial value of CPASS.

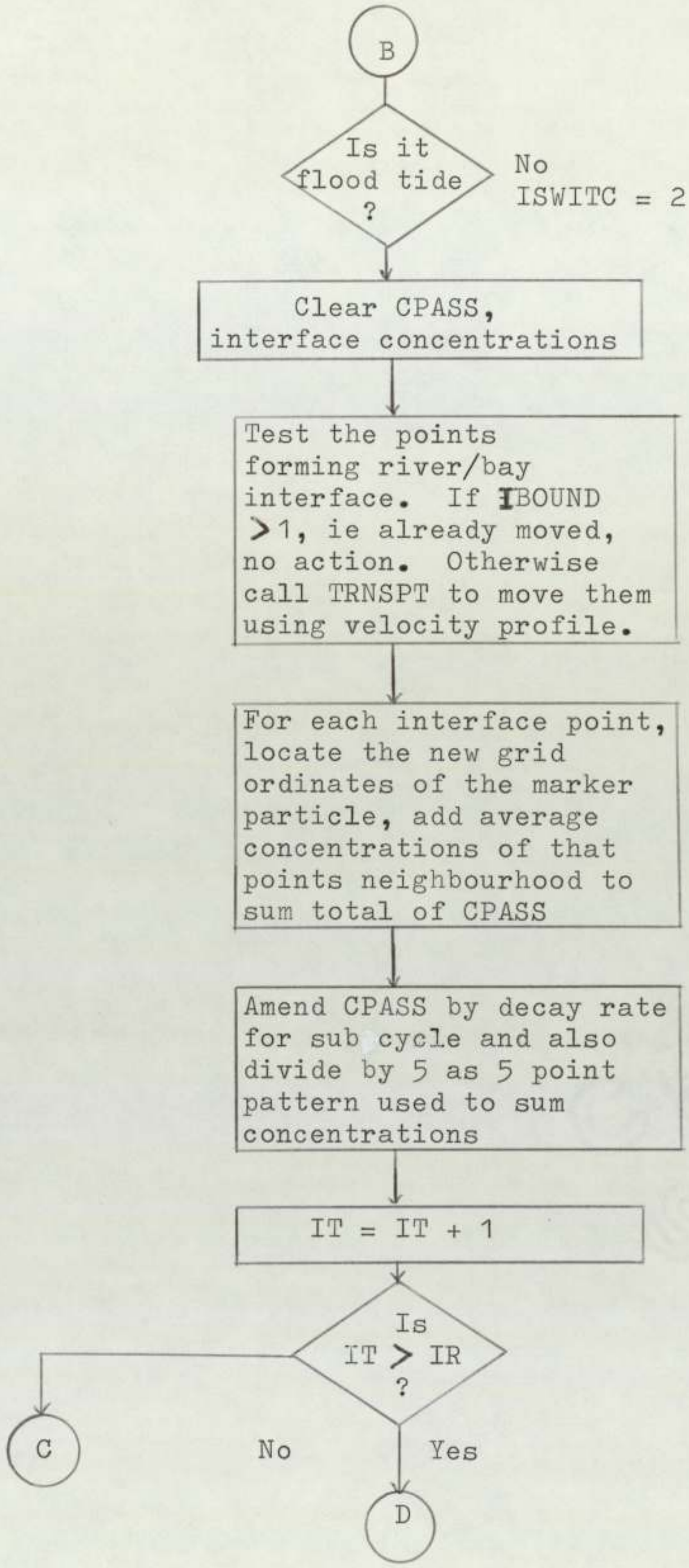
Once the movement of water has been completed for all IR hours in the subcycle, new values of C are computed with some diffusion allowances. There is some diffusion between river/bay interface. Finally, options to print and/or plot the results are available .

Flow Logic

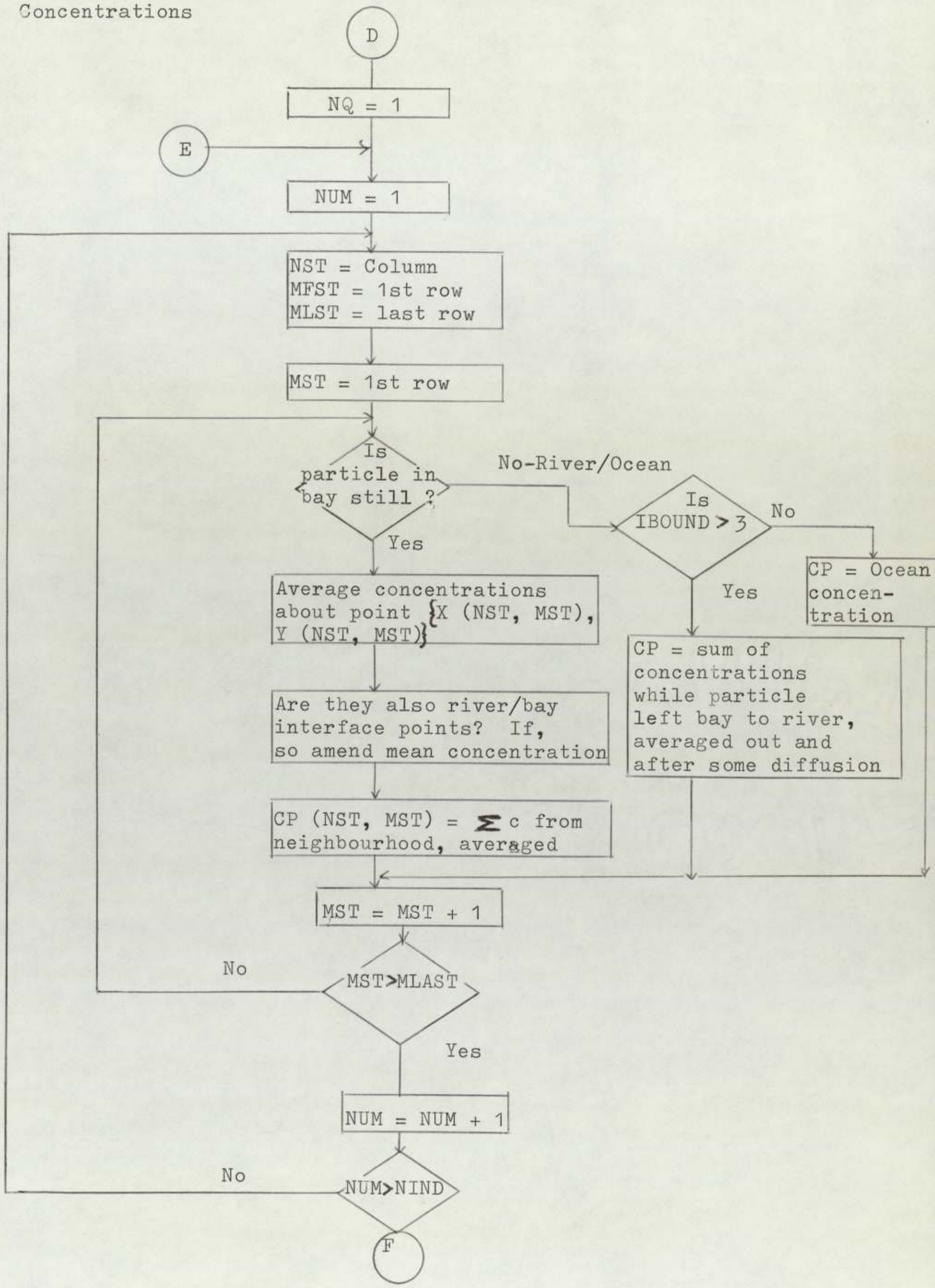

ENTER
Is it 1st Call
No
A
Yes
Read in grid size diffusive steps, plot dump options, interface points
Set FFF to seconds per hour/(NDH x no. of velocity readings summed)
Set FF to 1/5 divided by number of bay interface points. Reset call number
Set all C(N,M,NQ) to default land value
Set those values of C(N,M,NQ) that are in water to initial concentrations CONCGI
Print bay initial concentrations
RETURN
Initial Call Input and Initial Concentration Estimates

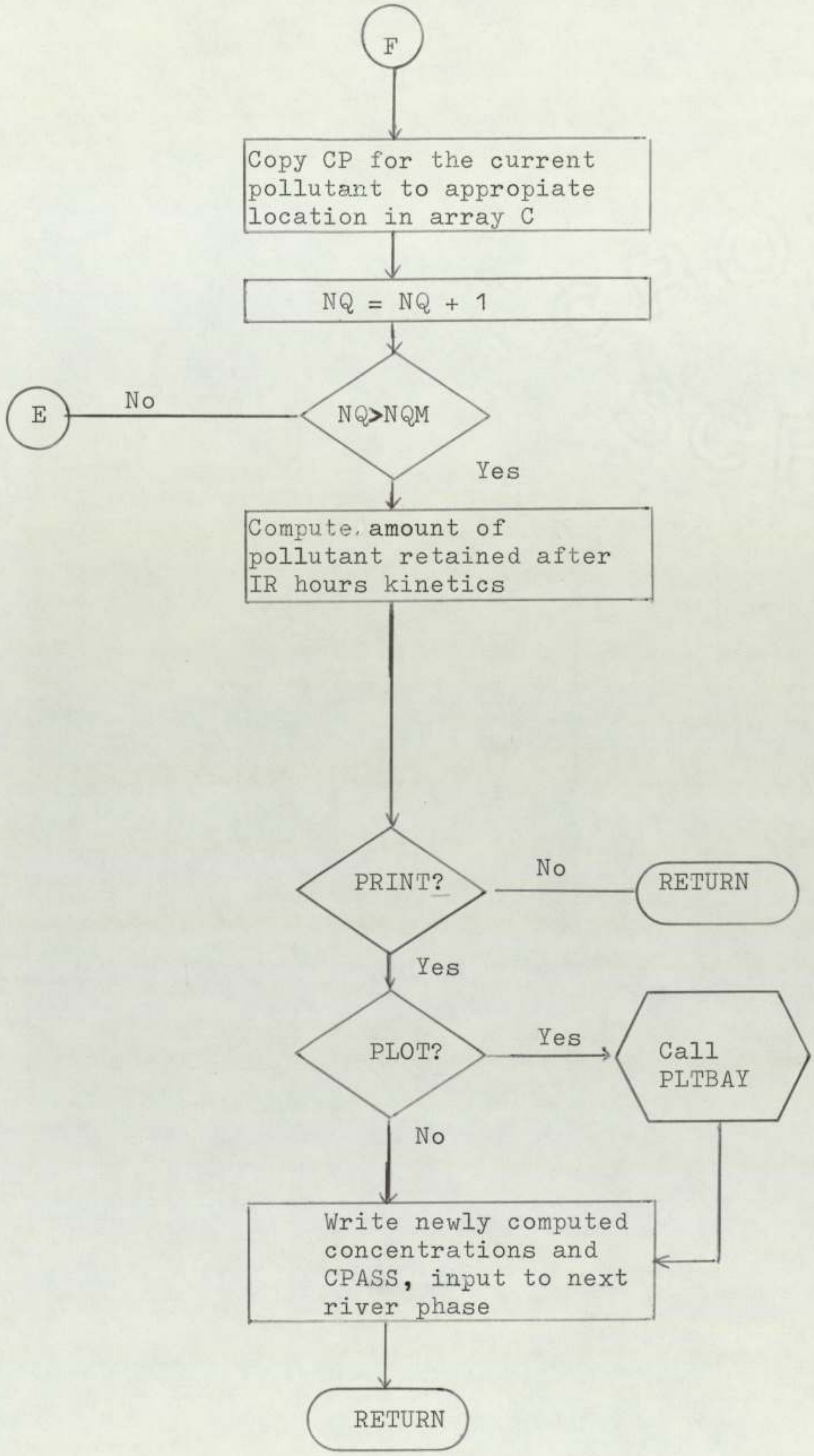

A
Calculate DIEOFF(NO) = e -(DECAY x IR /24)
Set up distances in X & Y arrays and set all IBOUND to 1
Slightly increase X element for river/bay interface for points going into the river
Main time loop IT = 1
Sub Cycle Loop
C
Read U & V off tape ITP2 for time IT
Is it flood tide?
ISWITC = 2
Yes
No
Write interface velocity
Set NUM = 1 (loop for all columns)
NST = Column (NUM)
MFST = first row
MLST = last row
Test for marker particle in each row MFST to MLST. If it is already out of bay, no action. If IBOUND = 1 call TRNSPT to move the particle
B

INTERFACE TRANSPORT AND CPASS ESTIMATES



New Estimate of
Concentrations




F
Copy CP for the current pollutant to appropiate location in array C
NQ = NQ + 1
E
No
NQ>NQM
Yes
Compute amount of pollutant retained after IR hours kinetics
PRINT?
No
RETURN
Yes
PLOT?
Yes
Call PLTBAY
No
Write newly computed concentrations and CPASS, input to next river phase
RETURN

C.11 Subroutine TAPECH (for PT1)

Function	To prepare an unformatted data file from the output of the F1/F2 models for use by the model PT1
System Functions	READ/WRITE I/O, REWIND
User Functions	None
Calling statement	CALL TAPECH (ID, ITREP)
Error Messages	STOP 9 - sub cycles > 9 hours duration
Common Areas	/RIV1/, /ALL/, /IO/
Common Variables Reset	Parameters for river/bay dimensions in /ALL/, NMAX in /RIV1/

Description

The following are input initially from the principle data stream:-

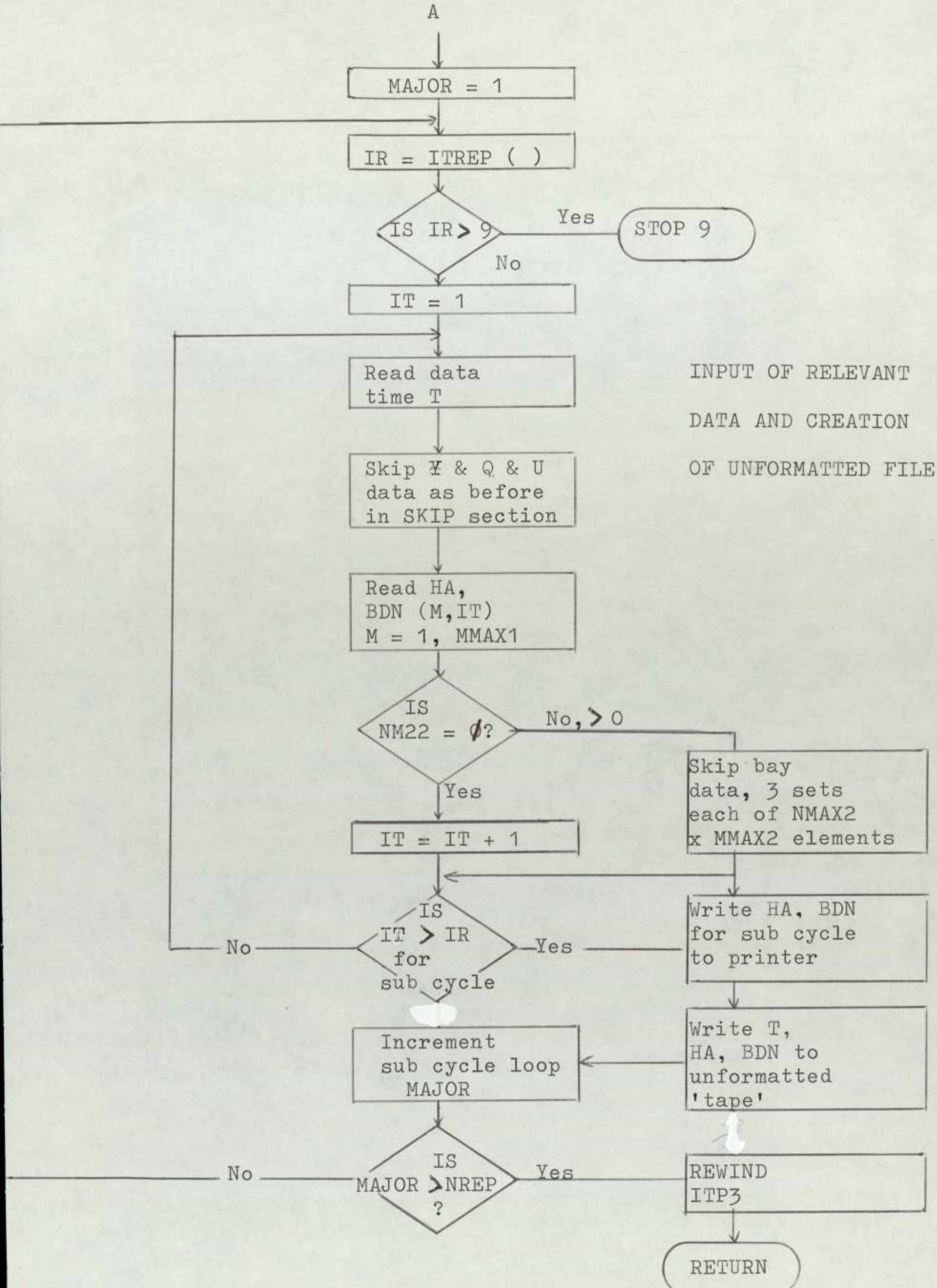
NREP	Number of sub cycles per 'tidal' cycle (2 or 4 usually)
ITREP	Hours in each sub cycle (NREP elements)
NSKIP	Number of data records to skip from F1/F2 file before data is to be retained

The data set reference of the F1/F2 file is transferred in /IO/ (name ITP1) and is assessed to read the initial two records:-

ID title - one card image

Segments in river MMAX1, Columns in bay NMAX2, Rows in bay MMAX2, grid points in each of NMAX1 segments in array NMAX - one card image.

A variable NM22 = NMAX2 + MMAX2 is set up to act as a flag for whether file is an F1 or F2 output (if NM22 = 0 file is from F1). If NSKIP is negative, the unformatted file has already been established and control returns to POLTRA for PT1. If NSKIP > 0 the appropiate number of records are skipped. If NSKIP = 0 none need be skipped and control goes to the DO 200 loop, reading in data and saving HA & BDN for each hour in the sub cycle. Then these values are rewritten onto formatted tape. This is done for all NREP sub cycles, the loop index being 'MAJOR'. Finally the file ITP3 so created is rewound to set indexing to start in the event of a disc file.



A
MAJOR = 1
IR = ITREP ( )
IS IR > 9
Yes
STOP 9
No
IT = 1
Read data time T
INPUT OF RELEVANT DATA AND CREATION OF UNFORMATTED FILE
Skip ¥ & Q & U data as before in SKIP section
Read HA, BDN (M,IT) M = 1, MMAX1
IS NM22 = ∅?
No, > 0
Skip bay data, 3 sets each of NMAX2 x MMAX2 elements
Yes
IT = IT + 1
IS IT > IR for sub cycle
No
Yes
Write HA, BDN for sub cycle to printer
Increment sub cycle loop MAJOR
Write T, HA, BDN to unformatted 'tape'
IS MAJOR > NREP ?
No
Yes
REWIND ITP3
RETURN

C.12 Subroutine TAPECH (for PT)

Function	To prepare two unformatted data files from the output of the F2 model for use by PT
System Functions	READ/WRITE I/O, REWIND
User Functions	None
Calling Statement	CALL TAPECH
Error Messages	STOP 9 - Over nine hours in a sub-cycle
Common Areas	/ALL/, /RIV1/, /IO/
Common Variables Reset	Parameter for river/bay dimensions in /ALL/, NMAX in /RIV1/

Description

This routine is very similar to TAPECH for PT1. The NM22 tests need not be carried out as only F2 data should be input, in which case NM22 > 0. If F1 data is input accidentally, there will be a read failure on the first input attempt.

In the data retention phase, bay elevations are skipped and arrays U and V are retained for each hour of the sub cycle. The consequent need for an array of dimensions NMAX2 x MMAX2 x 2 x 19 elements makes the routine costly and thus it should either be segmented or detached from the main program. A second file, reference ITP2 is set with bay data in reverse time sequence.

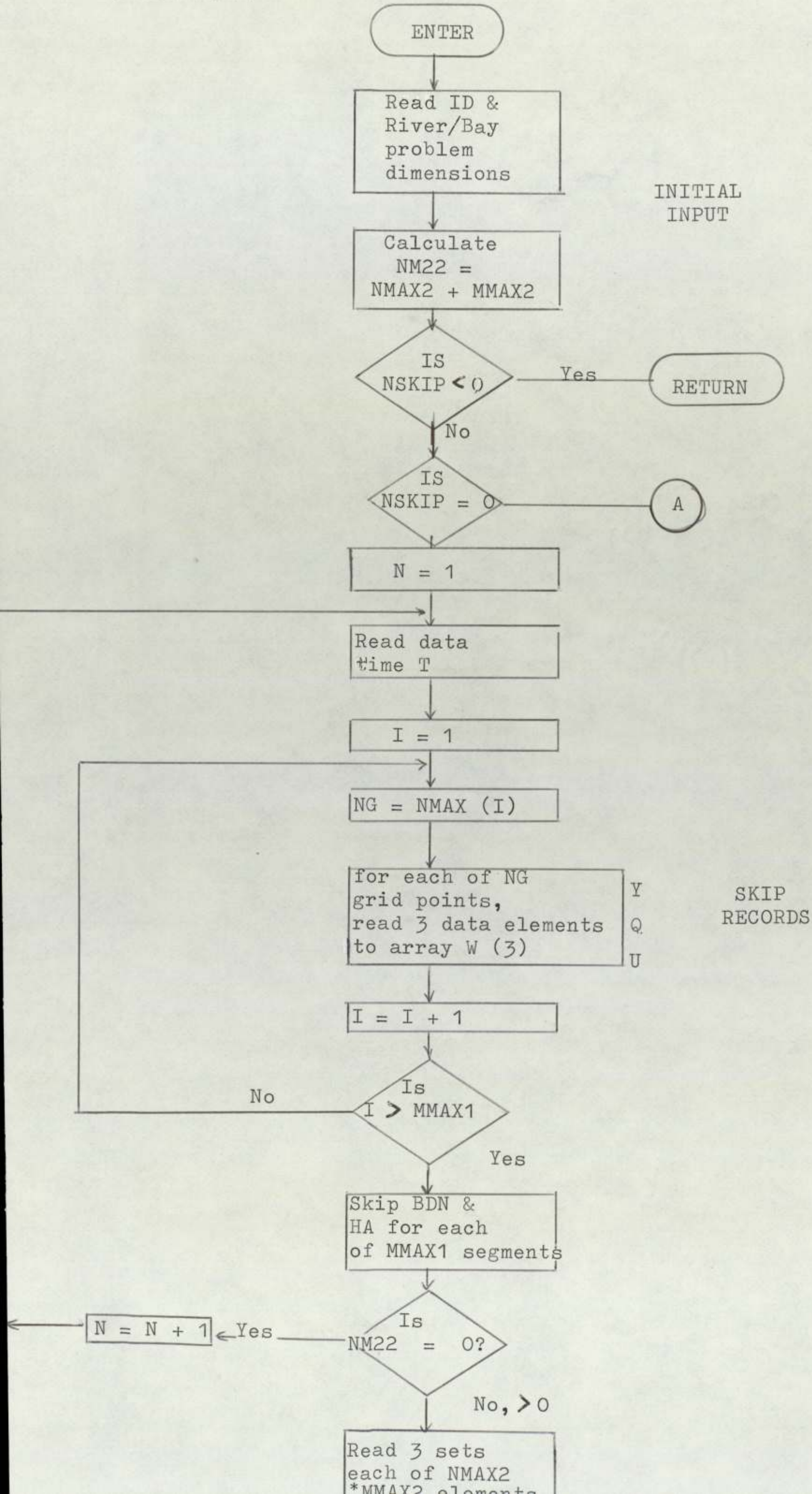
Flow Logic

With exceptions outlined above, very similar to TAPECH for PT1. Do 300 is an additional loop to set up file ITP2 which has as its first record ID & ITREP, hence the lack of a calling parameter list as in TAPECH for PT1.

C.13 Subroutine TAPECH (file PT2 TAP)

Note	This routine is identical to TAPECH for PT but the input file from F2 is also unformatted
Function	Refer to TAPECH (for PT)

Flow Logic


ENTER
Read ID & River/Bay problem dimensions
INITIAL INPUT
Calculate NM22 = NMAX2 + MMAX2
IS NSKIP < 0
Yes
RETURN
No
IS NSKIP = 0
A
N = 1
Read data time T
I = 1
NG = NMAX (I)
for each of NG grid points, read 3 data elements to array W (3)
Y
Q
U
SKIP RECORDS
I = I + 1
Is I > MMAX1
No
Yes
Skip BDN & HA for each of MMAX1 segments
Is NM22 = 0?
N = N + 1
Yes
No, > 0
Read 3 sets each of NMAX2

C.14 Subroutine BRANCH

Function	To accomplish the transfer of volumes of water from upstream to downstream segments through a node
System Functions	None
User Functions	None
Calling Statement	CALL BRANCH (M, M1, M2, M3, M4)
Common Areas	/RIV1/, /CONSTI/
Common Variables Reset	Arrays V and C in /RIV1/, element M of NMAX in /RIV1/

Description

There are four distinct segments to this routine. If QO and Q1 are zero for the segment M there is no net water movement and the routine returns unchanged values to RIVQAL routine.

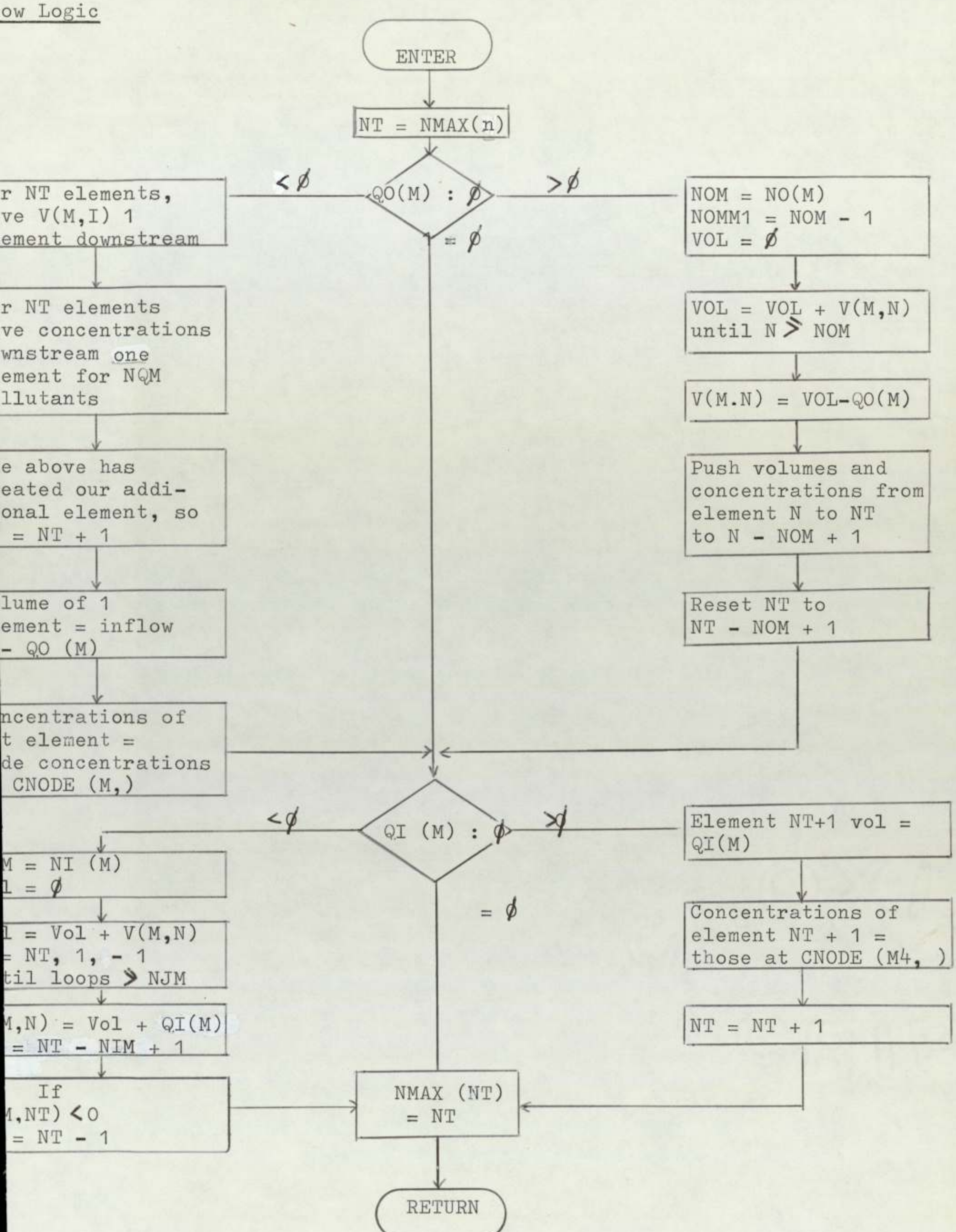
If QO (M) is negative, flow is seaward. All volumes and concentrations are moved one element downstream and a new element then set up with index NT + 1. That leaves the first element in each segment to be replaced by -QO(M), the inflow in volume units, ie V(M,1). The concentrations of the first element C(M,1,NQ) is obtained from the node concentrations CNODE (M, NQ).

If QO(M) is positive, flow is landward; and NO(M) is the elements required to produce the outward flow from the landward end of a segment. The volume represented by these NO(M) elements (element 1 to NO(M) in V(M,i)) is calculated and stored in VOL. These NO(M) elements leave this segment and so the volume/concentrations from element NO(M) + 1 to element NT are pushed down to index 1 to NT -(NOM(M)+ 1. NT is finally redefined to NT-NO(M) + 1.

If QI(M) is negative flow is seaward out of the seaward end of segment M. NI(M) holds the number of elements lost from this segment during the time step. The logic is similar to that of QO(M) > 0 with volume/concentrations moved in their indeces. If the final segment is negative volumed, the elements are further reduced by 1.

If QI(M) is positive, flow is landward into the seaward end of the segment and a volume element is created at NT + 1. Concentrations are taken from node concentrations of downstream segment connection.

ow Logic


ENTER
NT = NMAX(n)
QO(M) : ∅
< ∅
> ∅
= ∅
r NT elements,
ve V(M,I) 1
ement downstream
r NT elements
ve concentrations
wnstream one
ement for NQM
llutants
e above has
eated our addi-
onal element, so
= NT + 1
lume of 1
ement = inflow
- QO (M)
ncentrations of
t element =
de concentrations
CNODE (M,)
NOM = NO(M)
NOMM1 = NOM - 1
VOL = ∅
VOL = VOL + V(M,N)
until N ≥ NOM
V(M.N) = VOL-QO(M)
Push volumes and
concentrations from
element N to NT
to N - NOM + 1
Reset NT to
NT - NOM + 1
QI (M) : ∅
< ∅
> ∅
= ∅
M = NI (M)
l = ∅
l = Vol + V(M,N)
= NT, 1, - 1
til loops ≥ NJM
M,N) = Vol + QI(M)
= NT - NIM + 1
If
M,NT) < 0
= NT - 1
Element NT+1 vol =
QI(M)
Concentrations of
element NT + 1 =
those at CNODE (M4,)
NT = NT + 1
NMAX (NT)
= NT
RETURN

C.15 Subroutine DIFUSE

Function	To simulate diffusive transport in the river phases of the pollutant transport models, and control elements of segments
System Functions	WRITE, ABS
User Functions	None
Calling Statement	CALL DIFUSE (M)
Common Areas	/IO/, /XC/, /RIV1/, /RIV2/, /CONSTI/, /D/
Common Variables Reset	NE in /D/, some in /XC/, /RIV2/ has C, V, and possible NMAX(M) reset

Description

The first part of the program locates and tests to see if any elements in segment M are smaller if there are too many segments for the store limits (currently up to 30 elements are permitted). The diffusion factor, EFOR is adjusted for the current state of the system, net multiplied by:

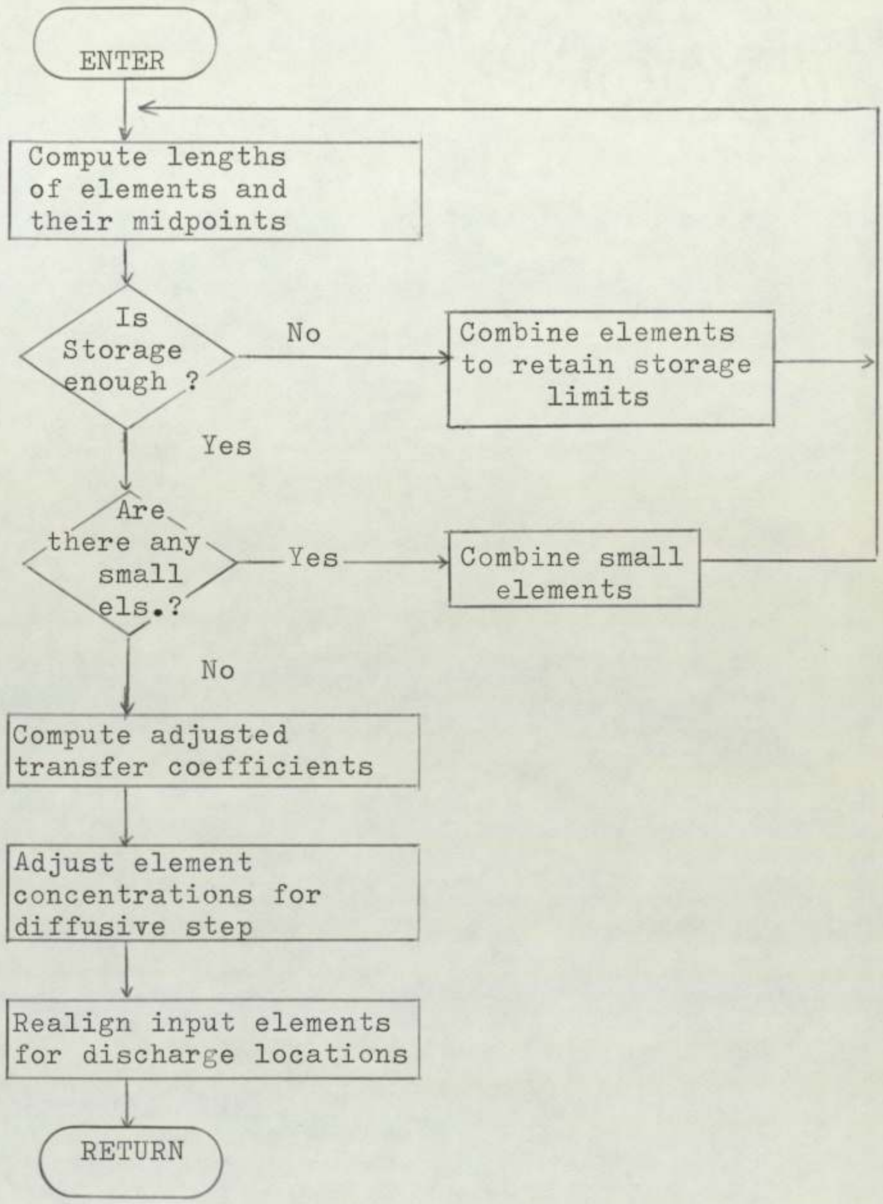
$$\left\{\frac{\text{width of segment at current time}}{\text{Length of Segment}}\right\} \cdot \left\{\frac{\text{distance travelled in a time step}}{\text{Length of Segment}}\right\}$$

The first factor accounts for transverse diffusion, the second for lateral diffusion.

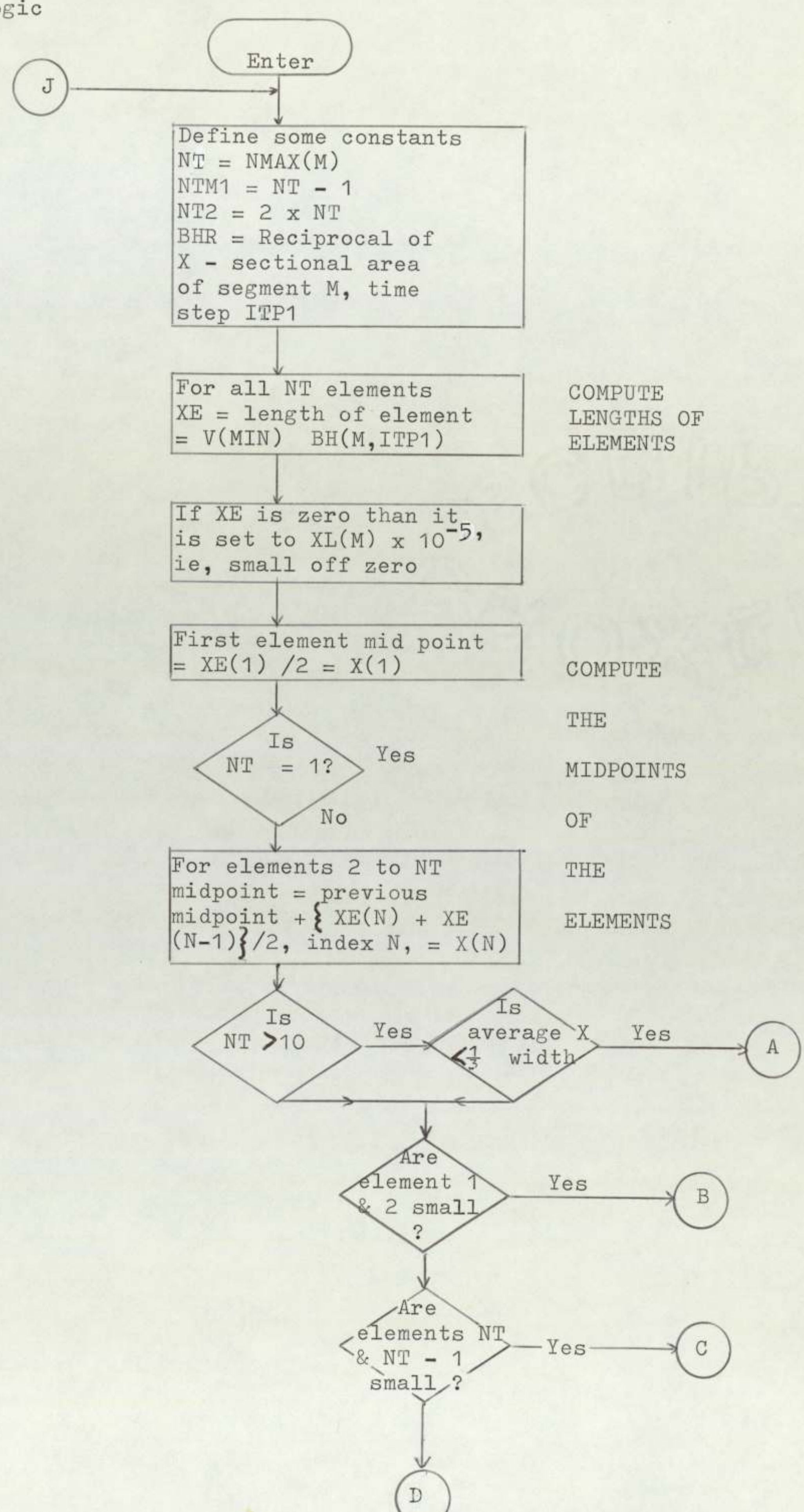
This adjusted value is then used to modify element concentrations for elements N in terms of diffusion to and from adjacent elements. As array X was evaluated in the first section, the discharge locations are now tested to see if they are now discharging to different elements within a segment.

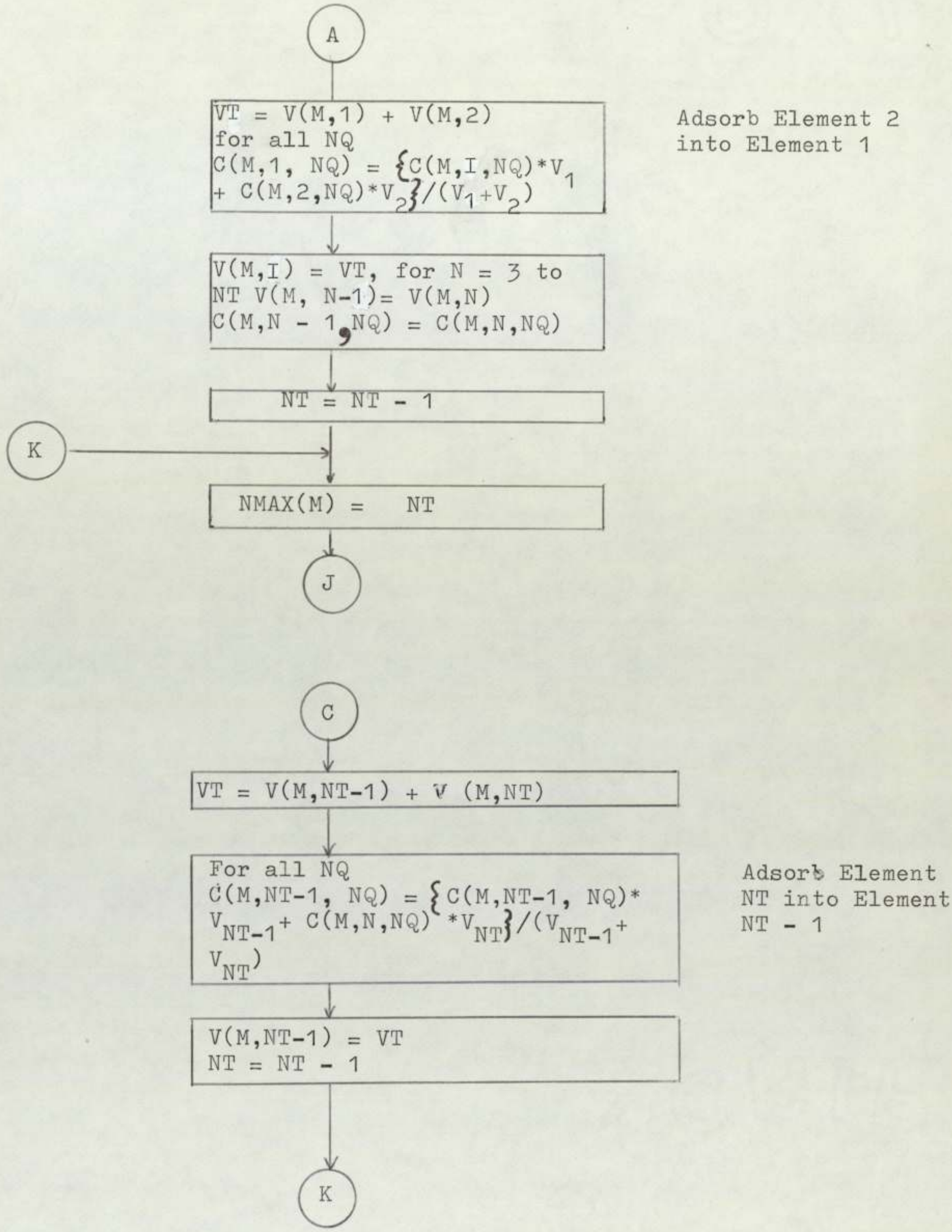
The rest of the coding is concerned with merging elements that are small in relation to some predefined criteria, in this case a length of less than 1/3 of the segment downstream boundary width.

Outline Flow Logic

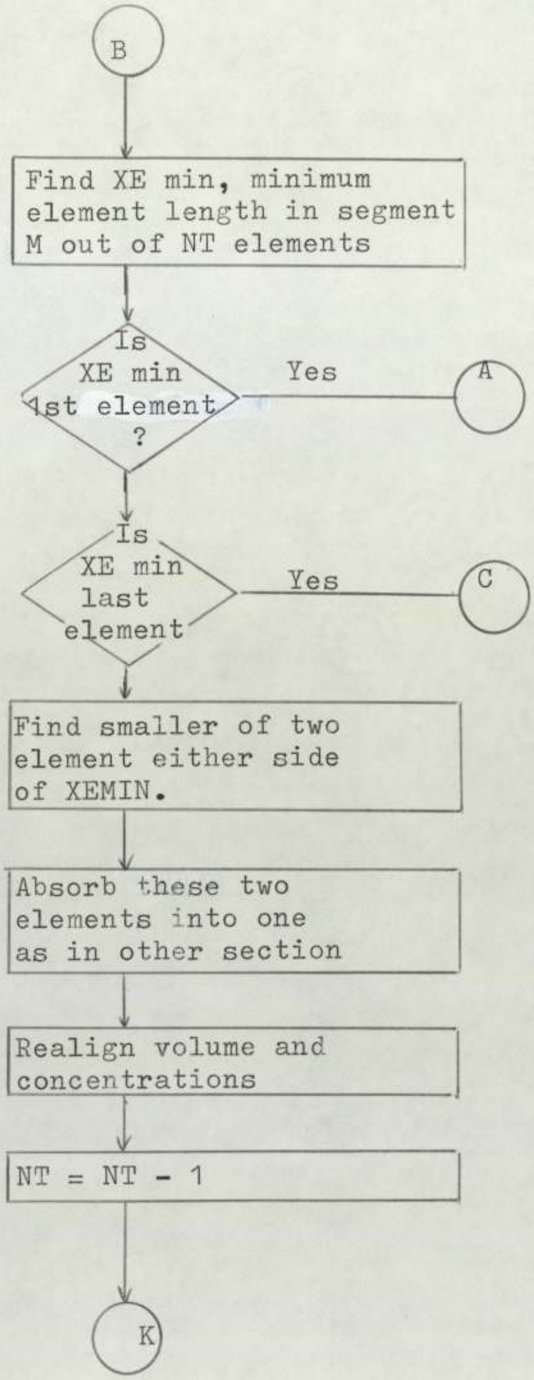


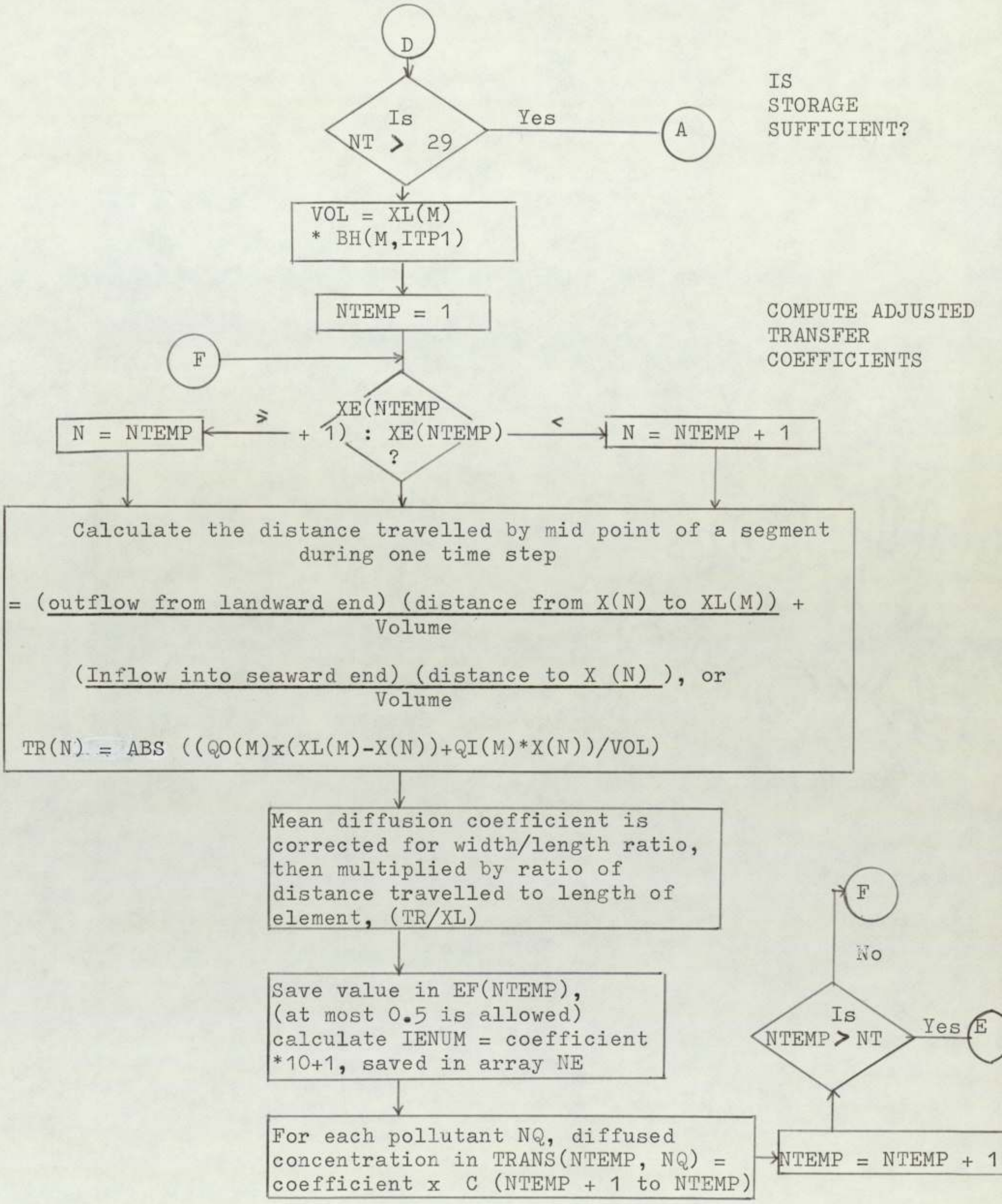
etailed Flow Logic



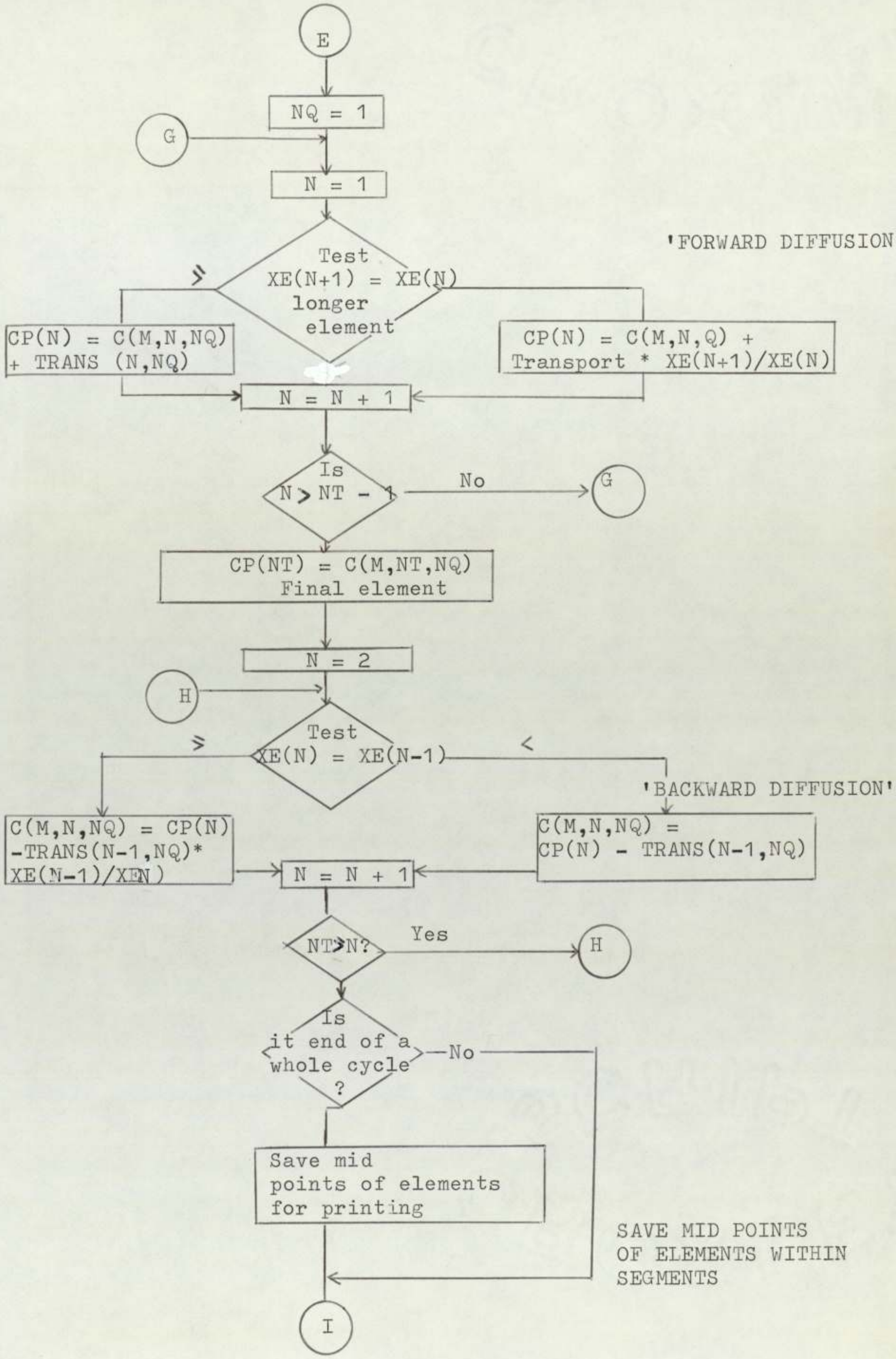

A
VT = V(M,1) + V(M,2)
for all NQ
C(M,1, NQ) = {C(M,I,NQ)*V1 + C(M,2,NQ)*V2}/(V1+V2)
Adsorb Element 2 into Element 1
V(M,I) = VT, for N = 3 to NT V(M, N-1)= V(M,N)
C(M,N - 1, NQ) = C(M,N,NQ)
NT = NT - 1
K
NMAX(M) = NT
J
C
VT = V(M,NT-1) + V (M,NT)
For all NQ
C(M,NT-1, NQ) = {C(M,NT-1, NQ)*VNT-1+ C(M,N,NQ) *VNT}/(VNT-1+ VNT)
Adsorb Element NT into Element NT - 1
V(M,NT-1) = VT
NT = NT - 1
K

REDUCTION OF ELEMENTS WHEN AN ELEMENT LENGTH IS $< \frac{1}{3}$ ITS WIDTH

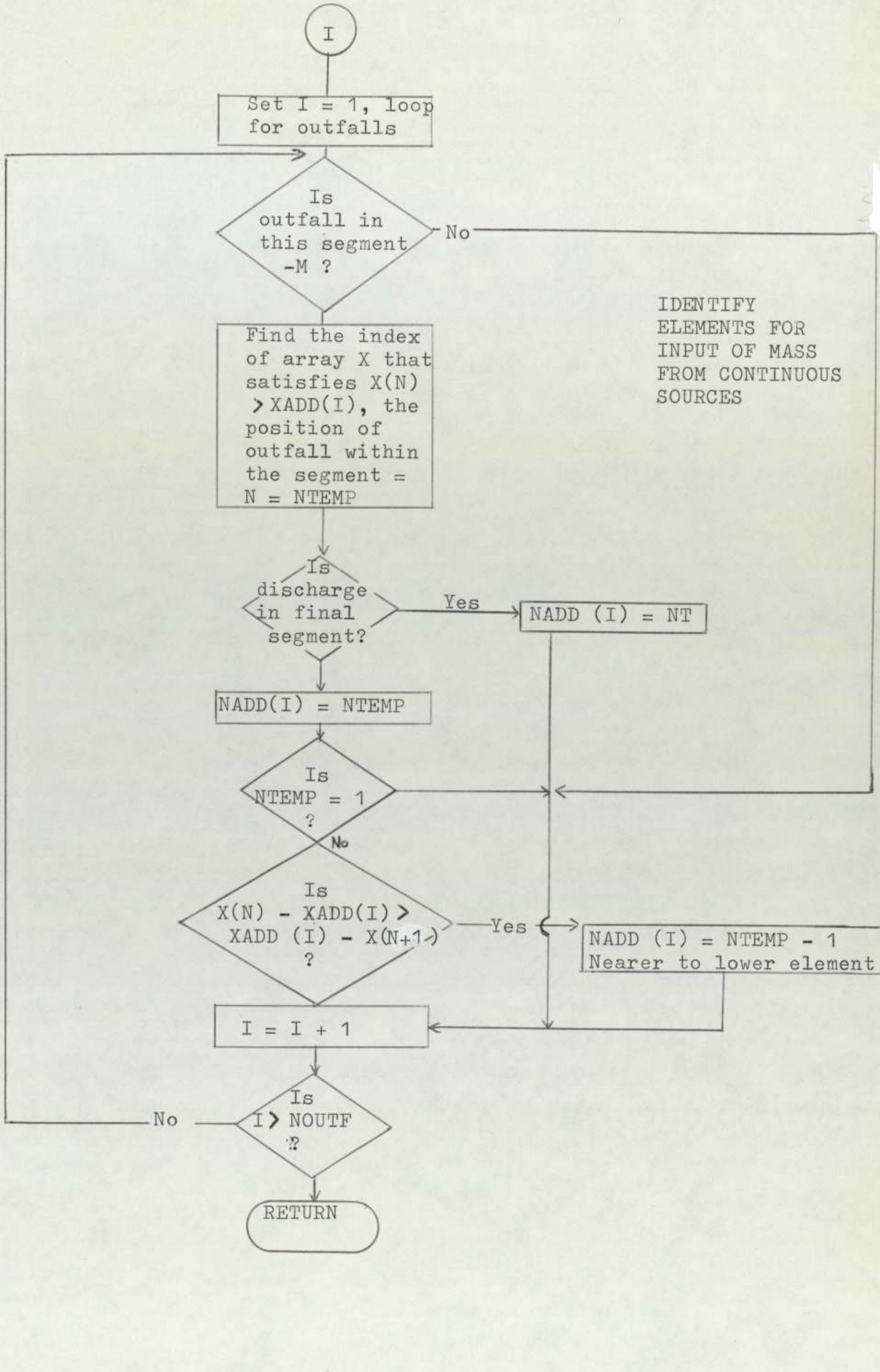




D
Is NT > 29
Yes
A
IS STORAGE SUFFICIENT?
VOL = XL(M) * BH(M,ITP1)
NTEMP = 1
COMPUTE ADJUSTED TRANSFER COEFFICIENTS
F
XE(NTEMP + 1) : XE(NTEMP) ?
≥
N = NTEMP
<
N = NTEMP + 1
Calculate the distance travelled by mid point of a segment during one time step
= (outflow from landward end) (distance from X(N) to XL(M)) / Volume +
(Inflow into seaward end) (distance to X (N)) / Volume, or
TR(N) = ABS ((QO(M)x(XL(M)-X(N))+QI(M)*X(N))/VOL)
Mean diffusion coefficient is corrected for width/length ratio, then multiplied by ratio of distance travelled to length of element, (TR/XL)
Save value in EF(NTEMP), (at most 0.5 is allowed) calculate IENUM = coefficient *10+1, saved in array NE
For each pollutant NQ, diffused concentration in TRANS(NTEMP, NQ) = coefficient x C (NTEMP + 1 to NTEMP)
NTEMP = NTEMP + 1
Is NTEMP > NT
No
F
Yes
E



E
NQ = 1
G
N = 1
'FORWARD DIFFUSION'
Test
XE(N+1) = XE(N)
longer
element
≥
CP(N) = C(M,N,NQ)
+ TRANS (N,NQ)
CP(N) = C(M,N,Q) +
Transport * XE(N+1)/XE(N)
N = N + 1
Is
N > NT - 1
No
G
CP(NT) = C(M,NT,NQ)
Final element
N = 2
H
Test
XE(N) = XE(N-1)
≥
<
'BACKWARD DIFFUSION'
C(M,N,NQ) = CP(N)
-TRANS(N-1,NQ)*
XE(N-1)/XEN)
C(M,N,NQ) =
CP(N) - TRANS(N-1,NQ)
N = N + 1
NT>N?
Yes
H
Is
it end of a
whole cycle
?
No
Save mid
points of elements
for printing
SAVE MID POINTS
OF ELEMENTS WITHIN
SEGMENTS
I


I
Set I = 1, loop for outfalls
Is outfall in this segment -M ?
No
Find the index of array X that satisfies X(N) > XADD(I), the position of outfall within the segment = N = NTEMP
Is discharge in final segment?
Yes
NADD (I) = NT
NADD(I) = NTEMP
Is NTEMP = 1 ?
No
Is X(N) - XADD(I) > XADD (I) - X(N+1) ?
Yes
NADD (I) = NTEMP - 1
Nearer to lower element
I = I + 1
No
Is I > NOUTF ?
RETURN
IDENTIFY ELEMENTS FOR INPUT OF MASS FROM CONTINUOUS SOURCES

C.16 Subroutine MAP

Function	To map inflow/outflow from segment to segment in terms of element transfer
System Functions	WRITE, DEBUG
User Functions	None
Calling Statement	CALL MAP (M, M1, M2, M3)
Common Areas	/IO/, /RIV1/, /CONSTI/
Common Variables Reset	Arrays QI, QO, CI, CO, NO, NI, CNODE in /RIV1/

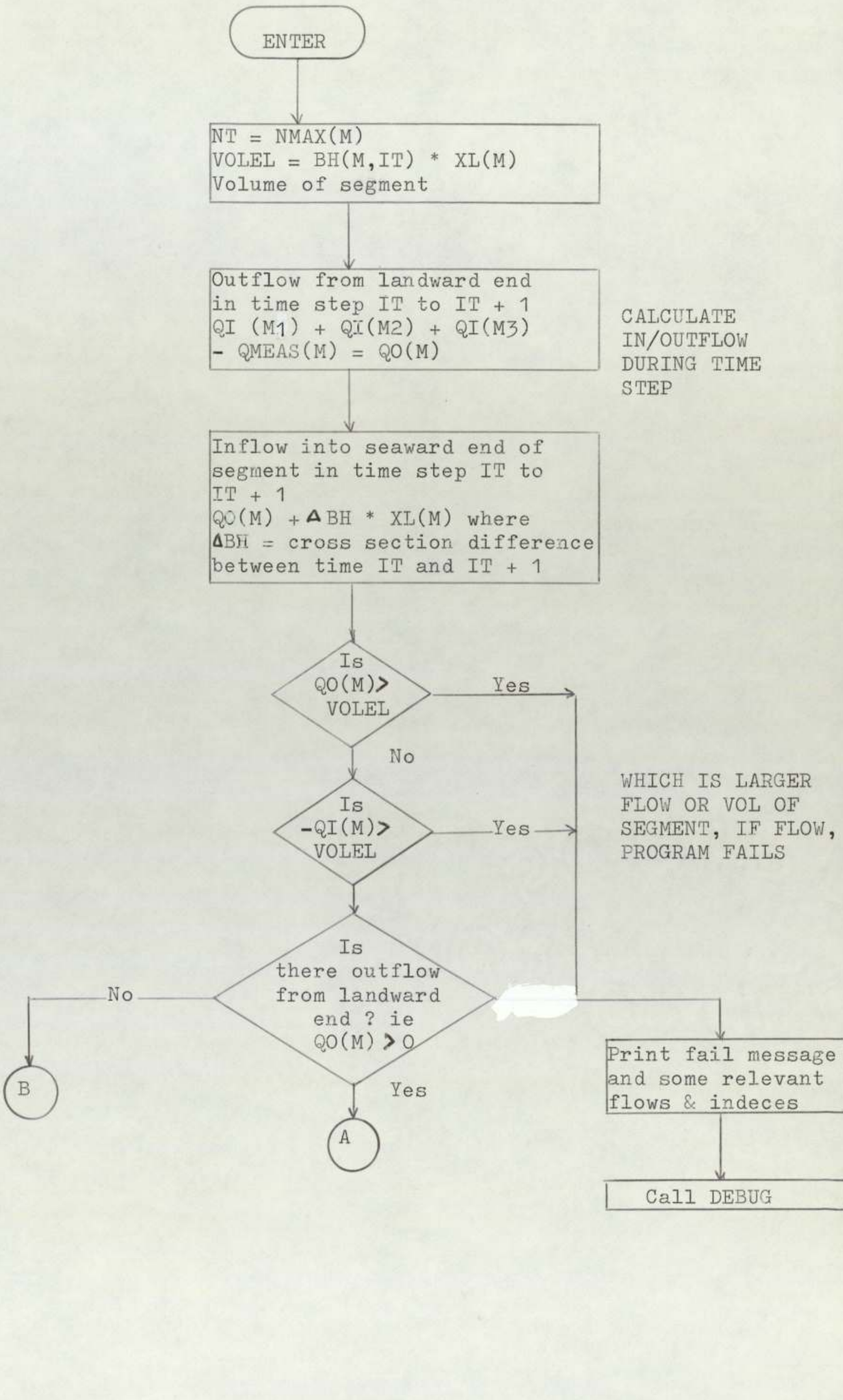
Description

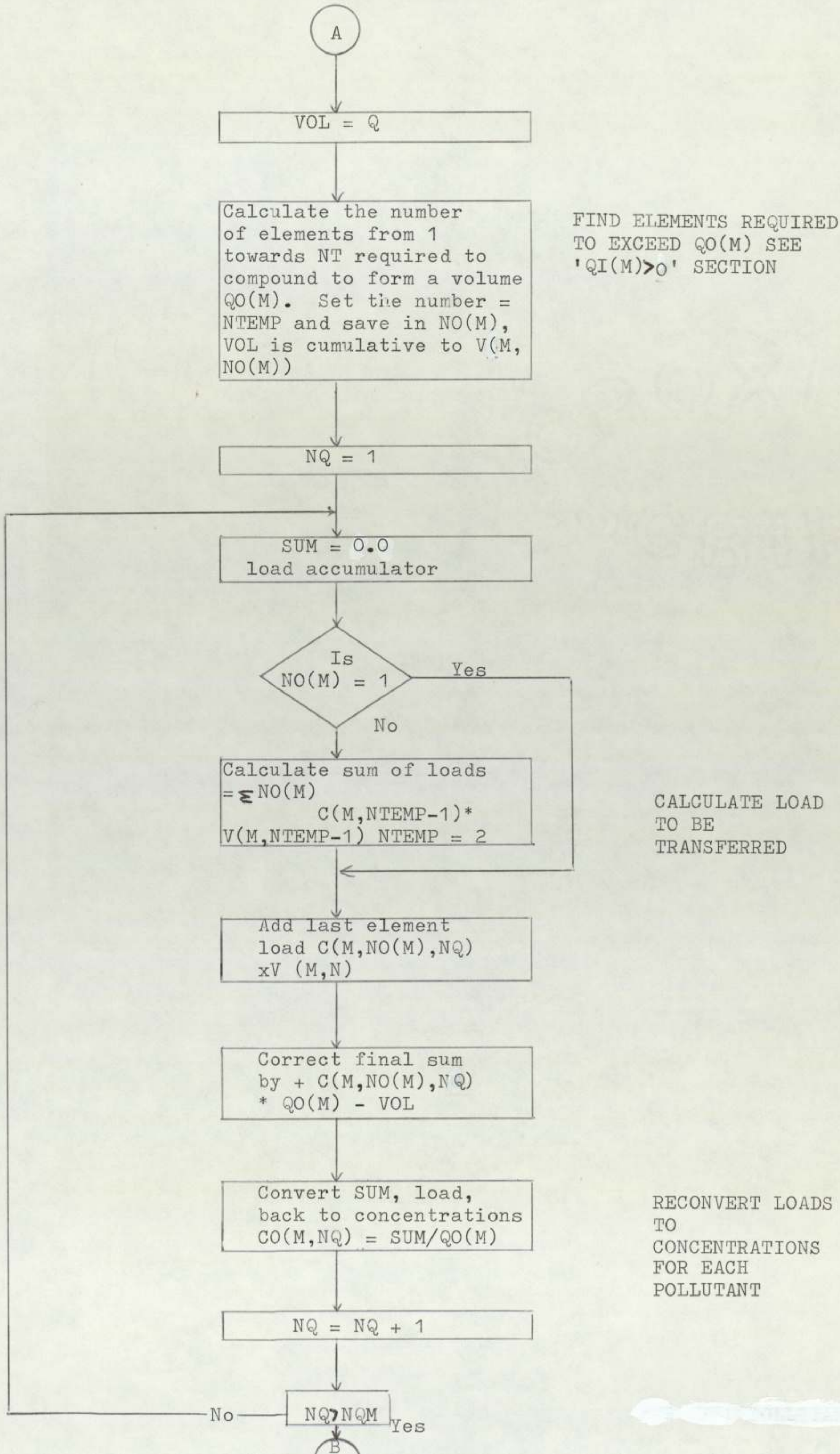
The program consists of a series of well defined sections setting up the number of elements to be moved in or out of a segment. As an initial test of feasibility, new values of QO(M) and QI(M) are calculated and tested against the volume of the entire segment. If the amount of flow is greater than the volume of the segment the algorithm will eventually fail. This is prevented by printing relevant parameters and calling the error trace routine provided by the system to dump the most relevant parameters.

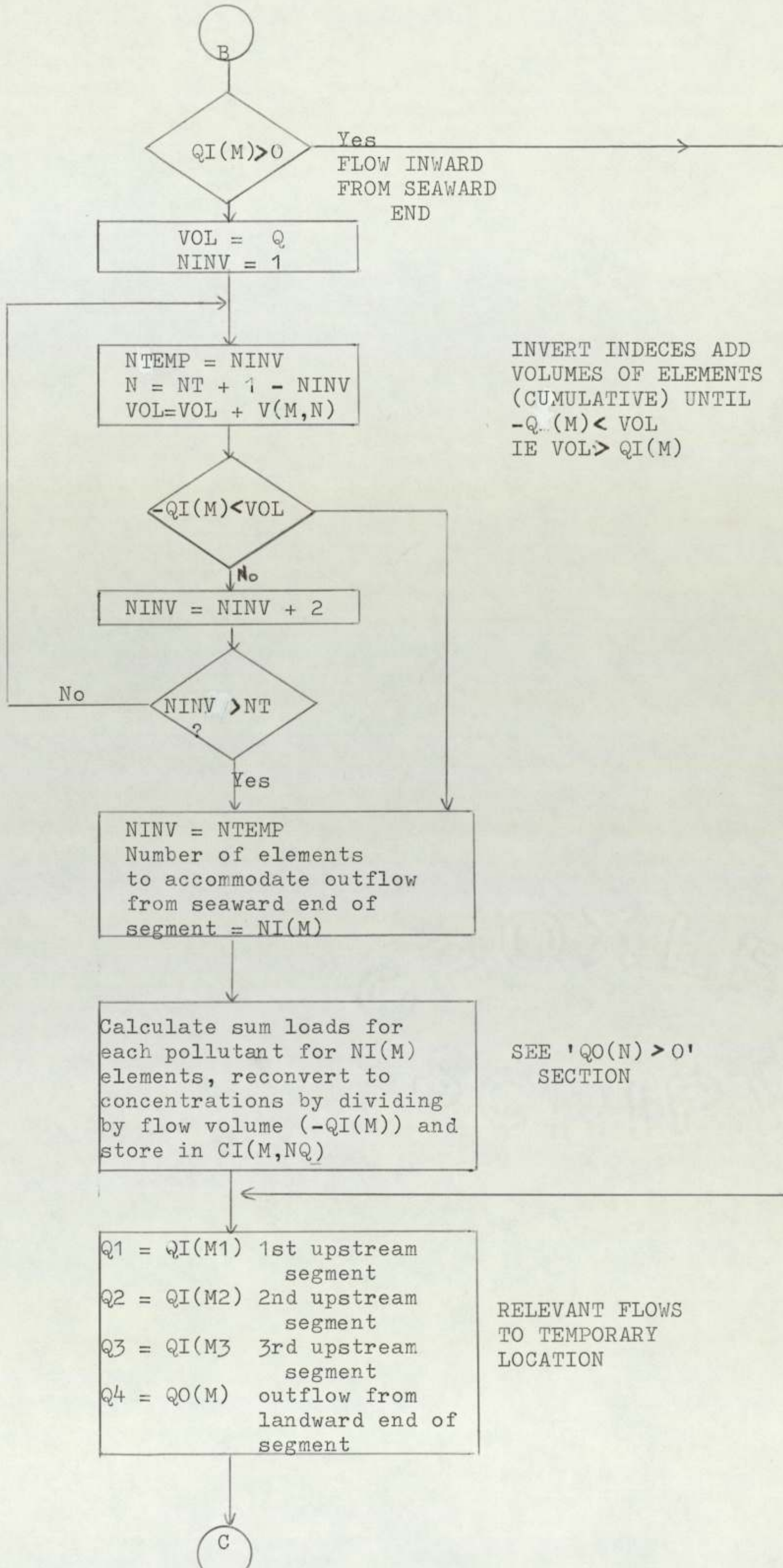
The second main segment computer concentrations of outflow from landward end of a segment, ie if QO(M) is >0. The third segment computes concentration of outflow from seaward end of segment.

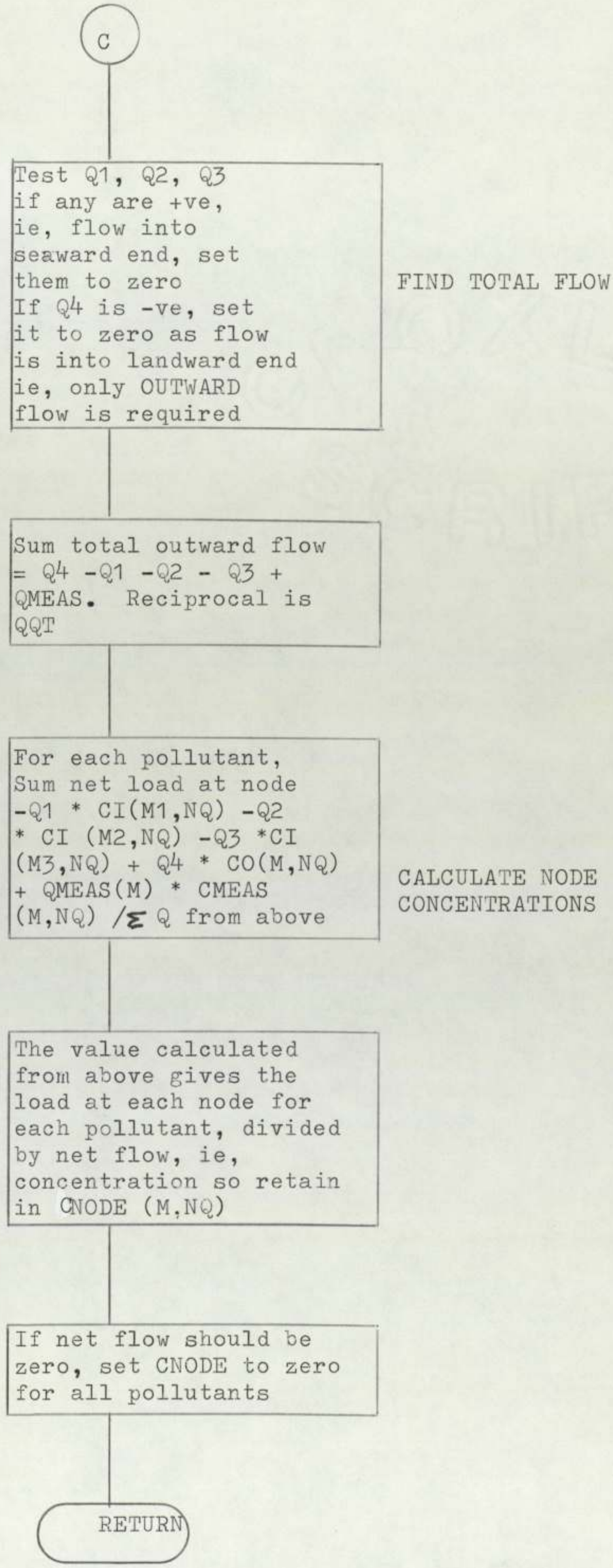
Finally, the new node concentration is computed from the weighted averages of the outflow concentrations in CI array and the additional discharge.

Flow Logic


ENTER
NT = NMAX(M)
VOLEL = BH(M,IT) * XL(M)
Volume of segment
Outflow from landward end
in time step IT to IT + 1
QI (M1) + QI(M2) + QI(M3)
- QMEAS(M) = QO(M)
CALCULATE
IN/OUTFLOW
DURING TIME
STEP
Inflow into seaward end of
segment in time step IT to
IT + 1
QO(M) + ΔBH * XL(M) where
ΔBH = cross section difference
between time IT and IT + 1
Is
QO(M)>
VOLEL
Yes
No
Is
-QI(M)>
VOLEL
Yes
WHICH IS LARGER
FLOW OR VOL OF
SEGMENT, IF FLOW,
PROGRAM FAILS
Is
there outflow
from landward
end ? ie
QO(M) > 0
No
B
Yes
A
Print fail message
and some relevant
flows & indeces
Call DEBUG


A
VOL = Q
Calculate the number of elements from 1 towards NT required to compound to form a volume QO(M). Set the number = NTEMP and save in NO(M), VOL is cumulative to V(M, NO(M))
FIND ELEMENTS REQUIRED TO EXCEED QO(M) SEE 'QI(M)>0' SECTION
NQ = 1
SUM = 0.0 load accumulator
Is NO(M) = 1
Yes
No
Calculate sum of loads = Σ NO(M) C(M,NTEMP-1)* V(M,NTEMP-1) NTEMP = 2
CALCULATE LOAD TO BE TRANSFERRED
Add last element load C(M,NO(M),NQ) xV (M,N)
Correct final sum by + C(M,NO(M),NQ) * QO(M) - VOL
Convert SUM, load, back to concentrations CO(M,NQ) = SUM/QO(M)
RECONVERT LOADS TO CONCENTRATIONS FOR EACH POLLUTANT
NQ = NQ + 1
No
NQ>NQM
Yes
B


B
QI(M)>0
Yes
FLOW INWARD
FROM SEAWARD
END
VOL = Q
NINV = 1
NTEMP = NINV
N = NT + 1 - NINV
VOL=VOL + V(M,N)
INVERT INDECES ADD
VOLUMES OF ELEMENTS
(CUMULATIVE) UNTIL
-Q (M)< VOL
IE VOL> QI(M)
-QI(M)<VOL
No
NINV = NINV + 2
No
NINV >NT
?
Yes
NINV = NTEMP
Number of elements
to accommodate outflow
from seaward end of
segment = NI(M)
Calculate sum loads for
each pollutant for NI(M)
elements, reconvert to
concentrations by dividing
by flow volume (-QI(M)) and
store in CI(M,NQ)
SEE 'QO(N) > O'
SECTION
Q1 = QI(M1) 1st upstream
segment
Q2 = QI(M2) 2nd upstream
segment
Q3 = QI(M3 3rd upstream
segment
Q4 = QO(M) outflow from
landward end of
segment
RELEVANT FLOWS
TO TEMPORARY
LOCATION
C


C
Test Q1, Q2, Q3 if any are +ve, ie, flow into seaward end, set them to zero If Q4 is -ve, set it to zero as flow is into landward end ie, only OUTWARD flow is required
FIND TOTAL FLOW
Sum total outward flow = Q4 -Q1 -Q2 - Q3 + QMEAS. Reciprocal is QQT
For each pollutant, Sum net load at node -Q1 * CI(M1,NQ) -Q2 * CI (M2,NQ) -Q3 *CI (M3,NQ) + Q4 * CO(M,NQ) + QMEAS(M) * CMEAS (M,NQ) /Σ Q from above
CALCULATE NODE CONCENTRATIONS
The value calculated from above gives the load at each node for each pollutant, divided by net flow, ie, concentration so retain in CNODE (M,NQ)
If net flow should be zero, set CNODE to zero for all pollutants
RETURN

C.17 Subroutine XCPRNT

Function	To provide an output routine for the river phase of model PT or PT1
System Functions	WRITE
User Functions	None
Calling Statement	CALL XCPRNT (MMAX1)
Common Areas	/IO/, /RIV1/, /CONSTI/, /XC/
Common Variables Reset	None

Description

The module is written to handle output for up to 5 pollutants. The main arrays are C in /RIV1/ holding concentrations of river phase and X in /XC/ holding mid points of elements.

The main loop is DO 2000, executed once for each segment. The length of the segments is printed in feet and miles. A second loop prints one line for each element in the segment, printing in the order.

LINE GENERATED FOR EACH ELEMENT

	Distance of mid point of element from top of segment in feet	As left but in feet	Concentration of pollutan 1	As left for 2	..for 3	..for 4	..for 5
M	1	13	20	33	46	59	71
	10	19	32	45	58	71	83

C.18 Subroutine PLTRIV

Function	To provide a routine for the output from the model PT1 or PT river phase for a CALCOMP platter
System Functions	GRAFOR, ABS, FLOAT, INT
User Functions	None
Calling Statement	CALL PLTRIV
Common Areas	/IO/, /ALL/, /RIV1/, /CONSTI/, /XC/
Common Variables Reset	None

Description

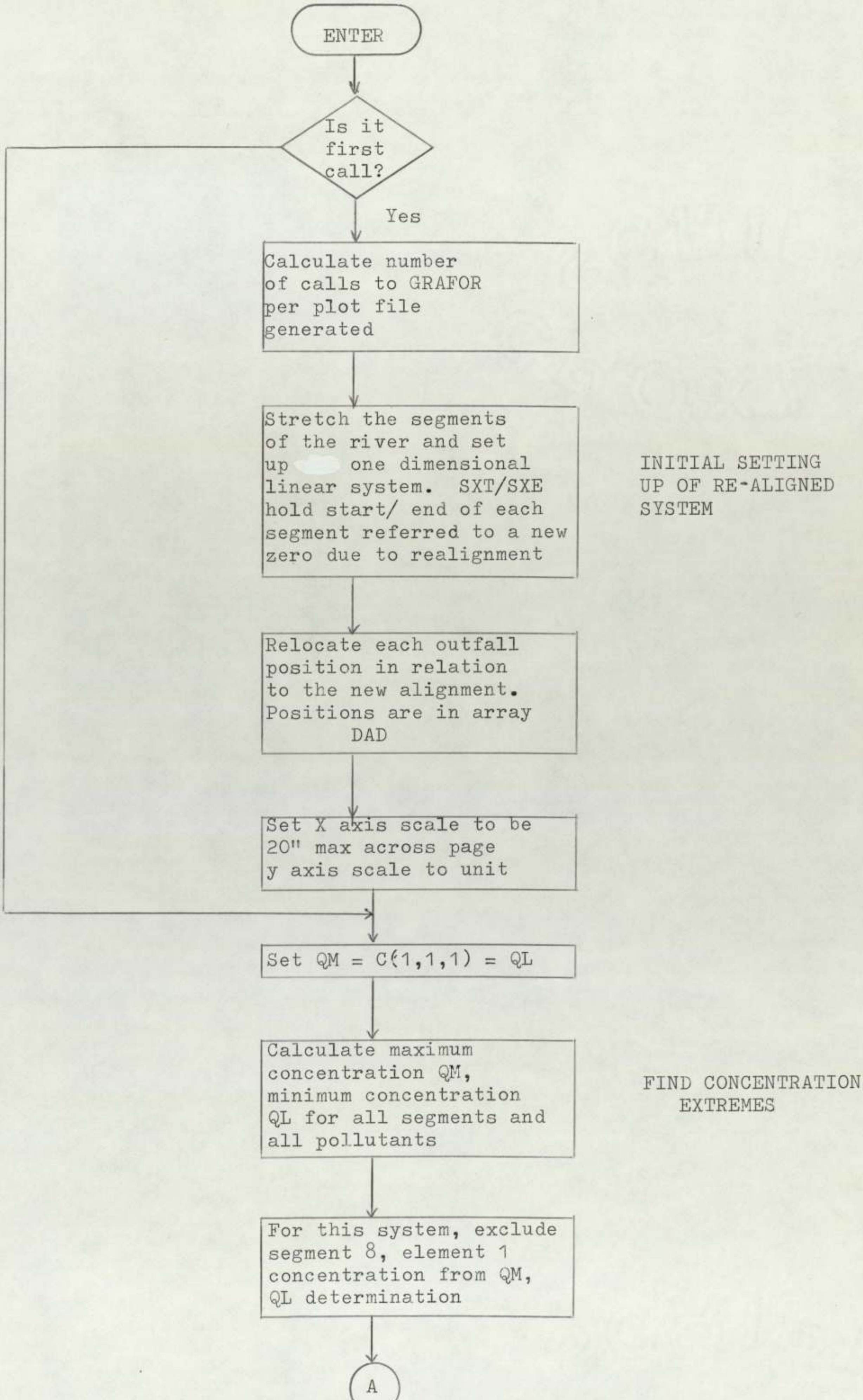
The program is called from the respective version of RIVQAL if two conditions are met: The logical variable RIVPLT is set to .TRUE., and PRT = .TRUE.ie, the concentrations of pollutants are also printed. Each cycle in PT/PT1 generates 2* NREP graphs from this routine.

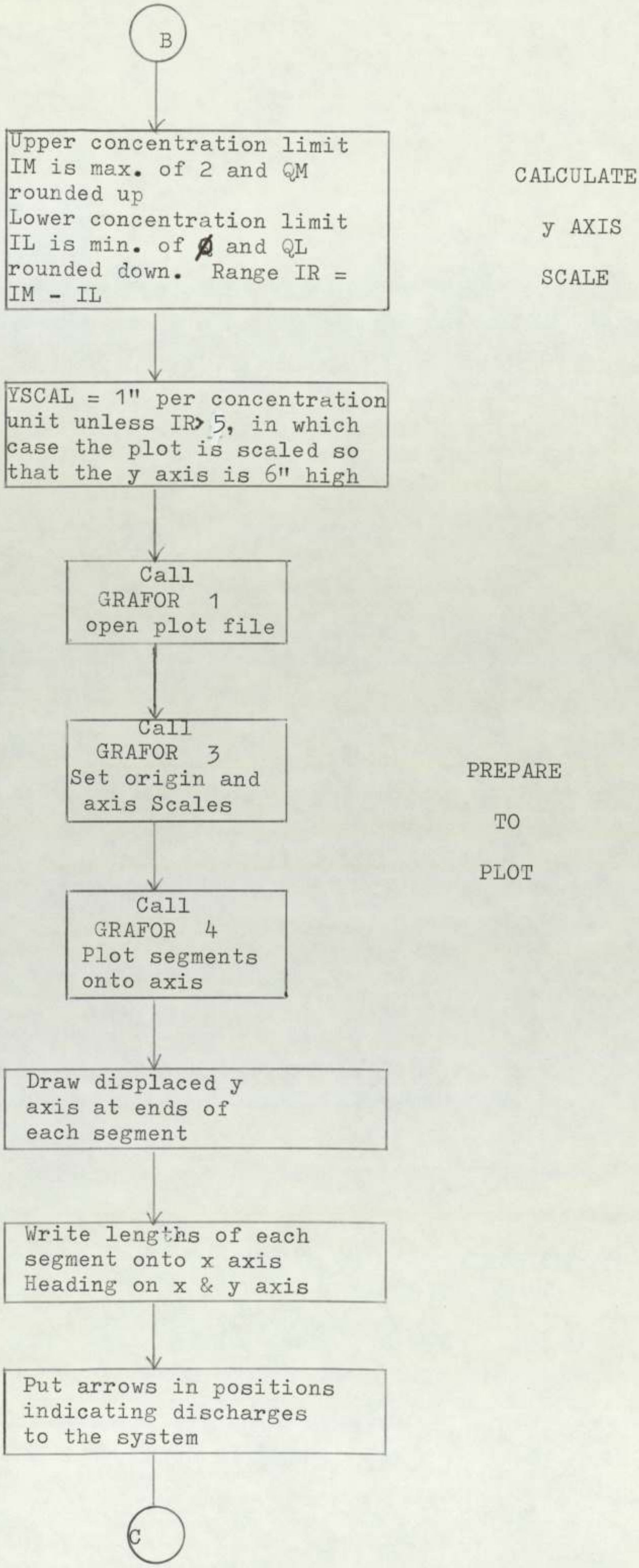
To accommodate a variable network the segments are aligned from left (EAST) for segment 1 to right (WEST) segment MMAX1. Each segment will have a local y axis. The plot routine requires a cohesive x axis and so arrays SXT and SXE are set up on the first call to retain the new x axis limits of start and end of a segment. These values are used later when plotting to provide a displacement factor for the element mid points XPRINT. The displacement at each segment XD, is SXE (I-1), starting for segment 1 at Ø.Ø. The x axis is scaled so that a width of 20" contains all segments.

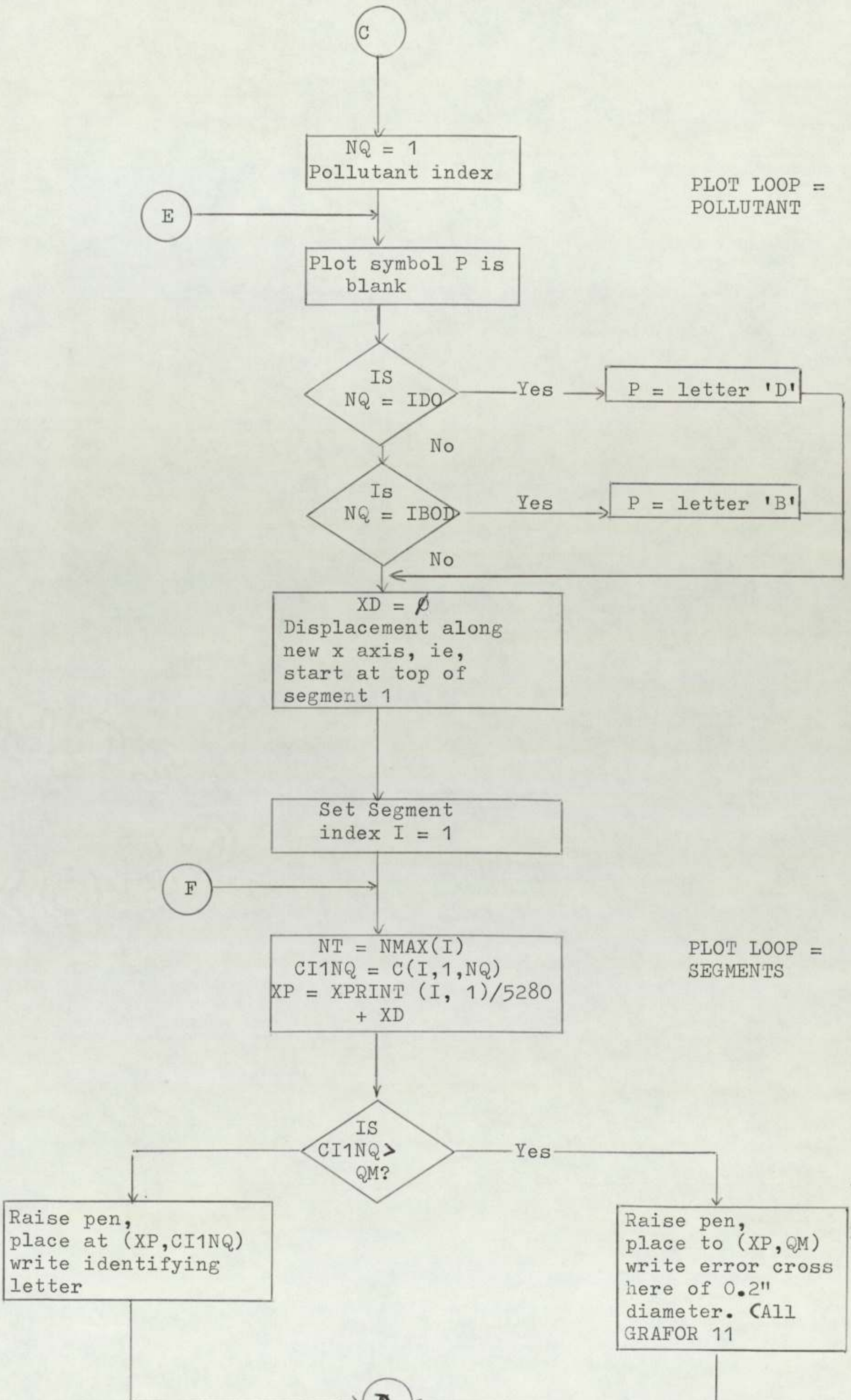
The x axis is constant for the problem, so can be set up in the first call to the routine. The y axis has to be flexible due to varying concentration levels. At each call, the extremes of concentrations are located and the y axis scaled so that if the range is <5, 1"≡1 p.p.m concentration unit. If the range is >5, the plot axis is scaled so that the y axis is 6" high.

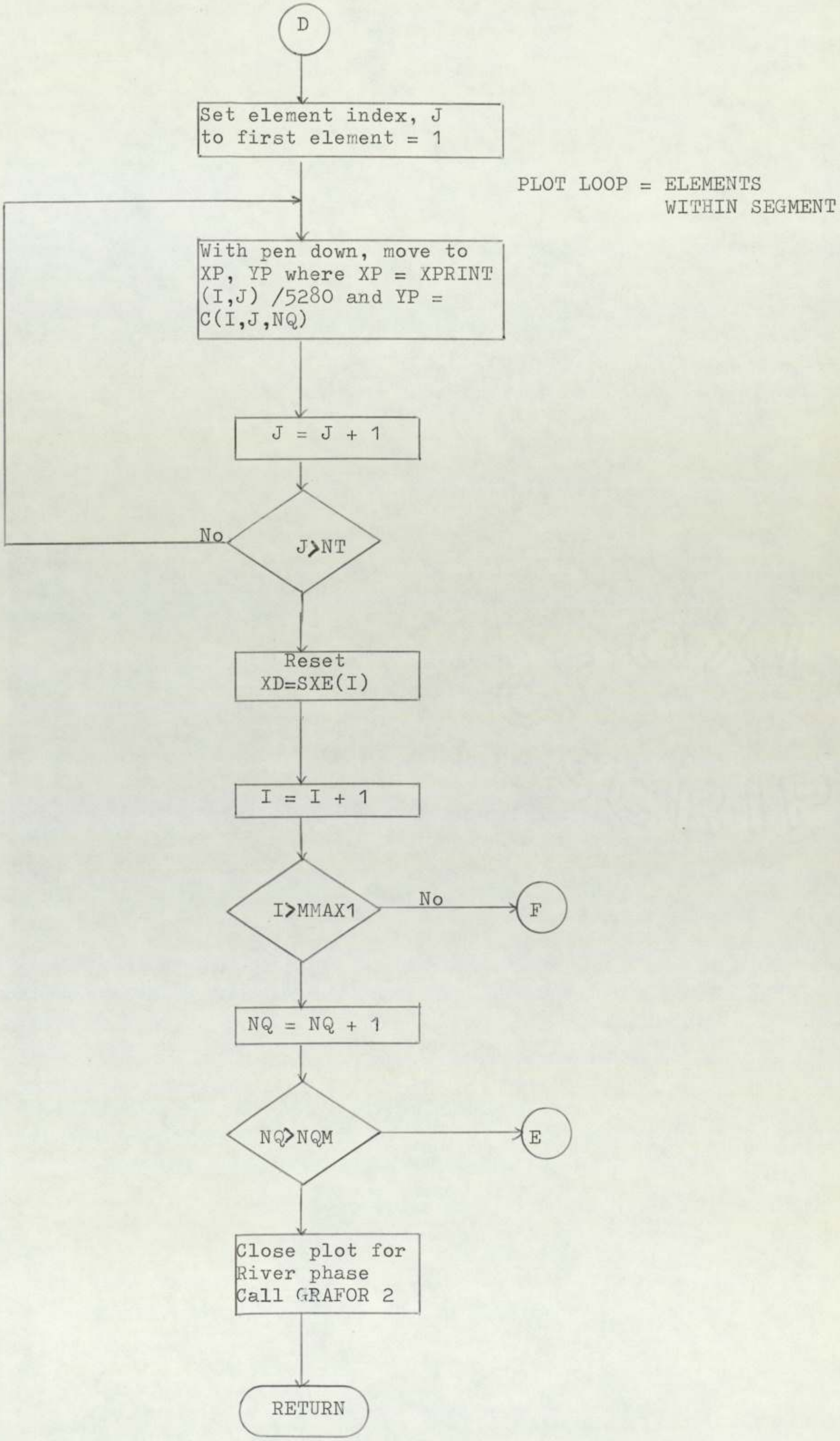
Along the top of the whole plot, arrows point to the positions in the segments where outfalls are physically located. Each pollutant is traced in turn. There is a break in the plot line across the y axis and as a label, the start of the O.D. and B.O.D. line have the letters 'D' and 'B' written on it.

The length of each segment in miles is written on the plot, as well as the segment index.


ENTER
Is it first call?
Yes
Calculate number of calls to GRAFOR per plot file generated
Stretch the segments of the river and set up one dimensional linear system. SXT/SXE hold start/ end of each segment referred to a new zero due to realignment
INITIAL SETTING UP OF RE-ALIGNED SYSTEM
Relocate each outfall position in relation to the new alignment. Positions are in array DAD
Set X axis scale to be 20" max across page y axis scale to unit
Set QM = C(1,1,1) = QL
Calculate maximum concentration QM, minimum concentration QL for all segments and all pollutants
FIND CONCENTRATION EXTREMES
For this system, exclude segment 8, element 1 concentration from QM, QL determination
A


B
Upper concentration limit IM is max. of 2 and QM rounded up
Lower concentration limit IL is min. of Ø and QL rounded down. Range IR = IM - IL
CALCULATE y AXIS SCALE
YSCAL = 1" per concentration unit unless IR> 5, in which case the plot is scaled so that the y axis is 6" high
Call GRAFOR 1 open plot file
Call GRAFOR 3 Set origin and axis Scales
PREPARE TO PLOT
Call GRAFOR 4 Plot segments onto axis
Draw displaced y axis at ends of each segment
Write lengths of each segment onto x axis
Heading on x & y axis
Put arrows in positions indicating discharges to the system
C


C
NQ = 1
Pollutant index
PLOT LOOP =
POLLUTANT
E
Plot symbol P is
blank
IS
NQ = IDO
Yes
P = letter 'D'
No
Is
NQ = IBOD
Yes
P = letter 'B'
No
XD = Ø
Displacement along
new x axis, ie,
start at top of
segment 1
Set Segment
index I = 1
F
NT = NMAX(I)
CI1NQ = C(I,1,NQ)
XP = XPRINT (I, 1)/5280
+ XD
PLOT LOOP =
SEGMENTS
IS
CI1NQ >
QM?
Yes
Raise pen,
place at (XP,CI1NQ)
write identifying
letter
Raise pen,
place to (XP,QM)
write error cross
here of 0.2"
diameter. CAll
GRAFOR 11
D


D
Set element index, J to first element = 1
PLOT LOOP = ELEMENTS WITHIN SEGMENT
With pen down, move to XP, YP where XP = XPRINT (I,J) /5280 and YP = C(I,J,NQ)
J = J + 1
No
J>NT
Reset XD=SXE(I)
I = I + 1
I>MMAX1
No
F
NQ = NQ + 1
NQ>NQM
E
Close plot for River phase Call GRAFOR 2
RETURN

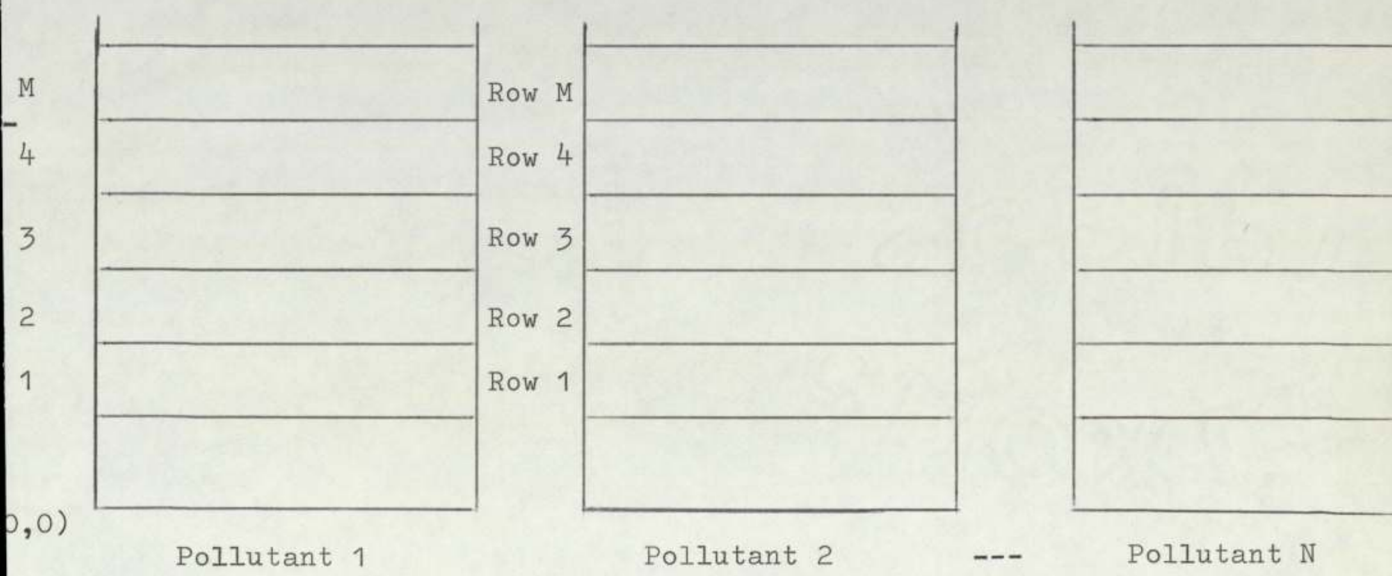
.19 Subroutine PLTBAY

Function	To provide a graphical display for predicted bay concentrations from model PT
System Functions	GRAFOR, FLOAT, READ/WRITE I/O, END & ERR options
User Functions	PTSUBY
Calling Statement	CALL PLTBAY
Common Areas	/IO/, /ALL/, /ESTOUT/, /BAYC/, CONSTI/, /DLIM/
Common Variables Reset	Those in /DLIM/, /ESTOUT/ is defined through input

Description

The variable geometry of the bay is constant for a run of simulations. In the first call of the routine the first pollutant is analysed for boundaries. Each row is scanned to locate the first and last column of the water field and the column numbers are retained in the arrays ISTRW, INDRW. These values are printed out for reference. Also on the initial call, a request is made to the main input stream for bayline description. If none is found the end option is switched on and the program continues. Also on an error input, the program will continue. If there is input, it is printed for reference. If no outline is present or an error on input occurs, NEST is set to zero.

A 3 level loop DO 103 I =, DO 103 J =, DO 104 N = searches for the extremes of the concentrations RMAX, RMIN for all pollutants. The x axis and y axis scale are calculated. The plot is split as shown in the disgram:



Each window has a y axis range from RMIN rounded down to RMAX rounded up. There are MMAX2*NQM lines, as each row for each pollutant gives rise to one line. Some displacement factors have to be calculated as the whole series is most easily drawn using a common axis referred to (0,0) at the concentration RMIN rounded down, for pollutant 1. As scales are the same for all sections of the plot, the plot file is opened and axis scales set in the main routine. The routine PTSUBY plots each substance in one of the sub sections as anove. The final call to GRAFOR closes the plot file, a GRAFOR 2 call.

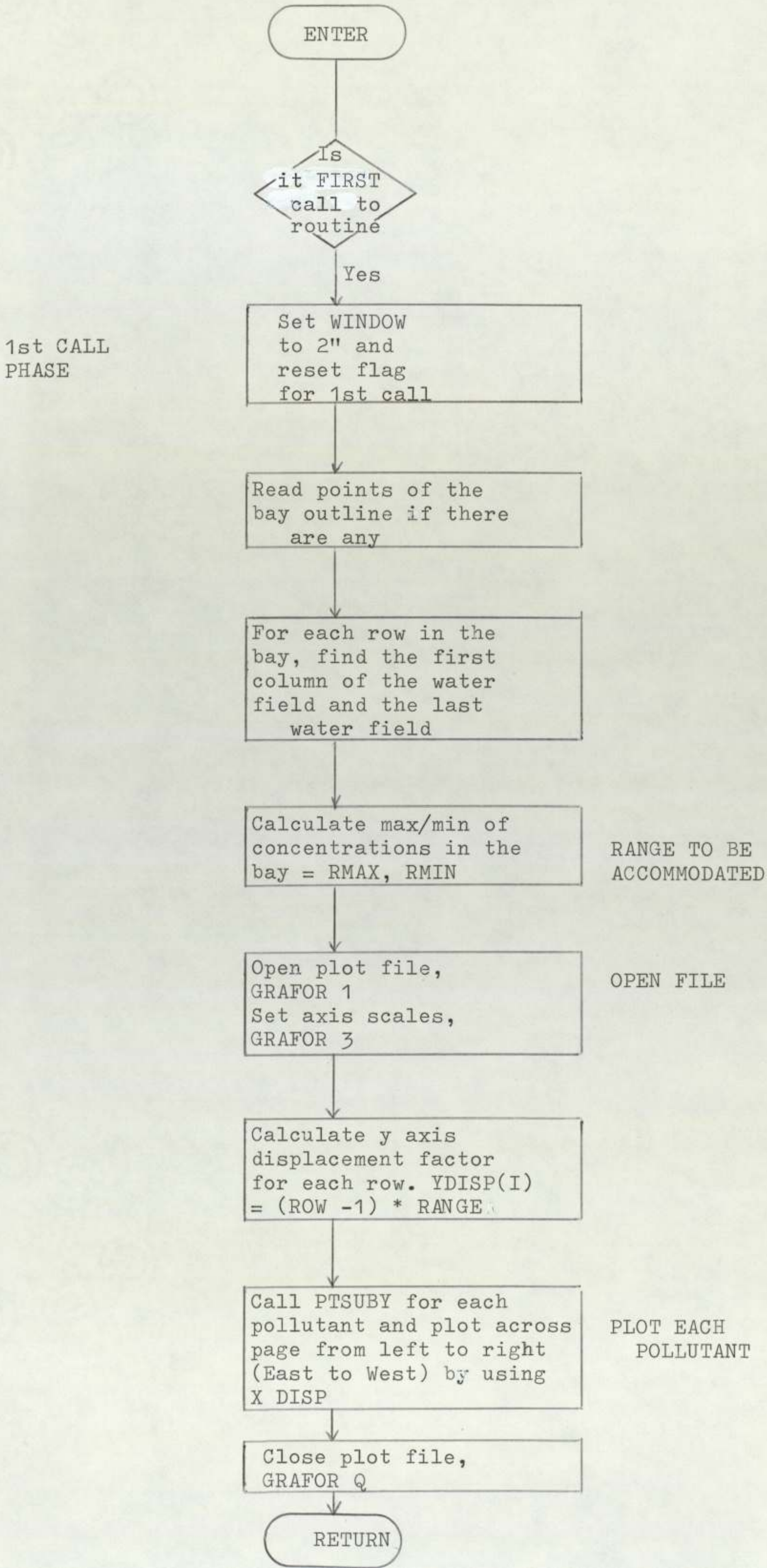
C.20 Subroutine PTSUBY

Function	To plot predicted bay concentrations for each individual pollutant on its own but aligned overall to each other and on a common axis and zero of data.
System Functions	GRAFOR
User Functions	None
Calling Statement	CALL PTSUBY(NQ, NMAX, MMAX, XDISP)
Common Areas	/IO/, /ESTOUT/, /BAYC/, /CONSTI/, /IF/, /DLIM/ .
Common Variables Reset	None

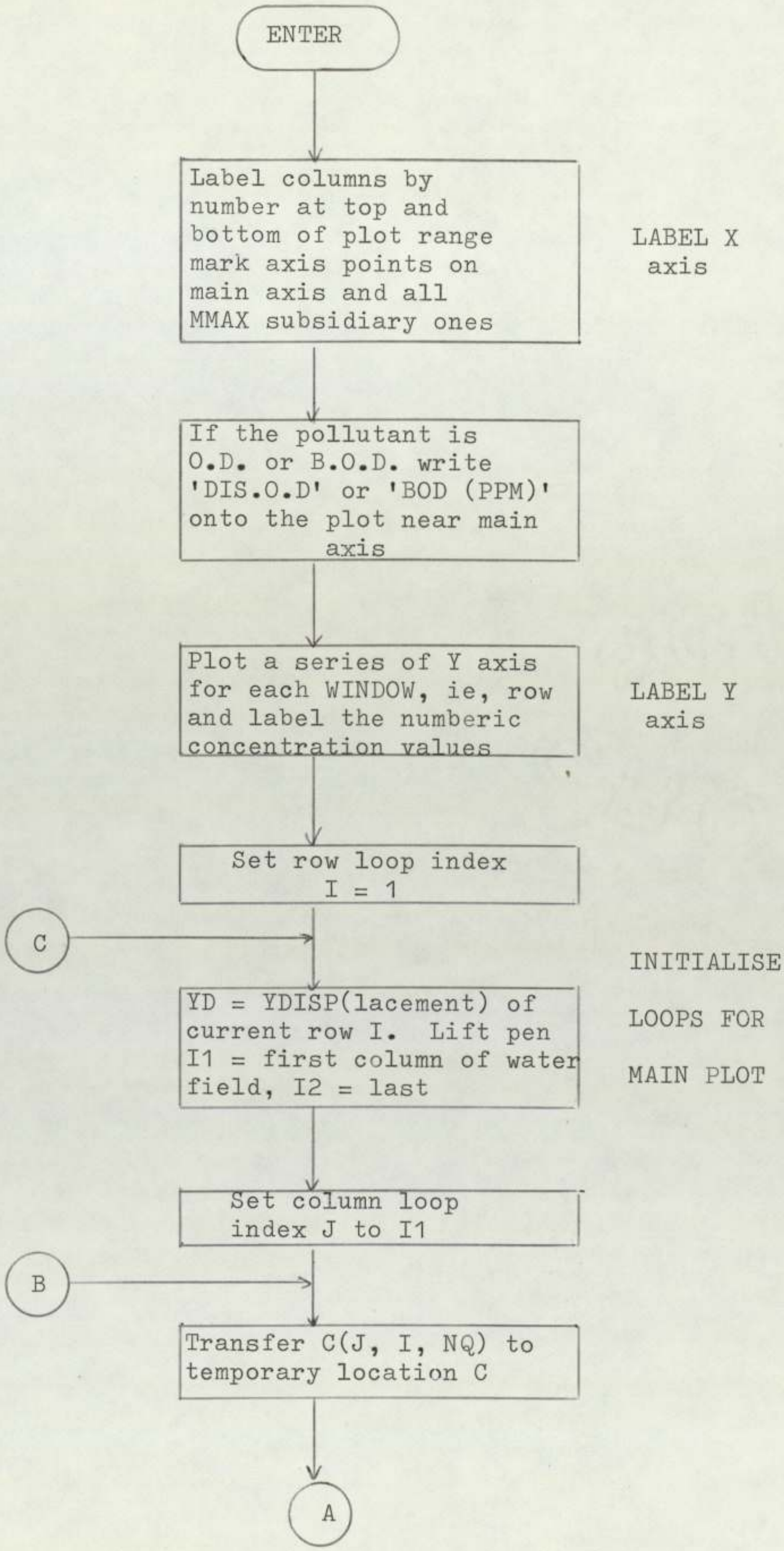
Description

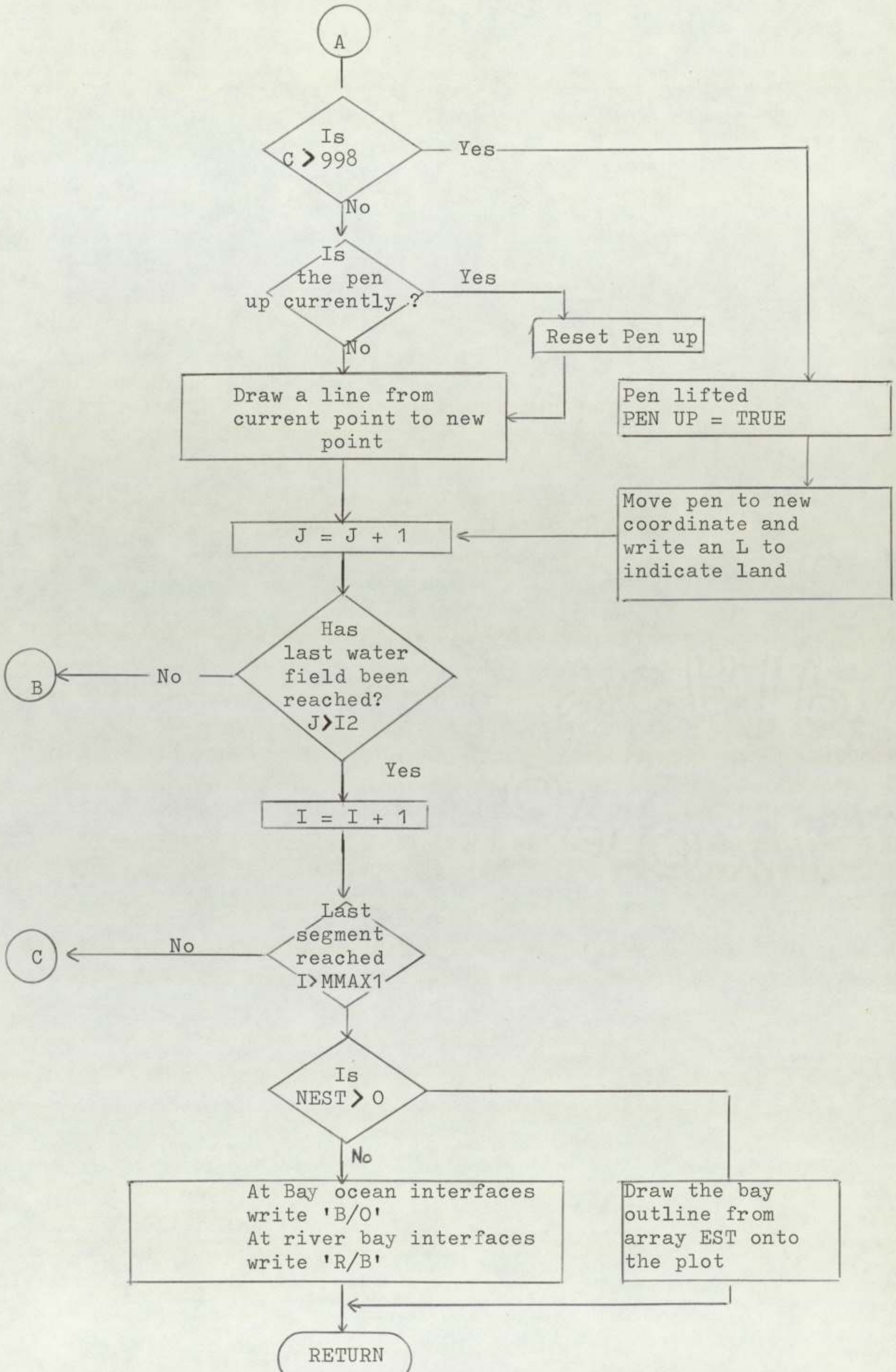
The routine plots on a CALCOMP using a common axis referred to (0,0) for row MMAX1 and pollutant 1. The X/Y axis are labelled by rows and columns and those points not in the water field are labelled 'L'. One line results for each row water field. An outline of the bay is drawn for ease of interpretation. If this is not available, then the points of the interfaces, bay/ocean, bay/river are marked as 'B/O' or'R/B' respectively.

As currently written, the plot resulting is large, some 20" by 30" for the Usk Estuary problem. This is required because of the amount of detail shown. Any great reduction would affect comprehension of the data and should only be attempted with a microfilm output facility. By manipulation of XDISP it is alternatively possible to plot all distributions on a single axis, in which case all extraneous detail is overwritten NQM - 1 times and thus be boldly outlined, whereas each concentration would produce only a single line.



ENTER
Is it FIRST call to routine
Yes
Set WINDOW to 2" and reset flag for 1st call
1st CALL PHASE
Read points of the bay outline if there are any
For each row in the bay, find the first column of the water field and the last water field
Calculate max/min of concentrations in the bay = RMAX, RMIN
RANGE TO BE ACCOMMODATED
Open plot file, GRAFOR 1
Set axis scales, GRAFOR 3
OPEN FILE
Calculate y axis displacement factor for each row. YDISP(I) = (ROW -1) * RANGE
Call PTSUBY for each pollutant and plot across page from left to right (East to West) by using X DISP
PLOT EACH POLLUTANT
Close plot file, GRAFOR Q
RETURN


ENTER
Label columns by number at top and bottom of plot range mark axis points on main axis and all MMAX subsidiary ones
LABEL X axis
If the pollutant is O.D. or B.O.D. write 'DIS.O.D' or 'BOD (PPM)' onto the plot near main axis
Plot a series of Y axis for each WINDOW, ie, row and label the numberic concentration values
LABEL Y axis
Set row loop index I = 1
C
INITIALISE LOOPS FOR MAIN PLOT
YD = YDISP(lacement) of current row I. Lift pen I1 = first column of water field, I2 = last
Set column loop index J to I1
B
Transfer C(J, I, NQ) to temporary location C
A


A
Is C > 998
Yes
No
Is the pen up currently ?
Yes
Reset Pen up
No
Draw a line from current point to new point
Pen lifted
PEN UP = TRUE
Move pen to new coordinate and write an L to indicate land
J = J + 1
Has last water field been reached? J>I2
No
B
Yes
I = I + 1
Last segment reached I>MMAX1
No
C
Is NEST > 0
No
At Bay ocean interfaces write 'B/O'
At river bay interfaces write 'R/B'
Draw the bay outline from array EST onto the plot
RETURN

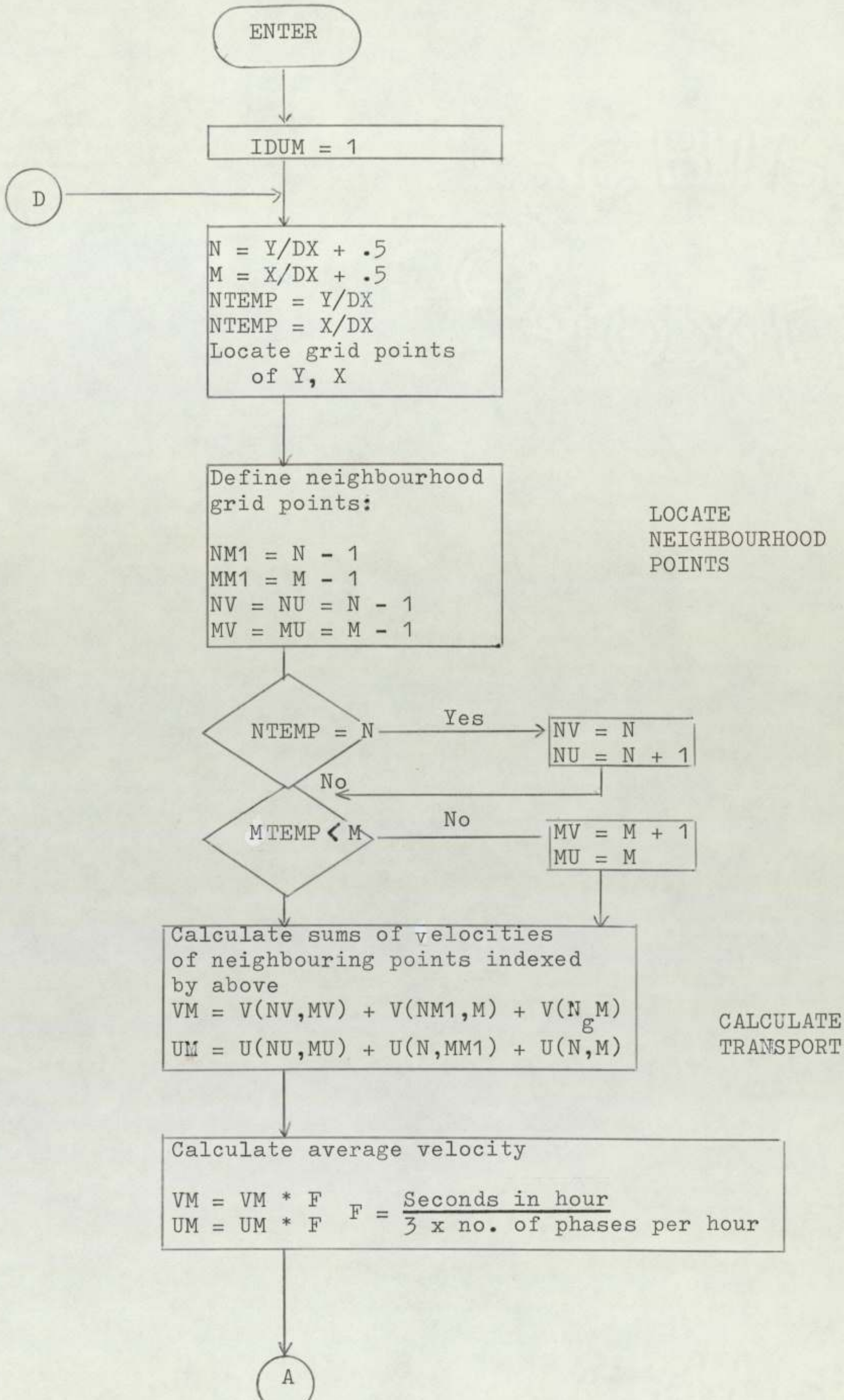
C.21 Subroutine TRNSPT

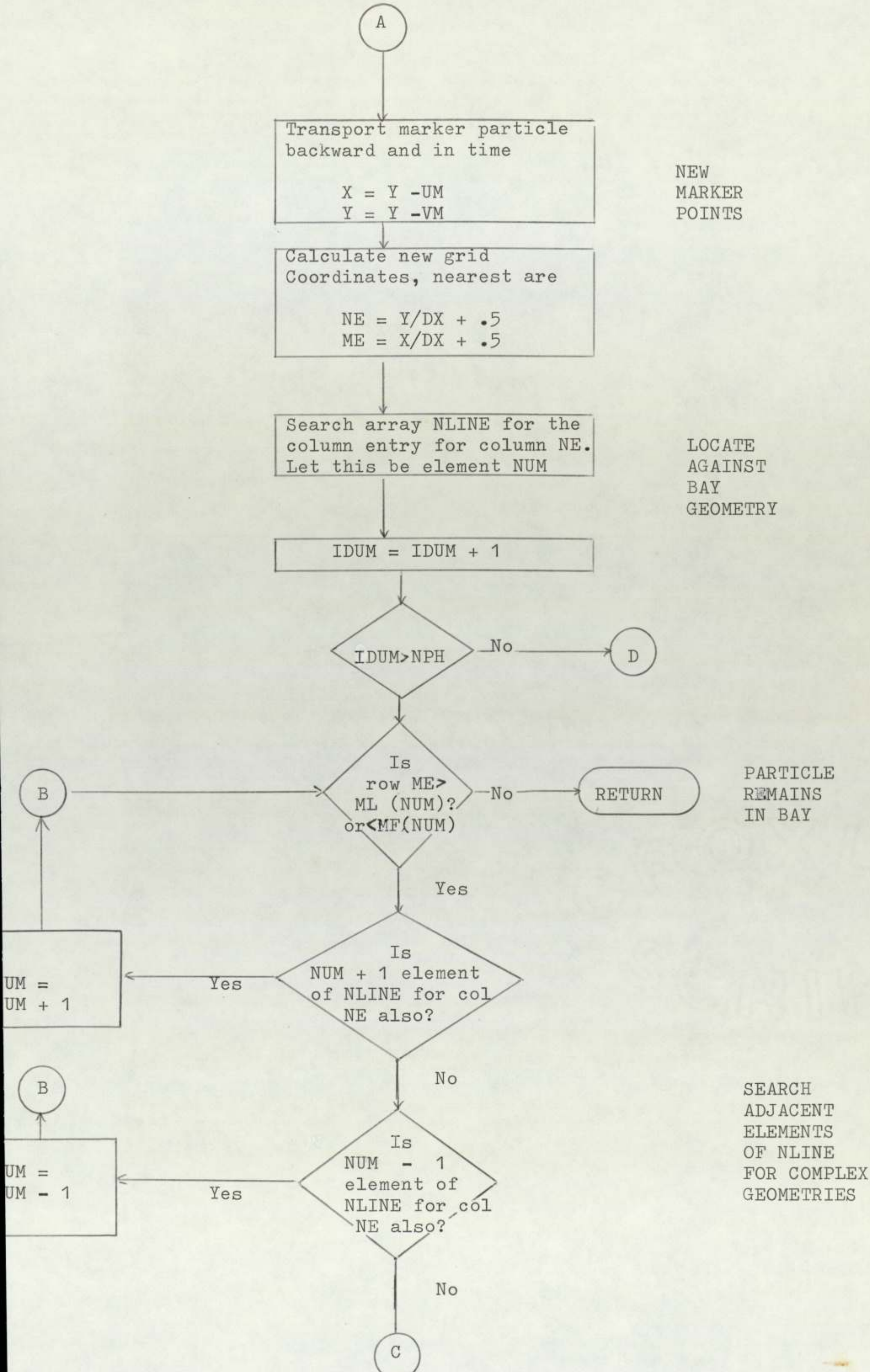
Function	To provide a transport step for the marker particles used to calculate pollutant concentrations
System Functions	None
User Functions	None
Calling Statement	CALL TRNSPT (Y, X)
Common Areas	/IO/, /UVCP/, /BAY1/, /TRANS/, /IF/, /ALL/
Common Variables Reset	Array IBOUND in /UVCP/

Description

This routine performs the movement of marker particles in an optional number of steps for a simulation period of one hour backwards in time for hour IT. (Y,X) are the coordinates of the marker particle at the start of the hour. This is located to a set of grid points from which corresponding mean velocities in X and Y directions are computed. The (X, Y) coordinate is then adjusted due to the backwards projection in time. A new grid point coordinate is calculated from the new position and interrogation of NLINE array will determine if the marker particle has left the bay across the ocean interface, across the river interface, or is still in the bay. The array IBOUND, element (NST, MST) ie, where the marker particle initiated, is labelled accordingly.

W LOGIC


ENTER
IDUM = 1
D
N = Y/DX + .5
M = X/DX + .5
NTEMP = Y/DX
NTEMP = X/DX
Locate grid points of Y, X
Define neighbourhood grid points:
NM1 = N - 1
MM1 = M - 1
NV = NU = N - 1
MV = MU = M - 1
LOCATE NEIGHBOURHOOD POINTS
NTEMP = N
Yes
NV = N
NU = N + 1
No
MTEMP < M
No
MV = M + 1
MU = M
Calculate sums of velocities of neighbouring points indexed by above
VM = V(NV,MV) + V(NM1,M) + V(N,M)
UM = U(NU,MU) + U(N,MM1) + U(N,M)
CALCULATE TRANSPORT
Calculate average velocity
VM = VM * F
UM = UM * F
F = Seconds in hour / 3 x no. of phases per hour
A

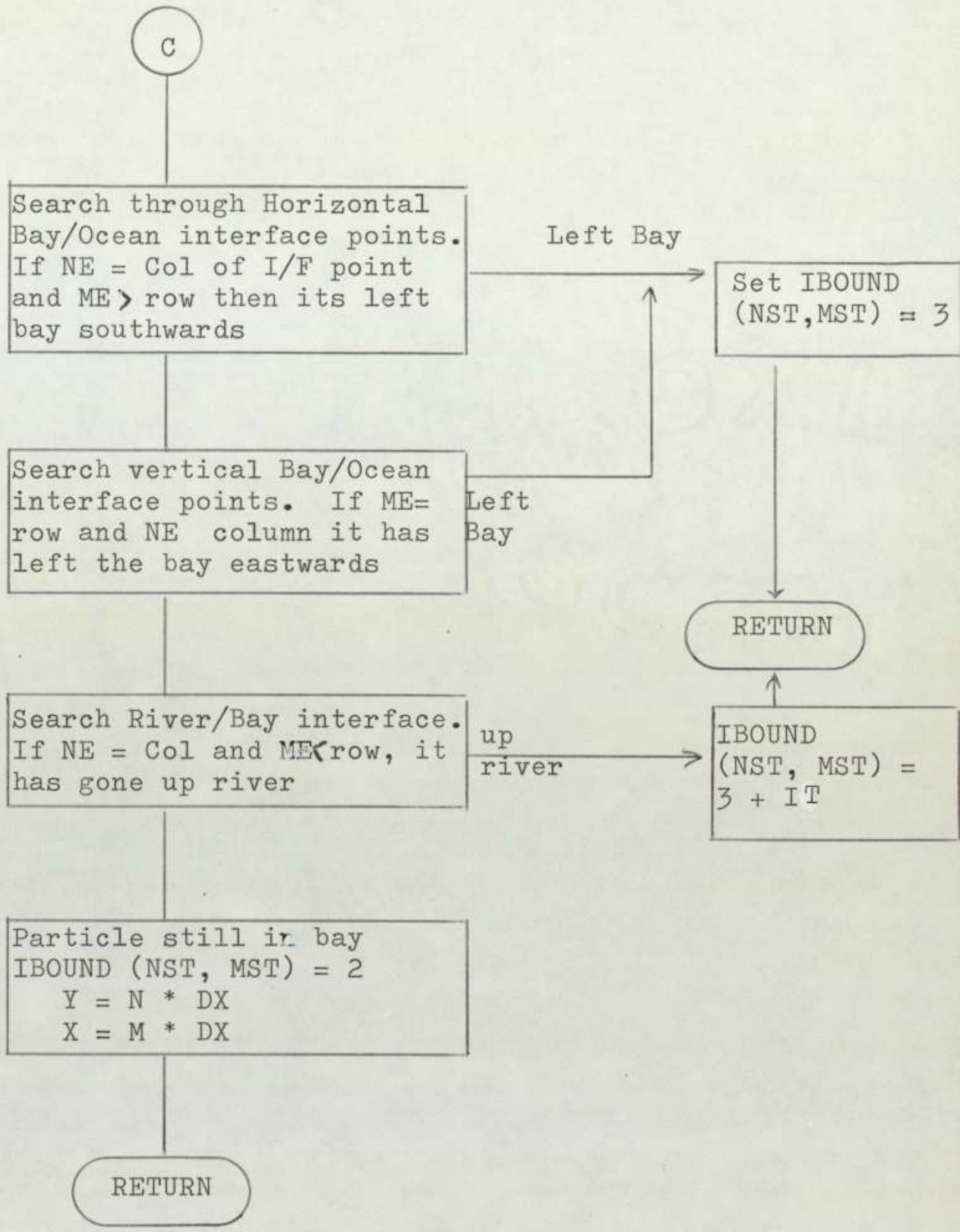

A
Transport marker particle backward and in time
X = Y -UM
Y = Y -VM
NEW MARKER POINTS
Calculate new grid Coordinates, nearest are
NE = Y/DX + .5
ME = X/DX + .5
Search array NLINE for the column entry for column NE. Let this be element NUM
LOCATE AGAINST BAY GEOMETRY
IDUM = IDUM + 1
IDUM>NPH
No
D
B
Is row ME> ML (NUM)? or<MF(NUM)
No
RETURN
PARTICLE REMAINS IN BAY
Yes
Is NUM + 1 element of NLINE for col NE also?
Yes
UM = UM + 1
No
B
Is NUM - 1 element of NLINE for col NE also?
Yes
UM = UM - 1
No
C
SEARCH ADJACENT ELEMENTS OF NLINE FOR COMPLEX GEOMETRIES

TRACE

PARTICLE

ACROSS

BOUNDARY



C.22 Timings

For the Usk Estuary investigation, a 16 x 11 Bay grid and a 9 segment one dimensional phase was used. Time steps of 10 - 15 minutes were used

PT1 About 130 etus (35 seconds) mill time
on 4/70 per main 25hr cycle

PT About 180 etus (45 seconds) mill time
on 4/70 per main 25 hour cycle

C.23 Data Input

Variable	Description
ISWITC	Set to 1 if initial phase is FLOOD tide. Set to 2 if initial phase is EBB tide.
NSKIP	If negative, the temporary files have already been established by a previous run and so TAPECH merely checks the problem parameters before control returns to POLTRA.
NQM (in PT1)	If negative, the number of pollutants is ABS (NQM) and the concentrations of pollutants at the downstream boundary are read in by subphase, hour in subphase and pollutant (Record 12B)
IDMPB (in PT)	If set to zero, computations performed in the bay section of the model will be traced to the line printer.
BAYPLT (in PT)	If set to TRUE the bay predictions are plotted when they are also due for printing.
NPASS 1 (in PT)	If zero, no report of initial height/width preparative computations in RIVQAL is sent to printer.
NPASS 1 (in PT1)	If zero is in PT. If >0, depth/width tables printed for all segments at intervals of NPASS1 time steps.
NPASS 2	If zero, on cycle 1, inflow and outflow, and travel distance, is printed for all segments every timestep.
IDMPR	If set to zero, a trace facility for river computations is switched on and results in much output
RIVPLT	If set to TRUE predictions for river are plotted when they are printed also.

Itemised Data Input Structure

rd e	For Model	Col from	Col to	Variable	Type	Description
	*PT	Y		ITP1	I	Main input tape from F1/F2
				ITP2	I	Bay temporary velocities file
				ITP3	I	River temporary data file
				ISWITC	I	Flood/Ebb phase initially?
				MMAX1	I	Segments in river phase
				NMAX2	I	Columns in bay
				MMAX2	I	Rows in bay
				IDO	I	Parameter index for O.D.
				IBO	I	Parameter index for B.O.D.
	*PT1	Y		ITP1	I)	Refer Rec Type 1
				ITP3	I)	
				ISWITC	I)	
				MMAX1	I)	
				IDO	I)	
				IBOD	I)	
		Y		NREP	I	Number of subphases
				ITREP	IA	Hours duration of each of NREP subphases
				NSKIP	I	Number of records to skip before date to be retained occurs
	*PT1	Y		NCYCLE	I	Number of main cycle to simulate
				N	I	Number of cyles to be printed
				IPRINT	IA	Array hold up cycles to be printed.
	*PT	Y		NCYCLE	I	Number of cycles to be simulated
				CONCGI	RA4	Concentration of each pollutan in ocean
				N	I	Number of cycles to be printed
				IPRINT	IA	Array holding cycles to be printed
		1	80	IDEN	AA20	Title array, 20x4 characters
		4A8		PARAM	AA4	8 character identification of pollutants
	*PT1	Y		NQM	I	Number of pollutants to be simulated
	*PT1	Y		DECAY	RA4	Decay coefficients of pollutants
	*PT	Y		NIND	I	Column identifiers require NNIND vectors
				NQM	I	Number of pollutants
				DECAY	RA4	Decay coefficients of pollutan
	*PT	Y		NLINE(M)	I	Column in bay
				MF(M)	I	First water field row for colm

NOTE Record 11 appears NIND times

rd e	For Model	Col from	Col to	Variable	Type	Description
	*PT1	Y		CONCGI	RA4	Constant downstream boundary concentration of pollutants
	*PT1	Y		CONCBY (I,J,NQ)	RA	Concentration of pollutant NQ at hour J for sub cycle I NQ = 1 NQM on any one record

NOTE Record 12B appears IR times of each of NREP sub cycles

*PT	Y	DX	R	Grid step in feet for bay
		NPH	I	Number of sub hourly transport steps
		IDMPB	I	Bay computation trace option
		BAYPLT	L	Bay data plotting option
*PT	Y	NRBIF	I	Number of river/bay interface points
		RBIF(I,1)	I	Column of I^{th} river bay interface point
		RBIF(I,2)	I	Row of I^{th} river bay interface point
		I = 1,NRBIF		
*PT	Y	NBOIFV	I	Number of vertical bay ocean interface points
		BOIFV(I,1)	I	Column of I^{th} bay ocean interface point
		BOIFV(I,2)	I	Row of I^{th} bay ocean interface point
		I = 1,NBOIFV		
*PT	Y	NBOIFH	I	Number of horizontal bay ocean interface points
		BOIFH(I,1)	I	Column of I^{th} horizontal bay ocean interface points
		BIOFH(I,2)	I	Row of I^{th} horizontal bay ocean interface points
	Y	EFOR	R	Diffusion coefficient for river
		NPASS1	I	See note in C.23
		NPASS2	I	Print flows/velocities on cycle 1
		IDMPR	I	Trace river computation options
		NIT	I	Time steps per hour
		ITSLUG	I	Spare
		NOUTF	I	Number of discharges
		RIVPLT	L	Plot river predictions option
	YOF8.0	QMEAS	RA9	Read fresh water inputs to head of each segment
	Y	CMEAS(M,NQ)		For pollutant NQ, initial concentrations in each segment
	Note Record 19 appears NQM times			
	Y	XL(M)	R	Length of segment M
		MA(M)	I	1st upstream segment
		MB(M)	I	2nd upstream segment
		MC(M)	I	3rd upstream segment
		MD(M)	I	1st downstream segment
	Note Record 20 appears MMAX1 times			
	Y	MINPUT(I)	I	Segment including outfall I
		XADD(I)	R	Distance from top of segment to outfall I
		HGHT(I)	R	Height of outfall above river bed
		DF(I)	R	Decay factor of discharge, if any
		ADD(I,NQ)	R	Role of addition of each pollutant per hour
	Note Record 21 appears NOUTF times			

C.25 Data Record Sequences for Models PT/PT1

Model PT

Record No:
1
3
5
6
7
10
11 REPEATED : NIND
13
14
15
16
17
18
19 REPEATED : NQM
20 REPEATED : MMAX1
21 REPEATED : NOUTF

Model PT1

Record No:
2
3
4
6
7
8
9
12A DEPENDING ON NQM
12B
17
18
19 REPEATED : NQM
20 REPEATED : MMAX1
21 REPEATED : NOUTF

APPENDIX D.

A DETERMINISTIC STOCHASTIC MIXED MODEL OF ESTUARINE POLLUTANT TRANSPORT.

APPENDIX D.

C O N T E N T S.

D1 Introduction.

There are two models described here, ST and ST2. They differ only in the first section, the computation of predicted means. Model ST uses the approximation that velocity is a superimposed sum of a constant freshwater velocity and a summesoidally varying tidal velocity. Model ST2 however, uses the predicted composite velocities from the models F1 or F2 via a preparative routine FSHCHA. Sections 2 to 17 deal with the determination of the mean. The subsequent sections with computing estimates of the stochastic coefficient (DELTA) and with combining the two ideologies together to give a predicted probability distribution either through time at one point or through space at a fixed tidal phase.
The DATE-TIME package used previously is also employed here. The latter sections were not used in the project because of lack of inputs available.

Executive Routine DIFEQ(forST)

Function: To provide the executive function for model ST

System Functions: READ/WRITE I/O,FLOAT,TIME

User Functions: INPUT,STPF,MULT,IPLANS,FUNCT,RKMI,POUT

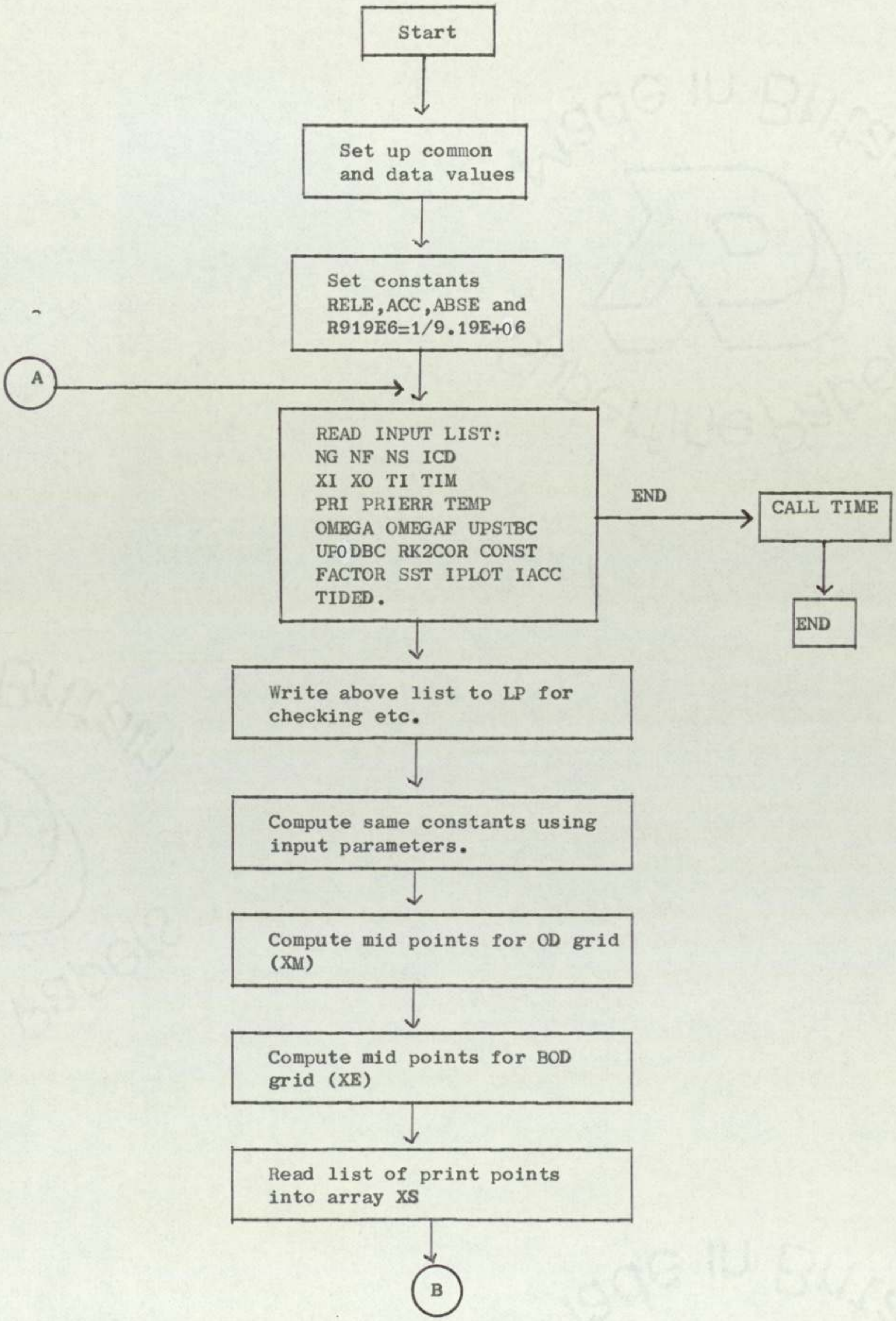
Common Areas: /A/,/B/,/C/,/D/,/E/,/F/,/G/

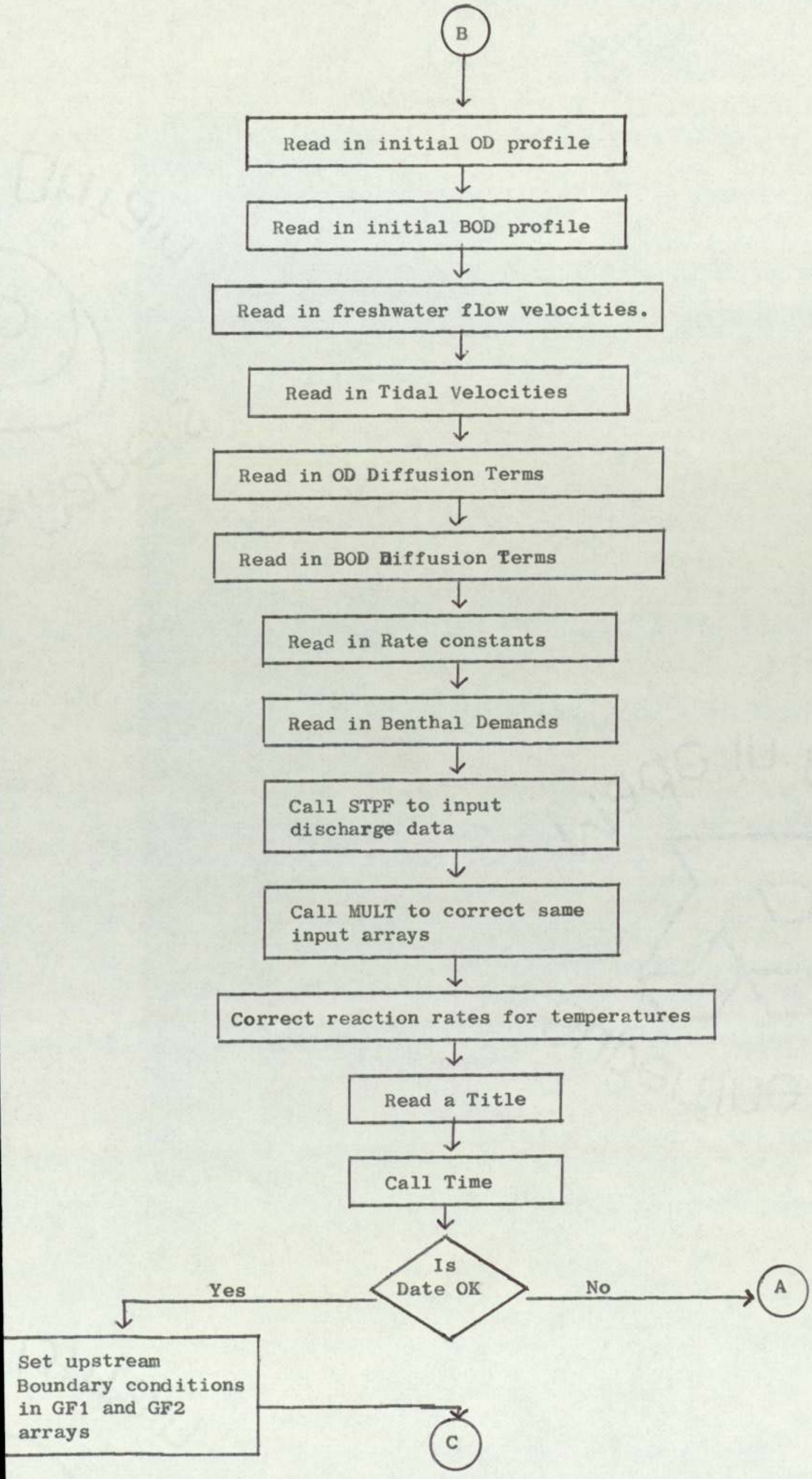
Description:

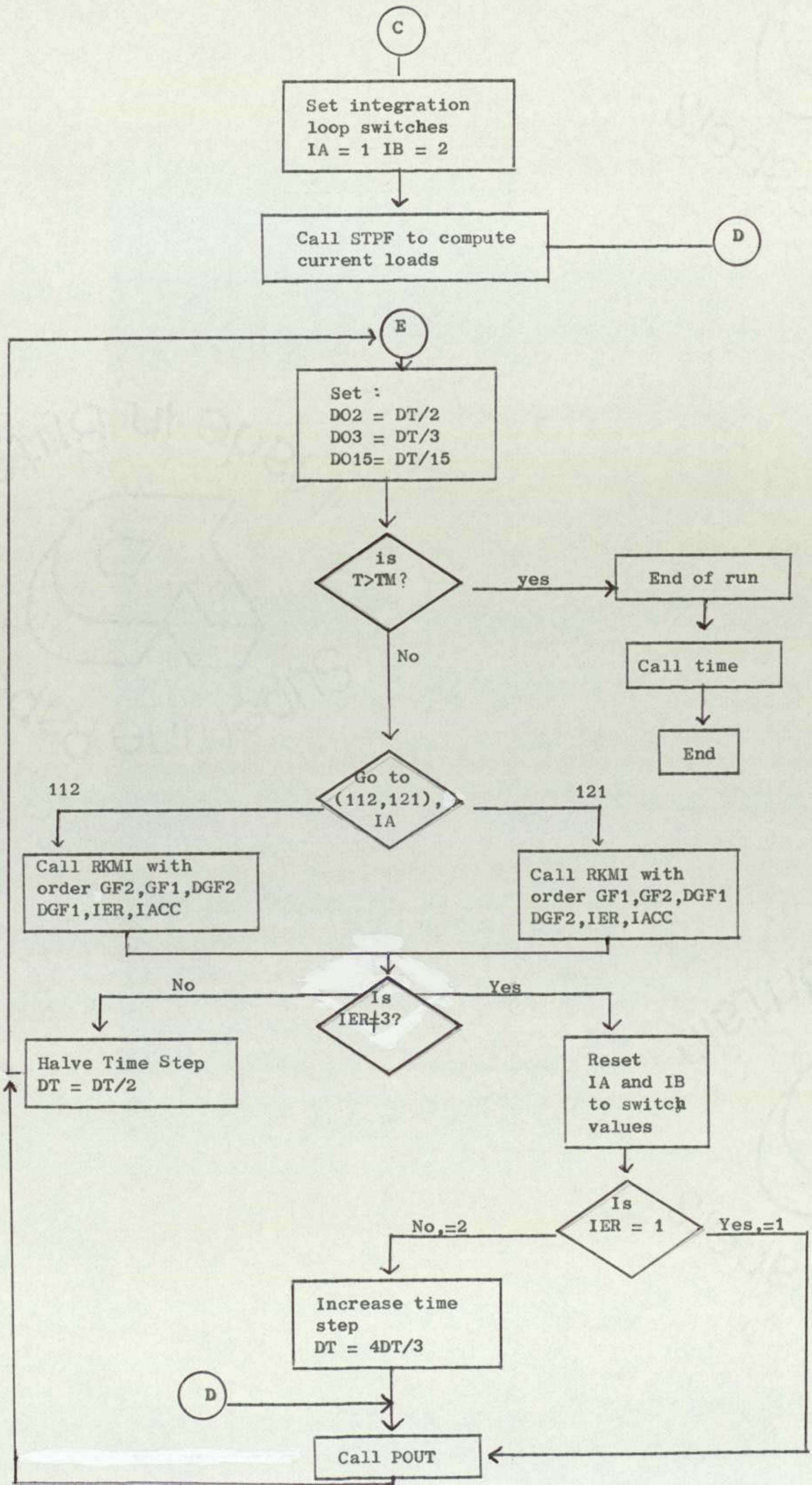
The initial problem parameters are read in to steer the routines. Certain variables are computed for use as constants and the estuary parameters are entered through the routine INPUT. A dummy call to STPF inputs discharge data if any. Calls to MULT ensure correct units. Dates are tested for plausibility and boundary conditions are set up. The routine then cycles through the integration loop with occasional adjustment of time steps for optimal use of mill time. When triggered by input parameters, POUT is called to print an output summary.

The SOLUTION COMPLETE message is generated when the set simulation time is exceeded.

Flow Logic


Start
Set up common and data values
Set constants RELE,ACC,ABSE and R919E6=1/9.19E+06
A
READ INPUT LIST:
NG NF NS ICD
XI XO TI TIM
PRI PRIERR TEMP
OMEGA OMEGAF UPSTBC
UP0DBC RK2COR CONST
FACTOR SST IPLOT IACC
TIDED.
END
CALL TIME
END
Write above list to LP for checking etc.
Compute same constants using input parameters.
Compute mid points for OD grid (XM)
Compute mid points for BOD grid (XE)
Read list of print points into array XS
B


B
Read in initial OD profile
Read in initial BOD profile
Read in freshwater flow velocities.
Read in Tidal Velocities
Read in OD Diffusion Terms
Read in BOD Diffusion Terms
Read in Rate constants
Read in Benthal Demands
Call STPF to input discharge data
Call MULT to correct same input arrays
Correct reaction rates for temperatures
Read a Title
Call Time
Is Date OK
Yes
No
A
Set upstream Boundary conditions in GF1 and GF2 arrays
C


C
Set integration loop switches IA = 1 IB = 2
Call STPF to compute current loads
D
E
Set :
DO2 = DT/2
DO3 = DT/3
DO15= DT/15
is T>TM?
yes
End of run
Call time
End
No
Go to (112,121), IA
112
121
Call RKMI with order GF2,GF1,DGF2 DGF1,IER,IACC
Call RKMI with order GF1,GF2,DGF1 DGF2,IER,IACC
No
Yes
Is IER≠3?
Halve Time Step DT = DT/2
Reset IA and IB to switch values
Is IER = 1
No,=2
Yes,=1
Increase time step DT = 4DT/3
D
Call POUT

D3 Executive Routine DIFEQ (for ST2)

Function: As DIFEQ (for ST).

Description:

The routine is identical to D2 with the exception of computation of velocities. Input of US and UT are not requested. Where velocities are required, control goes to a routine FILL. At any one time, two velocity profiles are in store; one for the current or just passed time, one for the next time that F1/F2 predicted velocities. At each call of FIU, a simple linear interpolation provides an estimated velocity profile for the current time. If there is a long period between F1/F2 predictions (in this study one hour was found to be the longest acceptable period) then the interpolation may have to be of a higher order. FILL also controls the reading in of new values as required. The values used are for a whole number of tidal cycles and are used repetitively by FILL to enable steady state to be reached. Alternatively the perturbations over tidal cycles in F1/F2 as they settle to steady state values filter through and could interfere constructively with ST perturbations.

Subroutine INPUT

D.4

Function	To act as the major input facility for models ST or ST2
System Functions	READ/WRITE I/O, ABS
User Functions	INTP, P
Calling Statement	CALL INPUT (IC, J, K, U, DX)
Common Areas	/F/, /C/, /INP/.
Common Variables Reset	All of /INP/, All of /C/

Description.

Calling List

IC	peripheral reference for input file.
J	1st element of array to be filled.
K	Last element of array to be filled.
U	Array to be filled.
DX	Grid size of distance mesh from XI -> XO

In a comprehensive estuary model, many variables are known to a different degree of accuracy and also have different characteristics.
This routine recognises four types of data input in the estuary modelling context; type switch set by variable IPT.

IPT Value	Type of Data
-2	Parameter has single value all through estuary. This value is copied to all array elements.
-1	Parameter values only occur at points, otherwise they are zero. Examples are effluent point discharges.
0	The parameter is constant within a segment, but variable over segments.
>1	The parameter is known as a smooth spatially distributed variable. The precise value of IPT is used to indicate the number of x-y data pairs available. These values are interpolated over the region XI to XO.

Each call to INPUT fills one array. The header card for each array has a fixed format and the layout is

Col from	Col to	Variable	Description
1	2	IPT	As above
3	10	-	Blank
11	60	D	Alphabetic Description field.
61	70	VAL	Floating point variable used in conjunction with IPT

The actual data values, when applicable, are read in under free format, col 1 to 70 with at most 5 pairs per card image.

<u>If IPT = -2</u> only the 'DO 2' loop is executed. In every element of U from J to K, the value of VAL is inserted.

<u>If IPT = -1</u> the 'DO 4' loop clears the array U within the limits of interest. The value of 'VAL' is truncated to an integer IVAL and a read request is issued for IVAL data pairs of distance - parameter value. Each distance is converted to a grid point by division by DX and truncation (note implicit assumption that XI = 0.0 in usual cases). The distance of the point source from the two nearest grid points are calculated and the point value is added to U at the nearest point. Should a source fall exactly halfway between two grid points, the parameter is halved and proportioned to the two grid points. Values need not appear in order of position.

<u>If IPT = 0</u> the value of VAL is truncated to give IV, the number of distance - parameter values in ascending distances order. The program considers a segment of the estuary to run between two consecutive values of distances in the input arrays.

For points within XI (assumed zero) to D1ST (1) the value of the parameter is set to D2ND(1).

For points from D1ST(I) to D1ST(I+1) the value of the parameter used is D2ND(I+1).

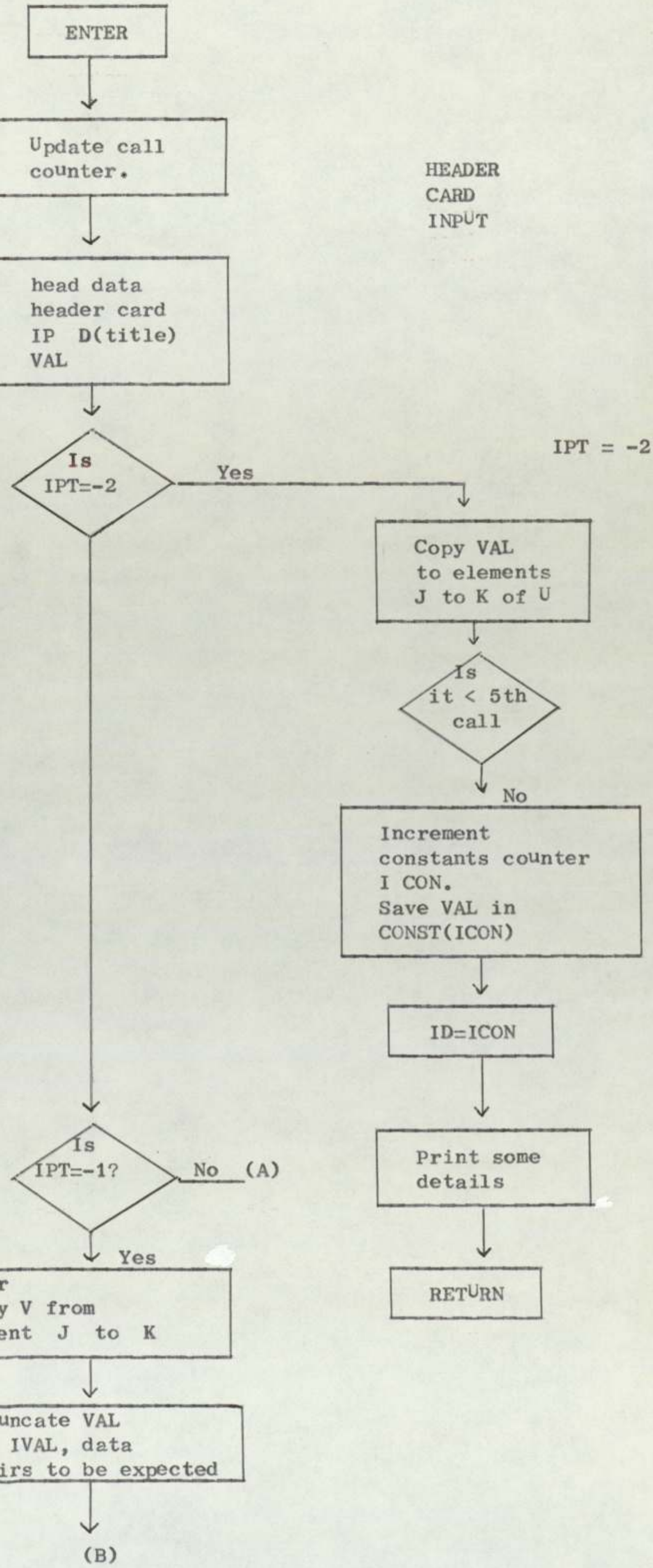
For points from D1ST(IV) to XO the value of the parameter used is D2ND(IV).

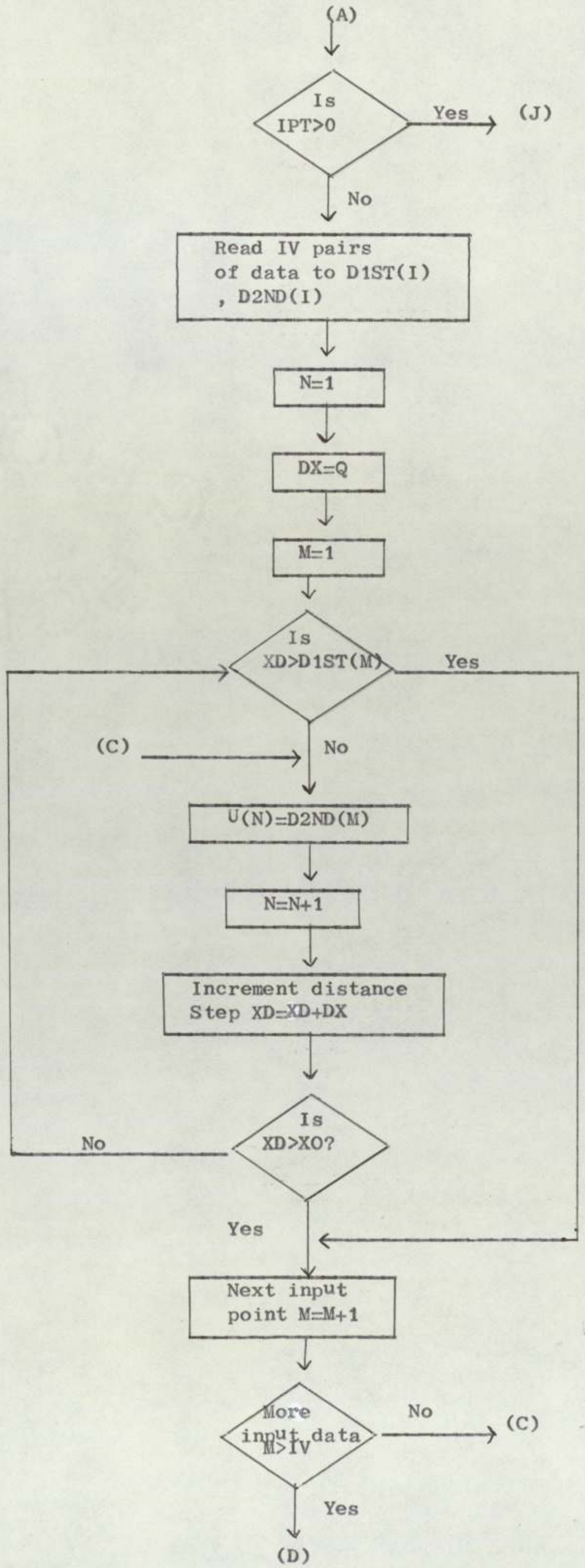
__If IPT>0__ This switch expects to find IPT data pairs of the type: (distance from zero) - (value of parameter). Then the routine INTP is used to interpolate the read in values to the grid points between XI and X0. The first values should be at a distance ≤XI and the last value ≥X0 to ensure stability of the interpolation process. To change the accuracy, reset the last parameter in the list.

Finally, the input data having been laid over the internal grid is printed for a visual check.

__Note__: When used by DIFEQ, an internal count of the number of IPT = -2 is made, if this is counter, (ID in /INP/) is seven at the end of input, a special switch is set to make functional evaluation more efficient.

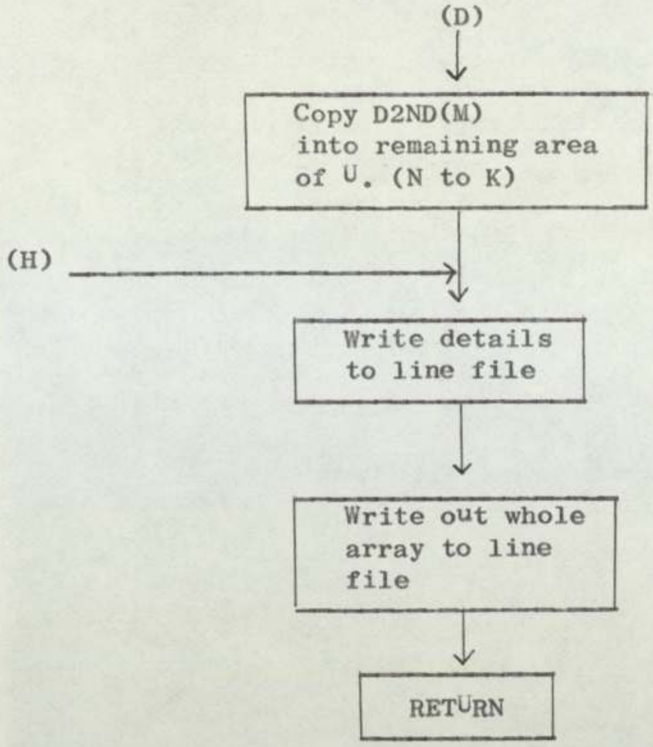
Flow Logic.


ENTER
Update call counter.
head data header card IP D(title) VAL
HEADER CARD INPUT
Is IPT=-2
Yes
IPT = -2
Copy VAL to elements J to K of U
Is it < 5th call
No
Increment constants counter I CON. Save VAL in CONST(ICON)
ID=ICON
Print some details
RETURN
Is IPT=-1?
No (A)
Yes
Clear array V from Element J to K
Truncate VAL to IVAL, data pairs to be expected
(B)

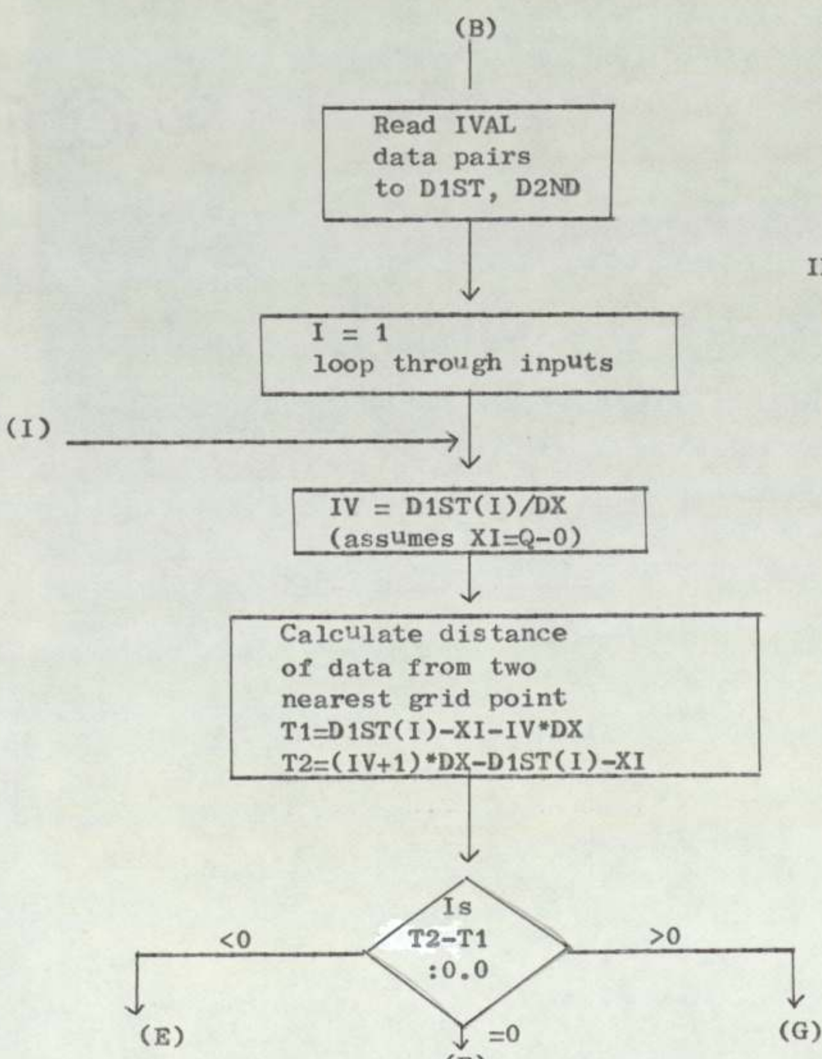


IPT = 0

CONSTANT WITHIN SEGMENTS

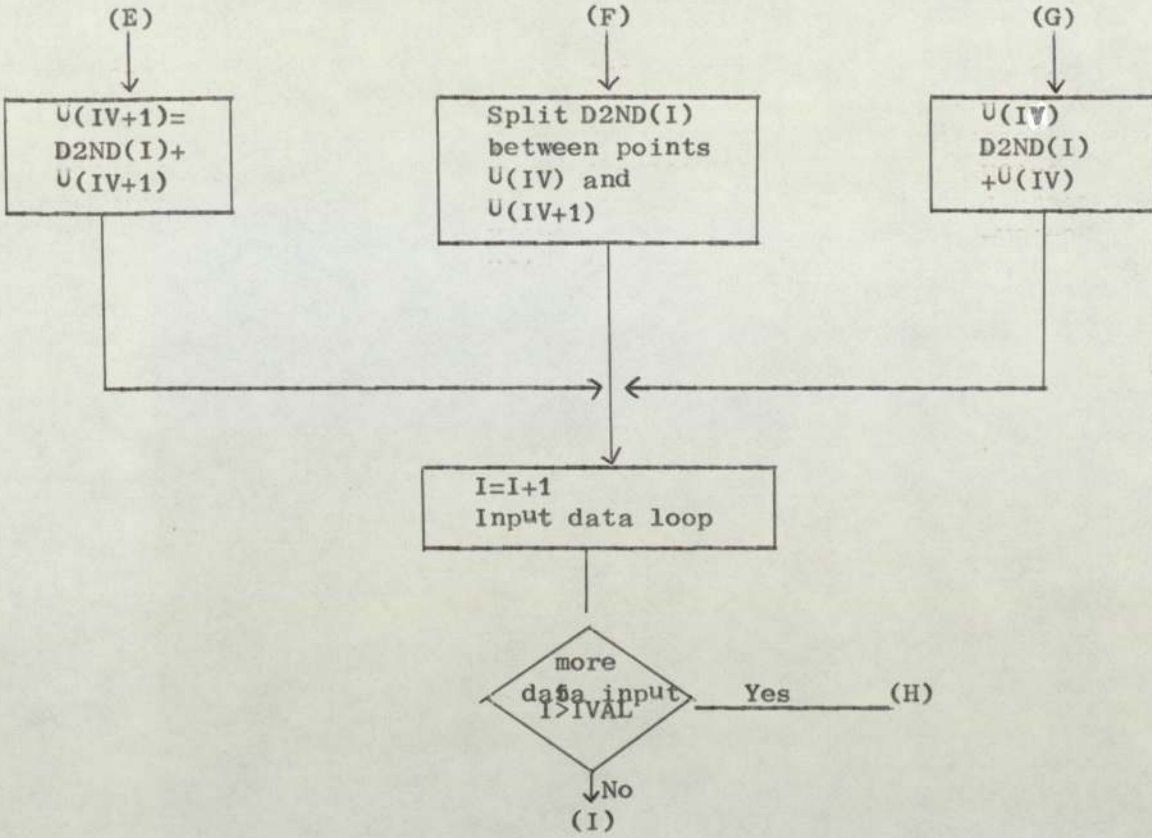


REPORT

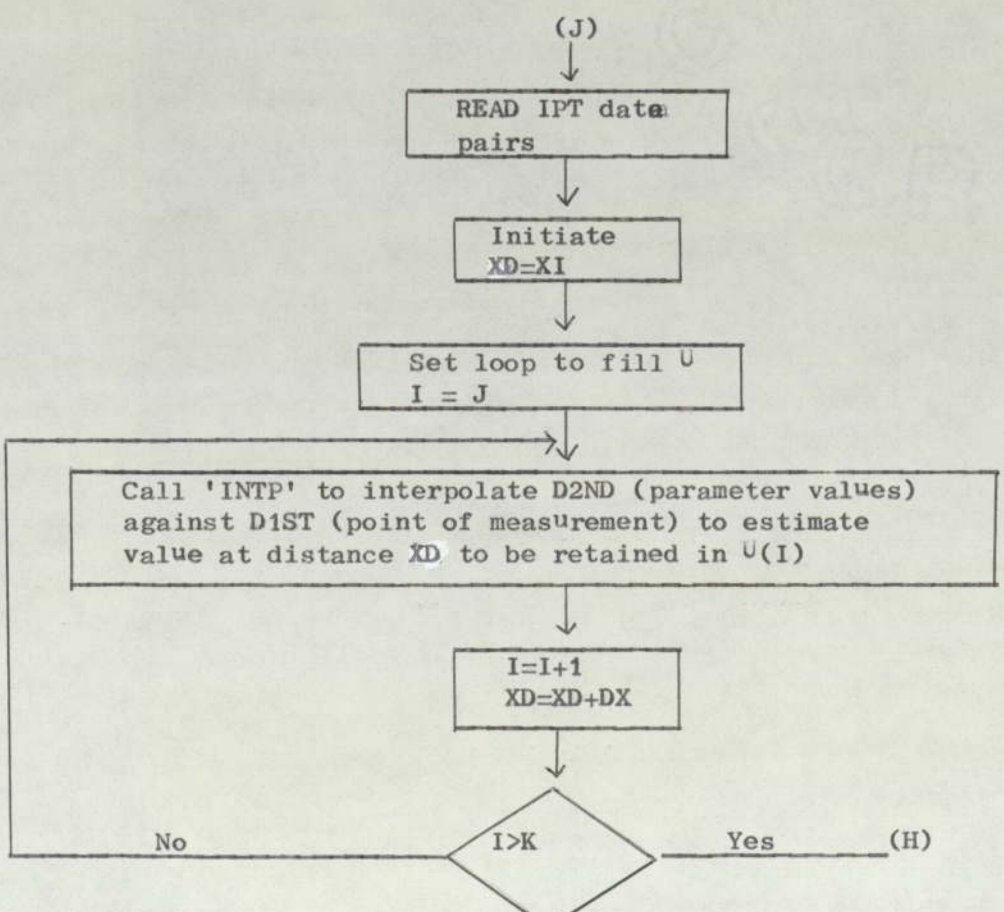


IPT = -1

LINEAR INTERPOLATION TO GRID POINTS FOR POINT DATA



ASSIGN VALUE TO GRID POINT



IPT >0 SMOOTH FUNCTION

Real Function DOXSAT

Function	To estimate the saturation level of D.O. in water as a function of temperature.
Calling Statement	X = DOXSAT (TEMP)
Error Messages	None
Common Areas	None
References	"Saturation levels of oxygen in water", W.P.R.L.

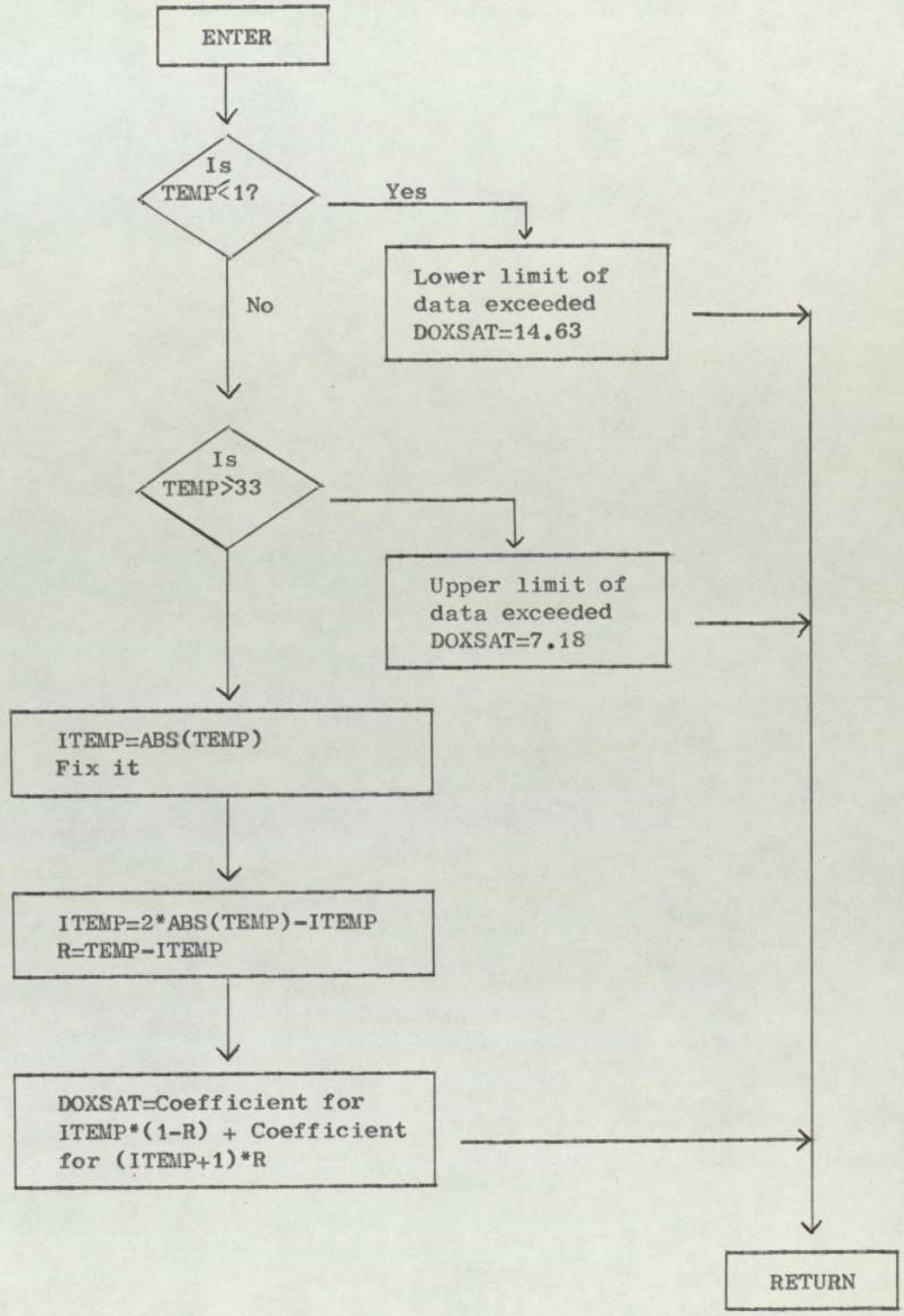
Description.

'TEMP' is the temperature at which the saturation level is required (in degrees Centigrade) Two tests check if the temperature is within the range 1 to 33°C. If outside the range, the values of the limits are used instead. This should not occur in estuary models.

For intermediate temperatures, the two nearest integer temperatures are located and the values interpolated for the enquiry value 'TEMP'.

A more comprehensive version of this routine is available, allowing for salinity variations (see model SSMOD).

Flow Logic


ENTER
Is
TEMP<1?
Yes
Lower limit of
data exceeded
DOXSAT=14.63
No
Is
TEMP>33
Upper limit of
data exceeded
DOXSAT=7.18
ITEMP=ABS(TEMP)
Fix it
ITEMP=2*ABS(TEMP)-ITEMP
R=TEMP-ITEMP
DOXSAT=Coefficient for
ITEMP*(1-R) + Coefficient
for (ITEMP+1)*R
RETURN

Subroutines FRSTD2 FRSTD6 SCNDD2 SCNDD6

Function:	All four routines are similar, and calculate:
	FRSTD2 First derivative of a function to 2nd order accuracy.
	FRSTD6 First derivative of a function to 6th order accuracy.
	SCNDD2 Second derivative of a function to 2nd order accuracy.
	SCNDD6 Second derivative of a function to 6th order accuracy.
System Functions	None.
User Functions	None
Calling Statement	CALL FRSTD2 (D, Y, N, M, DELX)
	" FRSTD6 "
	" SCNDD2 "
	" SCNDD6 "
Error Messages	None
Common Areas	None

Description:

Input List:

D the returned derivative defined from D(N) to D(M) inclusive

Y the function defined from Y(N) to Y(M) for which the derivative is required.

N,M Limits of definition of Y and D

DELX Step size of definition of Y.

The basic steps are the same for each routine:

1. Define constants associated with DELX and the order of accuracy.
2. Compute the derivatives using the estimates compatable with the order of accuracy desired.
3. Use lower order methods for boundary values.

FRSTD2

$$D_n = 0$$
$$D_m = (Y_m - Y_{m-1})/\Delta x$$
Loop limits : n + 1 to m - 1
$$D_i = (Y_{i+1} - Y_{i-1})/(2\Delta x)$$

SCNDD2

Loop limits : n+1 to m-1
$$D_i = (Y_{i+1} - 2\,Y_{i-1})/(\Delta x)^2$$
$$D_n = D_n+1$$
$$D_m = D_m-1$$

FRSTD6

$$D_n = 0$$
$$D_{n+1} = (-Y_{n+4} + 8\,[Y_{n=3} - Y_{n=1}] + Y_n)/(12\Delta x)$$
$$D_m = (Y_m - Y_{m-1})/(\Delta x)$$
$$D_{m-1} = (Y_m - Y_{m-2})/(2\Delta x)$$
$$D_{m-2} = (Y_m + 8\,[Y_{m-1} - Y_{m-3}] + Y_{m-4})/(12\Delta x)$$
Loop limits : n + 3 to m - 3
$$D_i = \frac{(Y_{i+3} - Y_{i-3} - 9[Y_{i=2} - Y_{i-2}] + 45[Y_{i+1} - Y_{i-1}])}{60\Delta x}$$

SCND6

$$D_n = (Y_{n+2} - 2Y_{n+1} + Y_n)/(\Delta x)^2$$
$$D_{n+1} = D_n$$
$$D_{n+2} = (-Y_{n+4} + 16*[Y_{n+3} + Y_{n+1}] - 30[Y_{n+2}-Y_n])/(12\Delta x^2)$$
$$D_m = (Y_{m-2} -2Y_{m-1} +Y_m)/(\Delta x)^2$$
$$D_{m-1} = (Y_m - 2Y_{m-1} + Y_{m-2})/(\Delta x)^2$$
$$D_{m-2} = (-Y_m +16*[Y_{m-1} + Y_{m-3}] - 30*\,[Y_{m-2} - Y_{m-4}])/(12\Delta x^2)$$
Loop limits : n+3 to m-3
$$D_i = (2*[Y_{i+3} + Y_{i-3}] - 27*\,[Y_{i+2}\;Y_{i-2}] + 270*\,[Y_{i+1} - Y_{i-1}] - 490*\,Y_i)/(\Delta x)^2$$

There are no path options in any of the routines, so flow diagrams are not required.

The accuracy of FRSTD2 is of order

$$-\frac{(\Delta x)^2}{6}\left\{ \cdot\ \frac{d^4y(\Upsilon)}{d\ x} \right\} \quad \text{where } xo-\Delta x < \Upsilon < xo + \Delta x$$

The accuracy of SCNDD6 is of order: $\left\{ \frac{(\Delta x)^2}{12} \right\}$

Subroutine MULT

Function: To multiply parts of an array by a fixed factor.

System Functions: None

User Functions: None

Calling Statement: CALL MULT (I, J, A, F)

Error Messages: None

Common Areas: None

Description:

Input List:

F is the required multiplier.

A is the array to be modified
from A(I) to A(J) inclusive.

The DO 99 K = I, J loop performs the indexing and the elements of A are multiplied by the factor F in turn. I, J are returned uncorrupted.

Subroutine STPF.

Function:	To handle point discharges for the model ST
System Functions:	READ/WRITE I/O
User Functions:	None
Error Messages:	None
Calling Statement:	CALL STPF(FF)
Common Areas:	/A/, /D/, /E/, /G/
Common Variables: Reset	None

Description.

The routine performs one function on the first call to it from DIFEQ (usually a dummy call to facilitate this); that of input of discharge data.

On all subsequent calls the function is to define the value of FF where it is non zero, i.e. at the points of discharge in relation to the grid laid over the system.

On the first call the variable NDIS is read in, the number of point sources to be considered in the system.

The loop 'DO 1LLL' is the discharge characteristic's input loop.

The parameters are outlined in the data input lists.
The loadings are converted to loads per area unit and LPOS is the nearest grid point to the point source and these points are saved in array NGP. Because of the possible variability of the source the array for loads DIS(I,J) is the loading per hour per area unit of discharge I during the J tenth of tidal cycle .

The times of closing of the outfall, TCL, and its reopening, TOP, are hours into the tidal cycle (0 being low water) and if TCL=TOP the discharge is continuous.

These times now are converted to 10th of a tiday cycle TIDED hours long, IC and IO.

The discharge rates for the deciles 1 to IC and IO to 10 can now be calculated. These will be greater than that calculated from HLOAD as there is a storage effect from IC to IO when no discharge occurs.

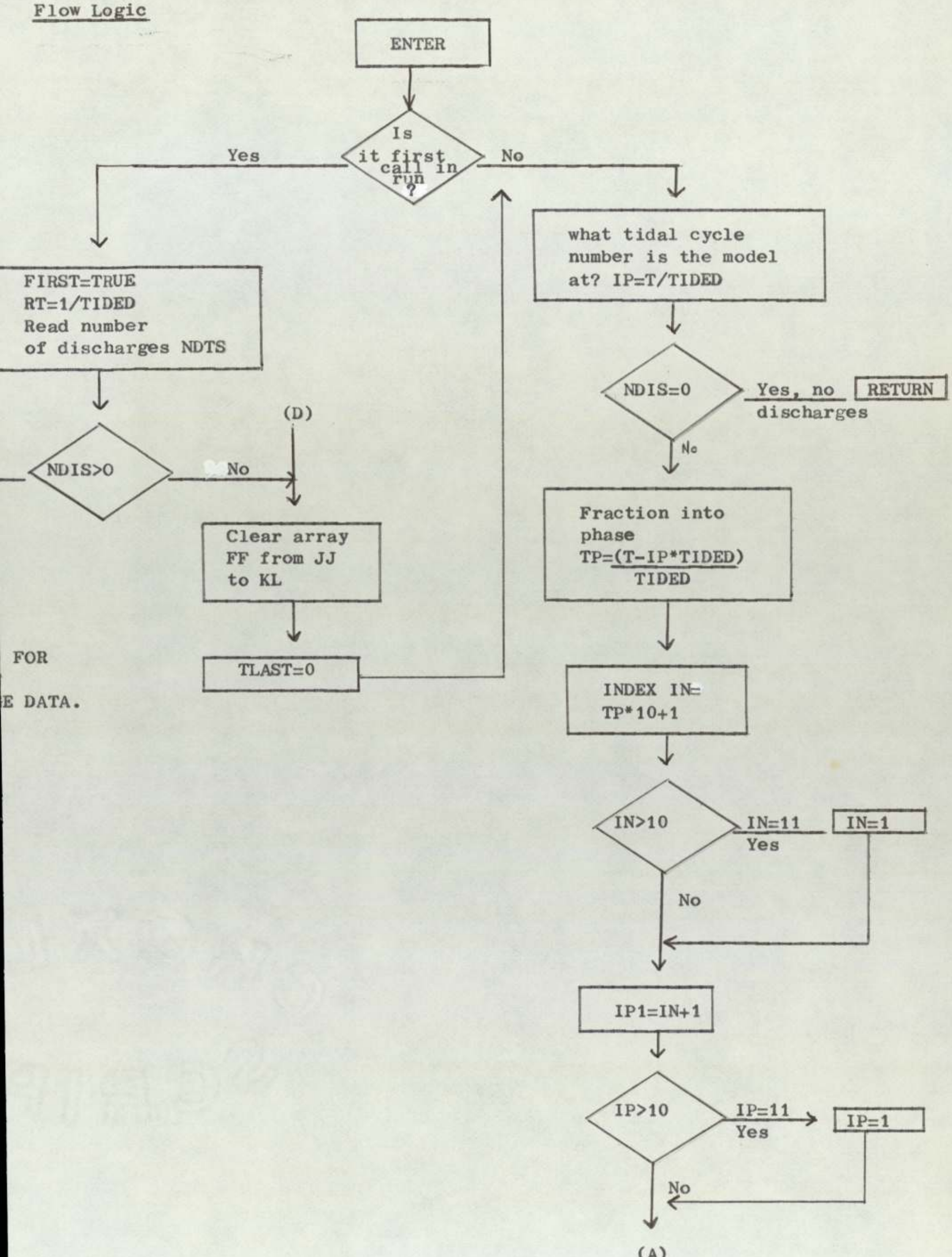
Details of these loadings are printed and the loop repeated until all NDIS discharges have been read in and stored.

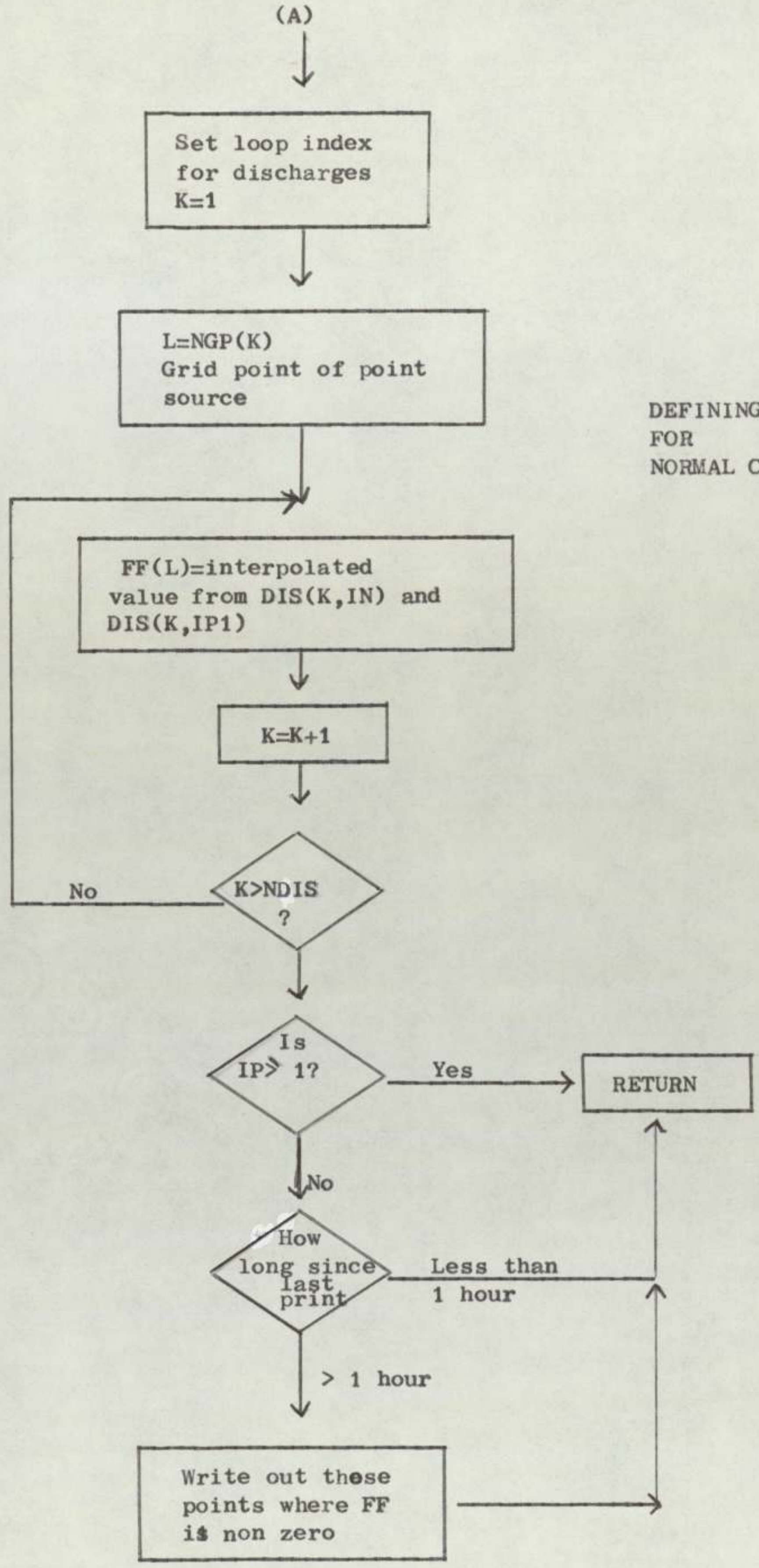
The array FF is cleared from JJ to KL on the first call only.

On all calls to the routine, the current time is converted to a fraction of a tidal cycle and a discharge level is computed for each non zero element of FF through interpolation. This is the 'DO 700' loop.

For the first cycle, i.e. 12½ hours, values of FF that are non zero are printed to the line file every simulated hour for inspection.

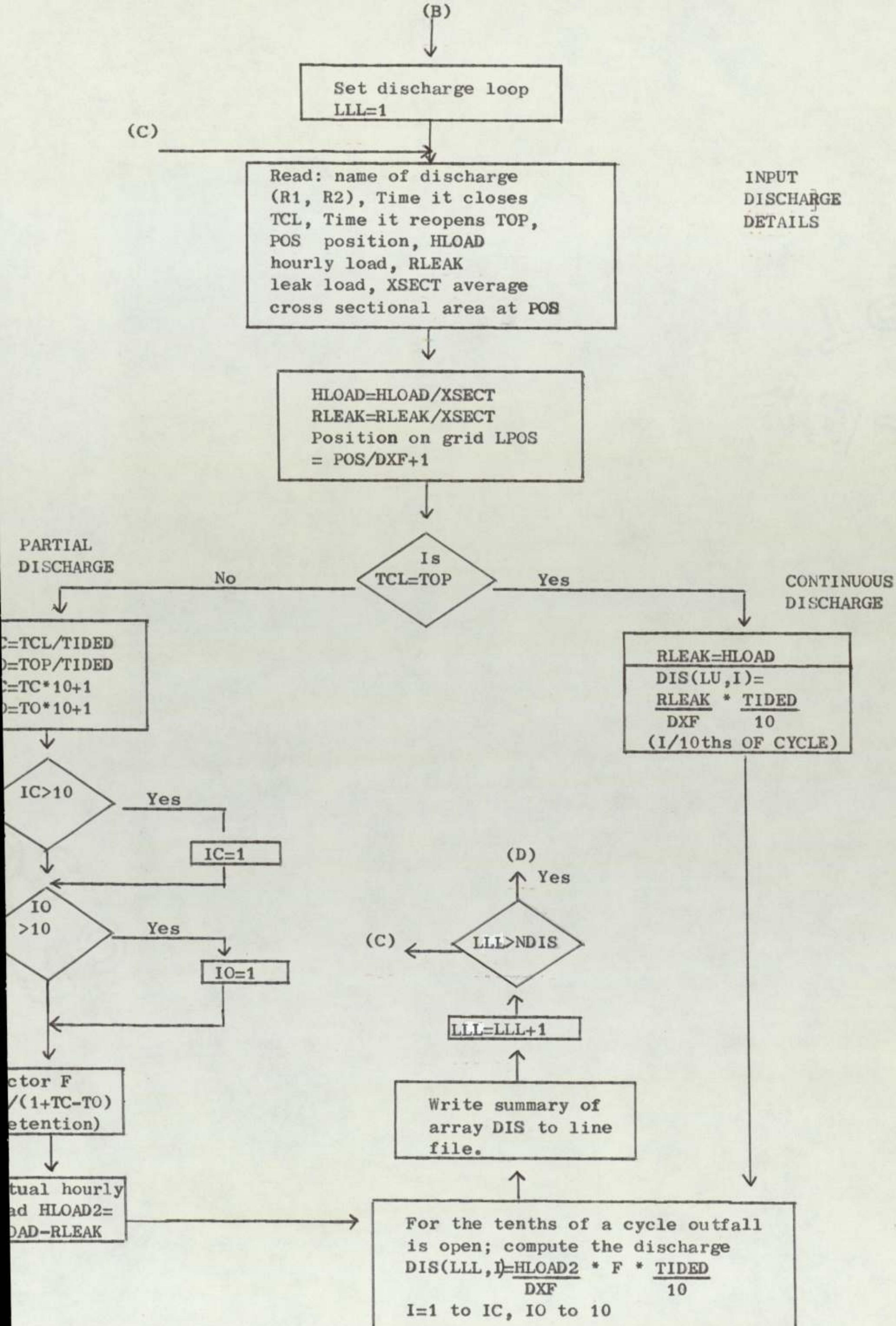
Flow Logic


ENTER
Is it first call in run ?
Yes
No
FIRST=TRUE
RT=1/TIDED
Read number of discharges NDTS
NDIS>0
No
(D)
Clear array FF from JJ to KL
TLAST=0
FOR
E DATA.
what tidal cycle number is the model at? IP=T/TIDED
NDIS=0
Yes, no discharges
RETURN
No
Fraction into phase TP=(T-IP*TIDED)/TIDED
INDEX IN=TP*10+1
IN>10
IN=11
Yes
IN=1
No
IP1=IN+1
IP>10
IP=11
Yes
IP=1
No
(A)



DEFINING FF
FOR
NORMAL CALL

PRINT FF
FOR FIRST
TIDAL CYCLE
AT HOURLY
INTERVALS


(B)
Set discharge loop LLL=1
(C)
Read: name of discharge (R1, R2), Time it closes TCL, Time it reopens TOP, POS position, HLOAD hourly load, RLEAK leak load, XSECT average cross sectional area at POS
INPUT DISCHARGE DETAILS
HLOAD=HLOAD/XSECT
RLEAK=RLEAK/XSECT
Position on grid LPOS = POS/DXF+1
Is TCL=TOP
No
Yes
PARTIAL DISCHARGE
CONTINUOUS DISCHARGE
C=TCL/TIDED
)=TOP/TIDED
C=TC*10+1
)=TO*10+1
RLEAK=HLOAD
DIS(LU,I)= RLEAK * TIDED / DXF 10
(I/10ths OF CYCLE)
IC>10
Yes
IC=1
IO >10
Yes
IO=1
ctor F
/(1+TC-TO)
etention)
tual hourly
ad HLOAD2=
OAD-RLEAK
(D)
Yes
(C)
LLL>NDIS
LLL=LLL+1
Write summary of array DIS to line file.
For the tenths of a cycle outfall is open; compute the discharge
DIS(LLL,I)=HLOAD2 * F * TIDED / DXF 10
I=1 to IC, IO to 10

Subroutine INTP2.

Function: A simple linear interpolation routine.

System Functions: None

User Functions: None

Error Messages: None

Calling Statement: CALL INTP2(L,NP, X, FX, XX, FXX, ACC)

Common Areas: None

Description:

Input List:

Comparable to that of subroutine INTP, but 'L' and 'ACC' are redundant.

The 'DO 1' loop tests to see that XX, the independent variable for which an estimate of the dependant variables value is to be made, is within the limits of X(NP).

If XX > X(NP) then FXX is set to FX(NP).

Also, if XX < X(1) then FXX is set to FX(1)

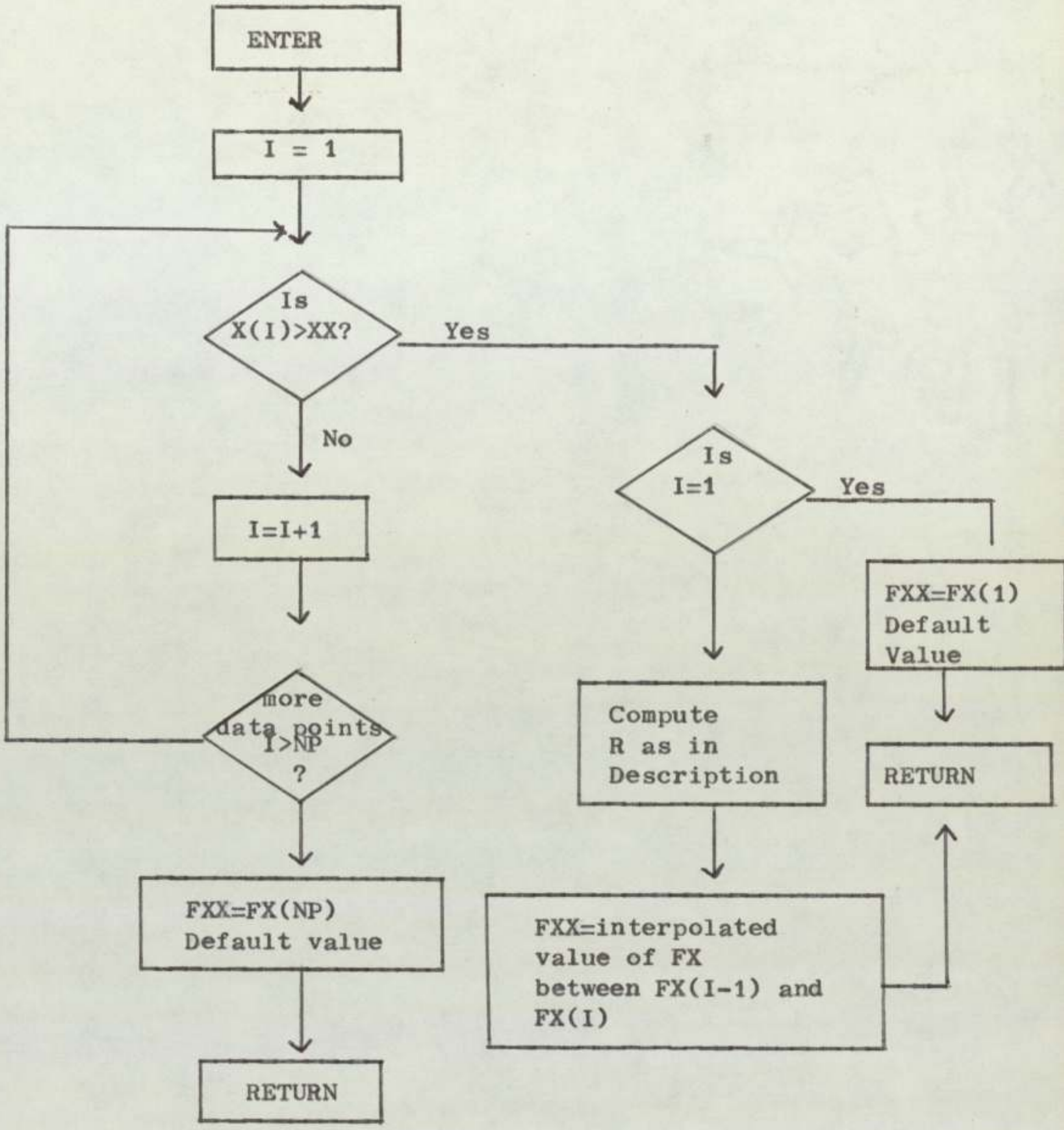
For usual cases X(1) ≤ XX ≤ X(NP)

Calculate R = (XX - X(I-1)/(X(I)-X(I-1))

and then FXX = R*FX(I-1) + (1-R) * FX(I)

Where I is the element index of the first element of X that is greater than XX.

Flow Logic.


ENTER
I = 1
Is
X(I)>XX?
Yes
No
I=I+1
more
data points
I>NP
?
FXX=FX(NP)
Default value
RETURN
Is
I=1
Yes
FXX=FX(1)
Default
Value
Compute
R as in
Description
RETURN
FXX=interpolated
value of FX
between FX(I-1) and
FX(I)

Subroutine INTP

Function: To perform non-linear interpolation.

System Functions: ABS

Calling Statement: CALL INTP(LN,NP,X,FX,XX,FXX,ACC)

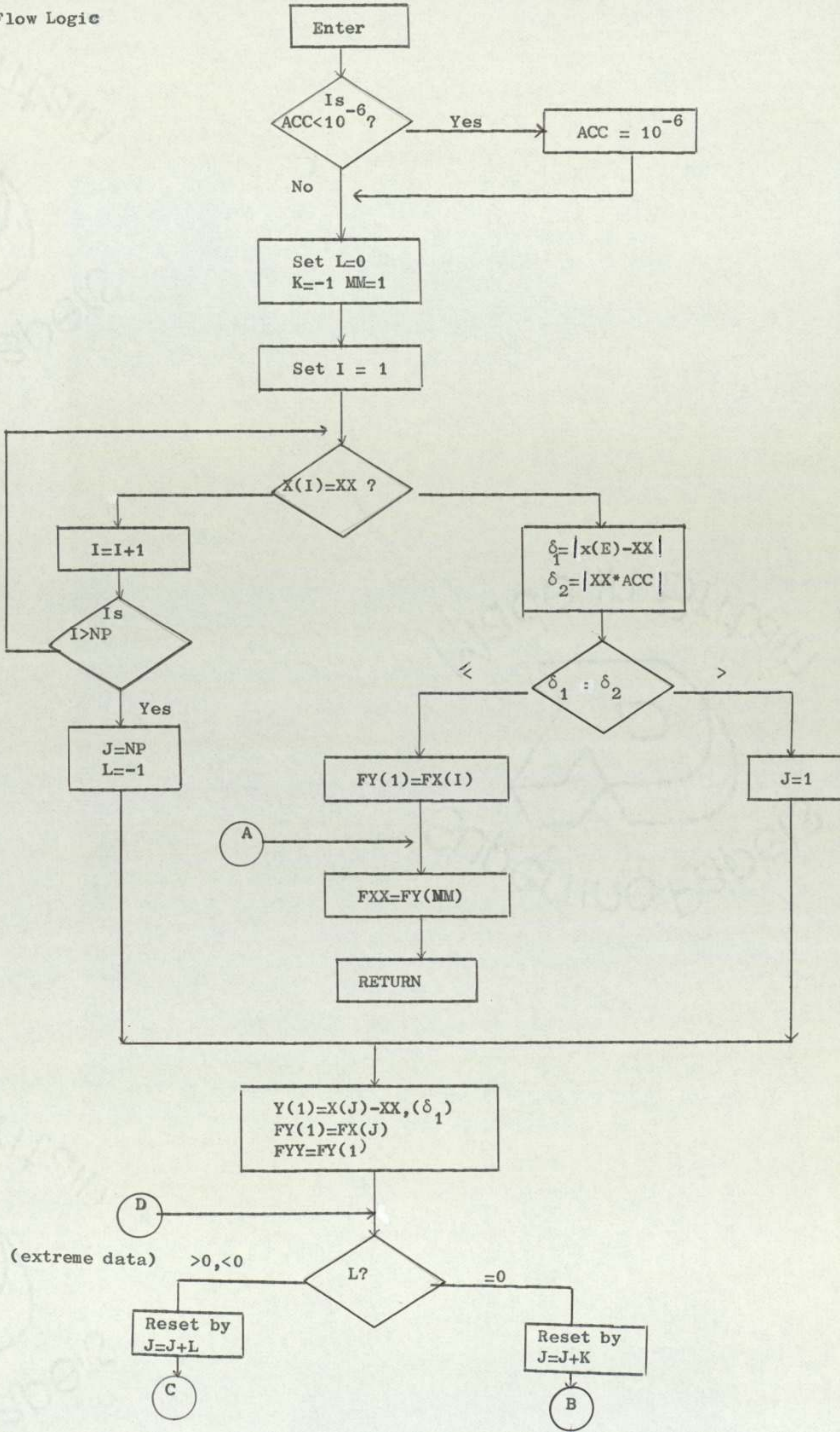
Description:

Calling List.

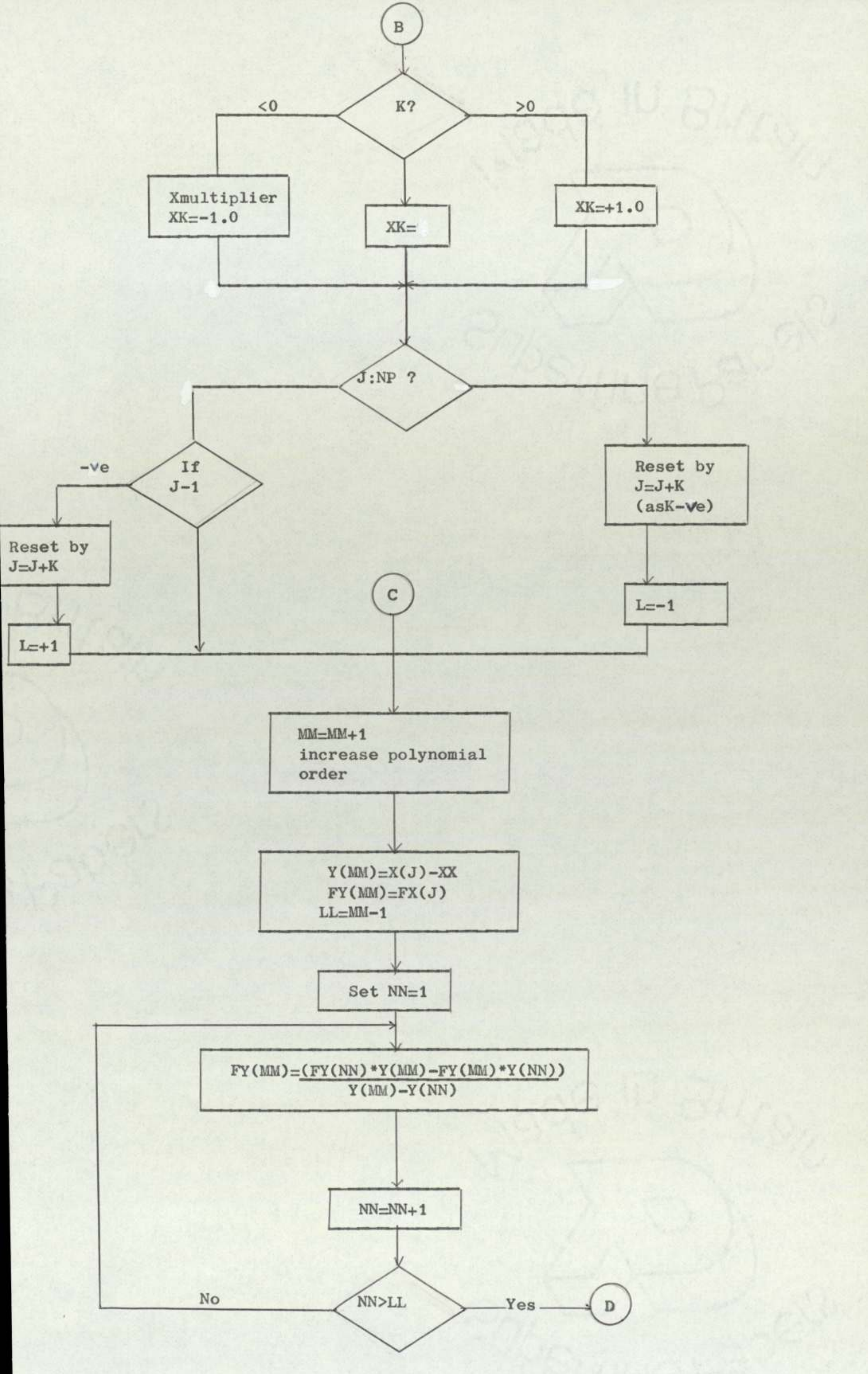
LN	Maximum Polynomial order of required interpolation $0<LN\leq 7$
NP	Number of well ordered pairs of points in the arrays X,FX
X	Values of the independent variable arranged in strictly ascending order.
FX	Values of the dependent variable corresponding to values in X where $FX_i = F(X_i)$
XX	Value of the independent variable for which a value of FX is sought.
FXX	The value of the dependent variable corresponding to XX i.e. FXX = F(XX)
ACC	Desired relative accuracy of FXX. $> 10^{-6}$ to safeguard against computational errors.

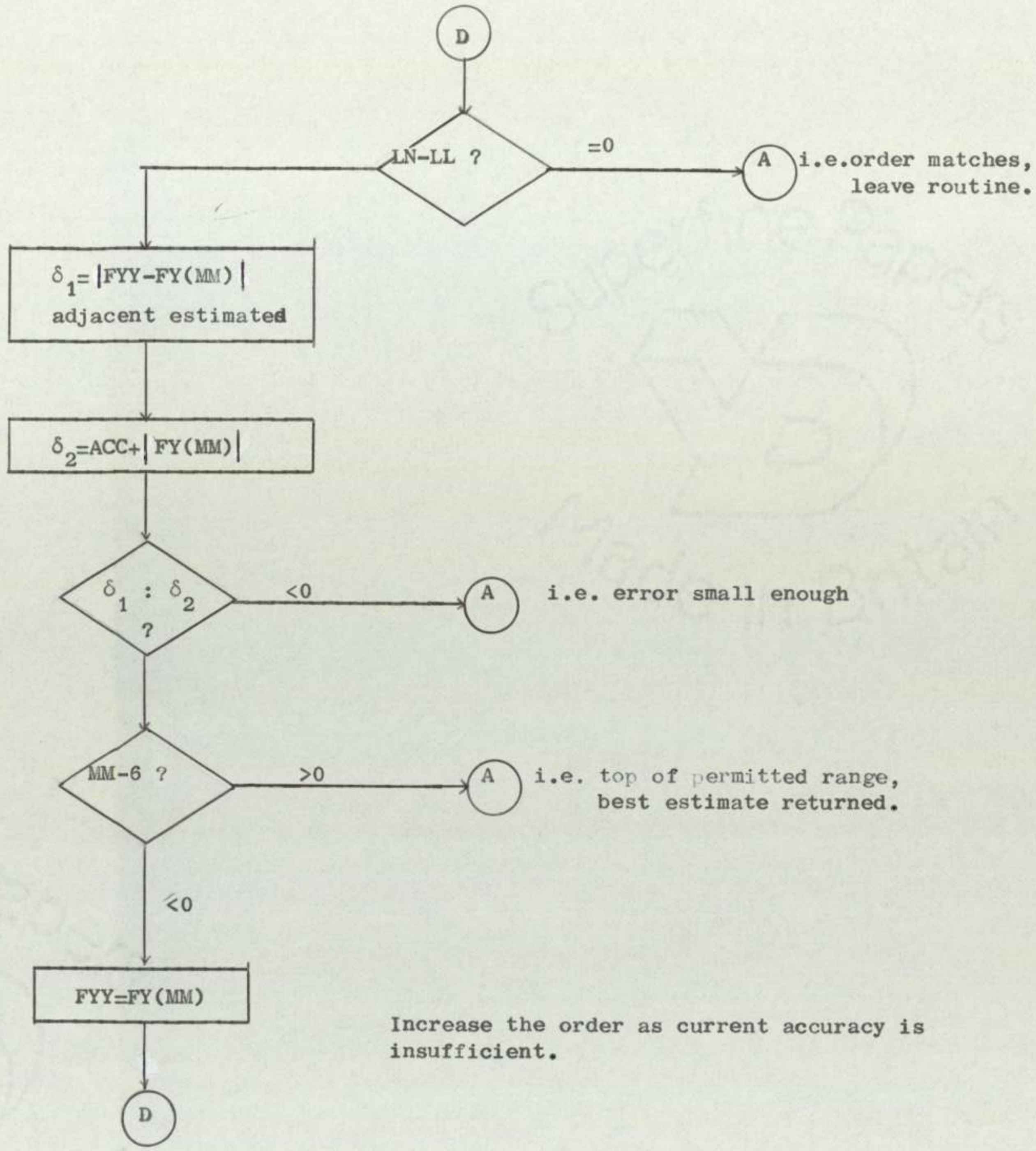
A Lagrangian method interpolation of increasingly higher order is performed until the accuracy is obtained or until the highest order is computed.

Flow Logic



Enter
Is ACC<10^-6 ?
Yes
ACC = 10^-6
No
Set L=0
K=-1 MM=1
Set I = 1
X(I)=XX ?
I=I+1
Is I>NP
Yes
J=NP
L=-1
δ1=|x(E)-XX|
δ2=|XX*ACC|
δ1 : δ2
≤
>
FY(1)=FX(I)
J=1
A
FXX=FY(MM)
RETURN
Y(1)=X(J)-XX,(δ1)
FY(1)=FX(J)
FYY=FY(1)
D
(extreme data)
>0,<0
L?
=0
Reset by
J=J+L
Reset by
J=J+K
C
B


B
K?
<0
>0
Xmultiplier
XK=-1.0
XK=
XK=+1.0
J:NP ?
-ve
If
J-1
Reset by
J=J+K
(asK-ve)
Reset by
J=J+K
C
L=-1
L=+1
MM=MM+1
increase polynomial
order
Y(MM)=X(J)-XX
FY(MM)=FX(J)
LL=MM-1
Set NN=1
FY(MM)=(FY(NN)*Y(MM)-FY(MM)*Y(NN))
Y(MM)-Y(NN)
NN=NN+1
No
NN>LL
Yes
D


D
LN-LL ?
=0
A
i.e.order matches, leave routine.
δ1=|FYY-FY(MM)|
adjacent estimated
δ2=ACC+|FY(MM)|
δ1 : δ2 ?
<0
A
i.e. error small enough
MM-6 ?
>0
A
i.e. top of permitted range, best estimate returned.
≤0
FYY=FY(MM)
Increase the order as current accuracy is insufficient.
D

Subroutine FUNCT

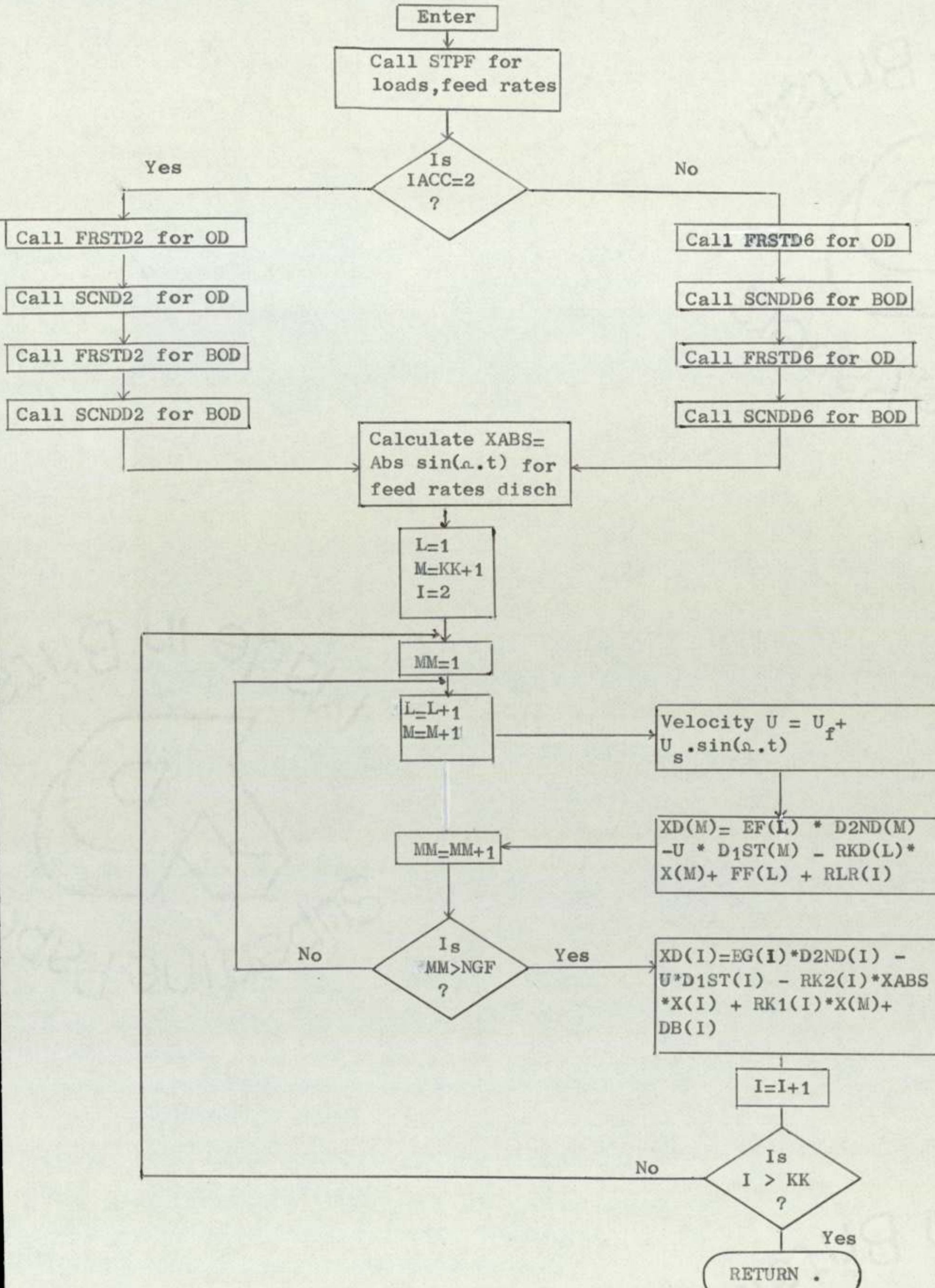
Function:	To evaluate the coupled differential equations for OD/BOD at one point in time.
Calling Statement:	CALL FUNCT(X,XD,IACC)
System Functions:	SIN,COS,ABS.
User Functions:	FRSTD6, SCNDD6, FRSTD2, SCNDD2, STPF
Common Areas:	/A/, /B/, /C/, /D/, /E/, /F/

Calling List.

X	Previous function evaluation.
XD	Returned function evaluation.
IACC	The accuracy code from DIFEQ

Description.

A call to STPF generates current loadings to the system. The derivature routines are called to evaluate the derivatives of the function dependent on whether 2nd or 6th order is desired. Some constants connected with the tidal phase are computed.
The function is evaluated for OD and BOD through a double loop with the BOD on a finer grid than the OD.


Enter
Call STPF for loads,feed rates
Is IACC=2 ?
Yes
No
Call FRSTD2 for OD
Call SCND2 for OD
Call FRSTD2 for BOD
Call SCNDD2 for BOD
Call FRSTD6 for OD
Call SCNDD6 for BOD
Call FRSTD6 for OD
Call SCNDD6 for BOD
Calculate XABS= Abs sin(Ω.t) for feed rates disch
L=1
M=KK+1
I=2
MM=1
L=L+1
M=M+1
Velocity U = U_f + U_s.sin(Ω.t)
XD(M)= EF(L) * D2ND(M) -U * D1ST(M) - RKD(L)* X(M)+ FF(L) + RLR(I)
MM=MM+1
Is MM>NGF ?
No
Yes
XD(I)=EG(I)*D2ND(I) - U*D1ST(I) - RK2(I)*XABS *X(I) + RK1(I)*X(M)+ DB(I)
I=I+1
Is I > KK ?
No
Yes
RETURN .

Routine POUT

Function: To provide an output facility for models ST and ST2

System Functions: OUTPUT to LP/CP, MOD, PBPLT1

User Functions: DOXAT, COPYDT, RKMI, ADDATE, INTP

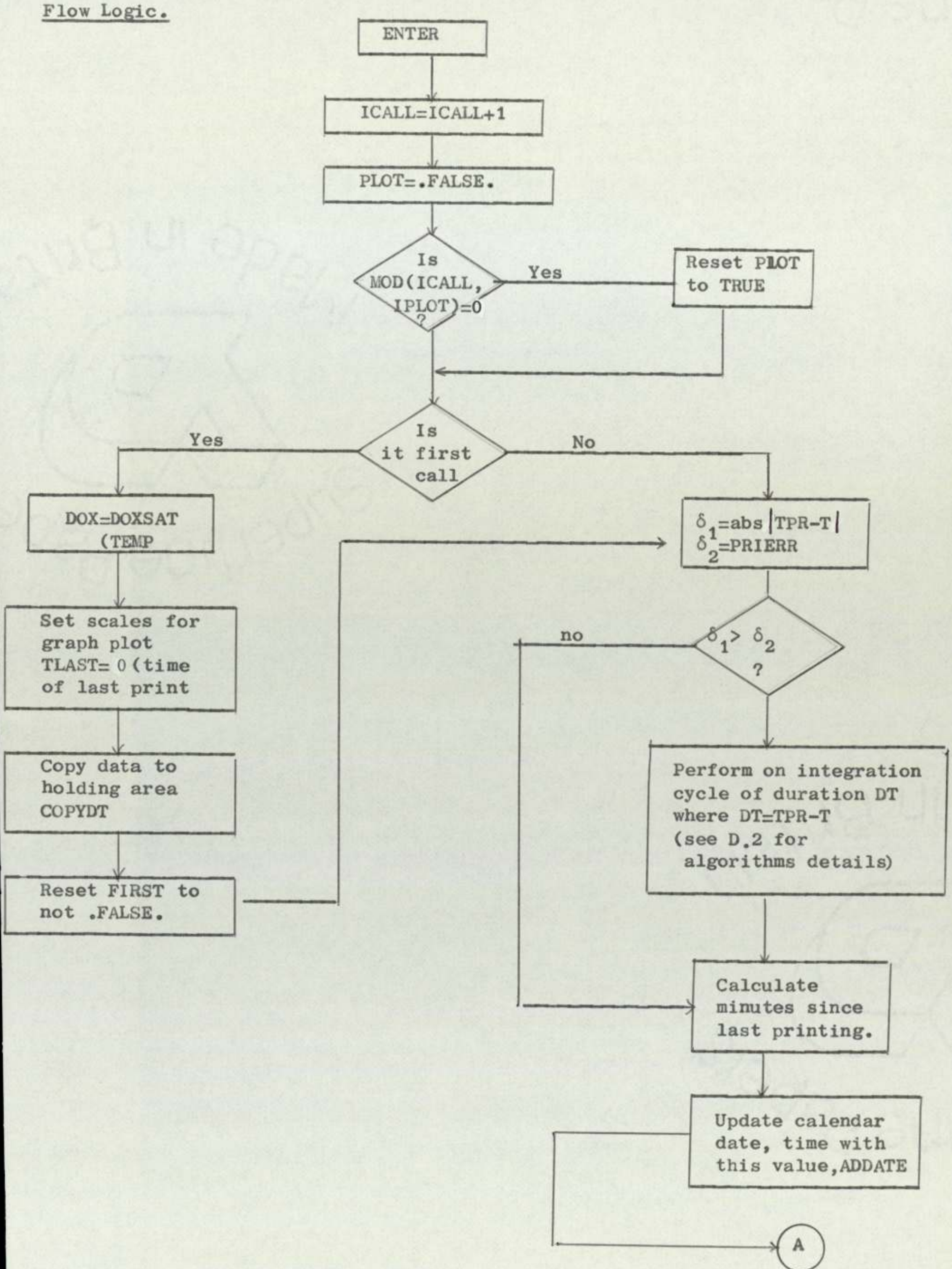
Common Areas: /A/, /B/, /C/, /D/, /E/, /G/.

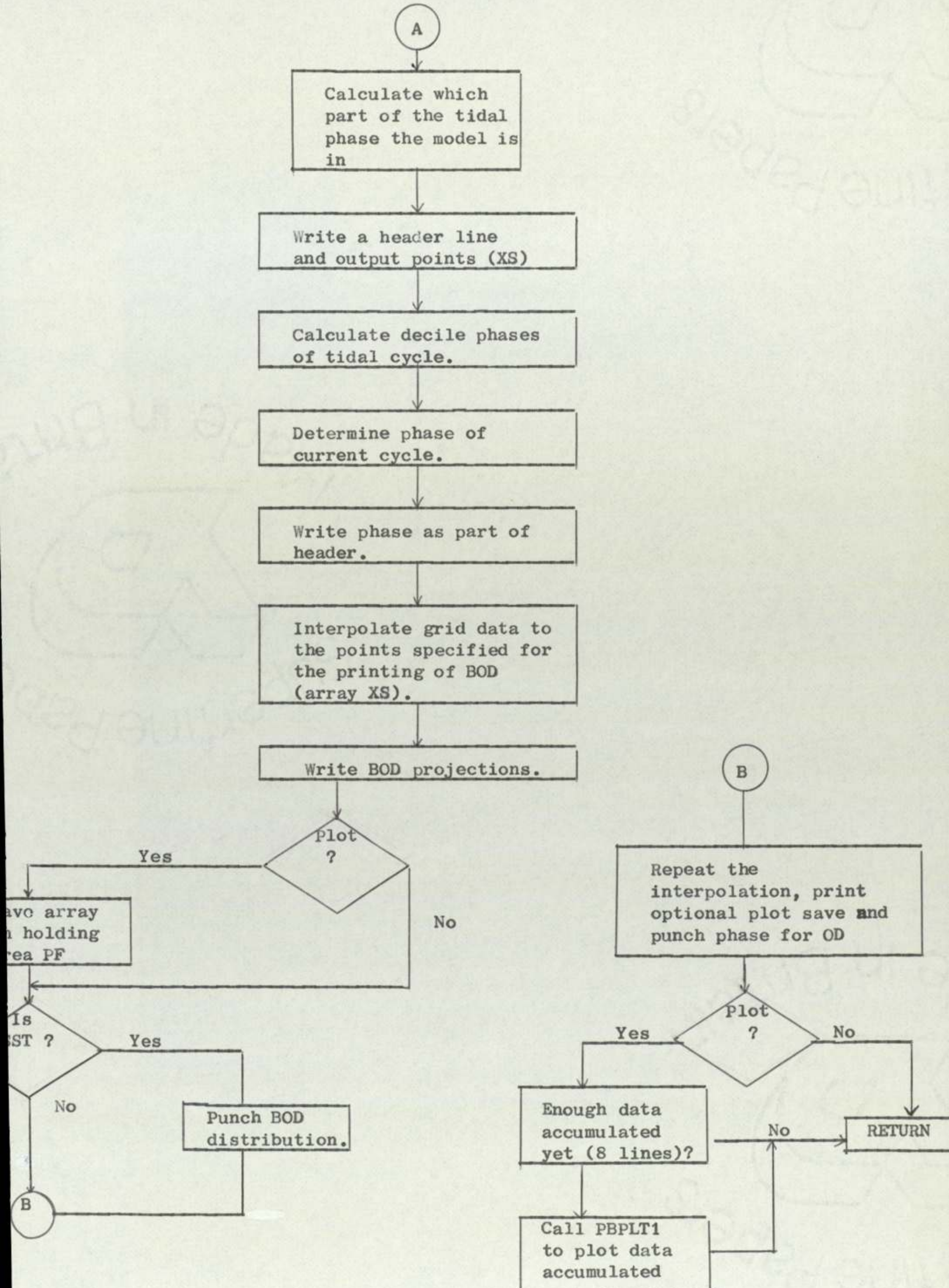
Calling Statement: CALL POUT(IPLOT,IACC)

Description.

IACC is 2 or 6 depending on which level of accuracy is required. IPLOT is a switch for graph plotting of output. Some initial constants are set. If the current time is not within the error bounds of the print-time a further integration step is required if the time is near a print. If the print is some time away, the routine returns control to the executive. The integration cycle is performed as in the executive routine. The elapsed time is computed, and added to the calendar date and time. The tidal phase currently being simulated is computed. The header information is printed. Using the interpolation routine the data is generated for BOD for the points specified in array XS. If plotting, the summary is retained in array PF. The procedure is repeated for the OD component. If the current time is in excess of the steady state time (SST) the two distributions are punched (to channel 8 on ICL 4/70 MJI500).

Flow Logic.


ENTER
ICALL=ICALL+1
PLOT=.FALSE.
Is MOD(ICALL, IPLOT)=0 ?
Yes
Reset PLOT to TRUE
Is it first call
Yes
No
DOX=DOXSAT (TEMP
Set scales for graph plot TLAST= 0 (time of last print
Copy data to holding area COPYDT
Reset FIRST to not .FALSE.
δ1=abs |TPR-T|
δ2=PRIERR
δ1 > δ2 ?
no
Perform on integration cycle of duration DT where DT=TPR-T (see D.2 for algorithms details)
Calculate minutes since last printing.
Update calendar date, time with this value, ADDATE
A



A
Calculate which part of the tidal phase the model is in
Write a header line and output points (XS)
Calculate decile phases of tidal cycle.
Determine phase of current cycle.
Write phase as part of header.
Interpolate grid data to the points specified for the printing of BOD (array XS).
Write BOD projections.
Plot ?
Yes
No
ave array
holding
rea PF
Is
ST ?
Yes
No
Punch BOD distribution.
B
B
Repeat the interpolation, print optional plot save and punch phase for OD
Plot ?
Yes
No
Enough data accumulated yet (8 lines)?
No
RETURN
Call PBPLT1 to plot data accumulated

D13. **Routine RKMI.**

Function:	To perform a onestep integration for the coupled OD/BOD equation.
Calling Statement:	CALL RKMI(X,Y,E,A, J1,IACC)
System Functions:	ABS
User Functions:	FUNCT
Common Areas	/A/,/C/,/D/,/E/

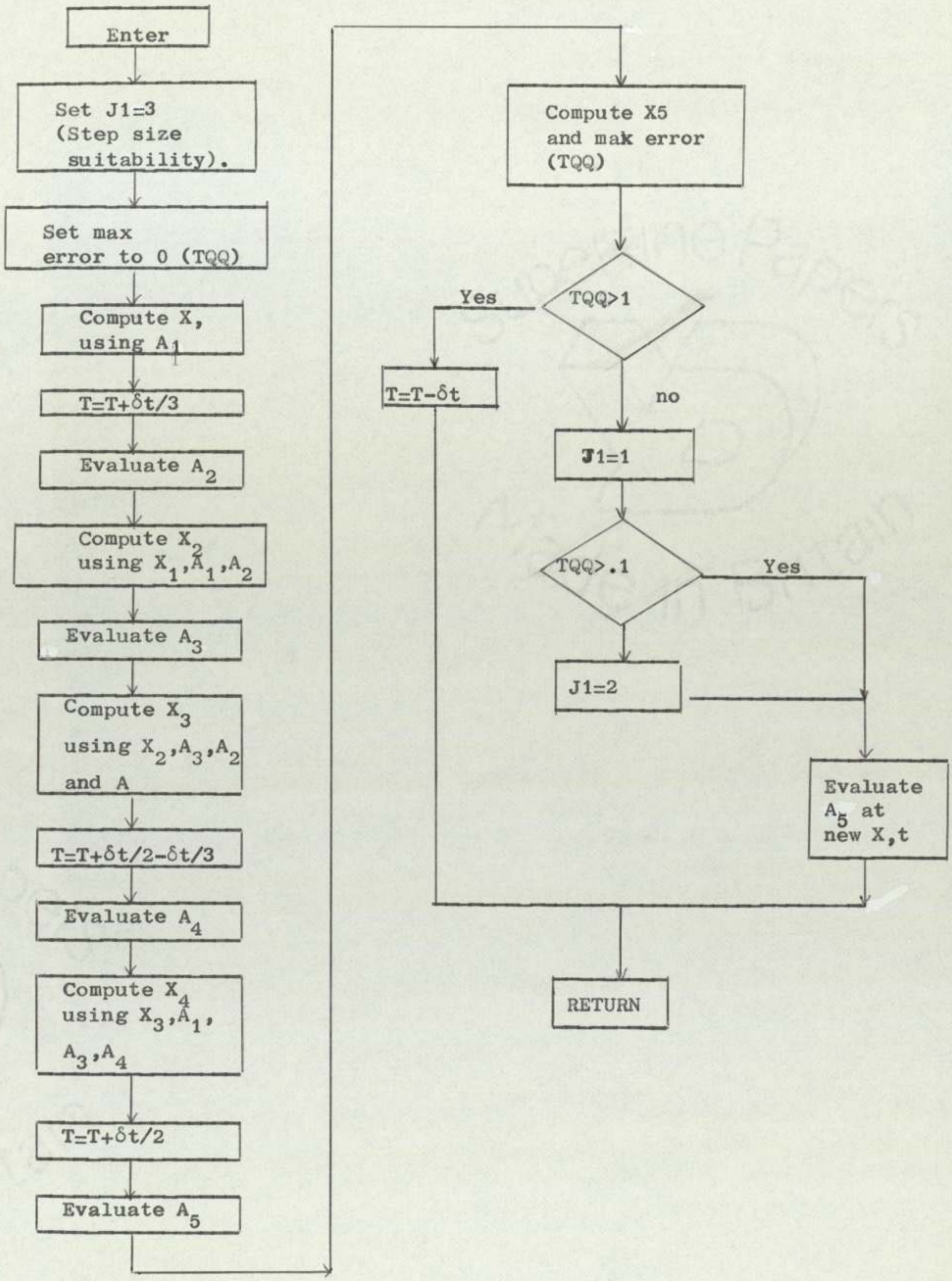
Description.

Calling List

X	-	Function one time step advanced.
Y	-	Function on previous time step.
E	-	Last function evaluation.
A	-	First function evaluation.
J1	-	Time step suitability for the next or current step.
IACC	-	2 or 6 depending on order of accuracy desired.

IACC is passed onto routine FUNCT. For precise details of the algorithm, see D14.

Flow Logic.



Enter
Set J1=3 (Step size suitability).
Set max error to 0 (TQQ)
Compute X, using A1
T=T+δt/3
Evaluate A2
Compute X2 using X1,A1,A2
Evaluate A3
Compute X3 using X2,A3,A2 and A
T=T+δt/2-δt/3
Evaluate A4
Compute X4 using X3,A1, A3,A4
T=T+δt/2
Evaluate A5
Compute X5 and mak error (TQQ)
TQQ>1
Yes
T=T-δt
no
J1=1
TQQ>.1
Yes
J1=2
Evaluate A5 at new X,t
RETURN

D14 **The Runge-Kutta-Mersion Method of Integration.**

This is a variant of the general 4th order Runge-Kutta method applied to solution of simultaneous first order equations with some estimate of errors:

$$\frac{dY_1}{dx} = f_1 (x, y_1, y_2)$$

$$\frac{dY_2}{dx} = f_2 (x, y_1, y_2)$$

In the context of the current model

$$\frac{dB}{db} = f_1(t,B,D) \text{ and } \frac{dD}{db} = f_2(t,B,D)$$

where
$$f_1 = E_B \cdot \frac{\partial^2 B}{\partial x^2} - \frac{U\partial B}{\partial x} - KB + L + F$$

$$f_2 = E_D \cdot \frac{\partial^2 D}{\partial x^2} - \frac{U.\partial D}{\partial x} - K_R D + D_B - P_S$$

The method briefly outlined in Chapter 5 is to evaluate:

$$k_{01} = \frac{\partial t}{3} f_1(t,B,D) \text{ and } k_{02} = \frac{\partial t}{3} f_2 (t,B,D) \qquad (A)$$

t is new set to $t + \partial t/3$

$$k_{11} = \frac{\partial t}{3} f_1(t, B+k_{01}, D+k_{02}) \text{ and } k_{12} = \frac{\partial t}{3} f_2(t,B+k_{01}, D+k_{02}) \qquad (B)$$

then

$$k_{21} = \frac{\partial t}{3} f_1(t,B + \frac{k_{01}}{2} + \frac{k_{11}}{2}) \; D + \frac{k_{02}}{2} + \frac{k_{12}}{2}) \quad \text{and} \qquad (C)$$

$$k_{22} = \frac{\partial t}{3} f_2 (t,B + \frac{k_{01}}{2} + \frac{k_{11}}{2}, D + \frac{k_{02}}{2} + \frac{k_{12}}{2})$$

then t is reset to $t + \partial t/2 - \partial t/3$

$$k_{31} = \frac{\partial t}{3} f_1 (t,B + + \frac{3k_{01}}{8} + \frac{9k_{21}}{8}, D + \frac{3k_{02}}{8} + \frac{9k_{22}}{8} \quad \text{and} \qquad (D)$$

$$k_{32} = \frac{\partial t}{3} f_2 (t,B + \frac{3k_{01}}{8} + \frac{9k_{21}}{8}, D + \frac{3k_{02}}{8} + \frac{9\,k_{22}}{8})$$

For the fifth evaluation, t advances the whole time step.

i.e. $t = tt+\partial t$

$$k_{41} = \frac{\partial t}{3} f_1(t_1 B+\frac{3k_{01}}{2} - \frac{9k_{21}}{2} + 6k_{31}, D+\frac{3k_{02}}{2} - \frac{9k_{22}}{2} + 6k_{32}) \quad \text{and}$$

$$k_{42} = \frac{\partial t}{3} f_2 (t_9 B+\frac{3k_{01}}{2} - \frac{9k_{01}}{2} + 6k_{31}, D + \frac{3k_{02}}{2} - \frac{9k_{22}}{2} + 6k_{32}) \qquad (E)$$

By laying f_2 after f_1 in the t,B,D space and accounting for the discontinuity at the B,D interface allows the software to be simplified.
Three work arrays are required: A_2, A_3, A_4 for internal use. Two further arrays are passed across: A_1 and A_5. Y is the current value at t and X will be the value at $t + \partial t$ when the routine returns control. A_5 is the final evaluation and returned as A_1 on the next call to the step integration

(A) to (E) can be written ($\partial = \partial t/3$)

$$k_0 = \partial f(t,Y)$$
$$k_1 = \partial f(t + \partial, Y + k_0)$$
$$k_2 = \partial f(t + \partial, Y + k_0/2 + k_1/2)$$
$$k_3 = \partial f(t + \partial t/2, Y + 9k_2/8 + 3k_0/8)$$
$$k_4 = \partial f(t + \partial t, Y + 6k_3 - 9k_2/2 + 3k_0/2)$$

and the prediction of $Y = f(t_1 Y)$ (i.e. $f_1(t,B,D)$ and $f_2(t,B,D)$) at $Y^1 = f(t + \partial t, Y)$(i.e. $f_1(t + \partial t, B,D)$ and $f_2(t + \partial t, B, D)$) is

$$Y^1 = Y + \tfrac{1}{2}(k_0 + 4k_3 + k_4)$$

In the detailed algorithm, this is achieved by the following sequences of evaluations and definitions.

$A_1 = f(t,Y)$ ($_1$ defined on entry)

therefore $k_0 = \partial . A_1$

$X_1 = Y + \partial . A_1 = Y + k_0$

Then $A_2 = f(t + \partial, X,) = f(t +], Y + k_0)$ therefore $k_1 = \partial . A_2$

$X_2 = X_1 + \partial/2 (A_2 - A_1)$

$= Y + (k_0 + k_1)/2$

$A_3 = f(t + d, X_2)$, i.e. $k_2 = \partial . A_3$

$X_3 = X_2 + \partial(9A_3/8 - A_2/2 - A_1/8)$

$= Y + 3k_0/8 + 9k_2/8$

$A_4 = f(t + \partial t/2, X_3)$, i.e. $k_3 = \partial . A_4$

$X_4 = X_3 + \partial(9A_1/8 - 45A_3/8 + 6A_4)$

$= Y + 6k_3 - 9k_2/2 + 3k_0/2$

$A_5 = f(t + \partial t, X_4)$, i.e. $k_4 = \partial . A\ _5$

$X_5 = X_4 - \partial(A_1 - 9A_3/2 + 4A_4 - A_5/2)$

$= Y + k_0/2 + 2k_3 + k_4/2$

$= Y^1$

X_5 is the required output. The evaluation A_5 is used as A_1 in the subsequent call of RKMI.

If R_E is the permitted relative error, and A_E is the maximum absolute error, the estimated error of the method is $(2k_0 - 9k_2 + 8k_3 - k_4)/10$

now $E = 2(X_4 - X_5)$

$= 2(k_0 - 9k_2/2 + 4k_3 - k_4/2)$

Therefore the error to be considered for magnitude taking into account relative error, absolute error and magnitude of the function at the current point in space/time is

$$E_Q = \left| \frac{\partial}{5} \cdot \frac{E}{(R_E \cdot |X_5| + A_E)} \right|$$

And the largest of these is retained for the estimation of the suitability of the time step.

If	$\text{Max}[E_Q] > 1$	Halve time step
If	$\text{Max}[E_Q] < .1$	Increase time step by 1/3rd.
If	$.1 < \text{Max}[E_Q] < 1$	Retain current time step

D15 Subroutine RUNGK2

Function: To provide a second order Runge-Kutta method of solutiion as an alternative to the fourth order Runge-Kutta Merson Method.

User Functions: FUNCT

System Functions: None.

Calling Statement: CALL RUNGK2(X,Y)

Common Areas: /D/, /E/.

Description.

Two evaluations at t and t+ $3\partial t/4$ are made by yield $k_0 + k_1$ as previously:

$$A = f(t,Y)$$

$$X_1 = Y + 3\partial t/4.f(t,Y)$$

$$B = f(t + 3\partial t/4, Y + 3\partial t/4f(t,Y))$$

$$X_2 = X_1 + \partial t/3.A + 2\partial t/3.B$$

therefore $Y^1 = Y + \partial t/3\ (A + 2B)$

No error estimate is given. This should be used only in slowly varying systems as inaccuracies are functions of higher order derivatives.
A reasonable comparison is the solution of this method to that using routine RKMI to see if it is cost effective.

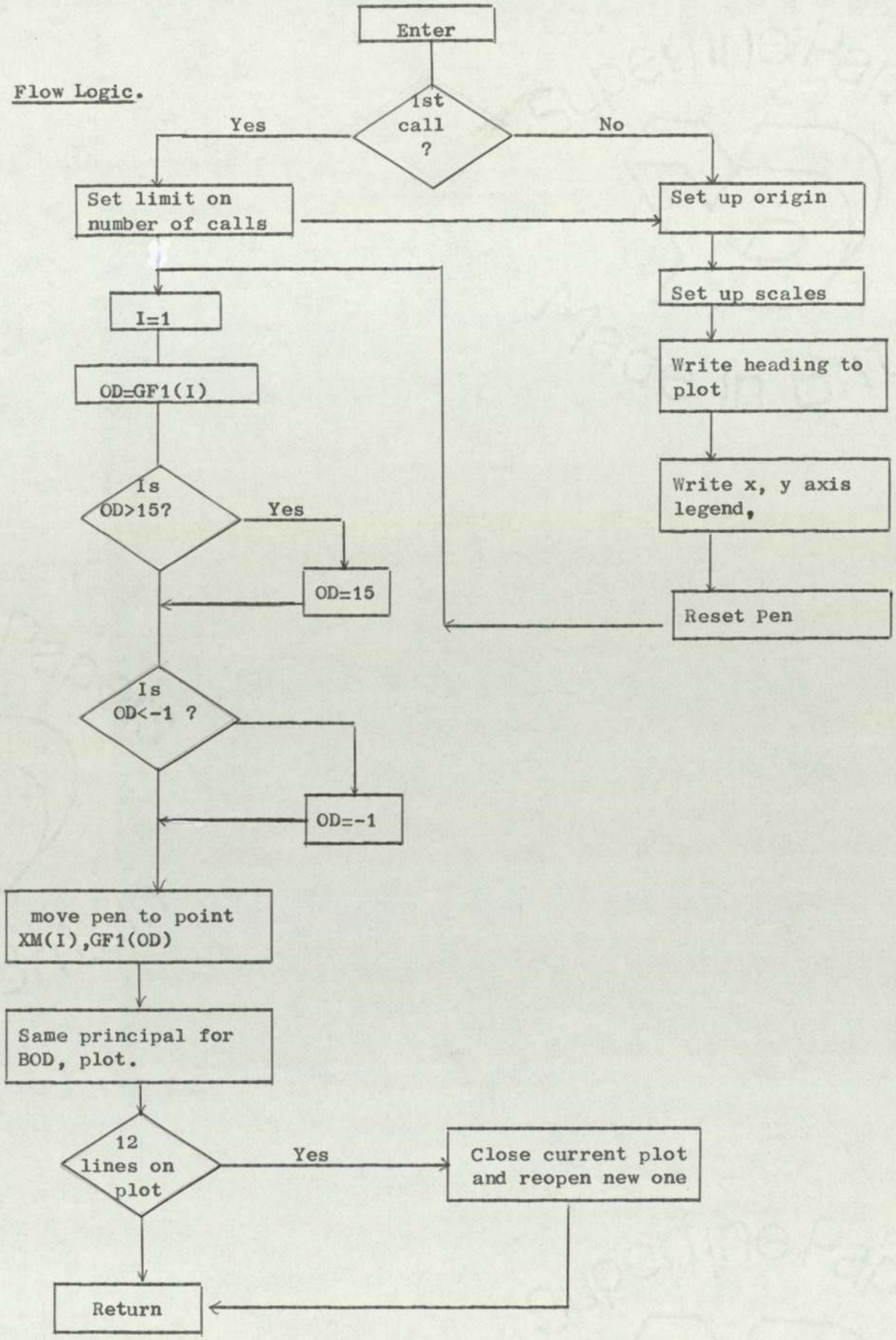
Subroutine SPLOT

Function:	To provide a graph plot facility.
User Functions:	None.
System Functions:	GRAEOR
Common Areas	/A/,/B/,/E/,/F/,/H/.
Calling Statement:	CALL SPLOT(GF)

Description.

GF is the array holding current estimates of BOD/OD from the stochastic means routine. The graph is set up if necessary and the OD/BOD lines plotted. Up to 6 pairs of lines may be plotted on one graph with closing and opening of graph files handled by the routine.
(Graph plotter is currently a CALCOMP 31" Drum plotter)
The option should be used selectively as the costs for a long simulation in terms of number of graphs generated could be costly.

Flow Logic.


Enter
1st call ?
Yes
No
Set limit on number of calls
Set up origin
Set up scales
Write heading to plot
Write x, y axis legend,
Reset Pen
I=1
OD=GF1(I)
Is OD>15?
Yes
OD=15
Is OD<-1 ?
OD=-1
move pen to point XM(I),GF1(OD)
Same principal for BOD, plot.
12 lines on plot
Yes
Close current plot and reopen new one
Return

Subroutine PRINT3

Function:	To provide a selective print and digital plot facility for use with ST or ST2
User Functions:	SMMTH (from SSMOD)
System Functions:	MAX,INT,WRITE/PRINT,MIN.
Calling Statement:	CALL PRINT3(GF,IPLOT,DOXS)
Common Areas	/A/,/B/,/E/

Description.

GF if the array holding the current similation of OD/BOD. DOXS is the array of D.O. (saturation levels for each of NS points at which printing is required. The digital plot is digitized every IPLOT points. The per cent saturation is evaluated using DOXS and local OD values. Three indeces are obtained:

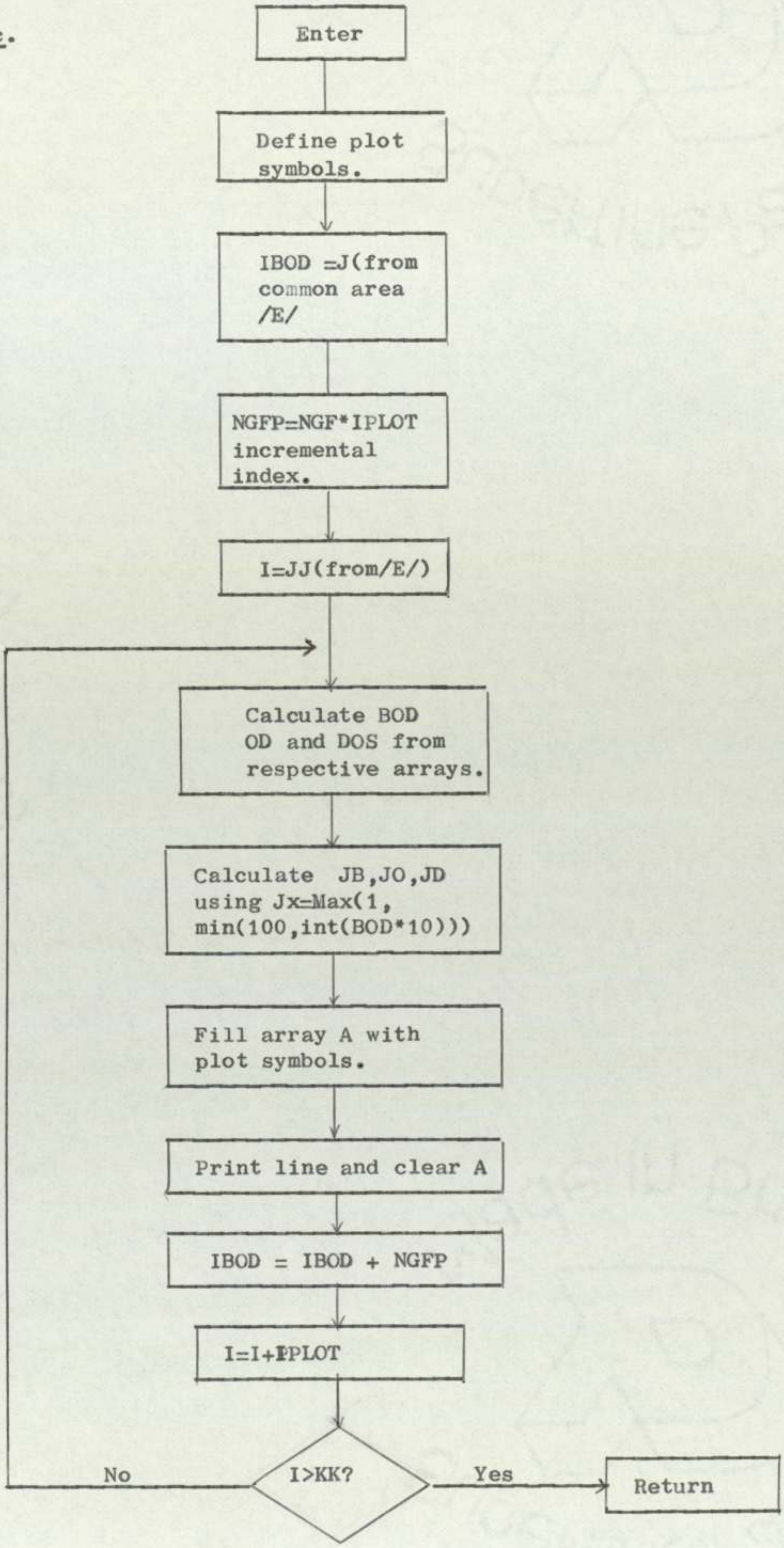
JB - BOD curve co-ordinate.

JO - O.D. curve co-ordinate

JD - D.O. (% sat) curve co-ordinate.

and plotting symbols put in the appropriate element of array A (one line of the plot). This option is considerably faster than the SPLOT option, although generating more volume output.

Flow Logic.


Enter
Define plot symbols.
IBOD =J(from common area /E/
NGFP=NGF*IPLOT incremental index.
I=JJ(from/E/)
Calculate BOD OD and DOS from respective arrays.
Calculate JB,JO,JD using Jx=Max(1, min(100,int(BOD*10)))
Fill array A with plot symbols.
Print line and clear A
IBOD = IBOD + NGFP
I=I+IPLOT
I>KK?
No
Yes
Return

Program FSHCHA (for ST2)

Function:	To act as an interface between models F1 or F2 and the stochastic model, ST2.
System Functions:	READ/WRITE I/o, REWIND, DEBUG, EXIT, FLOAT
User Functions:	INTP, FILL
Error Messages:	'E-OF-FILE PREMATURE' sets a return code to 255 to prevent sequential execution of next program (usually ST2).
Common Areas:	/AA/, /LIM/
Common Variables Reset:	Most of /AA/, /LIM/ reset from input from temporary file.

Description.

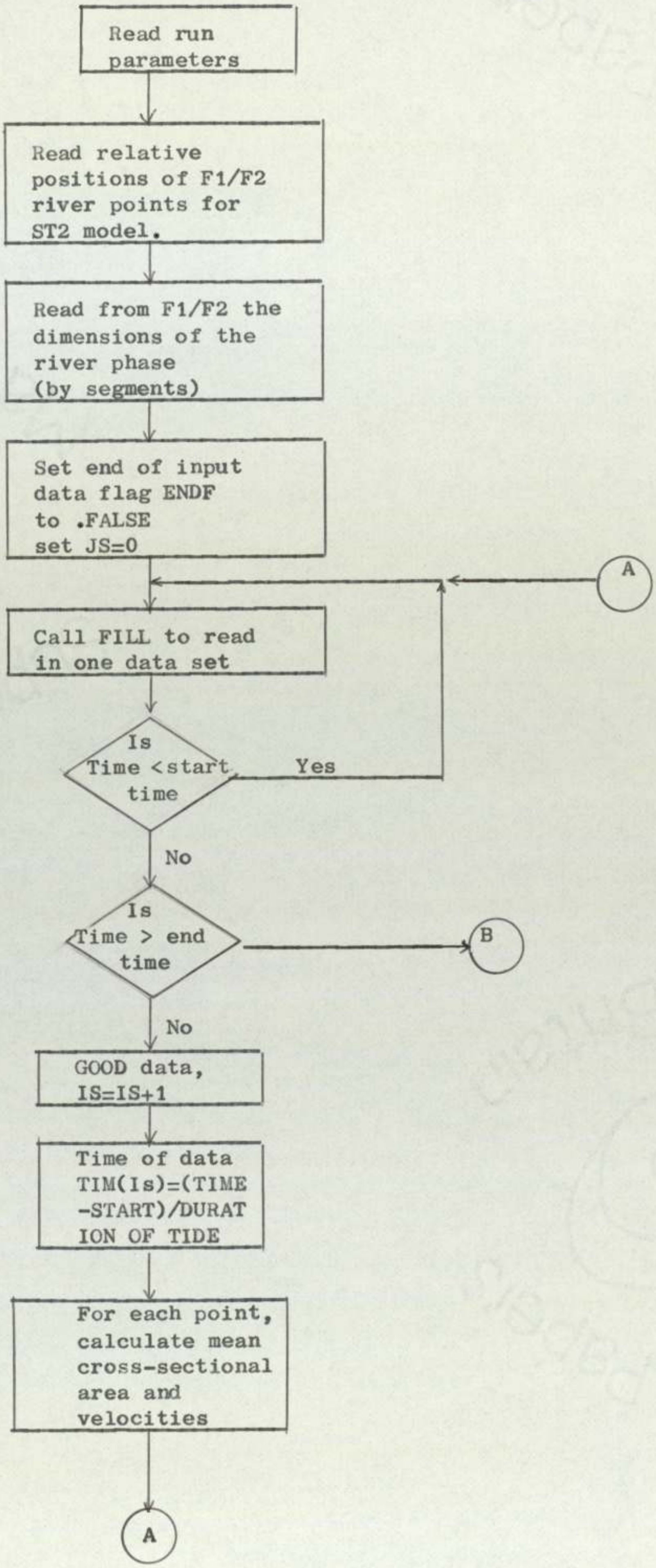
The two sections of the routine involve the input of data from models F1 or F2, and the estimation of intermediate grid points (as ST2 has a much finer mesh) from the sparse data thus provided.

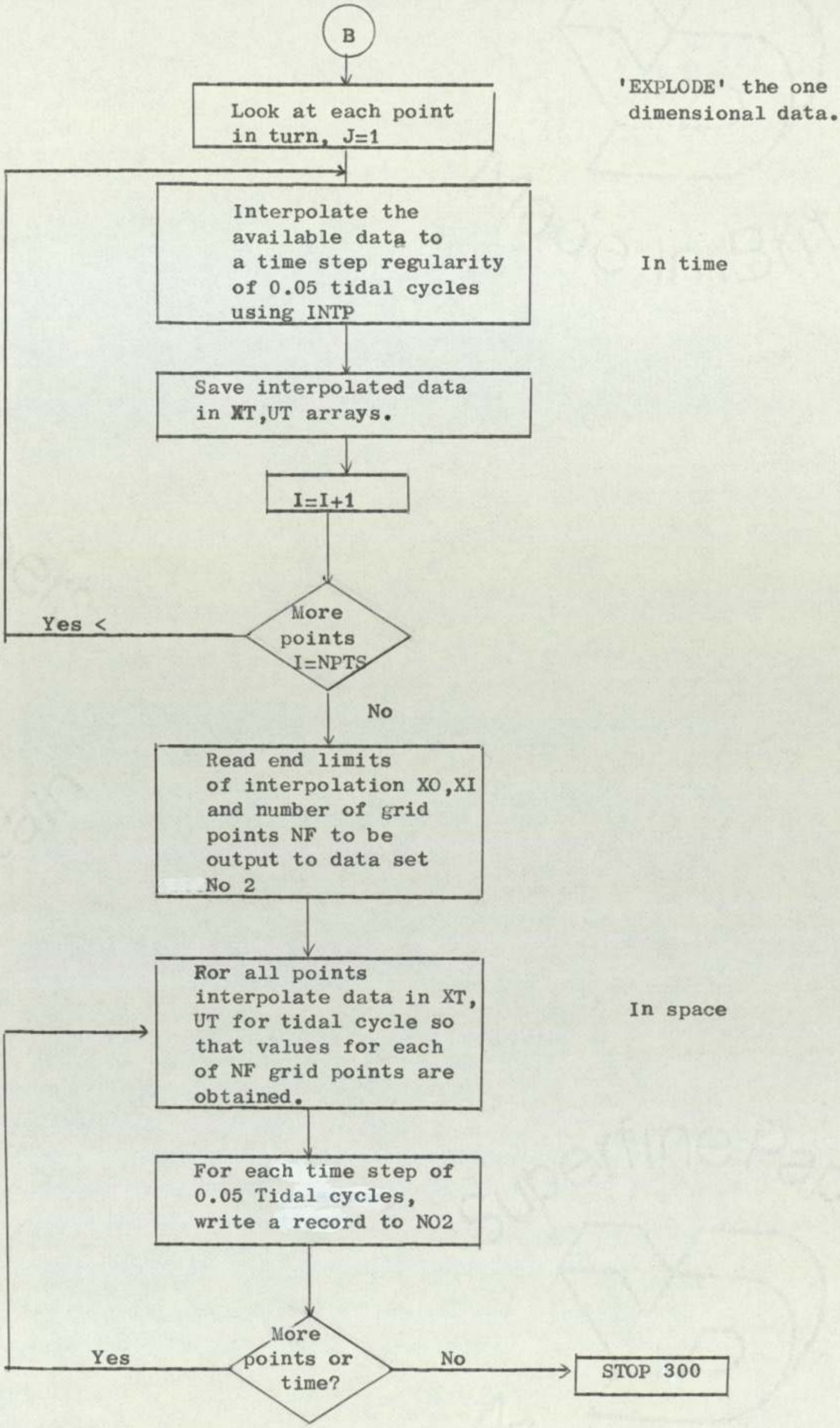
The first input record consists of problem parameters like times of start/end of useful data to be considered and scaling parameters, number of data points and interpolation scheme to be employed.

This is followed by a series of records, one for each useful point of models F1/F2. A distance from zero is labelled by a set of indeces which identify the point in F1/F2. For example: 8005.0 1 2 3 5 implies that segment 1 grid point 2, segment 3 grid point 5 coincide at a point 8005 units of distance from an overall zero. Only points on the main channel should be considered as the model ST2 will not handle networks. If it is desired to smooth data prior to interpolation, then in the list of grid indeces given above, for each point it is also possible to the neighbourhood upstream/downstream points and when data is transformed, averages of all these points will then be used.

The initial read requests from peripheral NO (the output data set from F1/F2) read the dimensions of the model in terms of bay grid size, river segments etc.

Flow Logic.


Read run parameters
Read relative positions of F1/F2 river points for ST2 model.
Read from F1/F2 the dimensions of the river phase (by segments)
Set end of input data flag ENDF to .FALSE set JS=0
A
Call FILL to read in one data set
Is Time <start time
Yes
No
Is Time > end time
B
No
GOOD data, IS=IS+1
Time of data TIM(Is)=(TIME -START)/DURAT ION OF TIDE
For each point, calculate mean cross-sectional area and velocities
A


B
Look at each point in turn, J=1
'EXPLODE' the one dimensional data.
Interpolate the available data to a time step regularity of 0.05 tidal cycles using INTP
In time
Save interpolated data in XT,UT arrays.
I=I+1
More points I=NPTS
Yes
No
Read end limits of interpolation XO,XI and number of grid points NF to be output to data set No 2
Ror all points interpolate data in XT, UT for tidal cycle so that values for each of NF grid points are obtained.
In space
For each time step of 0.05 Tidal cycles, write a record to NO2
More points or time?
Yes
No
STOP 300

Program ESTAN

Function:	To continue the stochastic and deterministic elements the model produces on probability distribution.
User Functions:	GROOM, LENGTH, TIMEX, TIDAL, PREDIC, LIMITJ, COUNT
System Functions:	READ/WRITE I/O
References:	Stochastic Modelling for Water Quality Management, DUH 2/71

Description.

The array XX has 3 indeces; i = 1 for BOD, = 2 for OD, j = point in space, k = time. Various sets of these predictions are read in from the output file of DIFEQ (see D2). Each reading is converted to phase within a tidal cycle. All such data is sorted into ascending order and concentration profiles obtained for phase 0, .25, .50 and .75 of the cycle. Profiles per station are also obtained. Actual data is then entered. Predicted means and an alpha % Pousson confidence limit is determined with the number of excessive values recorded.

D20

Program ESTLIM

Function:	To tabulate and graphically present the probability distribution from routine ESTAN.
User Functions:	GROOM,PREXXX,LIMXXX,ATSG,ACFI.
System Functions:	READ/WRITE I/o, FLOAT, 131-col LP
References:	Introduction to Numerical Methods, FB Hildebrand, McGraw-Hill

Description.

ATSG, ACFI are routines from the IBM SSPS Package. ACFI is a general interpolation routine, ATSG is a sort subroutine.
The data from DIFEQ and ESTAN is entered and sorted. Station by station probability profiles are generated for an entire tidal cycle using one sided distributions. Levels are tabulated and presented graphically for 9 different alpha % confidence limits.

D21 Subroutine GROOM.

Function: To arrange tide values and corresponding data values in strictly ascending order.

Calling Statement: CALL GROOM(XX,TIDE,IK,NKEEP)

System Functions: FIX, WRITE OUTPUT.

Description.

Calling List.

XX is the array described in ESTAN(D19). TIDE is the array retaining tide-cycle value for the appropriate XX values. IK are numbers of points to be sorted. NKEEP are the number of points retained after the sort (allowing for some identical TIDE values).

TIDE is converted to an array of values in the region $0 \leqslant TIDE(I) < 1$ i.e. the phase within a tide. This truncation gives rise to some duplicated values. The sort itself is a straightforward exchange between inconsistant positions, sifting out identical phase values.

Subroutine COUNT.

Function:	To signal if an observed data value is outside the predicted confidence limits.
Calling Statement:	CALL COUNT(XVALUE, XLIM1,XLIM2,ICOUNT)
Description:	ICOUNT is zero unless XVALUE falls within the range XLIM1 and XLIM2, the two tails of the predicted probability distributions.

D23 Subroutine LENGTH.

Function: To calculate OD profiles for tidal phases in the Estuary.

User Functions: ATSG, ACFI

System Functions: WRITE OUTPUT, FLOAT

Calling Statement: CALL LENGTH (XX,TIDE,XPOSIT, NKEEP)

Description.

XX,TIDE and NKEEP are as in D19, D21. XPOSIT is an array of data positions. Values throughout the estuary of OD levels at phases 0, 0.25, 0.50 and 0.75 are computed using ATSG/ACFI.

Subroutine TIMEX.

Function: To calculate OD profiles for various stations through the tidal phase.

User Functions: ATSG,ACFI.

System Functions: WRITE OUTPUT, FLOAT.

Calling Statement: CALL TIMEX(XX,TIDE,NKEEP).

Description.

As LENGTH but interpolation is for intermediate phase sections at each station in turn.

D25 **Subroutine PREDIC/PREXXX**

Function: For a given determined data value the routine calculates a predicted value from the output of DIFEQ.

User Functions: ATSG,ACFI.

Calling Statement: CALL PREDIC(XYZ,NKEEP,J,TIDE,PHASE,XMEAN) (PREXXX).

Description.

NKEEP, PHASE, TIDE are as in D21, XY7 is the array XX. To perform the calculation the routines ATSG,ACFI are used. J references the current station number being processed, from ESTAN. Routine PREXXX performs the same task from ESTLIM.

Subroutine LIMITJ.

Function: Calculates the Poisson variance and thus confidence limits on the predicted means from DIFEQ.

System Functions: EXP, FLOAT

Calling Statement: CALL LIMITJ(XMEAN,X1,X2,J).

Description.

J is the station number. XMEAN is the predicted mean. The stochastic coefficients are defined internally, as well as the level of significance of interest currently. X1 and X2 are the limits of the defined confidence limits.

Subroutine TIDAL.

Function: From time of day and date, the routine determines phase of tidal cycle.

User Functions: DATE-TIME package.

System Functions: FLOAT,FIX.

Description:

The elapsed time between the current date-time and that of a known high tide is calculated, and the modulus of this to be 12.4 used as the phase of the current time.

Program DELTA.

Function: To compute an estimate of the stochastic coefficient Δ, from field measurements of oxygen content.

User Functions: DOXSAT, TID12.

System Functions: READ/WRITE INPUT/OUTPUT

Description.

Input consists of some header data, and a start and end time. Only data between these times will be retained for the purposes of calculation to remove the effect of photosynhthetic activity.

Each data card contains the following data:

a) Sample station number.

b) Date of sample.

c) Time of sample.

d) Phase in tidal cycle (ratio, 0 to 1 -low water onwards)

e) measured D.O. (ppm)

f) Measured Temperature (deg. C)

g) Measured salinity (gms/litre)

Time limits of 0500 to 1000 can usually be used. Any very low D.O. levels are disregarded as often the biological processes have altered.

The tidal cycle is split into 12 phases of equal duration, and the stochastic coefficient is calculated for as many stations and as many phase segments as data allows. Also an overallΔ is computed per station.

Output consists of station-number, phase-segment and the associated estimate of Δ.

The overall Δ per station is also printed. All calculations are based on oxygen-deficit, computed from the measured D.O. levels, temperatures and salinities via the subroutine DOXSAT.

Input File Structure for Model ST/ST2

There are three phases of input:

Phase 1. Basic Run Parameters.

Record.	Col. from	Col. to	Variable.	Type.	Interpretation.
1	1	10	NG	I	Number of BOD Grid Points.
	11	30			Comment.
	31	50	NF	I	Number of O.D. Grid Points.
	51	80			Comment.
2	1	10	NS	I	Number of data print points.
	11	30			Comment.
	31	40	ICD	5I	Current start date/time for simulation.
	51	80			Comment.
3	1	10	XI	F	Start Point of data.
	11	20			Comment.
	31	50	XO	F	End Point of data.
	51	80			Comment.
4	1	10	TI	F	Start time of printing in hours.
	11	30			Comment.
	31	50	TM	F	End time of simulation in hours.
	51	80			Comment.
5	1	10	PRI	F	Print step in hours.
	11	30			Comment.
	31	50	PRIERR	F	Latitude in print times, fractions of an hour.
	51	80			Comment.
6	1	10	TEMP	F	Ambient temperature (degrees C)
	11	30			Comment.
	31	50	UTYP	F	Type of velocity data flag.
	51	80			Comment.
7	1	10	OMEGAF	F	Feed frequency (discharges)
	11	30			Comment.
	31	50	UPSTBC	F	Upstream BOD boundary condition.
	51	80			Comment.
8	1	10	UPODBC	F	Upstream O.D. boundary condition.
	11	30			Comment.
	31	50	RK2COR	F	Reaeration rate correction factor.
	51	80			Comment.
9	1	10	CONST	F	Feed constant (discharges).
	11	30			Comment.
	31	50	FACTOR	F	Feed Factor (discharges)
	51	80			Comment.
10	1	10	SST	F	Time at which steady state is reached.
11	1	10	IPLOT	I	Plot step option.
12	1	10	IACC	I	2nd or 6th order accuracy option.
13	1	10	TIDED	F	Duration of tidal cycle (in hours)
14	1	10	RELE	F	Relative Error in RKMI
15	1	10	ACC	F	Accuracy of interpolation INTP
16	1	10	ABSE	F	Absolute error in RKMI
17	1	10	PLIM1	F	New time step after PLIM1 hours (start-up)
18	1	10	PSTEP	F	New time step.
19	1	10	IPNTYP	I	Output options (one of 9)
20	1	10	IFOLO	I	One grid point to monitor closely.
21	1	10	IRKMTH	I	Order of Runge-Kutta method used.
22	1	10	DT	F	Initial time step in hours.
23	1	70	XS	nF	Array to hold points for printout of simulated values (NS values). May be spread over any number of records.

Phase 2. Input of Array Data.

The principal input is via the routine INPUT. However, each array input has a common first line which steers it through the rest of the subroutine. The common format is:

Col. from	Col. to	Variable.	Type.	Interpretation.
1	2	IPT	I	Input type switch.
3	10	blank		Value -2 - Parameter constant throughout estuary. -1 - Non zero at some points. 0 - Constant within segments. +1 - Smoothly varying function.
11	60	D	AA	Title of input data array.
61	70	VAL	RF	Constant value in -2 option.

If further records are required for the current input option, these follow on the free format, column 1 to 20.

Arrays so input are:

a) Initial OD profile.

b) Initial BOD profile.

c) Freshwater flow velocity.

d) Maximum tidal velocity.

e) OD diffusion coefficient.

f) BOD diffusion coefficient.

g) BOD decay rate constant.

h) Reaeration rate.

i) BOD decay rate, K_D processes.

j) Land run-off.

k) Benthol demand.

Phase 3. Input of Discharge Data.

The input is made into routine STPF. For this reason, a dummy call to the subroutine is made early on in the main executive program.
The first record contains NDIS on free format between cols 1 and 60.
This is the number of discharges to be read in.
Each subsequent line image deals with one discharge.

Col. from	Col. to	Variable.	Type.	Interpretation.
1	8	R1,R2	A	Title of discharge.
9	10			Blank.
11		TCL	R	Closing time (hours into tidal cycle)
		TOP	R	Opening Time ('ditto')
(Free)		POS	R	Position in distance from zero
		HLOAD	R	Load rate when open.
		RLEAK	R	Leak load rate when closed.
	60	XSELT	R	Cross-sectional area mean at point of discharge.

Closing/opening times allow tide locked simulations, times are in hours from low water.

D.30 PRINCIPAL VARIABLES USED.

A	Common Area. Holds mainly time parameters.	
A	RKMI Work array.	IR*4
BSE	Common /A/ Absolute error permitted in the integration scheme.	R*4
CC	INTP Relative permitted accuracy of interpolation	R*4
	Common Area Holds basic data arrays.	
	RKMI Work array	604R*4
BODD	POUT BOD Boundary condition, downstream and variable.	R*4
BODU	POUT BOD upstream boundary condition, constant.	R*4
ODD	POUT O.D. boundary condition, downstream and variable	R*4
ODU	POUT Upstream boundary condition for O.D.	R*4
D	PRINT3 BOD in ppm at point IBOD	R*4
	Common Area Holds main working arrays	

GF2 Common /C/ 604R*4

Evaluated function holding array.

IS STPF 25x10R*4

Array whose i,jth element is the discharge load of discharge i for the j tenth of the tidal cycle.

OX POUT R*4

Level of dissolved oxygen in system at TEMP degrees to achieve 100% saturation.

015 Common/D/ R*4

= (time step)/15

02 Common/D/ R*4

=(time step)/2

02D3 RUNGK2 R*4

=(2Δt)/3

3 Common/D/ R*4

=(time step)/3

3D4 RUNK2 R*4

=3(Δt)/4

3H Common/D/ R*4

=(time step)/6 = (D03)/2

318 Common/D/ R*4

=(time step)/3x 1/8

2 RKMI R*4

=(2Δt)/3

4 RKMI R*4

=(Δt/3)x4

45 RKMI R*4

=(Δt/3)x9/2

Common area.

Holds main working arrays

RKMI 604 R*4

Work array

ONST Common/F/ R*4

Redundant factor for cyclic discharges

ONST INPUT 7I*4

Contains the constant parameters as single values.

Common Area

Holds time interval constants for integration.

INPUT 13A4

Array to hold title of input parameter

POUT 60x3R* 4

Array containing time/OD/BOD data for everytime step
for one grid point (pre selected and optional)

RKMI 604R*4

Work array

ST Common/C/ 604 R*4

First derivative holding area

ND Common/C/ 604 R*4

Second derivative holding area

Common/B/ 302R*4

Array of Benthal demands

C Common /INP/ R*4

Constant value of array DB

'1 Common/C/ 604R*4

Evaluated function holding array.

0345	RKMI =(Δt/3)x9/2	R*4
0356	Common/D/ =(time step)/3x5/6	R*4
036	RKMI =2(Δt)	R*4
0387	Common/D/ =(time step)/3 x8/7	R*4
OS	PRINT3 % saturation levels of D.O.	R*4
T	Common/D/ Time step of integration method	R*4
X	Common/D/ Grid size of both meshes	R*4
XF	Common/D/ Grid size of O.D.	R*4
XG	Common/D/ Grid size of BOD	R*4
75	Common/D/ Second location for time step of integration for routine POUT	R*4
	Common Area Holds indeces for delimiting arrays.	
	Common/B/ Diffusion coefficient array of O.D.	302R*4

FC	Common /INP/ Used for constant value of array EF	R*4
G	Common /B/ Diffusion coefficient array of BOD	302R*4
GC	Common/INP/ Used for constant value in array EG	R*4
	Common Area Holds factors and limits of the system.	
	STPF Tidelocking load magnification =1/(1 + TC - TO)	R*4
ACTOR	Common/F/ Redundant factor for cyclic discharges.	R*4
	Common/B/ Point source discharges.	302R*4
RST	POUT Logical switch for first call of routine.	L*1
X	INTP2 The estimate of the dependent variable from an independent variable value XX	R*4
	INTP Estimates of interpolation see FXX in INTP2	7R*4
	Common Area Holds tide duration and work words.	
	DOXSAT Array holding.	34R* 4
	Common /C/ Predictions array.	604R*4

F2	Common/C/ Predictions array.	604R*4
HLOAD	STPF Hourly load of outfall.	R*4
HLOAD2	STPF Net load = HLOAD-RLEAK	R*4
I1	P Lower limit of array X to be printed.	I*4
I2	P Upper limit of array to be printed in subroutine.	I*4
A	Common/E/ Switch valued at 1 or 2. When valued at 1; GF2 contains data predicted for T and GF1 will be used to hold T + Δt. For 2 other way around.	I*4
ACC	Common/F/ Accuracy of the method of solution employed. Should be two or six.	I*4
ACC	RKMI Accuracy of derivation required, should be 2nd or 6th order.	I*4
B	Common/E/ A switch similar to IA	I*4
	Common/E/ A switch similar to IA	I*4
	Program DIFEQ Input card reader = 5	I*4
	INPUT Input peripheral number.	I*4
	Common/E/ An integer array holding the current start date. Format Y-M-D-H-Min.	5I*4

ICI	INPUT	I*4
	Call counter to the routine.	
ICON	INPUT	I*4
	Number of constant parameters counted in calls to routine.	
ID	P	I*4
	Numeric identifier of the location of the calling statement to reach P.	
IFOLO	Common/J/	I*4
	If zero, no action. If > 0, the predictions for grid index IFOLO are stored every time step to allow the development of the method to be followed.	
IND	POUT	I*4
	Index to array D, a trace of one point in the grid when requested.	
IPLOT	Common/F/	I*4
	Every IPLOT calls of POUT output routine, a graphical display of predictions are made.	
IPNTYP	Common/F/	I*4
	Type of output required.	
IPT	INPUT	I*4
	Input code for data type.	
IR	POUT	I*4
	Number of tidal cycles wholly simulated.	
IREM	POUT	I*4
	Part of IPRNT type code used to determine type of output	
IRKMTH	Common/J/	I*4
	A variable to select either a second order or fourth order self correcting Runge Kutta method. Should be 2 or 4	

T Common/E/ 40A4

Array for up to two card images of a title.

TEMP DOXSAT I*4

Truncated value of TEMP

Common Area

Holds information on order of method and output options.

Common/E/ I*4

Lower index for BOD data. Set to NG+2

1 RKMI I*4

Convergence of method flag.

CD POUT 5I*4

Array holding current simulation time from start date/ time in array ICD.

R POUT I*4

Error flag from integration in POUT

Common/E/ I*4

Lower index for O.B. data elements. Should be unity at all times.

Common/E/ I*4

Upper limit of BOD indexes. Set NF+NG+2

Common/E/ I*4

Upper limit of O.D. data. Set to NG+1

Common/E/ I*4

Upper limit for BOD data for arrays only of BOD factors - set to NF+1

. POUT I* 4

Degree of interpolation used in calls to INTP.

S STPF I*4

Grid point of discharge.

DIS STPF I*4

Number of discharges to be read in.

GP STPF 25R*4

Array holding the grid points to which the various discharges have been allocated.

G DIFEQ I*4

Number of grid points in the BOD grid.

GF Common/E/ I*4

The ratio of NF/NG - should be >1 and a whole number.

DIFEQ I*4

Number of points on the O.D. grid.

INTP2 I*4

Number of points in calling list vectors.

Common/E/ I*4

Number of print positions requested for predictions - should be <25

PRINT3 R*4

Oxygen deficit at point I at current simulation time.

GA Common Area/A/ R*4

Tidal frequency = 2 π/TIDED

GAF Common Area/A/ R*4

A flexible frequency for use in calculating values of FF.

DIFEQ R*4

Value of PI = 3.1416

I1 DIFEQ R*4

The simulation time up to which the print interval is PRI.

POUT L*1

Logical variable set to .TRUE. if the current call of POUT is also to be plotted. Every IPLOT calls are plotted.

OS STPF R*4

Distance of outfall from datum of grid measurement.

RI Common/A/ R*4

Interval, in units of hours, between printing predictions to the line file.

RIERR Common/A/ R*4

Latitude permitted in units of PRI to print predictions to the line file.

STEP DIFEQ R*4

The print stop after the simulatipn time had reached PLIM1

INTP2 R*4

The ratio of interpolation weighting for Δx in relation to value of XX

P A8

Eight character identifier of the array to be printed.

1 Common/G/ R*4

Work area.

2 Common/G/ R4

Work area.

19E6 Common/A/ R*4

Conversion factor for p.p.m. to lbs. per cubic mile (9.19×10^6)

LE Common/A/ R*4

Maximum relative error permitted in the integration scheme.

D Common/B/ 302R*4

Array of rate constants for removal of BOD by non oxygen consuming processesand by oxygen consuming processes

DC Common/INP/ R*4

Constant value of array RKD.

:1 Common/B/ 302R*4

BOD utilization rate constants.

:1C Common/INP/ R*4

Used for constant value of RK1

:2 Common/B/ 302R*4

Array of reaeration rates.

:2COR Common/F/ R*4

Correction multiplier for array RK2

:2C Common/INP/ R*4

Constant value of array RK2

EAK STPF R*4

Background leak rate operative even while discharge is closed.

R Common/B/ 302R*4

Array of land run off.

RC Common/INP/ R*4

Constant value of array RLR

STPF R*4

Reciprocal of tide duration.

FUNCT R*4

Current phase of tide = sin(OMEGA*T)

Common/E/ R*4

The time in hours, after which the system is considered to have reached steady state conditions. Will precipitate hard copies of predictions after that time.

Common Area/A/ R*4

Current time into simulation starting at TI, units are in hours.

C STPF R4

TCL in units of tides.

CL STPF R*4

Time in hours into a flood cycle before the discharge ceases.

EMP Common/A/ R*4

Average temperature in system during simulation in degrees centigrade.

EMP DOXSAT R*4

Temperature at which DO saturation is required to be estimated.

I DIFEQ R*4

Start time of simulation. Usually zero.

IDE Common Area/A/ R*4

Current time into simulation starting at TI in units of TIDED, i.e. in units of completed tides.

IDED Common/G/ R*4

The duration of one tide in hours.

LAST STPF R*4

The time in hours of simulation since the last print of discharge levels.

I FEQ T*4

End time of simulation, in hours.

D STPF R*4

TOP in units of tides.

P STPF R*4

Time into the tidal cycle when a discharge recommences, having ceased at TCL

RKMI R*4

Error in integration of element I in grid for step size Δt

QQ	RKMI	R*4	Maximum error in integration over whole grid over one time step.
YP	POUT	4A4	Tidal states RUSH/HIGH/EBB/LOW
	INPUT	IR*4	Array to hold read in data.
	FUNCT	R*4	Total particle velocity at a grid point at current simulation times.
F	Common /B/	302R*4	Array of freshwater velocities
PODBC	DIFEQ	R*4	The constant upstream boundary oxygen deficit in ppm
PSTBC	DIFEQ	R*4	The constant upstream boundary condition for BOD in ppm
S	Common /B/	302R*4	Array of tidal flood velocities
TYP	DIFEQ	R*4	Type of velocity input switch. If set to >0 , US is a conglomerate velocity, if <0 US is split into UF and US
AL	INPUT	R*4	Either a constant parameter value or an index
	P	I2*R*4	Array to be printed
BS	FUNCT	R*4	Absolute value of SNT
	INPUT	R*4	Distance along grid from XI (assumed zero)

XI	Common /F/ Start position of switch .Stretch to be modelled. Should be zero usually.	R*4
XM	Common /B/ Points on the BOD grid	302R*4
Xo	Common /F/ End point of stretch to be modelled.	R*4
XS	Common /B/ Points in the system at which prints of predictions are required	25R*4
XSECT	STPF Average Cross-sectional Area of estuary at point POS	R*4
XX	INTP2 Value of independant variable for which dependant parameter is to be estimated.	R*4
XZ	Common /B/ Points on the O.D. Grid	302R*4

APPENDIX E

STEADY STATE MODEL COMPUTER ALGORITHM

APPENDIX E - CONTENTS

E.1 INTRODUCTION

The Steady State Model Program is the least intricate of the three models being presented.The algorithm is executed in the same routine as the executive functions. The additional subroutines are extensions or additional input/output options. As this model will be extensively used by management, a good deal of attention was given to data presentation.Output is effected through line printer and Visual Display Unit or Graph Plotter (current version for use with CALCOMP Models). Routine Matset solves explicitly for the four non-interactive components.

E.2 Model Composition

To satisfy the composition/consolidation phase of the object code generation,all subroutine must be included in the model.However, by choice of certain parameters these routines can be switched in or out as required.

E.3 Executive/Main Routine SSMOD

Objective : To execute the algorithm of the Steady State Model and control input/output and ancilliary functions.

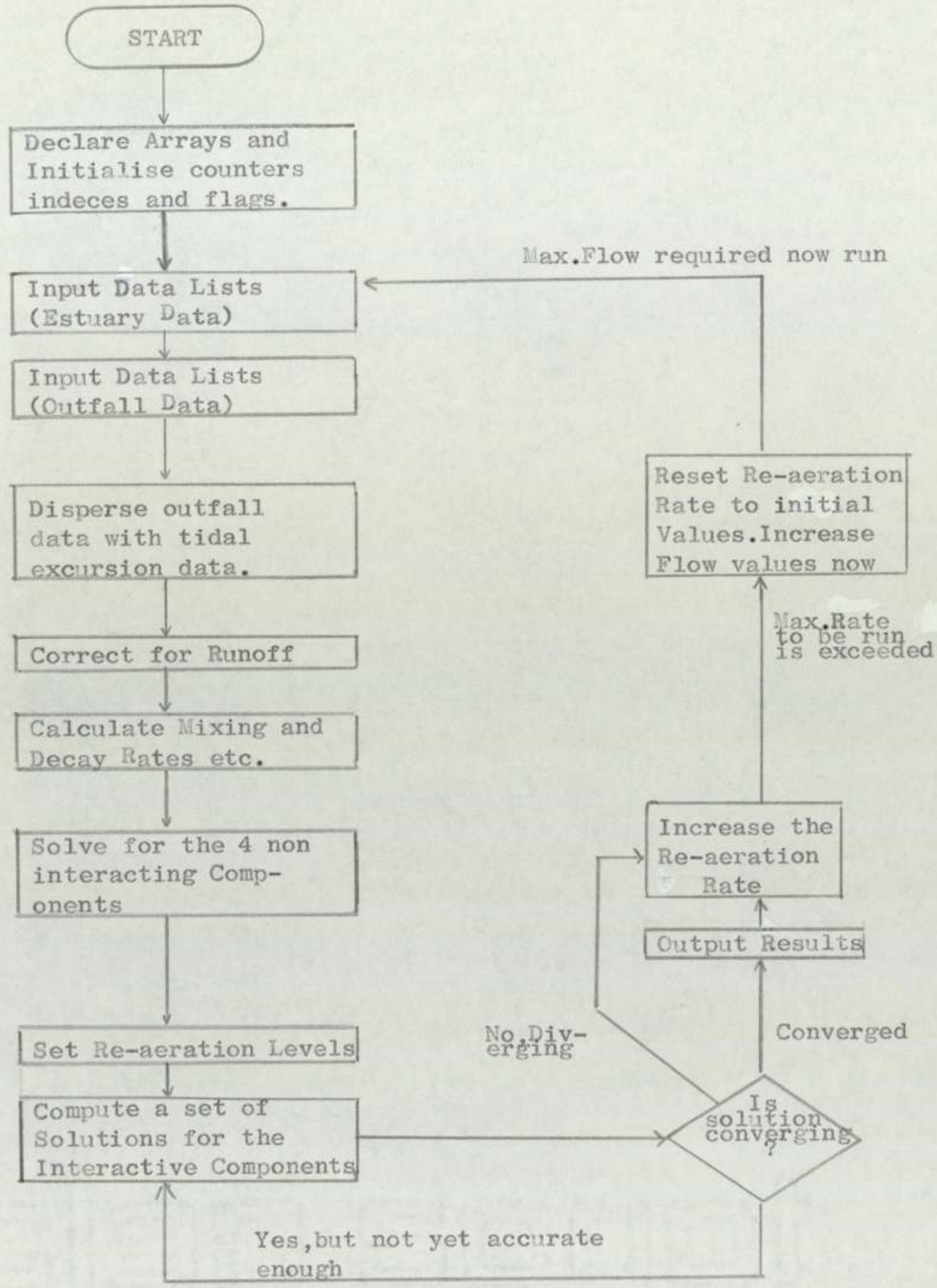
Subroutines Used : EXMINE MATSET OUTPUT PLOT PLOTGP RUNOFF SMMTH SUM TEXRE

System Routines Used : I/O,SQRT,ABS,GRAFOR,INT

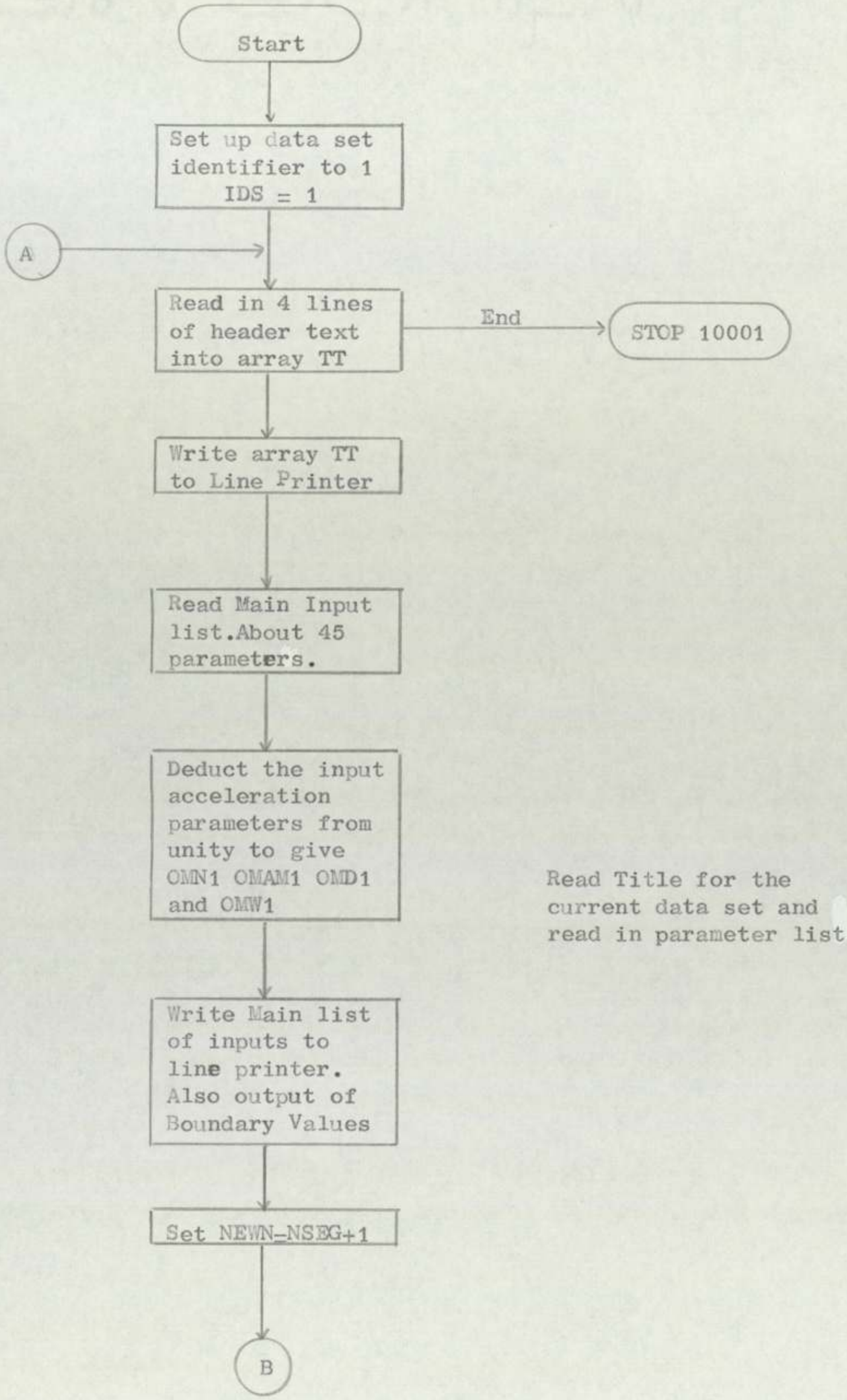
Method : See E.4

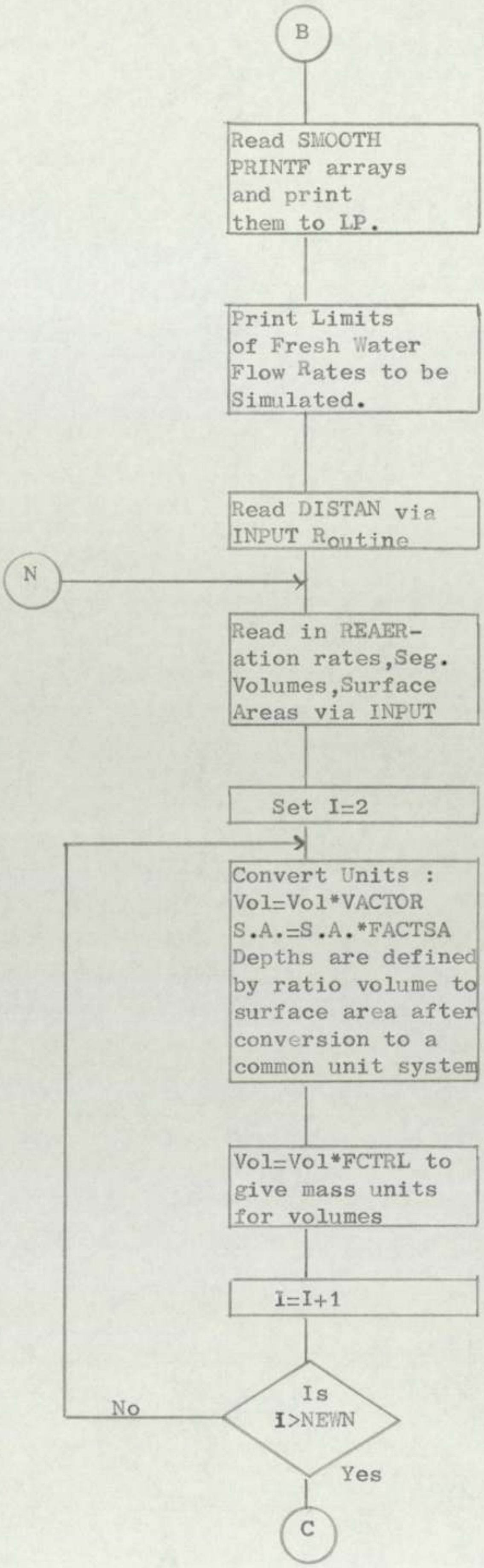
Time : On an ICL4-70 a simulation for data as specified in Ch.6 required 10 cpu units ,mainly by production of a digital plot and grph plot.Much depends on the choice of acceleration parameters.

Flow Diagram - Schematic Outline Logic



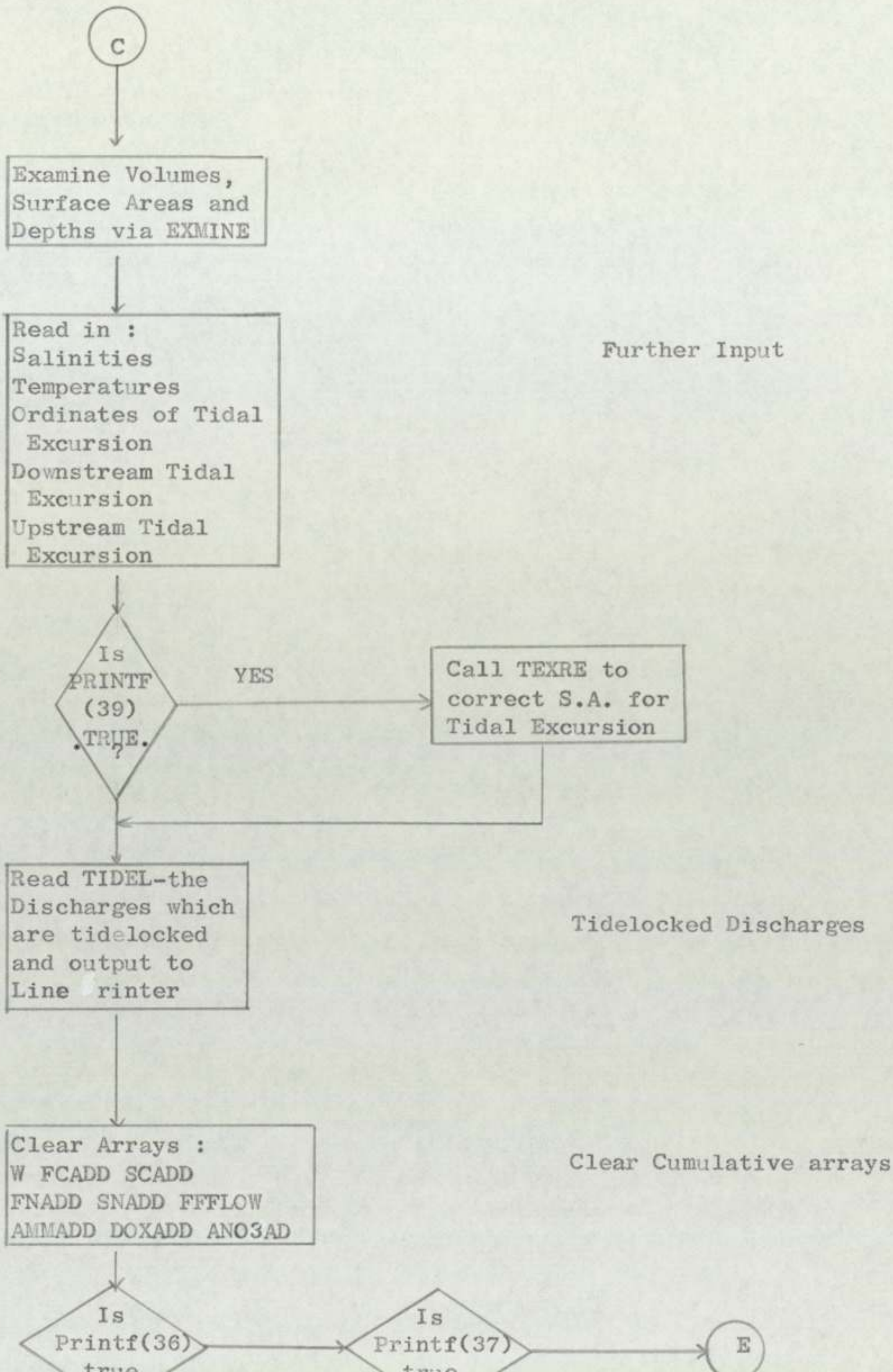
Detailed Flow Diagram for SSMOD-The major routine of the Steady State Model

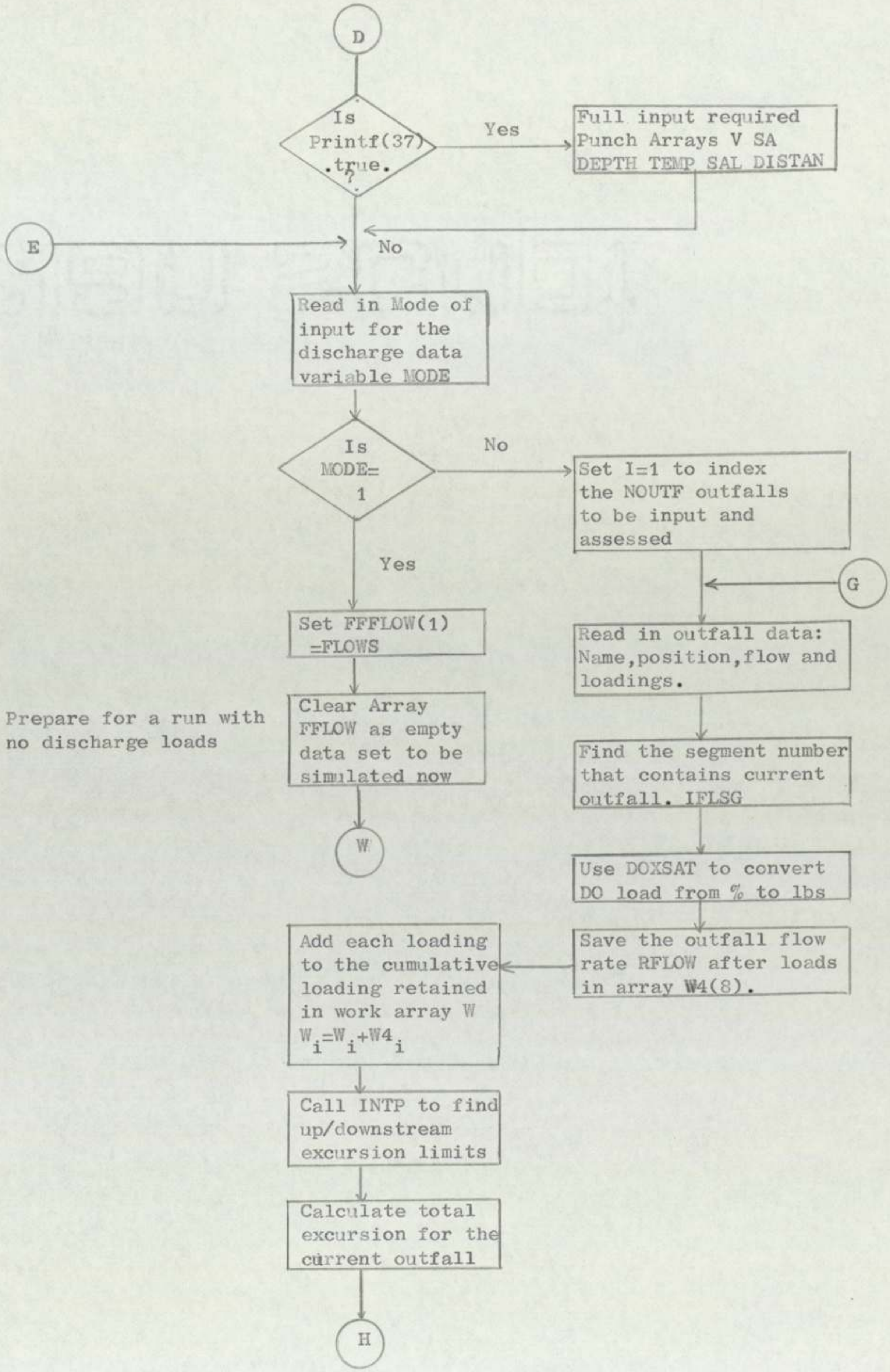




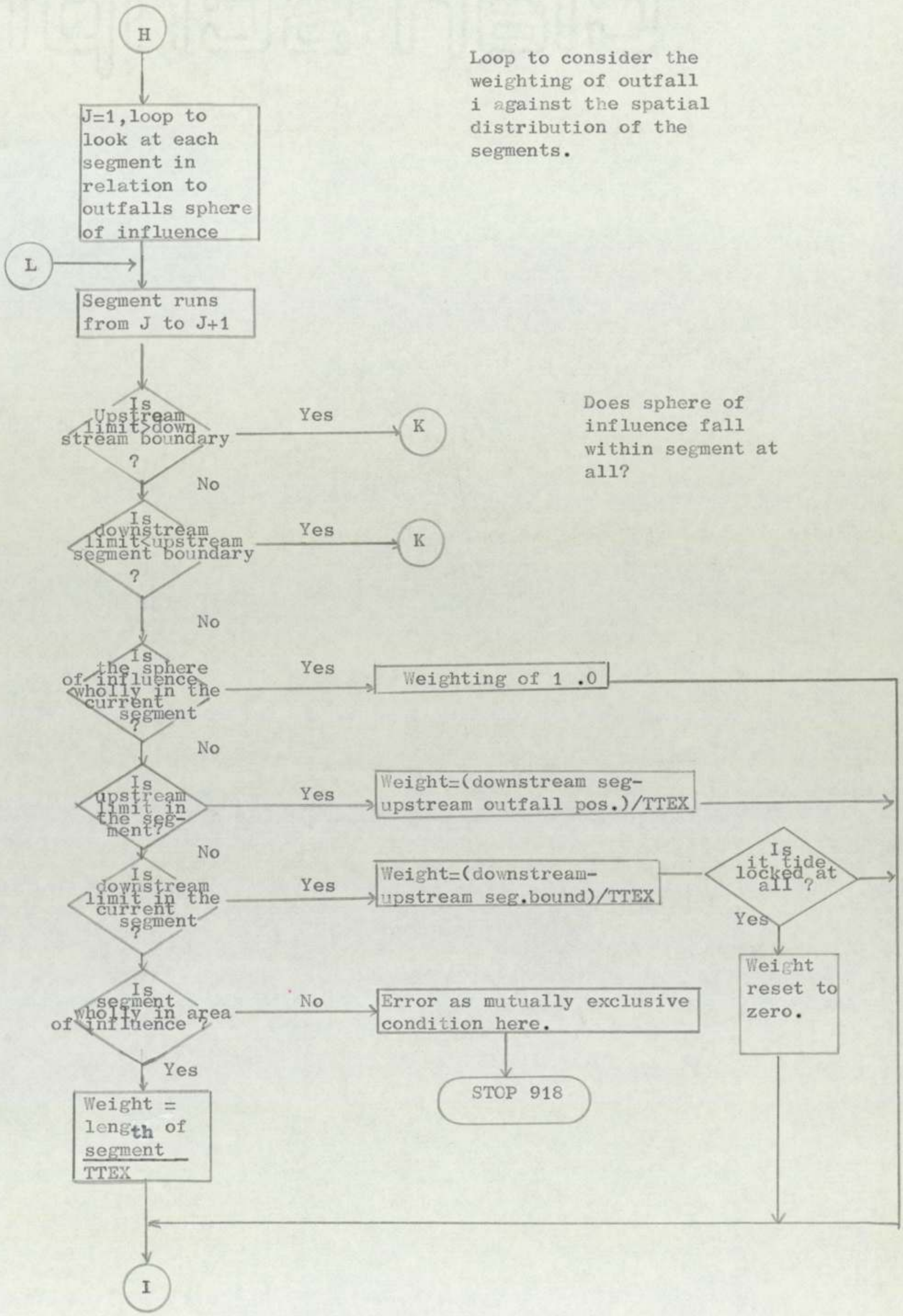
Program
Trace
I/O Control

Conversion of
Physical Data to a
Uniform system.

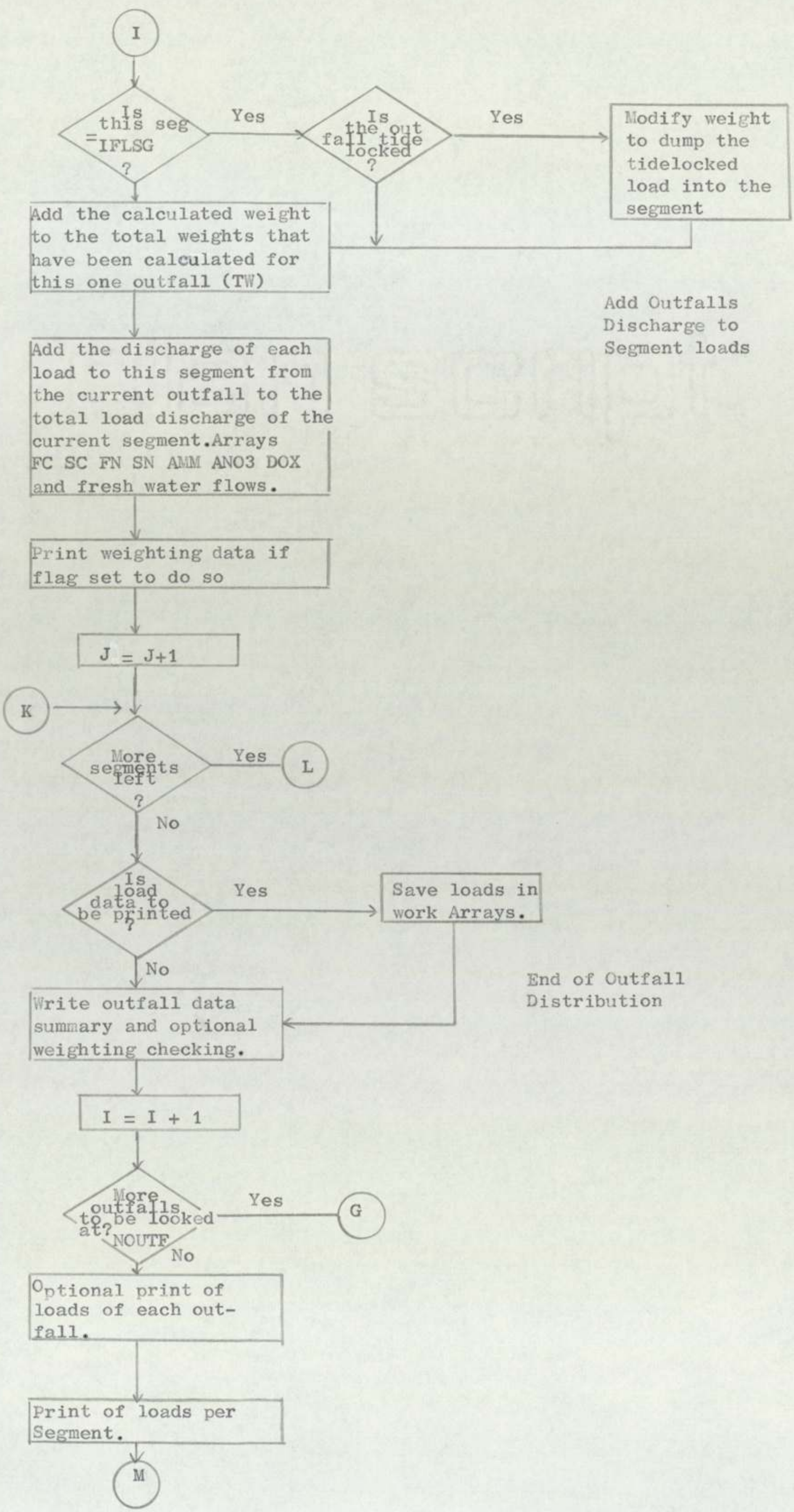

C
Examine Volumes, Surface Areas and Depths via EXMINE
Read in : Salinities Temperatures Ordinates of Tidal Excursion Downstream Tidal Excursion Upstream Tidal Excursion
Further Input
Is PRINTF (39) .TRUE. ?
YES
Call TEXRE to correct S.A. for Tidal Excursion
Read TIDEL-the Discharges which are tidelocked and output to Line rinter
Tidelocked Discharges
Clear Arrays : W FCADD SCADD FNADD SNADD FFFLOW AMMADD DOXADD ANO3AD
Clear Cumulative arrays
Is Printf(36) .true. ?
Yes
Is Printf(37) .true. ?
Yes
E
Punch title Punch Basic run Parameters
D

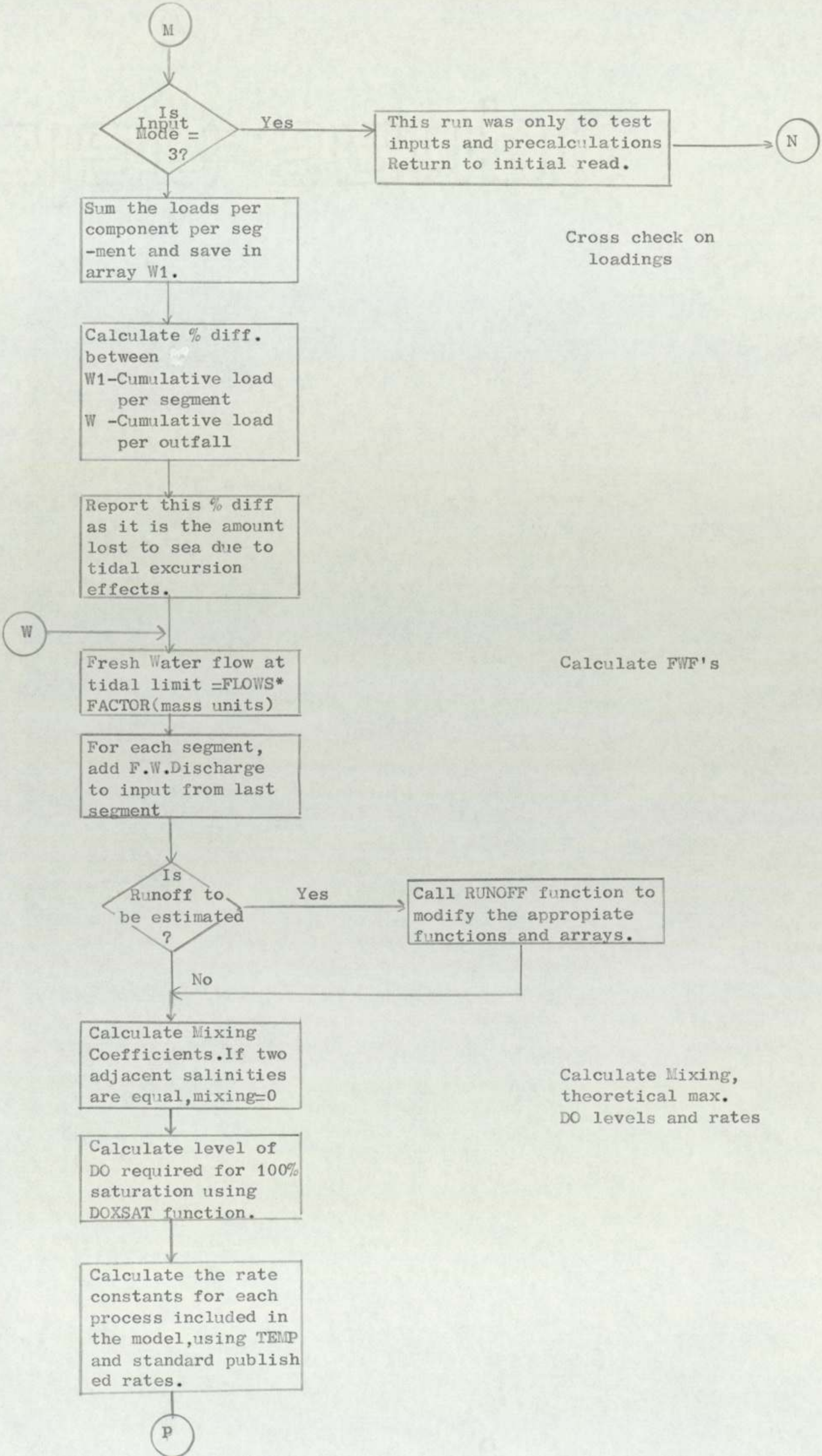


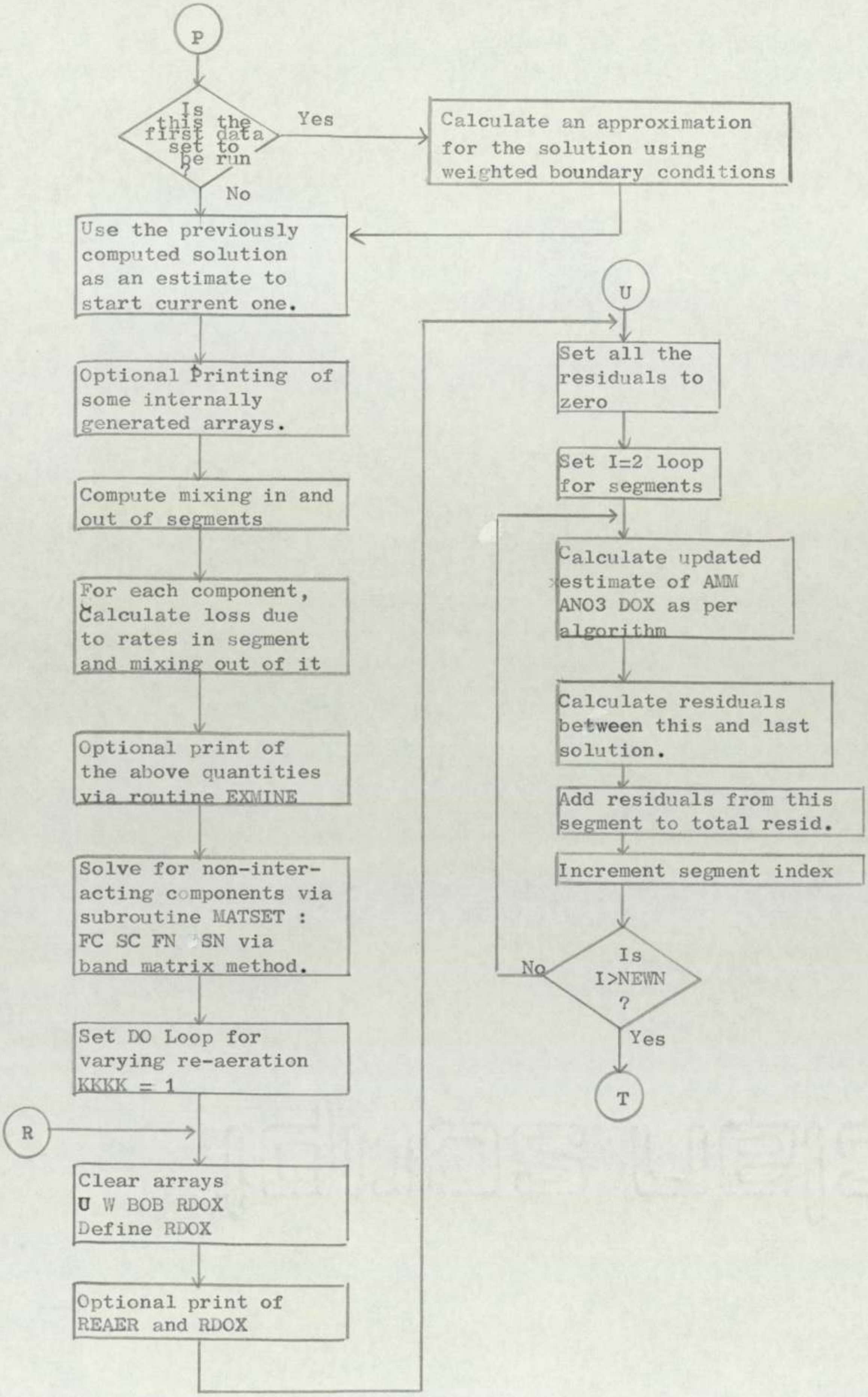
D
Is Printf(37) .true. ?
Yes
Full input required Punch Arrays V SA DEPTH TEMP SAL DISTAN
E
No
Read in Mode of input for the discharge data variable MODE
Is MODE= 1
No
Set I=1 to index the NOUTF outfalls to be input and assessed
G
Yes
Set FFFLOW(1) =FLOWS
Read in outfall data: Name,position,flow and loadings.
Prepare for a run with no discharge loads
Clear Array FFLOW as empty data set to be simulated now
Find the segment number that contains current outfall. IFLSG
W
Use DOXSAT to convert DO load from % to lbs
Add each loading to the cumulative loading retained in work array W $W_i=W_i+W4_i$
Save the outfall flow rate RFLOW after loads in array W4(8).
Call INTP to find up/downstream excursion limits
Calculate total excursion for the current outfall
H

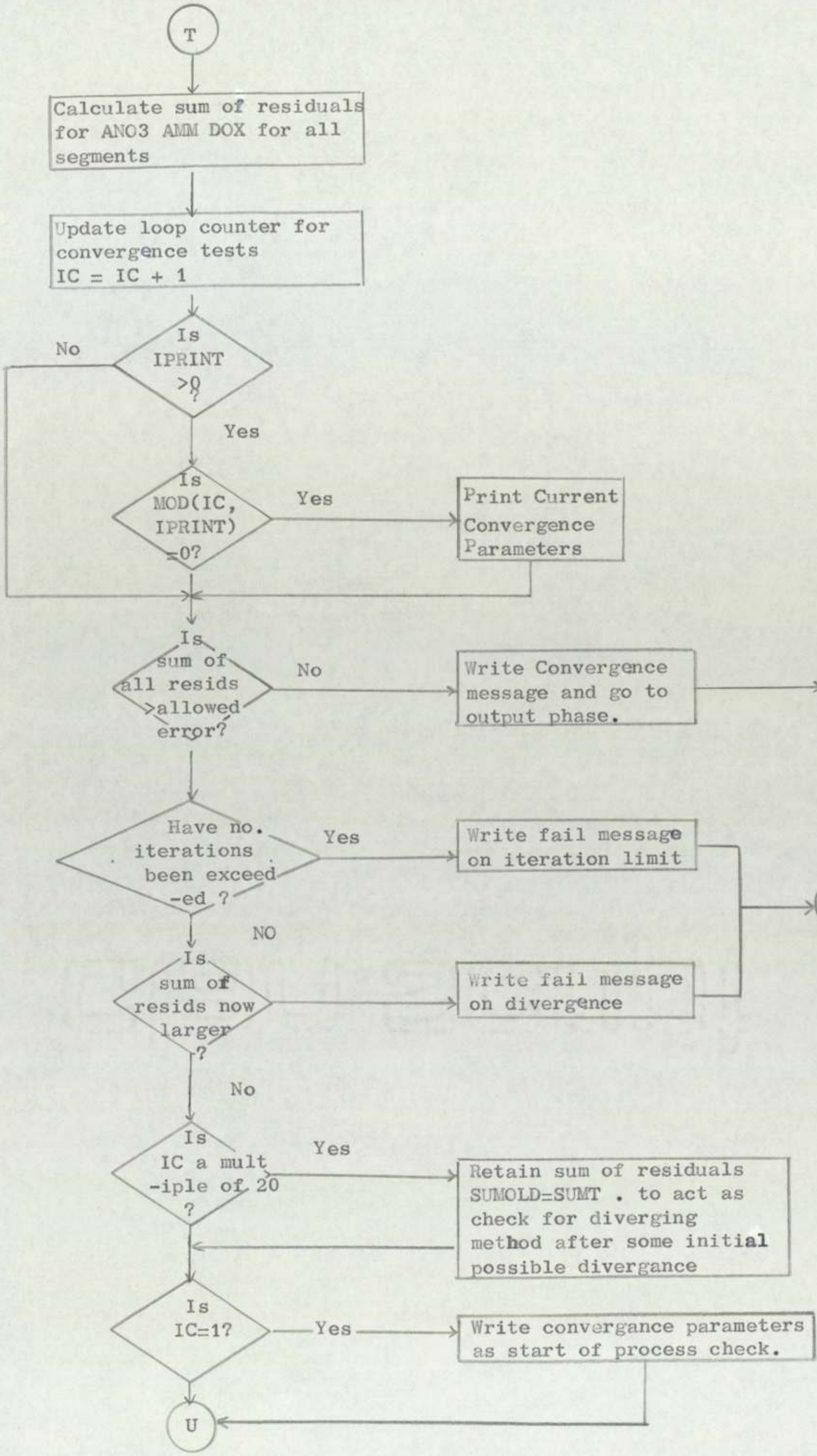


H
J=1, loop to look at each segment in relation to outfalls sphere of influence
Loop to consider the weighting of outfall i against the spatial distribution of the segments.
L
Segment runs from J to J+1
Is Upstream limit>down stream boundary ?
Yes
K
No
Does sphere of influence fall within segment at all?
Is downstream limit<upstream segment boundary ?
Yes
K
No
Is the sphere of influence wholly in the current segment ?
Yes
Weighting of 1 .0
No
Is upstream limit in the segment?
Yes
Weight=(downstream seg-upstream outfall pos.)/TTEX
No
Is downstream limit in the current segment ?
Yes
Weight=(downstream-upstream seg.bound)/TTEX
Is it tide locked at all ?
Yes
Weight reset to zero.
Is segment wholly in area of influence ?
No
Error as mutually exclusive condition here.
STOP 918
Yes
Weight = length of segment / TTEX
I

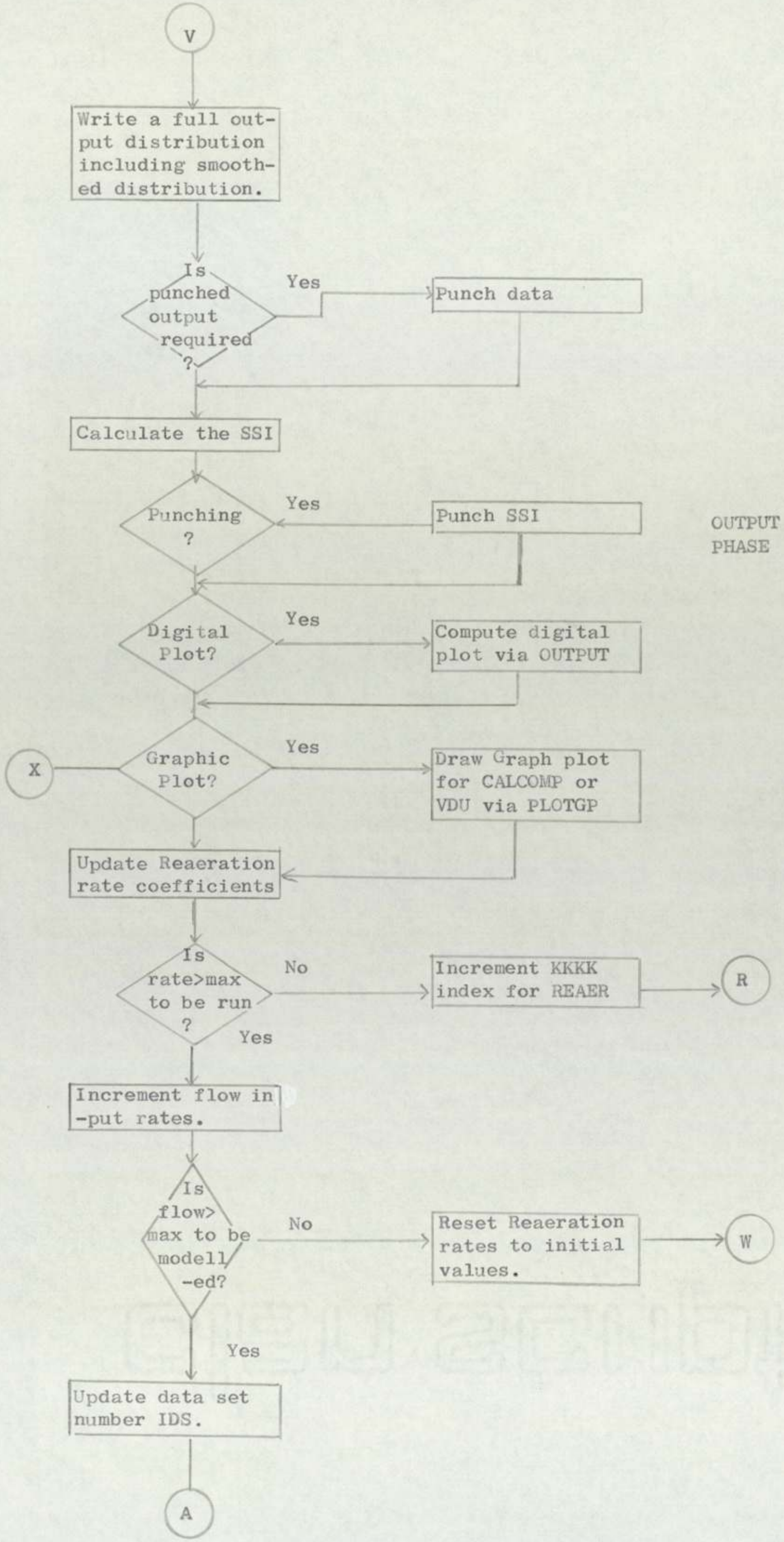

I
Is this seg =IFLSG ?
Yes
Is the out fall tide locked ?
Yes
Modify weight to dump the tidelocked load into the segment
Add the calculated weight to the total weights that have been calculated for this one outfall (TW)
Add Outfalls Discharge to Segment loads
Add the discharge of each load to this segment from the current outfall to the total load discharge of the current segment.Arrays FC SC FN SN AMM ANO3 DOX and fresh water flows.
Print weighting data if flag set to do so
J = J+1
K
More segments left ?
Yes
L
No
Is load data to be printed ?
Yes
Save loads in work Arrays.
No
End of Outfall Distribution
Write outfall data summary and optional weighting checking.
I = I + 1
More outfalls to be looked at? NOUTF
Yes
G
No
Optional print of loads of each outfall.
Print of loads per Segment.
M


M
Is Input Mode = 3?
Yes
This run was only to test inputs and precalculations Return to initial read.
N
Sum the loads per component per seg -ment and save in array W1.
Cross check on loadings
Calculate % diff. between W1-Cumulative load per segment W -Cumulative load per outfall
Report this % diff as it is the amount lost to sea due to tidal excursion effects.
W
Fresh Water flow at tidal limit =FLOWS* FACTOR(mass units)
Calculate FWF's
For each segment, add F.W.Discharge to input from last segment
Is Runoff to be estimated ?
Yes
Call RUNOFF function to modify the appropiate functions and arrays.
No
Calculate Mixing Coefficients.If two adjacent salinities are equal,mixing=0
Calculate Mixing, theoretical max. DO levels and rates
Calculate level of DO required for 100% saturation using DOXSAT function.
Calculate the rate constants for each process included in the model,using TEMP and standard publish ed rates.
P


P
Is this the first data set to be run ?
Yes
Calculate an approximation for the solution using weighted boundary conditions
No
Use the previously computed solution as an estimate to start current one.
Optional Printing of some internally generated arrays.
Compute mixing in and out of segments
For each component, Calculate loss due to rates in segment and mixing out of it
Optional print of the above quantities via routine EXMINE
Solve for non-interacting components via subroutine MATSET : FC SC FN SN via band matrix method.
Set DO Loop for varying re-aeration KKKK = 1
R
Clear arrays U W BOB RDOX Define RDOX
Optional print of REAER and RDOX
U
Set all the residuals to zero
Set I=2 loop for segments
Calculate updated estimate of AMM ANO3 DOX as per algorithm
Calculate residuals between this and last solution.
Add residuals from this segment to total resid.
Increment segment index
Is I>NEWN ?
No
Yes
T



T
Calculate sum of residuals for ANO3 AMM DOX for all segments
Update loop counter for convergence tests IC = IC + 1
Is IPRINT >0 ?
No
Yes
Is MOD(IC, IPRINT) =0?
Yes
Print Current Convergence Parameters
Is sum of all resids >allowed error?
No
Write Convergence message and go to output phase.
V
Have no. iterations been exceed -ed ?
Yes
Write fail message on iteration limit
X
NO
Is sum of resids now larger ?
Write fail message on divergence
No
Is IC a mult -iple of 20 ?
Yes
Retain sum of residuals SUMOLD=SUMT . to act as check for diverging method after some initial possible divergance
Is IC=1?
Yes
Write convergance parameters as start of process check.
U


V
Write a full output distribution including smoothed distribution.
Is punched output required ?
Yes
Punch data
Calculate the SSI
Punching ?
Yes
Punch SSI
OUTPUT PHASE
Digital Plot?
Yes
Compute digital plot via OUTPUT
X
Graphic Plot?
Yes
Draw Graph plot for CALCOMP or VDU via PLOTGP
Update Reaeration rate coefficients
Is rate>max to be run ?
No
Increment KKKK index for REAER
R
Yes
Increment flow in -put rates.
Is flow> max to be modell -ed?
No
Reset Reaeration rates to initial values.
W
Yes
Update data set number IDS.
A

E.4 ALGORITHMIC DETAILS OF THE STEADY STATE MODEL PROGRAM

START

The initial 70 statements are essentiallty comment cards and type declarations. All real arrays except TT are of identical lengths - at least NSEG + 2 words long. All arrays are single precision. Then the variable IDS is set to unity to utilise the reference number of the data set being read in.

199 READ

Reads in 2 lines of heading input through temporary array TT. If the last date has been processed, the control jumps to statement 10001 to go to normal EOJ.

The header is 4 lines long and read in 2 lines at a time and printed to NOUT under format 3ØØ.

READ (IN, 400) ....

This is a long read in statement with the basic data parameters. There are 8 continuations to this read statement. The variables are read in as described in the section in data input.

Now 4 variables are calculated which are 1.0 - (acceleration parameter). The next batch of comment cards describe the working of array PRINTF and SMOOTH.

400 FORMAT....

Is the read in format for the basic data parameters.

WRITE (NOUT, 500)

Prints out the essential data read in by the previous read statement. Then a table is printed out of boundary values for the various constituents. This uses format 600.

READ (IN,700)....

The arrays SMOOTH & PRINTF are read in under format 700 and printed under 701.

CALL IN SEG....

The routine INPUT is called to read in segment boundaries. The final sea boundary is set to a high value internally by using the last read in value and doubling it.

Then the Reaeration, Volumes and Surface areas are read in through routine INPUT.

DO6100 I = ...

This is a units conversion loop and runs from i = 2 to NSEG + 1. The order of computation is

V_i (NEW) = V_i (OLD) * VACTOR

SA_i (NEW) = SA_i (OLD)* FACTSA

$DEPTH_i$ = V_i (NEW) / SA_i(NEW)

V_i = V_i (NEW) * FCTRL

where the subscript OLD and NEW occupy the same words in store, i.e., they are in the FORTRAN IV sense equivalent quantities. These arrays are then examined for dumping and smoothing.

C INPUT SAL....

Calls to routine INPUT now reads in

A Salinities from 1 to NSEG + 2

B Temperatures from 2 to NSEG + 1

C Ordinate of tidal excursion from 1 to NOL

D Downstream Tidal Excursion from 1 to NOL

E Upstream Tidal Excursion from 1 to NOL

IF PRINTF(30)....

Here an optional call to TEXRE allows for correction of surface areas due to tidal excursion.

C READ WHICH.....

The array TIDEL is read in now under format 7ØØ and printed under format 1Ø99 & 11ØØ.

DO 1200 I = 2, NEWN

This loop clears the cumulative addition arrays FCADD, SCADD, FNADD, SNADD, AMMADD, ANO3AD, FFFLOW (addition of fresh water flows) and DOXADD. The work array W is cleared from 1 to NSEG.

C PUNCH....

Any punched output of input data is now handled depending on value of PRINTF(37)

C READ IN OPTION....

The mode of input is read in under format 8ØØ. In the version being documented, only three options are available. If mode is not equal to one normal data input occurs. The program computes dispersion etc., up and including paragraph 98Ø1 BOB(I) = PSTN. Then control returns to statement 5Ø92 which is a dummy statement prior to reading in REAR array.

If mode = 3 control returns to input REAR at a stage prior to modelling but after all inputs have been processed. It is a data validation made of input.

If mode = 1 then zero loads are assumed.

DO 9ØØ I = 1, NOUTF

This is the loop to read in each outfalls loads, position, flows and evaluate its additive contribution to each segment of the estuary.

Read 91Ø....

This reads the name of an outfall, its position, the daily rate of discharge and loads.

DO 6Ø1Ø J = 1, NEWN

This loop ascertains the segment index where the current outfall is situated.

6010 CONTINUE

IFLSG is set to the required index.

6017 W4 (7) = ...

This statement converts the input D.O. load of a discharge from percentage saturation to load in pounds per day. Then the rate of discharge is stored in W4 (8), the work array for this section.

An IF statement now asks whether this is a normal run or a contingency test (i.e. if NOUTF = 4). If the run is not a normal run control jumps to 914.

DO 913 K = ...

The array WW is filled from element 1 to 6 with the discharge loads in 000's of pounds per day.

PRINT 912

This prints the input data for the outfall with the transformed loads for D.O. using format 912. Each outfall gives 3 lines of print.

914 UB = PSTN - ....

This statement and the next one (916 DB = ...) compute the upstream and downstream limits of the tidal excursion from the point PSTN, which is where the outfall discharges. Comment cards explain the legend.

RINTER is used here and is an interpolation routine in this case, of the 1st order (i.e., linear).

Then the total tidal excursion, TTEX can be evaluated. If this is zero (or very small) it is set to a large default value which should run the rest of the data as if it were a zero load run. The variable TW is set to zero to hold the cumulative weights evaluated in the next level loop.

DO 98ØØ J = 1, NEWN

For a given discharge with a now known tidal excursion:

There are 6 IF statements here that cover all possibilities of how UB and DB could be orientated with respect to the segment boundaries.

The first two are

$$\text{if distan}_{j+1} > \text{UB}$$

and

$$\text{if distan}_{j} > \text{DB}$$

then the current discharge does not affect the load on segment j (with array index j+1) and the process jumps out of the loop with zero weighting.

The third possibility is a tidal excursion so small that the outfall only discharges to one segment. In this case the segment weight is 1.

The next two possibilities are where either UB or DB falls within the segment boundary in which case a partial weighting factor has to be determined. The final IF statement deals with the case where both UB and DB are outside the segment but the segment boundaries are within the range UB to DB i.e.

$$UB < DISTAN_i < DISTAN_{i+1} < DB$$

If none of these if statement branch then an error dump is given consisting of current outfall position and array W4, followed by error flag - STOP 918

919 WGHT = ...

The next few statements calculate weighting factors for the segment with respect to the outfall.

922 IFLSEG = ...

This section deals with tidelocking of the discharges. If the discharges are freeflowing no action is taken and control jumps to 980. If the discharge is tidelocked two modifications are made dependent on where the current segment is in relation to the outfall.

1. If the segment is downstream of the outfall the weight is set to zero.
2. If the segment is that into which the outfall discharges, the weight for that segment is so modified as to cause all that part of the load that would normally fall in the downstream part of the tidal excursion to be deposited in the one segment that contains the outfall.

938 CONTINUE

The weighting of a particular segment has now been established and it is added to TW, a variable in which total weights are summed and provide a check total later.

Statements 939 FC ADD.... to 946 FFLOW ... add the weighted load of an outfall to segment with index JP1. If the weight is zero this section is by-passed for efficiency by a previous if statement.

IF (PRINTF (6....

If the above flag is true, the program prints the segment number and its load weighting factor for each outfall. This option can be heavy on LP output if a large number of outfalls or segments are involved.

9800 CONTINUE

End of load distributing Do loop.

98Ø1 BOB (I) = PSTN

The position of each outfall is saved in array BOB to be printed later. The element PRINTF (14) is tested and if TRUE then the loads of each outfall are saved in temporary storage locations in arrays RFC, RSC, RFN, RSN, RDOX, RNO3 and RAMM.

Then for each outfall the following is printed:

Up & Downstream boundaries

Total Tidal Excursion

Segment in which outfall occurs

Whether tidelocked (T) or freeflowing (F).

If PRINTF (6) is true then the sum of weights for each outfall is printed. This should usually be 1.0, but when an outfall is near the lower end of an estuary, the tidal excursion may be sufficient to remove part of the load from the estuary system. In this case TW < 1.0. In no case should TW > 1.0.

9ØØ CONTINUE

End of outfalls read in loop. The loads from all NOUTF outfalls are now processed and the cumulative loads per segment are in arrays FCADD, SCADD, FNADD, SNADD, AMMADD, ANO3AD & DOXADD.

IF (MODE. EQ. 3) ....

This option now branches back to reading in REAR array.

Now the Sum totals of loads in the arrays suffixed.. ADD are calculated and compared to the sum of loads from individual outfalls.

Discrepancy in % terms is calculated and this is the amount discharged effectively to the open sea and so removed from the estuarine system.

CALCULATE FRESH ....

The loop for varying fresh water flows commences here.

The 3 statements subsequent to the above comment card convert fresh water flows at the head of the estuary to flow in lbs per day, and fills the array FFLOW with the cumulative fresh water flow in lbs per day due to F.W.F. at head of estuary plus dispersed input flows from discharges and tributaries. The latter is calculated in the tidal excursion distribution section. The routine EXaMINE is then called for an optional print dump dependent on variable PRINTF (25). Also, an optional correction for Land Runoff is made to FFLOW at this stage.

CALCULATION OF MIXING....

The DO 1999 loop calculates the estuarine mixing coefficients from the measured salinity distribution through the formula

$$F_i = \frac{Q_i \quad (S_i - S_1)}{(S_{i+1} - S_i)}$$

If $S_{i+1} = S_i$ then F_i is set to zero as a default value. Optional printing of F occurs through setting PRINTF (24).

CALL SATURATED ....

The loop DO 6300 uses the salinity and temperature distributions to evaluate the saturation level of D.O. in the segments. These values are used to calculate deficits and leter useful in % saturation calculations. The boundary values have to be approximated using the nearest data available.

Optional printing follows through setting PRINTF (26).

C TEMPERATURE - DEPENDENCE

The loop DO 6400 sets up two kinds of arrays. Arrays AKCT, AKNT, AKNO3T and AKAMMT are the temperature corrected rate constants.

Also, if the data set currently being analysed is the first data set (i.e., IDS = 1), an initial guess at the solution to NH_3 - NO_3 - D.O. distributions has to be made using

$$D_i = D_1 \cdot \left(1 - \frac{i}{NSEG + 1}\right) + D_{NSEG + 2} \cdot \left(\frac{i}{NSEG + 1}\right)$$

This proved to be a better guess than

$$D_i = (D_1 + D_{NSEG + 2}) / 2$$

and so saved considerable iteration time.

If the data set is not the first data set to be processed in the run, then the initial guess for the run is taken to be the final solution for the previous data set. This again saves considerable time on the iteration stage.

<u>64ØØ CONT....</u>

End of do loop.

<u>64Ø1 CALL</u> to <u>64Ø7 CALL</u>

These statements are all calls to routine EXaMINE for optional printing of the arrays defined above in DO 64ØØ in the order AKCT, AKNT, AKNO3T, AKAMMT (dependent on value of PRINTF (27)), then AMM, ANO3 and DOX (dependent on value of PRINTF (28)).

<u>COMPUTE X</u>

The loop DO 65ØØ computes the mixing volumes of the segments.

$$X_{i-1} = F_{i-1} + Q_{i-1}$$

i.e., what arrives in segment i from segment i-1

$$XX_i = F_i + f_{i-1} + Q_i$$

i.e., what leaves segment i.

<u>DO 66ØØ I = 2, NEWN</u>

This loop evaluates arrays used as dividing factors in the distribution iteration or direct solution. The factors are all derived from the equation

$$RD_i = XX_i + V_i \cdot AK_i$$

$= F_i + F_{i-1} + Q_i + V_i \cdot AK_i$

= 'Flow out' + 'decay mass'

where RD_i is respectively RFC, RSG, RFN, RSN, RNO3 and RAMM. AK_i is respectively AKCT, AKCT/5 (for the slow carbon rate), AKNT, AKNT/5 (for the slow nitrogen rate), RKNO3 and AKAMMT.

After the loop a few boundary values are set to 10^{50} for printing purposes.

Then the arrays X and XX can be printed by setting PRINTF (29) to .TRUE.

C EVALUATE STEADY STATE SOLUTION

The statement 6610, 6620, 6630, 6640 each define one call to the subroutine MATSET to directly solve the distributions of Fast and Slow Carbon and Nitrogen as they are non-interacting constituents and can be solved using a simple band matrix method, slightly modified in the indexing that is normally associated with FORTRAN IV to take account of the fact that the indeces run from 2 to NSEG + 1. The routine is described later.

The equation and parameters transmitted to the subroutine are sufficient to solve the resulting band matrix in routine MATSET.

If this section is completed without obvious errors, a message is output:

FC-SC-FN-SN SOLUTIONS COMPUTED

and the time taken is dumped to the LP.

DO 9ØØØ KKKK = 1, NODELS

This is the last major loop which calculates the distribution of the interacting substances for varying reaeration rates. As the non-interacting substances are independent of the reaeration rates it is an efficient means of doing several runs in one program pass and only this loop is repeated. Also, the final solution for one cycle can be used as the initial estimate of the next solution and the iterations required to converge the problem are small.

DO 7ØØ5

This loop clears the arrays U, W and BOB which had earlier been utilized as work arrays. The array RDOX is also defined here as it has to be redefined for every cycle as it is dependent on the reaeration coefficient.

The new values of RDOX can be printed by setting PRINTF (3Ø). The array of reaeration coefficients REAR can be printed by setting PRINTF (31).

CONVERGENCE FOR INTERACTING COMPOUNDS

From this statement to 73ØØ IF (SUMT ....) is the iterative solution loop for interacting compounds in the estuary.

IC, the number of iterative cycles completed, is set to zero. SUMOLD is set to a high value (10^{50} as this has already been used), and SUM1, SUM2, SUM3, SUM4 and SUM5 are set to zero. These SUMn variables are the cumulative differences between successive approximations i.e.,

$$SUMn = \sum_{i=2}^{NSEG+1} G_i^{k+1} - G_i^k$$

where the superscript is the iteration cycle number. The first three deal with ammonia, nitrate and D.O. and the last two deal with the case of restricted oxygen supply in the estuary.

DO 72ØØ I = 2, NEWN

This is the do loop to update each segments solution for the different interactive components. Some variables are used several times and are put into a temporary location.

VTMP = V(I)

FTMP = F(I)

XTMP = X(I-1)

C AMMONIA

The solution of this distribution is again based on the resulting equation in the theory section. This equation can be rewritten as

$(F_{i-1} - F_i). C_i + F_i$

$(Q_{i-1} + F_{i-1}). C_{i-1} + (F_{i-1} - F_i - Q_i - K_i \Delta V_i)$

$C_i + F_i C_{i+} + M_i = 0$

Now $(Q_{i-1} + F_{i-1}) = X_{i-1}$

Also $(F_{i-1} - F_i - Q_i - K_i \Delta V_i) = RAMM_i$

A further term is introduced, of magnitude U_i/(oxygen equivalent of ammonia), to allow for the case of restricted oxidation.

A similar equation results for the nitrate distribution solution, with an additional term involving the oxygen equivalent of nitrate. The solution for dissolved oxygen content is essentially similar.

The array BOB contains the maximum amount of oxygen in the segment from mixing reaeration and also addition of free oxygen. From this value, the oxygen consuming terms are subtracted, i.e., mainly carbonaceous oxidation and ammonia breaking down.

C TEST FOR

This tests the trigger DOXMIN on the type of chemical breakdown considered to be taking place. If DO_i DOXMIN then the restricted process takes place.

36 A = OXA

From this statement to the one preceding 37 U (I) = Ø., the chemical process is limited. DO_i is set to the trigger point. Arrays U and W are used to transmit data on the modified distribution of ammonia and nitrate to the next cycle.

37 U (I) = Ø

As the limiting process has not come into operation, set U and W to zero.

42 IF (U(I))

If U(I) O reset both U and W to zero as in 37.

44 Z = Z - DOX (I)

The three residuals for D.O. and the two restricted processes are now evaluated and cumulatively added on to variables SUM3, SUM4 and SUM5.

7200 CONTINUE

End of segments do loop.

SUMT is the total difference between iteration n and n-1 for all segments and for all constituents.

$$\text{SUMT}= \sum_{I=2}^{NSEG+1} \sum_{J=1}^{NOSUBS} C(I,J)^{Nth} - C(I,J)^{(N-1)^{th}}$$

The iteration counter IC is updated and the print option variable IPRINT tested. If it is zero, or negative, the intermediate history of convergence option is not tested. If IPRINT is greater than zero, the test

$$\text{MOD} \ \frac{IC}{IPRINT} = \emptyset \ ?$$

is applied. If it is satisfied, the history of convergence is printed.

7300 IF (SUMT - ERROR)

This is the test to determine if the overall residual is less than the permitted error term ERROR.

77ØØ WRITE (NOUT, ...

An end of Convergence Parameter set is printed to NOUT. The output consists of IC, the Cumulative SUMn per constituent and the total SUMT. Control then jumps to 7900 CONTINUE

70330 IF (IC. GT. MAXLOO)

Here the convergence has not converged sufficiently. A series of tests follow:-

1) If IC > MAXLOO means that the iterations exceed the maximum number permitted. In this case control jumps to 7799.

2) If SUMOLD > SUMT means that the process is a divergent or stationary one as opposed to a rapidly converging one. In this case control goes to 7801.

3) If mod (IC, 20) = Ø reset SUMOLD to the new SUMT. This could be done every iteration but in practice the most efficient solutions have an initial divergence period.

4) If this is the first loop, print a set of iteration parameters.

7799 WRITE (....

Print iteration parameters and a failure message.

78Ø1 WRITE C

If control is here, the iterative process has failed. A message indicating the diverging nature of the solution in output and control goes to the failure branch.

7900 CONTINUE

Iterative process has converged.

The final distributions are now printed in:

7Ø77 DO 7086 I = 1.

This converts D.O. values of mg/l into % saturation. It also evaluates the minimum D.O. value in the estuary and the mean distances of each evaluated distribution. Smoothing is carried out on the D.O. function and printed later.

If a copy of the predictions are required, a summary of the predicitions are written to data set 98 for retention.

DO 75ØØ I = 1.

Prints the results with good format control.

RI = Ø

From this statement to WRITE (NOUT, 7088) the Sag Severity Index is calculated and printed below the complete distributions. If the output is to be saved, a copy of this is written to data set 98 (Card Punch).

8900 CONTINUE

If the print flag PRINTF (33) is set to .TRUE., then a one page graphical distribution of the D.O. in % sat. terms will be printed through subroutine OUTPUT.

If PRINTF(34) £3 set is set then plot options will give two types of graph plotter output.

DO 8999 KK = 2, NEWN

This loop updates the reaeration coefficients in array REAR.

9000 CONTINUE

END OF INTERACTIVE COMPONENTS

Repeat process or call time.

If repeat, back to DO 9000. (New REAR data)

Increase fresh water flow at top of estuary (FLOWING rement)

If less than maximum (FLOWMAX) then repeat the cycle for a new fresh water flow.

If no further computation required, close any open plot files and end with message STOP1001

E.5 Routine EXMINE

Function : Optional printing and or smoothing of input data from subroutine INPUT

System Functions : I/O

Calling Statement : CALL EXMINE(X,I1,I2,A1,A2,L1,L2)

Description

The routine is to consider the elements I1 to I2 of the single precision array X. A1 and A2 are alphanumeric descriptions of the data in the array and will be printed out.L1 and L2 are logical flags.

If logical flag

L1 = .TRUE. the array elements will be passed to SMMTH for processing before printing.

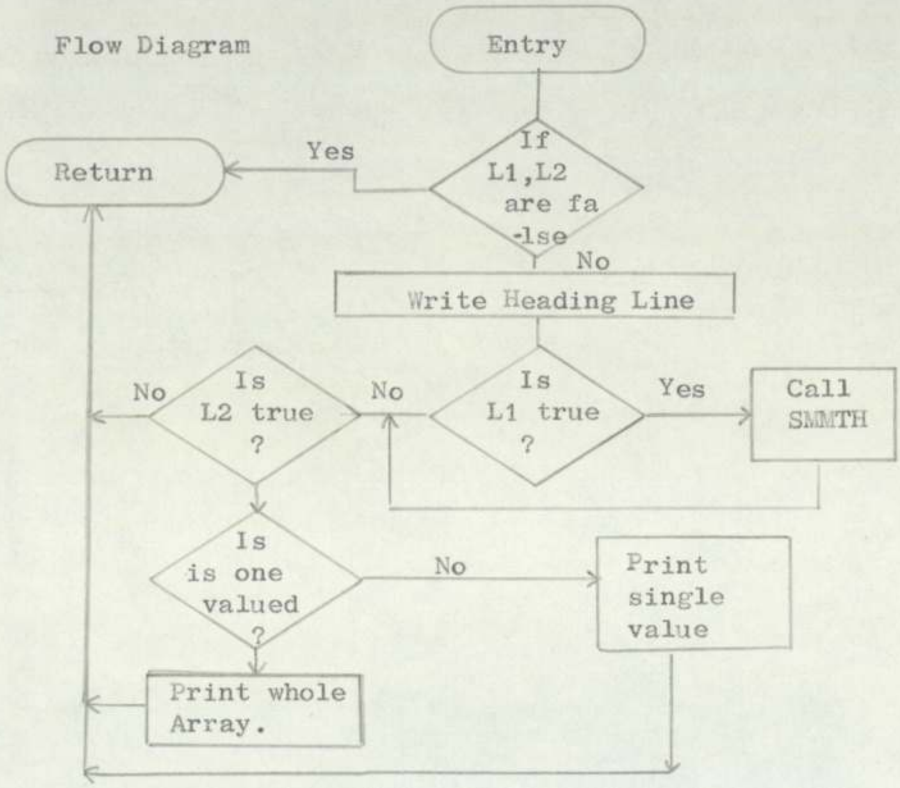
L1 = .FALSE. array elements are not smoothed.

L2 = .TRUE. the array elements are printed to peripheral channel NOUTF.

L2 = .FALSE. the array is not printed,but a print supression message is sent to channel NOUTF.

On entry,a line outputting the name of the data array and limits to be considered is sent to NOUTF. If a full print is required, the routine tests to see if X is a single valued array , so reducing lines actually printed. Else 6 values per line are output with accompanying indeces.

Flow Diagram



E.6 Routine MATSET

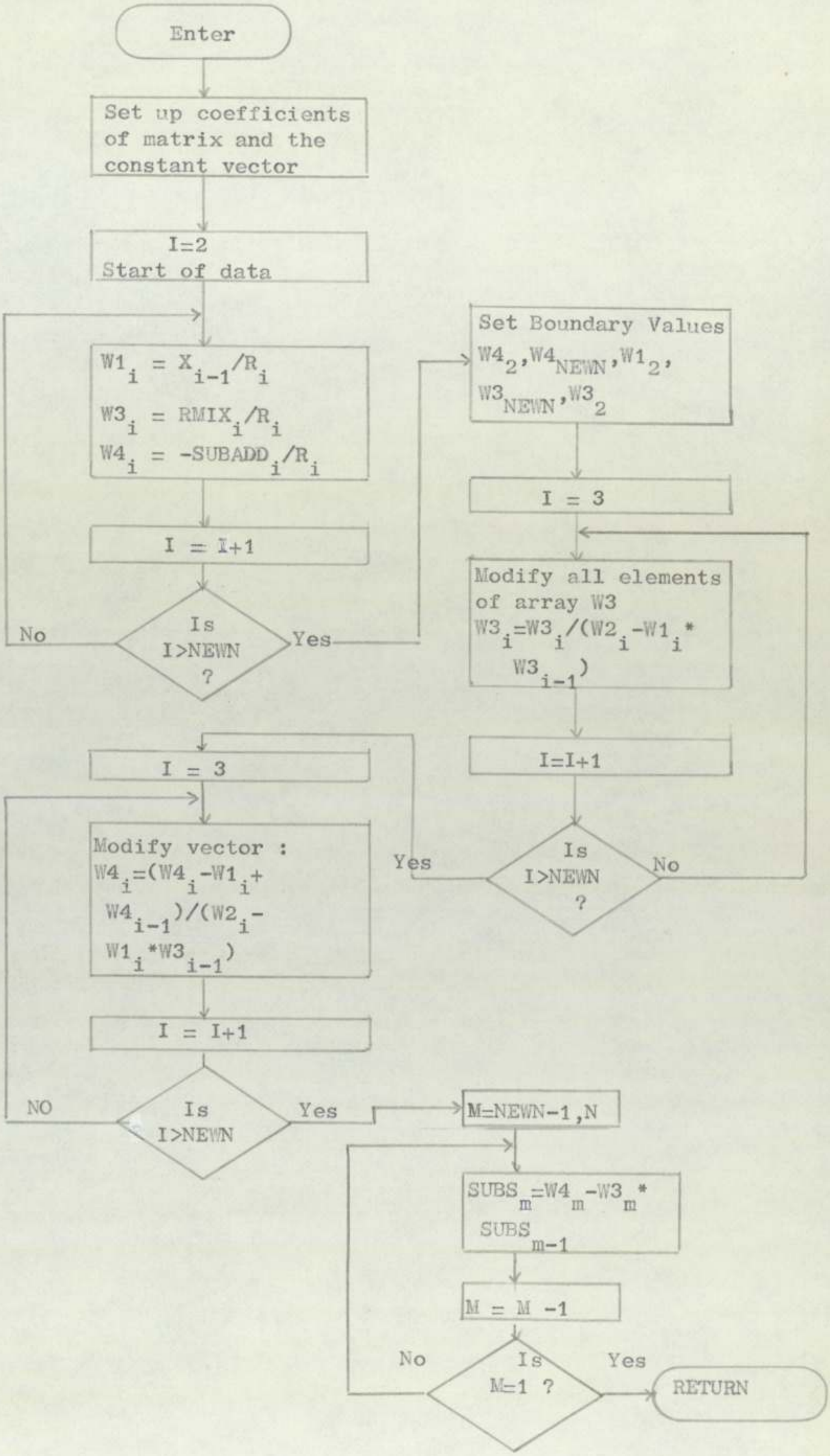
Function : To solve a band matrix set up in SSMOD for the non-interacting components of the model.

Calling Statement : CALL MATSET(X,SUBS,RMIX,SUBADD,R,NEWN,W1, W3,W4)

Description

X,RMIX,SUBADD,R are all arrays required to set up the elements of the matrix and are peculiar to the equations derived for the distributions of non-interacting components in the main model routine.SUBS is the holding array for the returned solution. W1,W3,W4 are work arrays and are all overwritten initially so need not be cleaned upon entry.W2 is implicit as it is a constant.All arrays are single precision. The routine solves $A.x = b$ by $A^{-1}.(A.x) = A^{-1}.b \Rightarrow x = A^{-1}.b$ for the specific case of a 3 element band matrix. The elements below the diagonal are in W1, the diagonal element is 1 , the above diagonal element is in W3,and the constant b in array W4.
The first DO-loop sets up the elements in the work arrays.By dividing each array by R(i), the diagonal element is reduced to unity.Some special values have to be inserted into end terms from boundary values The method is a straightforward elimination/back substitution(ref : Numerical Methods,Buckingham,Pitman 1966,chapter 11). The DO 1007 loop modifies the above diagonal elements,the DO 1010 the constant. Then an explicit loop around statement 1013 completes the back-substitution.The process terminates at row 2 as the indexing method used in the main model requires this.

Flow Diagram



E.7 Routine OUTPUT

Function : To produce a digital plot of the DO profiles produced by SSMOD

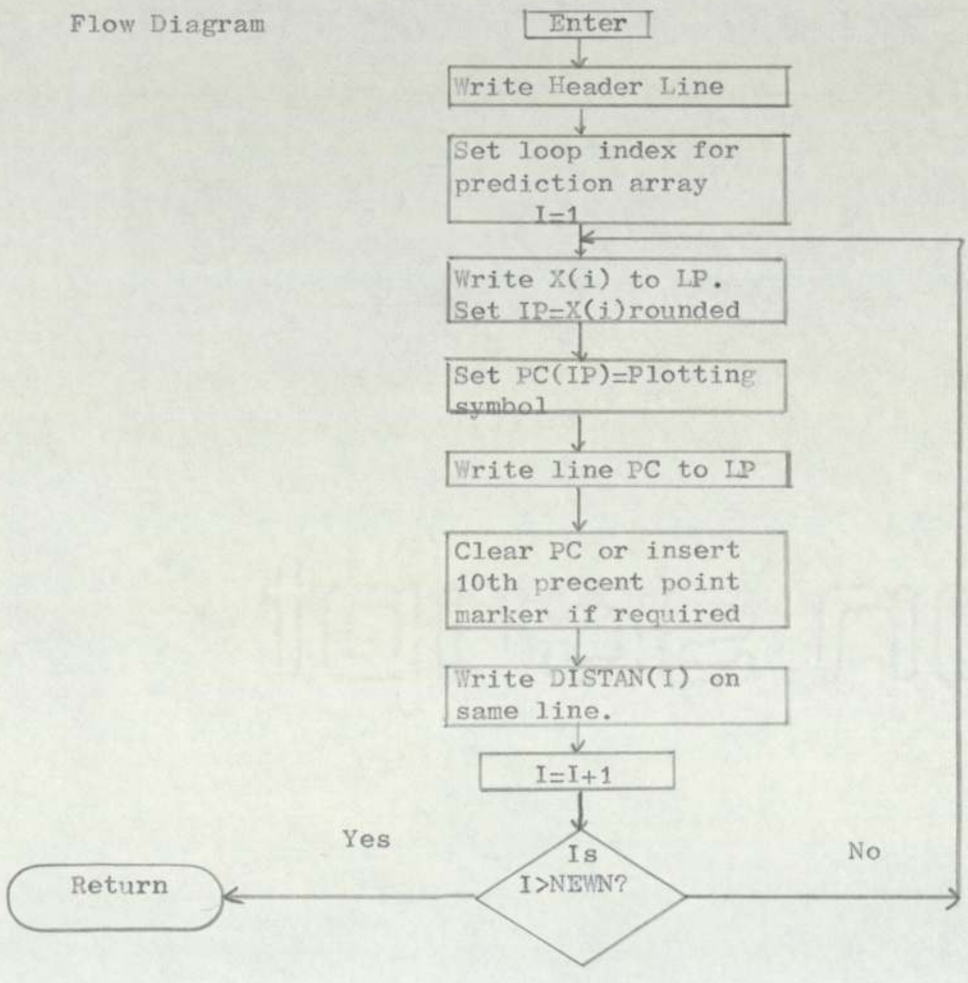
System Functions : I/O

Calling Statement : CALL OUTPUT(X,DISTAN,NEWN)

Description

The main output from the model is a prediction of dissolved oxygen levels vs. spatial points in the estuary.To facilitate interpretation, this is output on a one page digital plot at low mill-time overheads as it is a specifically written and thus fast routine.X is the array containing the predicted levels,and DISTAN the points at which the levels are predicted. Each line uses 110 characters for the plot,and 20 more for printing of data values.Every 10th precentage point is flagged by a symbol which is printed on all lines except when the prediction coincides with one of these points.Spatial points are orientated down the page.PC is a work array holding the line to be plotted each time.

Flow Diagram



E.7(a) Routine INPUT

Function : To act as input module to the SSMOD Routine

System Functions : Object-time Format,I/O with END and ERR options,DEBUG

User Functions : EXMINE

Calling Statement : CALL INPUT(IN,FMT,I1,I2,A1,A2,L1,L2)

Error Messages : 'END OF DATA ERROR' System I/O has detected end of data before the I/O list has been exhausted

'ERROR IN INPUT DATA' Usually format does not match data layout or vice-versa

Description

Input list legend is :

IN is the reference number of the input peripheral channel.Can be cards,paper tape or magnetic tape currently

FMT is the object time format of the input list,necessary because of differing input peripheral types.

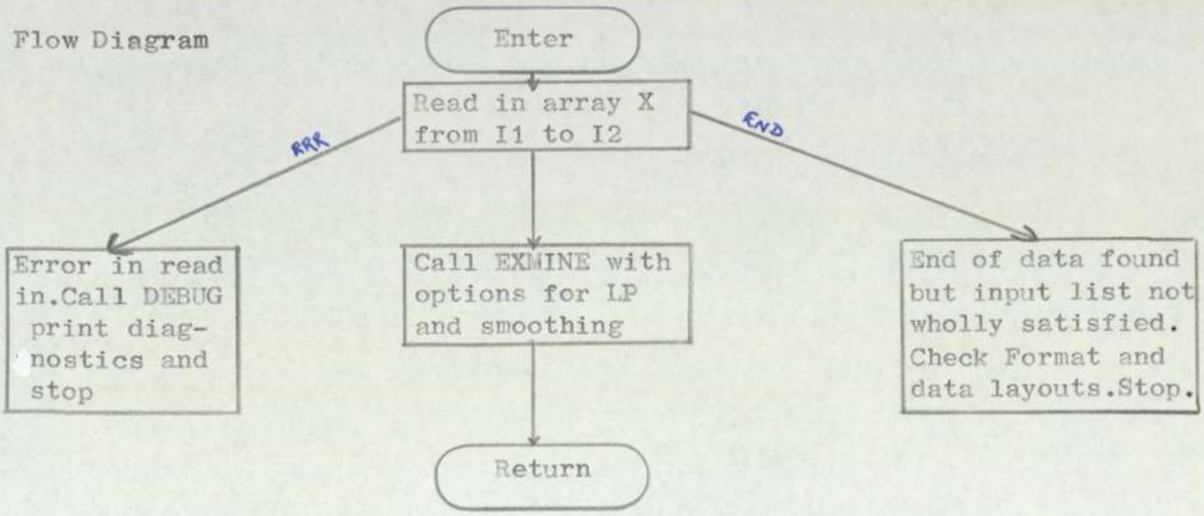
X,I1,I2 X is a single precision array,values to be read into it from elements I1 to I2 inclusive

A1,A2 are alphanumeric labels up to a total of 8 characters for array X

L1,L2 are one byte logical flags for routine EXMINE if called

An attempt to read the data into appropiate locations is made.The system will locate most gross errors, and if any are located,routine DEBUG will provide maximal diagnostic information.Otherwise EXMINE is called to check/smooth.

Flow Diagram



E.8 Routine PLOT

Function : Performs curve plotting on a CALCOMP Plotter

System Function : GRAFOR

Calling Statement: CALL PLOT(S,D,I1,I2,R,RMAX,IPOS)

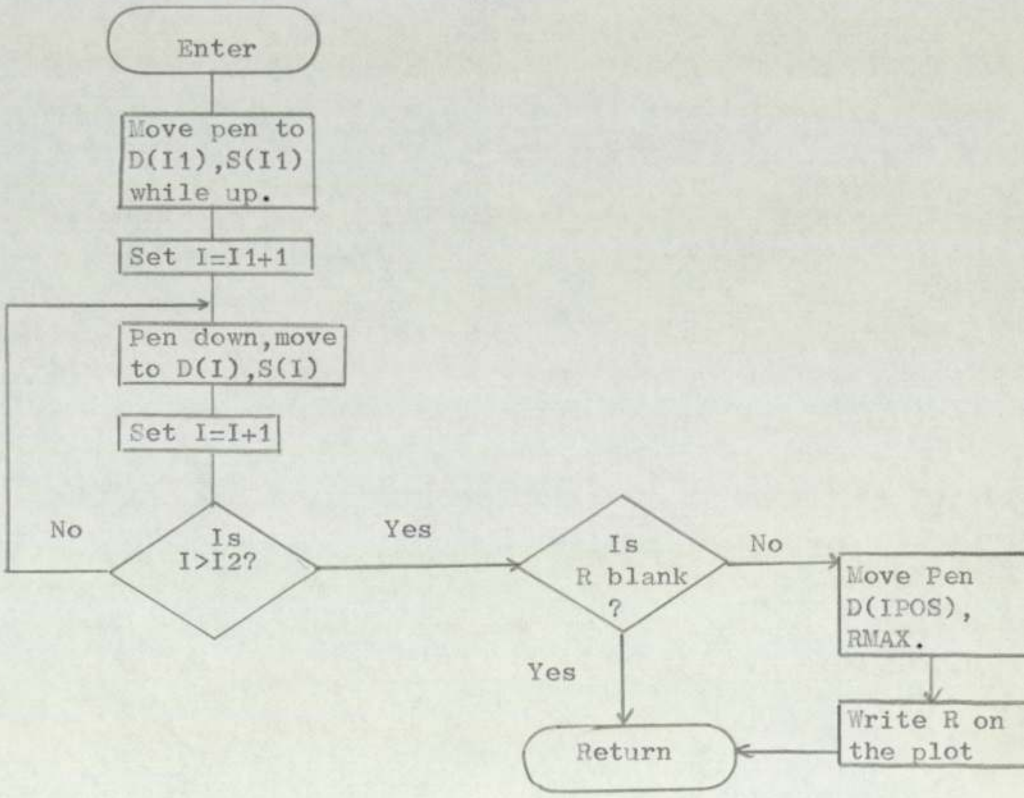
Description

S and D are single precision vectors containing the values to be plotted, and the x-ordinates at which they occur.I1 and I2 mark the inclusive limits of S to be plotted.R is an up to 4 character identifier which appears on the curve.This should be kept as short as possible if mult le curves are put onto one frame.RMAX is the maximum value reached by S and the point where R will appear.D(IPOS) is the spatial co-ordinate where the text appears.

The pen is up and positioned at the initial set of co-ordinates,(D(I1),S(I1).

A Loop the plots all elements with the pen down.Lastly the pen returns to RMAX in the up position to write the text on the line above those co-ordinates.

Flow Diagram



E.9 Routine PLOTGP

Function : To provide CALCOMP graphical output options for the plotting of predictions generated

System Functions : GRAFOR,I/O

User Functions : PLOT,EXMINE

Calling Statement : CALL PLOTGP(NN,FC,SC,FN,SN,AMM,DOX,DOXS,DISTAN,L1,L2,K4,IDS,L3)

Description

Input list legend :

NN	Equivalent to NEWN
FC	Fast Carbon.
SC	Slow Carbon.
FN	Fast Nitrogen.
SN	Slow Nitrogen.
AMM	Ammonia
ANO3	Nitrate
DOX	Dissolved Oxygen
DOXS	Saturation levels for DO
DISTAN	Segment Boundary Array
L1,L2	One byte logical flags,see below
K4	Index of re-aeration cycle currently being run.
IDS	Index of data set currently being run
L3	One byte logical switch for post-run processing of plots.

Logical Options on Entry to PLOTGP

L1 is PRINTF(34) in the executive routine.If .TRUE. , a complete graph is generated for every call,ie for every IDS and every K4 data set.A very expensive option

L2 is PRINTF(35) in the executive routine.If .TRUE. only DO predictions in terms of %sat are plotted.If L1 and L2 are set, then L1 is switched off in preference to L2.

L3 is PRINTF(39) in the executive program.If .TRUE., under MULTIJOB on the ICL4-70,it will cause the plot files generated while it is set to 'on' to be automatically queued for plotting,and deleted when the plot is completed.If set to 'off' the files have to be queued for plotting and erased manually.L3 is transferred through the routines to the operating system via GRAFOR(1,...

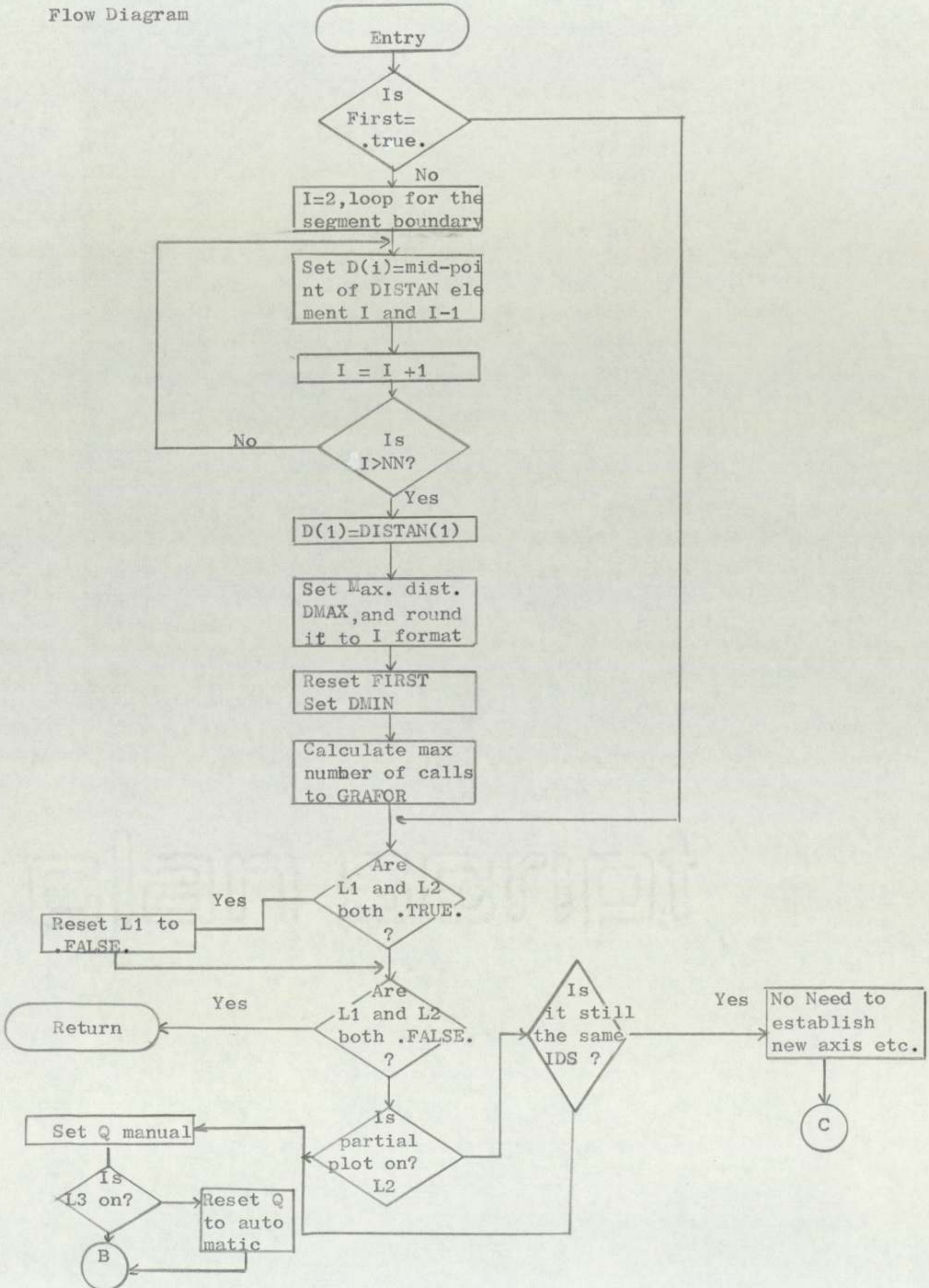
Program Logic

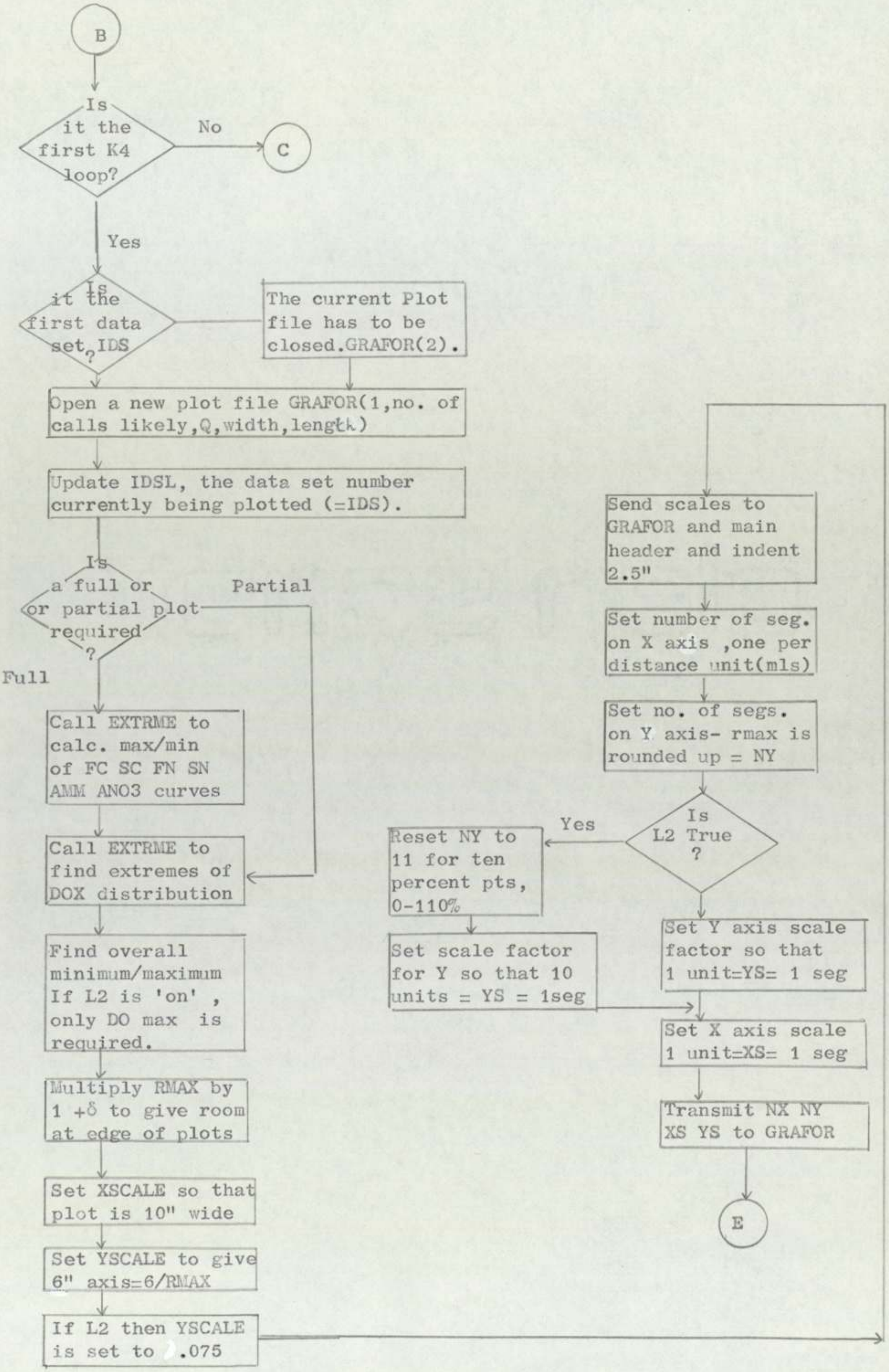
FIRST is a logical flag so the statement prior to 10 IF are only executed on the very first entry to the routine by the executive program.The maximum and minimum distances are extracted from the array DISTAN, and the mid point list in array D.An IF test checks to see that L1 and L2 are not both 'on'.L3 is checked and any current plot files closed prior to opening of the new one.

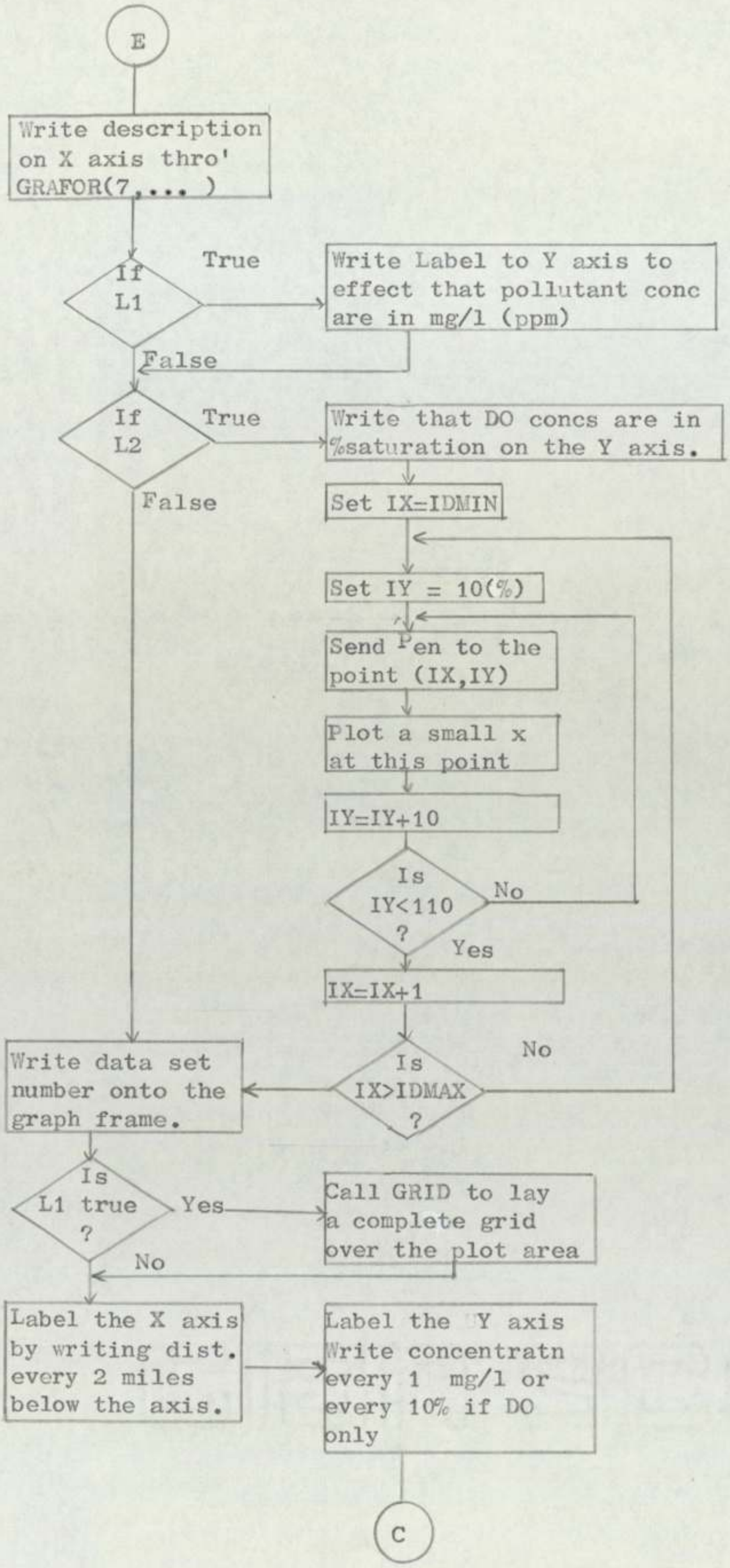
The routine EXTRME is called for each distribution to determine the upper limit for each distribution.The overall maximum value (RMAX) is then found and used to calculate scale factors.Axis are established for x,y axis with reference to plot options choden.Each axis is labelled with comment and a grid of + at intersections is laid over the area if requested.The data set number is written to the east side of the plot for reference.If a new file has been opened during this call of the routine, the non-interacting substances can be plotted via PLOT , as they remain constant during any one loop of K4. The Ammonia,nitrate and DO values are plotted and a short one-line message generated to the LP.Array DOX with DOXS are used to plot %sat levels.The minimum of this curve carries the label.

References Users Guide to the Multijob GRAFOR Routine,Bath University Computer Unit of the South West Universities Computer Network.

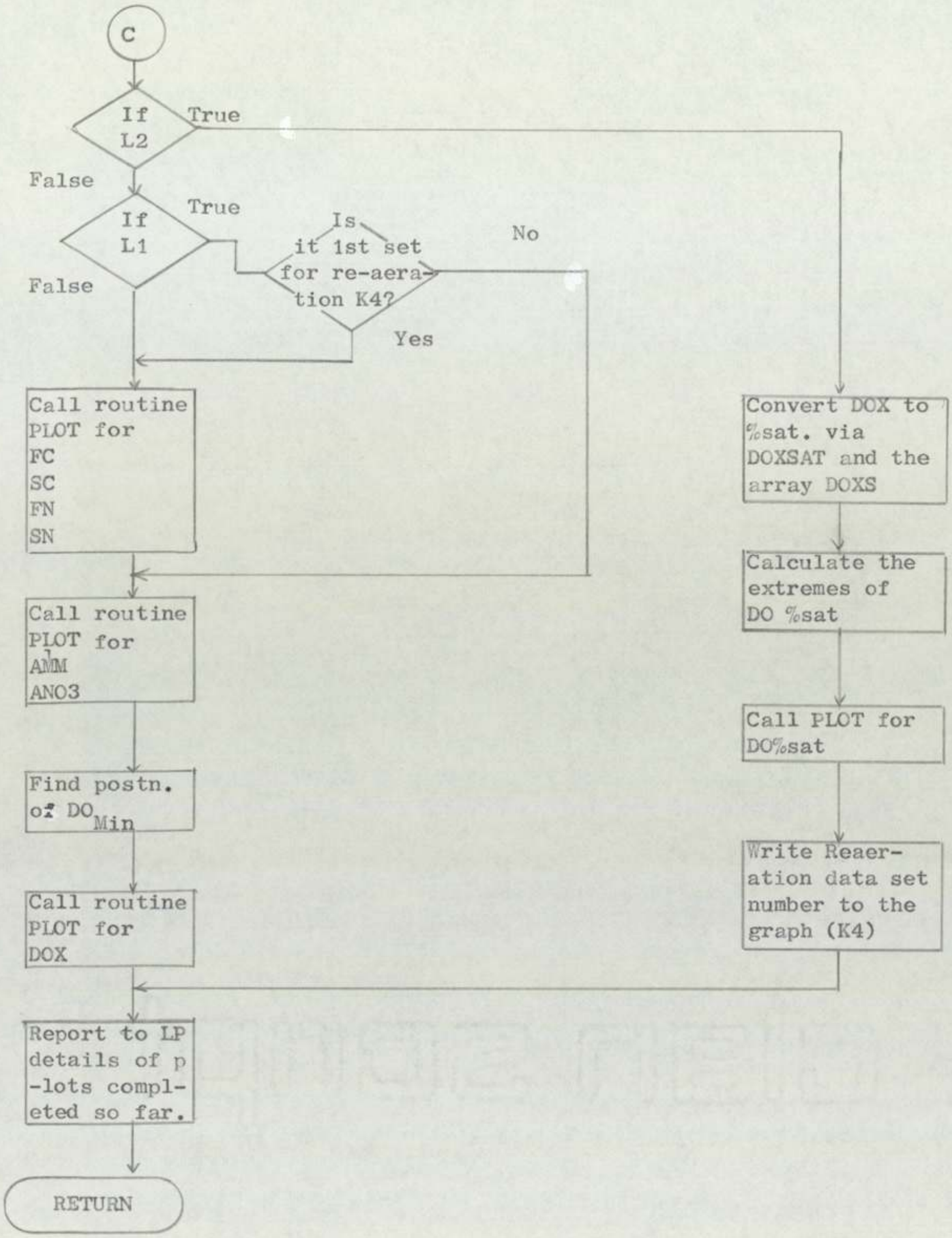
Flow Diagram




B
Is it the first K4 loop?
No
C
Yes
Is it the first data set IDS ?
The current Plot file has to be closed.GRAFOR(2).
Open a new plot file GRAFOR(1,no. of calls likely,Q,width,length)
Update IDSL, the data set number currently being plotted (=IDS).
Is a full or or partial plot required ?
Partial
Full
Call EXTRME to calc. max/min of FC SC FN SN AMM ANO3 curves
Call EXTRME to find extremes of DOX distribution
Find overall minimum/maximum If L2 is 'on' , only DO max is required.
Multiply RMAX by 1 +δ to give room at edge of plots
Set XSCALE so that plot is 10" wide
Set YSCALE to give 6" axis=6/RMAX
If L2 then YSCALE is set to .075
Send scales to GRAFOR and main header and indent 2.5"
Set number of seg. on X axis ,one per distance unit(mls)
Set no. of segs. on Y axis- rmax is rounded up = NY
Is L2 True ?
Yes
Reset NY to 11 for ten percent pts, 0-110%
Set scale factor for Y so that 10 units = YS = 1seg
Set Y axis scale factor so that 1 unit=YS= 1 seg
Set X axis scale 1 unit=XS= 1 seg
Transmit NX NY XS YS to GRAFOR
E



Lay a grid over the plot area.


C
If L2
True
False
If L1
True
Is it 1st set for re-aeration K4?
No
Yes
False
Call routine PLOT for FC SC FN SN
Call routine PLOT for AMM ANO3
Find postn. of DO Min
Call routine PLOT for DOX
Report to LP details of p-lots completed so far.
RETURN
Convert DOX to %sat. via DOXSAT and the array DOXS
Calculate the extremes of DO %sat
Call PLOT for DO%sat
Write Reaeration data set number to the graph (K4)

E.10 Routine RUNOFF

Function : to provide an option for correction for additional water source terms to the model along its axis.

System Functions : I/O

Calling Statement : CALL RUNOFF(FFLOW,DISTAN,NEWN,RUNFAC)

Description

The module is entirely at the discretion of the current user.In the current case , the variable RUNFAC was used to provide an occasional factor for estimating the effects of runoff on the model predictions. To include additional loadings imposed by the runoff, the loadings per segment would also have to be transfer red in the calling vector.

E.11 Routine SMMTH

Function : To smooth input data arrays

Calling Statement : CALL SMMTH(X,I1,I2)

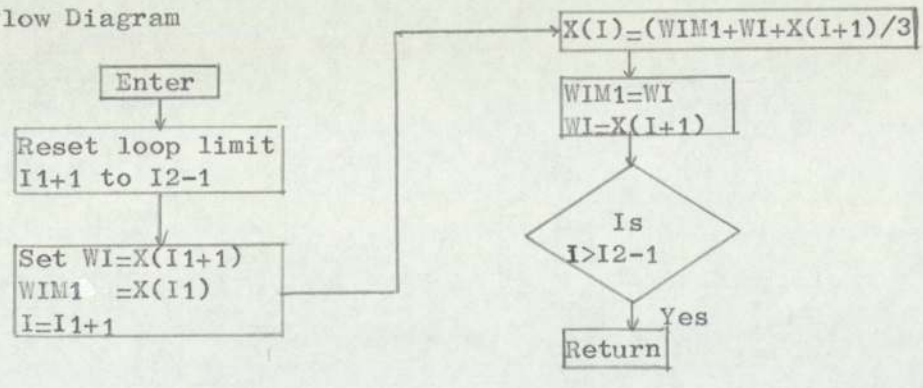
Description

The array X is the input array from elements I1 to I2 to be smoothed using the simplest formula

$$X_i = (X_{i+1} + X_i + X_{i-1})/3.$$

Each element is smoothed once , from index I1+1 to I2-1 in order to preserve the end (and possibly boundary) values . It may be necessary to weight the smoothing function, in which case only one statement need be altered .

Flow Diagram



E.12 Function SUM

Function: To return the sum of a subset of a real, single precision, array.

System)
Functions: None
User)

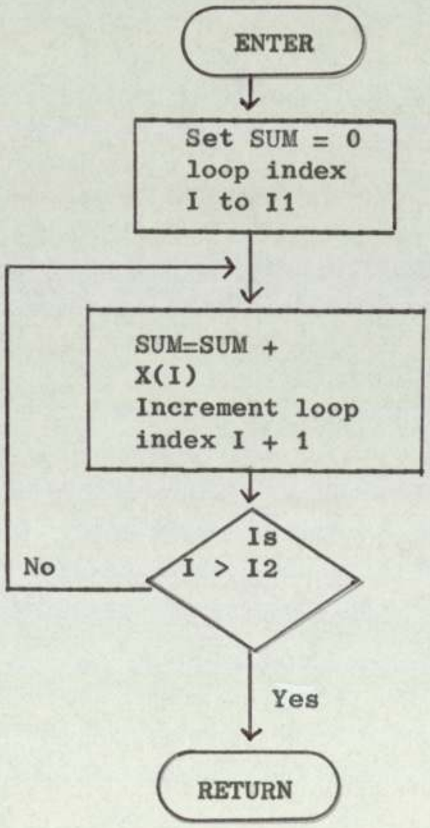
Calling Statement: Variable - SUM(X, I1, I2)

Commons: No common areas.

Errors: No error branches.

Description: The array X is to be summed from element I1 to I2 inclusive. The function name is set to zero and the summation proceeds via the only do loop. The sum is returned in the function name SUM. If I1 > I2 then SUM is set to X(I1), unless I1 is outside the defined limits of the array in which case the returned value is not guaranteed.

Flow Diagram:



E.13 Subroutine TEXRE.

Function: To adjust surface area data for the Steady State Model to allow excursion from the considered point.

System Functions: I/O

User Functions: INTP

Calling Statement: CALL TEXRE(D,SA,NEWN,NOL,W1,W2,W3)

Error Messages: None

Common Areas: None

Description: Input list

D spatial co-ordinates of segment boundaries (array)

SA Surface Areas of segments (array)

NEWN Number of segments plus one.

NOL Number of tidal excursion data pairs

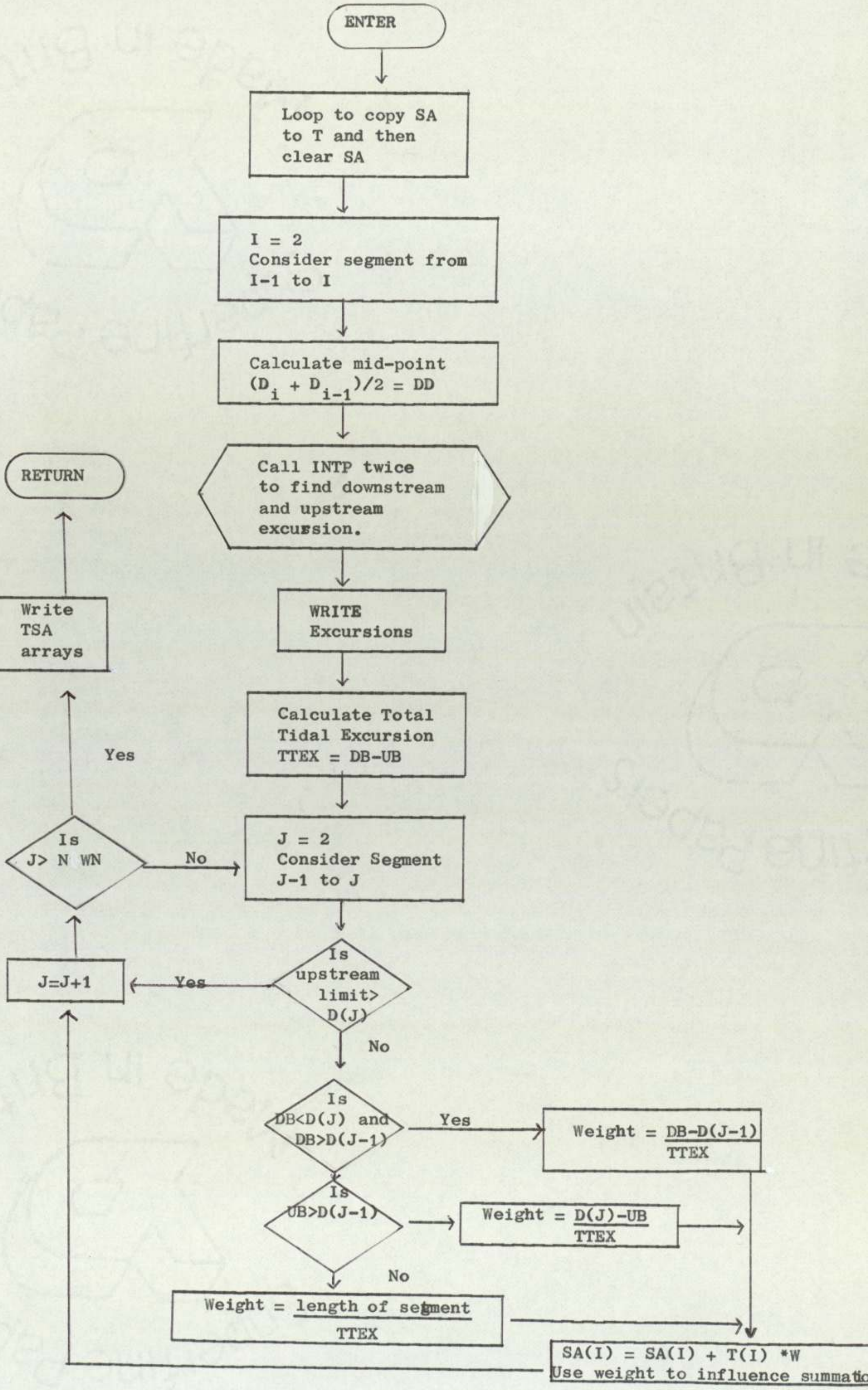
W1 Points at which tidal excursions are measured (array)

W2 Downstream tidal excursion (array)

W3 Upstream tidal excursion (array)

All arrays are single precision. T is a working area array localised to this subroutine. The first loop copies the raw data in SA to array T and clears array SA in preparation for cummulative refilling it. The DO Q loop is the main loop. The mid point of the segment under consideration is calculated, and then its estimated upstream/downstream tidal excursions obtained through use of routine INTP. UB and DB, the upstream/downstream boundary as far as the current segment is concerned, is calculated. The total excursion of the segment is the difference between these two limits. The DO 4 loop reconsiders each segment in light of the tidal excursion of the segment under scrutiny through the DO 2 loop. If the segment is outside the limits of excursion, the next segment is considered. If the limits of excursion fall partially within the excursion a partial weight is calculated, otherwise the weighting for the surface area summation is the length of the segment divided by the tidal excursion. The sum of weights will be unity unless the excursion extends over the physical limits of the system as defined by the first and last segment.

A listing of corrected vs. original surface areas is printed and control returned to the calling program.


ENTER
Loop to copy SA to T and then clear SA
I = 2
Consider segment from I-1 to I
Calculate mid-point
$(D_i + D_{i-1})/2 = DD$
Call INTP twice to find downstream and upstream excursion.
WRITE Excursions
Calculate Total Tidal Excursion TTEX = DB-UB
J = 2
Consider Segment J-1 to J
Is upstream limit> D(J)
Yes
No
J=J+1
Is J> N WN
No
Yes
Write TSA arrays
RETURN
Is DB<D(J) and DB>D(J-1)
Yes
Weight = DB-D(J-1) / TTEX
Is UB>D(J-1)
Weight = D(J)-UB / TTEX
No
Weight = length of segment / TTEX
SA(I) = SA(I) + T(I) *W
Use weight to influence summatio

E.14 DETAILED LAYOUT FOR THE STEADY STATE MODEL DATA.

Card Seq. No.	Column From	Column To	Alphanumeric Description of Variable Col 21 - 73	Col 74 - Col 80
1				TITLE1
2				TITLE2
3	1	20		TITLE3
4		(every 1-4 line)		TITLE4
5			NUMBER OF SEGMENTS	NSEG
6			HISTORY OF CONVERGENCE FLAG	IPRINT
7	1	20	MAXIMUM NUMBER OF ITERATIONS FOR CONVERGENCE	MAXLOO
8		(every 5-47 line)	NUMBER OF INCREMENTS OF REAERATION COEFFICIENTS	NODELS
9			NUMBER OF OUTFALLS IN ESTUARY	NOUTF
10			NUMBER OF READINGS FOR TIDAL EXCURSION	NOL
11			ACCELERATION COEFFICIENT FOR NITROGENOUS PHASE	OMN
12			ACCELERATION COEFFICIENT FOR AMMONIA PHASE	OMAM
13			ACCELERATION COEFFICIENT FOR D.O.PHASE	OMD
14			ACCELERATION COEFFICIENT FOR DE-NITRIFICATION	OMW
15			DO LEVEL BELOW WHICH DENITRIFICATION SETS IN	DOXMIN
16			FINAL MULTIPLIER FOR VOLUMES	FCTRL
17			MULTIPLIER FOR VOLUMES 10E6CU.FT.	VACTOR
18			MULTIPLIER FOR SURFACE AREAS TO 10E6SQ.FT.	FACTSA
19			MULTIPLIER FOR FRESH WATER FLOWS	FACTOR
20			INITIAL F.W.F. IN M.G.D.	FLOWS
21			FINAL MAXIMUM F.W.F. IN M.G.D.	FLWMAX
22			INCREMENT IN F.W.F. IN M.G.D.	FLWINC
23			FACTOR FOR RUN-OFF SUBROUTINE	RUNFAC
24			CARBONACEOUS RATE CONSTANT	RKC
25			NITROGENOUS RATE CONSTANT	RKN
26			AMMONICAL RATE CONSTANT	RKAMM
27			NITRATE RATE CONSTANT	RKNO3
28			OXYGEN EQUIVALENT OF NITRATE	CODNO3
29			AMMONIA EQUIVALENT FOR NITROGEN	AN
30			NITRATE EQUIVALENT FOR AMMONIA	ANO3A
31			OXYGEN EQUIVALENT FOR AMMONIA	OXA
32			MAXIMUM ERROR ALLOWED IN ITERATIVE PROCESS	ERROR
33			INCREMENT IN REAERATION RATES	DELRE
34			UPSTREAM FAST CARBONACEOUS LOAD IN PPM	FC(1)
35			DOWNSTREAM FAST CARBONACEOUS LOAD IN PPM	FC END
36			UPSTREAM SLOW CARBONACEOUS LOAD IN PPM	SC(1)
37			DOWNSTREAM SLOW CARBONACEOUS LOAD IN PPM	SC END
38			UPSTREAM FAST NITROGENOUS LOAD IN PPM	EN(1)
39			DOWNSTREAM FAST NITROGENOUS LOAD IN PPM	FN END
40			UPSTREAM SLOW NITROGENOUS LOAD IN PPM	SN(1)
41			DOWNSTREAM SLOW NITROGENOUS LOAD IN PPM	SN END
42			UPSTREAM AMMONIA LOAD IN PPM	AMM(1)
43			DOWNSTREAM AMMONIA LOAD IN PPM	AMMEND
44			UPSTREAM NITRATE LOAD IN PPM	ANO3(1)
45			DOWNSTREAM NITRATE LOAD IN PPM	ANO3END
46			UPSTREAM DISSOLVED OXYGEN LEVEL IN PPM	DOX(1)
47			DOWNSTREAM DISSOLVED OXYGEN LEVEL IN PPM	DOXEND

Card Seq. No.	Column From	Column To	Variable or Alphanumeric Comment (In ' ')
48	1	50	SMOOTH,LOGICAL ARRAY
	51	80	' = SMOOTHING OF DATA ARRAY L*1'
49	1	50	PRINTF,logical array
	51	80	'= PRINTING OF ARRAY INFORMATION'
50 and	1	70	DISTAN array Free Format
rptd.as necessary.	71	80	'SEG.BDRYS'
51 "	1	70	REAR array Free Format.
	71	80	'REAERATION'
52 "	1	70	Volumes of Segments in free format.
	71	80	'VOLUMES'
53 "	1	70	SA.Segment surface areas in free format.
	71	80	'SURFACE AR'
54 "	1	70	SAL Salinities in free format
	71	80	'SALINITIES'
55 "	1	70	TEMP Temperatures in free format
	71	80	'TEMPERATUR'
56 "	1	70	W1.Tidal Excursion measurement ordinates in free format
	71	80	'TIDE. EXC'
57 "	1	70	W2. Downstream Tidal Excursions
	71	80	'DOWNSTREAM'
58 "	1	70	W3. Upstream Tidal Excursions
	71	80	'UPSTREAM'
59 "	1	50	TIDEL Logical array
	51	80	'LOGICAL TIDELOCKED DISCHARGES'
60	1	1	MODE Single integer input mode for discharge data.
61	1	8	NAME - character name of discharge.
This format is to be repeated once for every outfall in the simulation.			
	9	10	TWO spaces for user identification of discharge (or sequence or version No).
	11	80	PSTN, RFLOW and (LOADS) in that sequence.

The entire file can be repeated several times and several simulations of the model therefore run in one overall run of the model. Output will be identified by variables to indicate data set and data subsets.

SUMMARY OF INPUT DATA.

This section describes the variable names of each parameter, the format number under which it is read in, the kind of variable it is and the number of elements it has under the Usk River Division standard program versions.

TABLE showing Name of Variable, it's function, the format it is read in with and type and size of data.

Variable Name.	Variable Function.	Input Format No.	Type of Variable.	No. of Elements.
TT	Title	200	Alphn	80
NSEG	No. Segments	400	I	1
IPRINT	Print Flag	"	"	"
MAXLOO	Iterative Loops	"	"	"
NODELS	Reaeration Loops	"	"	"
NOUTF	Outfalls	"	"	"
NOL	Tidal Excursion points	"	"	"
OMN	Convergance Accelerators	"	R	"
OMAN	"	"	"	"
OMD	"	"	"	"
OMW	"	"	"	"
DOXMIN	Min D.O. Trigger	"	"	"
FCTRL	Units converters	"	"	"
VACTOR	" "	"	"	"
FACTSA	" "	"	"	"
FACTOR	" "	"	"	"
FLOWS	Initial F.W.Flow	400	R	1
FLWMAX	Max.F.W.Flow	400	R	1
FLWINC	F.W.Flow increment	400	R	1
RUNFAC	Run off function factor	400	R	1
RKC	Rate constants C	"	R	"
RKN	Rate Constants N	"	"	"
RKAMM	" NH3	"	"	"
RKNO3	" NO3	"	"	"
COXNO 3	Equivalents	"	"	"
AN	"	"	"	"
ANO3A	"	"	"	"
OXA	"	"	"	"
ERROR	Max error of solution	"	"	"
DELRE	Reaeration increment	"	"	"

Name	Function	Format No.	Type of Variable.	No. of Elements.
FC(1)	Boundary Valves	"	RAE	"
FC(NSEG + 2)	"	"	"	"
SC(1)	"	"	"	"
SC(NSEG + 2)	"	"	"	"
FN(1)	"	"	"	"
FN(NSEG + 2)	"	"	"	"
SN (1)	"	"	"	"
SN(NSEG + 2)	"	"	"	"
AMM (1)	"	"	"	"
AMM (NSEG + 2)	"	"	"	"
ANO3(1)	"	"	"	"
ANO3(NSE) +2)	"	"	"	"
DOX (1)	"	"	"	"
DOX (NSE) +2)	"	"	"	"
SMOOTH	Smoothing of data options array	700	LA	50
PRINTF	Printing of data options array	"	"	"
DISTAN	Segment Boundary Positions	800	RA	NSEG +1
REAR	Reaeration rates per segment	800	RA	NSEG
V	Segment Volumes	800	RA	NSEG
SA	Segment Surface Areas	800	RA	NSEG
SAL	Salinities	800	RA	NSEG + 2
TEMP	Segment Temperatures	800	RA	NSEG
W1	Tidal Excursion Ordinates	800	RA	NOL
W2	Downstream Excursions	800	RA	NOL
W3	Upstream Excursions	800	RA	NOL
TIDEL	Tidelocked outfall option.	800	LA	50
MODE	Input data layout option.	800	I	1

then, for each outfall, under format 914

NAME	Name of outfall	Alphn - up to 8 characters.
PSTN	Position of outfall from zero	
RFLOW	Flow of outfall (m.g.d.)	
W4	Loads of components in the order RA - 7 elements.	

1. F.C. (lbs per day)
2. S.C. "
3. F.N. "
4. S.N. "
5. NH3 "
6. NO3 "
7. D.O. (Average % Saturation)

End of input

E.16 INDEX OF VARIABLES THAT APPEAR IN THE STEADY STATE MODEL

A

Tempoary location in the section dealing with restricted oxidation.

AKAMMT

Final rate of reaction for oxidation of ammonia

AKCT

Final rate of carbonaeceous oxidation of the BOD component

AKNO3T

Final rate of reaction of nitrates

AKNT

Final rate of nitrogenous oxidation of the BOD component

AMM

Final distribution of predicted ammonia levels

AMMADD

Ammonia load added in each segment through outfalls

AMMIT

New estimated ammonia distribution for the segment in question

AN

The ammonia equivalent for Nitrogen

ANO3

The nitrate equivalent of Ammonia

ANO3A

Nitrate equivalent for Ammonia

ANO3AD

Nitrate load added in each segment

ANO3T

New value for estimated nitrate levls in segment in question

B

Work area involving nitrates in the restricted oxidation of pollutants.

§BOUNDARIES

For each array, the 1st element contains the upstream boundary, and the last element the downstream(open-sea) boundary value in ppm. The computed distribution occurs in elements 2 to NEWN+1. These values must be constant and apply to arrays FC SC FN SN AMM DOX ANO3 only

COXNO3

The oxygen equivalent of nitrate.

DB

The downstream tidal excursion limit for the current discharge

DECR

Total increment in reaeration rates from start to end of the simulation, i.e. incremental step x number of increments

DELRE

Stepwise increment of the reaeration rates

DEPTH

Array holding depths of each segment

DISTAN

An array of dimension NSEG+2 containing the limits of the segments in distance units from a zero which is usually the tidal limit of the estuary in question.The 1st segment lies between distan(1) and distan(2) and those distributions calculated, say element i of a component, is the value computed for the point midway between distance elements i and i+1

DIV

A division factor in the calculation of mixing terms

DOX

The final distribution of dissolved oxygen levels

DOXADD

An array holding inputs of oxygen load from outfalls

DOXIT

Work area for the new estimated level of D.O. in the segment in question

DOXMIN

The level below which anaerobic kinetics/chemistry switch in,this is set to o.4 ppm D.O. currently

§DOXSAT

Function to return the saturated level of D.O. given a salinity & temperature

ERROR

This variable defines the limit of the accuracy of the iterative scheme. If there are m components being assessed over n segments, the process is considered to have converged iff

$$\text{ERROR} = \sum_{i=1,j}^{m,n} C_{i,j}^{k+1} - C_{i,j}^{k}$$

where $c_{i,j}^{k}$ is the concentration of the ith component in the jth segment at the kth iterative loop.

This implies that the average residual is approximately :

$$\frac{\sum_{\text{all } i} \sum_{\text{all } j} C_{i,j}^{k+1} - C_{i,j}^{k}}{m.n}$$

which is usually two orders of magnitude lower than the summed error term.

§ESTFIN

Main program name

§EXMINE

Subroutine to examine an array and optional smoothing/printing

F

An array of dimension NSEG+1 containing the mixing coefficients as estimated from the supplied salinity distribution.

$f_i = FFLOW_i . (SAL_i - SAL_1)/(SAL_{i+1} - SAL_i)$

FACTOR

Multiplier to convert input flow from any units to millions of pounds per day.Usually FFLOW is input in m.g.d. and converted

FACTSA

Multiplies the surface ares input to a common unit

FC

Array holding the final computed distribution of the fast carbon pollutant component

FCADD

Fast Carbonaeceous load input array in each segment

FCTRL

Factor for further conversion of volumes after depths computation.

FFF

A one byte logical false flag(constant)

FFFLOWS

Initial storage of data for flow parameters

IN

Input channel number.Usually defined by a DATA statement

§INPUT

Subroutine to handle general data input to the model

§INTP

Subroutine to perform general multiorder interpolation

IPRINT

This is a print dump flag.The details of the iterative scheme are dumped to the output channel every IPRINT loops.The larger this value is set,the smaller the output it generates.

KKKK

Index for increments in reaeration rates simulations

§MATSET

Subroutine to solve a band matrix

MAXLOO

Maximum number of loops of the iterative scheme to be attempted before the process is abandoned as divergent

MODE

An input option variable.See section on inputs for details

NAME

An array holding the name of a particular discharge

NEWN

Defined as NSEG+1

NODELS

The number of times the reaeration coefficients are incremented within one run of a data set.

NOL

Number of points at which up/down stream tidal excursions are known

NOUT

The output channel number,usually defined in a DATA statement

FFLOWS

An array of dimension NSEG+2 containing the limits of the segments Fresh water flow. The content of the 1st element reflects the fresh water input at the head of an estuary.Subsequent elements contain the total fresh water input to all upstream elements plus the estimated input from discharges which are considered to 'see' that particular segment.

FLOWS

The initial fresh water input to the head of the system

FLWINC

Incremental stages of fresh water flow levels to be simulated. This is the step size for progressing from FLOWS to FLWMAX

FLWMAX

Maximum fresh water flow for which the system is to be simulated

FN

Final distribution of Fast nitrogenous pollutant component

FNADD

Fast nitrogenous loads added in each segment

FTMP

Work area to hold current mixing term

§GRAFOR

System supplied plotting routine for a CALCOMP Plotter

IC

Counter for the iterative process

IDS

Index of the current data set being processed.For use with multiple run versions.

IFLGSG

IFLSG   Index of the segment containing the current outfall

IFLSEG

NOUTF

Number of discharges in the system

NSEG

Number of segments in the system

OMAM

This is the acceleration parameter for the iterative solution for NO3/NH3/DO. This coefficient, along with OMN,OMD,OMW can be varied as input parameters and tuned for the particular situation being modelled. Values of unity for all four will guarantee convergence eventually, although a large number of iterations may cause a normal convergence to an erroneous solution through round-off errors.In this investigation, values of unity required 960 iterations to converge. Very fine tuning reduced this to 12.The degree of tuning depends on the nature of the amendments to be run.If the variants are severe, less tuning should be employed-fine tuning will prevent the variant solution from converging.

OMAM1

Defined as 1. - OMAM

OMD

See OMAM

OMD1

Defined as 1. - OMD

OMN

See OMAM

OMN1

Defined as 1. - OMN

OMW

See OMW.Only used in the anaerobic situation (ie infrequently)

OMW1

Defined as 1. - OMW

§OUTPUT

Subroutine to produce a digital plot of the predicted DO distribution Geared for a 132 character line printer facility.

OXA

Oxygen equivalent of Ammonia

§PLOTGP

Subroutine to graphically interpret the predicted distributions. Written for a 31" CALCOMP.

PRINTF

A logical array, with an element set to .TRUE. if a particular item is to be printed. The list below itemises that section of PRINTF that is active. The same indeces apply to array SMOOTH:

Index	Is smoothing allowed here	Array for Action or cause of element set to 'on'
1	yes	Distan
2	yes	W1
3	yes	W2
4	yes	W3
6		Segment index, tidal excursion weight(outfalls)
15	yes	Reaeration rates
16	yes	Volumes
17	yes	Surface Areas
18	no	Converted volumes
19	no	Converted surface areas
20	yes	Depths
21	yes	Salinity
22	yes	Temperature
24	yes	Mixin coefficients
25	no	Flow additions
26	no	DOXS-DO saturation levels

RKN

Nitrogenous oxidation rate at 15 deg.C.

RKNO3

Nitrate rate constant at 15 deg.C.

RP

In the calculation of the sag severity index,it is a measure of the amount of oxygen theoretically possible in the system at the ambient salinity,temperatures

RSC

Slow Carbon rate constant. Usually .2 of RFC

RSN

Slow Nitrogenous rate constant.Usually .2 of RFN

RUNFAC

A factor allowing for a correction for land run-off to be made in the calculation of fresh water flows.

§RUNOFF

Subroutine to correct for land runoff

RZ

Reciprocal of NEWN times the index of current segment

RZ1

Defined as 1. - RZ

SA

Array holding surface ares of each segment

SAL

Array holding mean salinities in each segment

SC

Array holding final distribution of slow carbon pollutant

SCADD

Array holding slow carbonaeceous load for each segment

§SMMTH

Subroutine to smooth input data if requested

SMOOTH

A logical array of one byte elements. If set to .TRUE. the array is smoothed prior to use or output.See PRINTF for details of use.

SN

Array holding final distributions of the slow nitrogenous pollutants

SNADD

Slow Nitrogenous loads added per segment array

§SUM

Function to perform fast summation of vectors

SUMOLD

This contains a historic value of SUMT. Periodically, the newly computed value of SUMT is tested against this to see if the process is continuing to converge.

SUMT

Sum total of absolute differences between n^{th} and $n+1^{th}$ estimate for all segments and all interacting pollutants

SUM1,SUM2,SUM3,SUM4,SUM5

Summations of absolute differences between successive estimates of individual contaminants

TDEX

Downstream tidal excursion of the discharge being considered

TEMP

Array of dimension NSEG+1 holding the temperatures of each segment in degrees centigrade

TEMPX

The difference between the segment temperature and the 15 deg.C. standard

TEMPZ

Defined as 1.088^{TEMPX}

§TEXRE

Subroutine to allow correction for tidal excursion of surface areas

TIDEL

A logical array, each element corresponding to one discharge. A value of T on the input list implies that the outfall it refers to is a tidelocked outfall, acting under hydrostatic head pressure. A value of F implies that the discharge is constant and freeflowing

§TIME

System function to give elapsed CPU time used since the start of simulating.

TTEX

Total tidal excursion of the current discharge.

TTT

A one byte logical constant set to .TRUE.

TUEX

Upstream tidal excursion from the site of the discharge.

TW

The sum of weights for loads from the current outfall to the parts of the system within its tidal excursion. Should be 1.0 unless a part of the discharge load is lost through the downstream boundary.

U

Array in the de-nitrification section

UB

Upstream tidal excursion limit for the current discharge

V

Array holding volumes of each segment, stored from element 2 on, of dimension NSEG+1

VACTOR

Multiplier for array of volumes to achieve common units.

VTMP

The work location for volume of the current segment

W

Work array

§WORK ARRAYS

The following are used as work areas in the preliminary stages of computation : W W1 W2 W3 W4 WW TT RFC RSC RFN RSN RDOX RNO3 RAMM

WGHT

Weighting for all loads from the current discharge for the current segment

W1

Ordinates of tidal excursion input data

W2

Downstream tidal excursion data holding array

W3

Upstream tidal excursion data holding array

X

Flow into segment array

XTMP

Work area to hold X_{i-1}

XX

Array holding flow out of segments,computed internally

Z

Reciprocal of NEWN and work area

Note. Items prefixed by § are not strict variables within the object deck.

27	no	AKCT AKNT AKNO3T AKAMMT
28	no	AMM ANO3 DOX,initial attempts at solution
29	no	X XX,flows in and out of a segment
30	no	RDOX Rate of reaction,DO
31	no	Further reaeration rates
32	yes	Final DO predictions
33	-	Graphical output of DO curves

Variants are itemised in versions of the object deck

PSTN

The position of the outfall from the zero of measurement

RA

In the calculation of the sag severity,it is a measure of the amount of oxygen absent

RAMM

Rate of oxidation of Ammonia at 15 deg.C.

RDOX

Reaeration rate term.

REAER

Array holding the reaeration rate constants per segment

RFC

Fast Carbon rate term array

RFLOW

The flow of the current discharge to the system

RFN

Fast Nitrogen rate term array

RI

In the calculation of the sag severity,this is a measure of the total amount of oxygen present

RKAMM

Rate terms for ammonia

RKC

Carbonaeceous oxidation rate at 15 deg.C.

APPENDIX F

ADDITIONAL SOFTWARE DESIGNED FOR THE PROJECT

F.1	DIVNDX	Biological Diversity index
F.2	AVDATA	Metereologival data analysis routine
F.3	FISHPT	Interrogation of output from F1/F2
F.4	Program suite for lexical analysis of files (13 routines)	
F.5	A Graphics Package for a CALCOMP Plotter (31") (21 routines)	
F.6	The Date-Time Handling Package.(17 routines)	
F.7	Brief description of additional routines : LINREG,SORT,EIGEN,MULTMA,SERIES,CUMULA and TEXT.	
F.8	HYDOR	Hydrological Data Processing Routines

F.1 **Program DIVNDX**

Function	Calculation of Biological Diversity Index.
System Functions	Log to base e, FLOAT, I/O, Statement Functions (user supplied)
Reference	"One Year", IMD Report No.2, M.W.Rogers, Oct 1973, Vol.1, Chapter 5.

Description

The program was written in conjunction with Mr.S. Lambert, Biologist to the Usk River Division, and reference the Journal of the Water Pollution Control Federation, Vol. 43, No.5, May 1971.

Each set of readings are considered separately. Each set starts with Col 1-12 of the first data card used as a heading. The remaining 68 columns of this card and all 80 columns of subsequent cards are used to read in NSP - no. of species in sample and COUNT(I), I=1,NSP - the number of occurrences of species I.

The total species recorded is calculated and saved in TNSP. Probability occurrences are then calculated and summed

$$D = -\sum_{i=1}^{NSP} P_i \frac{\log_e(P_i)}{\log_e 2}$$

where P_i is the probability of occurrence of species I

This is the raw index. By partial differentiation of this expression, a maximum and minimum Theoretical Index can be calculated. The REDUNDANCY is then calculated by

$$\text{REDUNDANCY} = \frac{D_{Max} - D_{Raw}}{D_{Max} - D_{Min}}$$

The Index is then grated into clean/moderately polluted/heavily polluted waters and the statistics are printed. Another data set is sought. If none found, control jumps to statement 99 and a normal job termination. Run times for NSP less than 100 are negligible. The lising is in Appendix H.

F.2 **Program AVDATA**

Function: To re-align a set of data from fixed sampling points to a different set of application points on a varied grid.

System Functions: I/O, Tape I/O, DEBUG

Use Functions : None

Error Messages
1 "Data for n already present"
2 "Test to see if all stations read in fails"
3 "Error on reading in Met Office Data card"
4 "Premature EOF on card reader"
5 "Error in distance matrix input"

Common Areas : None.

Description: Data suplied:

NOFS No. of field station in the system.
NOMS No. of sampling stations where parameters are collected
NOPAR Number of parameters sampled
IDC Sampling station reference number allocated by The Meteorological Office Bracknell
FMT Format of Met.Office Data records
D Array containing weighting factors
IRPG Reporting frequency switch
IID Array holding field station identifiers.
DS18 Logical switch, on to produce a tape copy of output.

The program inputs the run parameters in the following sequence:

1st record in free format,

NOFS NOMS IRPG NOPAR DS18

2nd record under 16I5,

IID(I),I=1,NOMS

Reads in identifier of each sampling point.

3rd/4th record under 20A4

FMT , for layout of the input records

5th record under I3,10F5.0, reads in D(I,J), being the geographic distance from the ith field station to the jth sampling station.The first integer field is the integer reference number of the field station. This record is then repeated as necessary to accomodate the required NOFS and NOMS Stations.

6th record et seq.

These records are the data supplied by the Met. Office and one record may be more than one physical record or more than one tape block.The layout must be in accordance to that in array FMT. The initial integer must be the station IDC.This can be followed by up to 30 parameters. All records fo all stations for one point in time must appear before a new,later data time is considered. The program fails if data for any station for any sample time is absent as the latter half essentially requires complete data.

Logic Flow: PHASE 1

The data is read in as outlined above.The initial calculations involve estimated weighting factors of data for each field station in relation to each sampled station.

PHASE 2

The sum of distances from each field station to each sampling station is calculated. Then the data weight given to each sampling station for a field station, is the ratio of

$$\frac{\left[\begin{matrix}\text{Sum of distances to}\\ \text{all sampled points}\end{matrix}\right]-\left[\begin{matrix}\text{Distance to this}\\ \text{sampled station}\end{matrix}\right]}{\left[\begin{matrix}\text{Sum of distances to}\\ \text{all sampled points}\end{matrix}\right]}$$

This is a direct inverse law and is a convenient method for averaging such data. Variations can be introduced by the transformation of the distance input matrix,as it need not be logically consistent, and thus terrain singularities can be accomodated.

PHASE 3

This section deals with the input of the sampled data and its validity tests. The data is assembled in groups of NOMS logical records.Within each group, all records must refer to the same time, and one set of data must appear for each sampling point.

As each record is entered, its identification is checked against the list of expected points.The master list is in array IID. If this list is not exhausted by the time the next sample date is encountered,error message 2 appears. If more than one set of data appears for any one point in the same time interval then message 1 is output.Either of these errors terminates program execution.
Having read one set of data for each sampling point, the field point data values are calculated using weighted means calculated previously.The weights are in array D.For each sampling time and for each field station , one record is generated to a tape output file.The sequence of this record is :
parameter1,parameter2,...parameter NOPARA,Field station no., sequence number of sample in data.
Phase 3 continues until all groups of cards have been processed. When an EOF marker is reached, the tape file will contain estimated readings for each user field station.

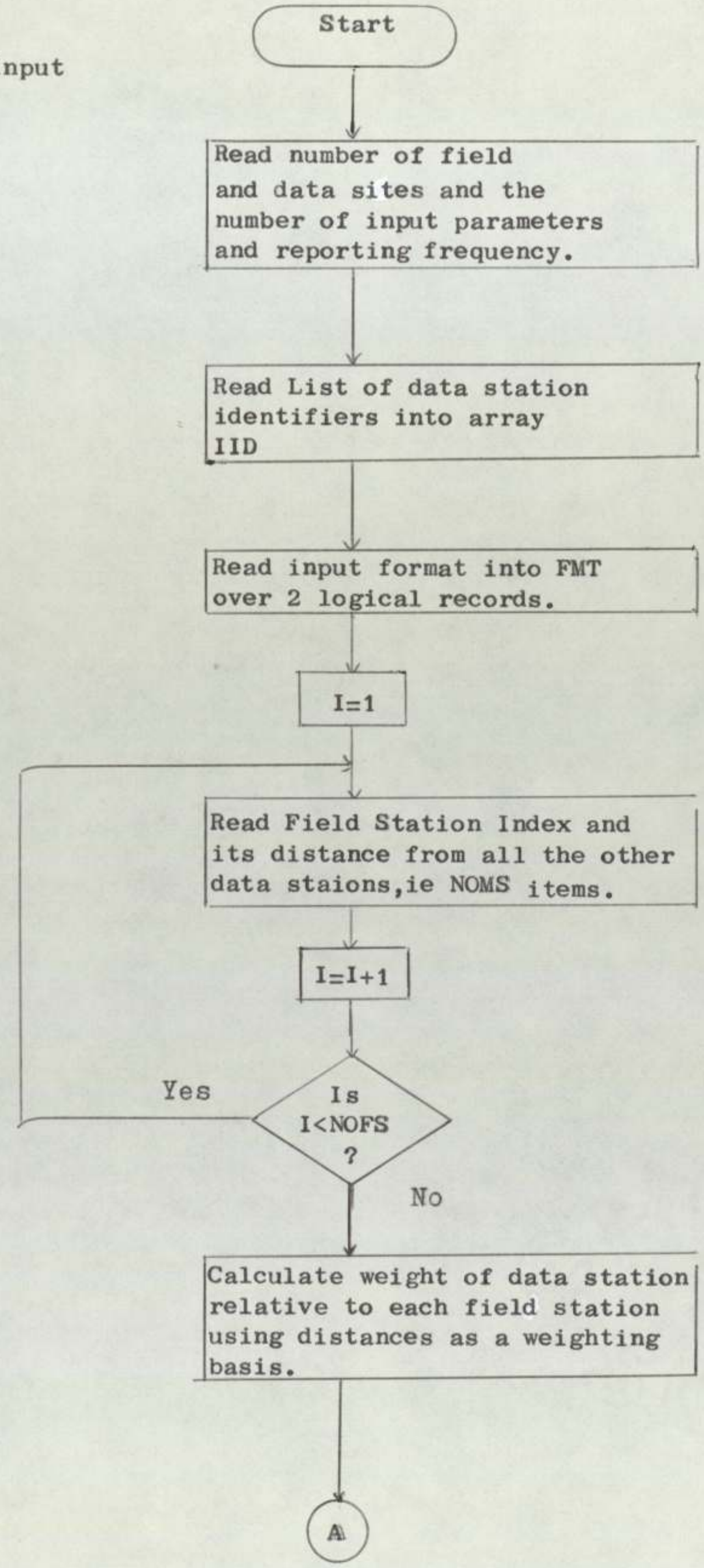
PHASE 4

For each field station, the data so estimated is now to be averaged.The time span for this process is defined by IRPG units of the basic sampling interval of the raw input data. If the input is daily data (as in this case), to produce weekly means, integer IRPG would have to be set to 7.Also, means are calculated for a time span of 4*IRPG. If the data is daily then weekly means can be compared to monthly totals. To alter the time multiple of 4, lines 11500,11600,11800 require to be changed.For monthly/annual statistics , the 4 would be replaced by a 12.
One pass of the tape is required for each field station.In a large system this might require reprogramming , but in practice such a system is more economic as the smaller size of program thus required costs less in terms of core-occupancy. If DS18 is .TRUE. then a copy of the calculated means is made to the device 18 (ICL4-70 under MJ=magnetic tape unit).This tape has output format compatible to the input and so can be readmitted to calculate successively longer interval means.

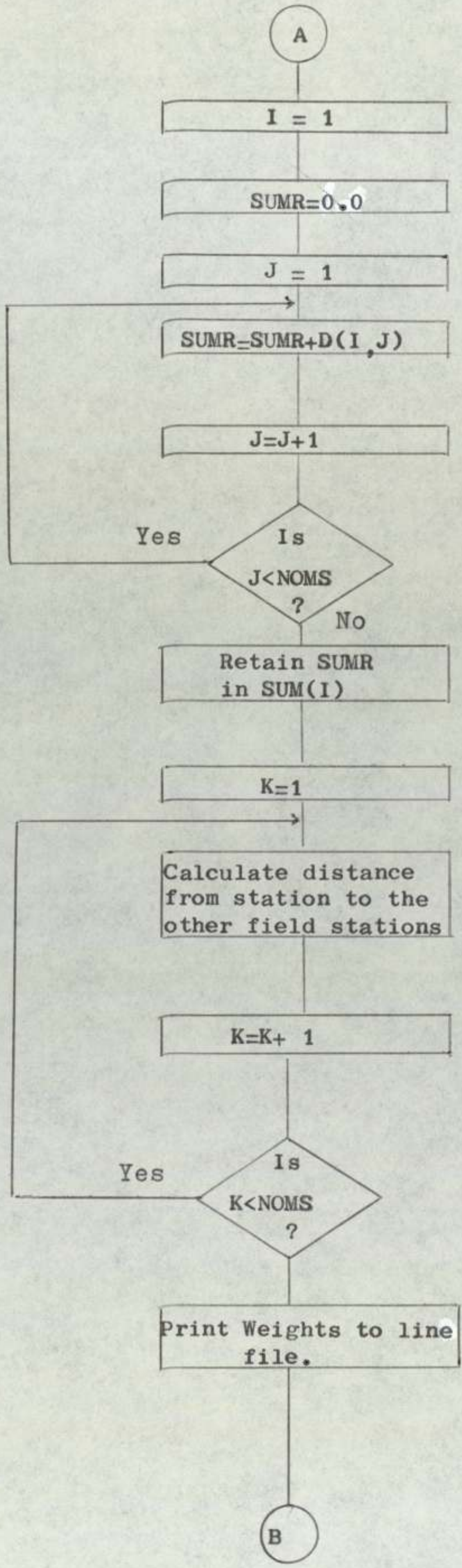
Applications: The program was written initially for the inclusion of weather records into a regressive quality model. As this was soon discarded, it was used for the TAFF and EBBW Projects for the Usk River Division and the U.C.Cardiff Botany Dept. on the re-establishment of diverse communities in dead systems.

Flow Logic

Phase 1-2
Parameter Input



Detailed Flow Logic of Phase 2-Calculation of weighting factors.



F.3

Program FISHPT

Function	:	A comphrehensive program to interrogate output from the F1/F2 models and present data in graphical form. FISHPT is the main organising program calling other routines and abstracting relevant data.
System Functions	:	READ/WRITE I/O, END & ERR options, SQRT, INT, NETLIB:ROUTIN PBPLTϕ & 1
User Functions	:	BAYOUT, PLOTH, PLOT4, BAYP, COPYDT, INTP, ADDATE, FILL (INP)
Error Messages	:	STOP 9999 Error on Read in or end of file reached STOP ϕϕϕ4 Level ϕ command not recognised against dictionary STOP ϕϕ13 Level 1 command not recognised against dictionary STOP ϕ2ϕ3 Level 3 command not recognised against dictionary
Common Areas	:	/AA/, /LIM/, /IO/, /DATE/
Common Variables reset	:	/IO/ variables are established, also /LIM/ and /DATE/.
References	:	Engineering classification BENGF=PBPLT1.ϕ12.ϕ1, P.R. Binding, School of Enginerring, University of Bath. For modules NETLIB:ROUTIN. PBPLTϕ & 1 details not given in this document.

Description

The program is structured to a series of command levels. Each command is a four character mnemonic for a particular action.

Commands Available

LEVEL ɸ

The most general level of command, used for setting up the run and selecting segments of the program to be read in.
All commands appear in Col.1 - 4 of a card, the remainder can be used for comments.

COMMAND	DESCRIPTION
'NOUT'	Next item read in is the data set reference number of the F1/F2 data to be interrogated. Must appear once before data to be assessed.
'DATE'	Next items read in are current start date of data (only required for F2 as an option)
'PRIN'	Select segment that only reads and prints F1/F2 data off data set NOUT
'BAYA'	Call routine that only deals with plotting of bay data - BAYP
'GRAP'	Call segment that reads in level 1 commands for plotting of data via standard graph plot package
'DATA'	Call segment that reads level 2/3 commands that select what data is to be retained from the input file
'PLOT'	Plot selected data thorough supplied package
'END'	End of level ɸ commands implies end of job. STOP 9999 is normal termination code.

LEVEL 1 Commands set up parameter list for a standard plotting package

Command	Description
'XAXI'	Next input item is the type of x axis to be selected, see below
'YAXI'	Next input item is the type of y axis to be selected, see below
'TITX'	The remaining 76 columns of input on this card image hold a title for the x axis
'TITY'	The remaining 76 columns of input on this card image hold a title for the y axis
'TITG'	The remaining 76 columns of input on this card image hold a general title to be given to the whole plot
'LENX'	Next input item is length of x axis in inches
'LENY'	Next input item is length of y axis in inches
'ALEN'	Reference length for plots, see below
'ISWI'	Next input item is switch to control plot and print, see below
'END'	Level 1 commands have terminated. Seek next level Ø command

Type of axis allowed

'LINZ' Linear scale, zero origin
'LINF' Linear scale, false origin computed internally
'LOGD' Decade logarithmic scale

Options are possible mixed in x and y axis

ALEN option

This is a real number input to the standard routine. If >0 the distance between two successive plot points is greater than ALEN inches, the line between the two points is not drawn, but symbols marking the points are drawn. Plotting/deletion is instigated automatically.

If = ϕ all lines are drawn and when the plot file is complete, it is automatically plotted and deleted upon completion of the plot

If <0 the plot file so generated will have to be explicitly queued to plot and manually deleted

ISWI option

A switch used to control output options

If <1 Only lines on plot are plotted, no point symbols and in graph plotter output

If = 1 Symbols occur on each plot line to differentiate lines

If = 2 Line printer output of plot points is made, only lines plotted

If >2 Both line printer ouput occurs and symbols marking plot lines occur

LEVEL 2 Commands restrict the retention of input data to that being plotted.

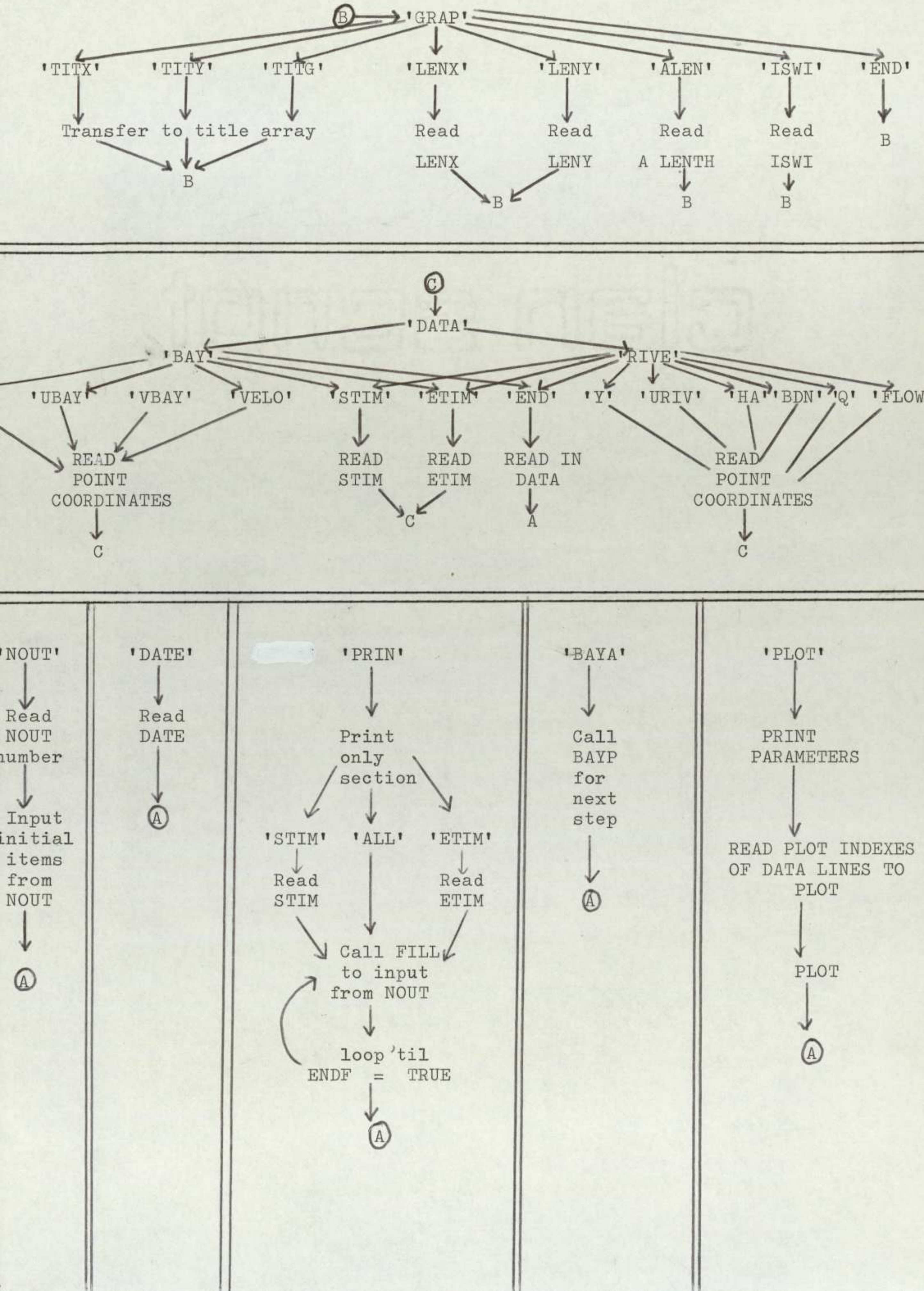
Command	Description
'STIM'	Next input item is start time to retain input
'ETIM'	Next input item is end time to retain input
'BAY'	Bay data to be retained, find level 3 commands
'RIV'	River data to be retained, find level 3 commands

LEVEL 3 Selected data items to be plotted later

for Bay data the following commands are required

Command	Description
'SE'	Elevations in bay
'UBAY'	U velocities in bay
'VBAY'	V velocities in bay
'VELO'	Velocity calculated from U & V components in bay
'Y'	Elevations in river
'URIV'	Velocity in river
'HA'	Mean depths in segments of river
'BDN'	Downstream boundary widths in river
'Q'	) Flow rates in river
'FLOW'	)

Hierarchy of commands for all Level ϕ commands, showing options in main routine



Program Structure

There are 4 sections in the routine:

1. Reading in and checking commands and retaining a coding system
2. Reading in data off files and retaining relevant items
3. Plotting graph options
4. Dealing with Bay plots through BAYP routine

1. Reading in commands etc.

The data array is set to - 1 initially and some default options are set. The statement 9 READ (5... is the main return statement when coming out of sub levels. A command is read in and tested against the dictionary. If not found, STOP 4 is issued.

Control transfers to the appropriate segments.

2. Reading in data

Data is input from the files generated by F1/F2 model simulations. Each block is prefaced by a time in hours, and this is checked against time limits for retention. Up to 5ΦΦ data items for each of 8 data points may be retained. A logical flag set in routine FILL returns the end of file message to switch back to a level Φ command.

3. Plotting Graph Options

This uses either the system routine PBPLTΦ & 1 or own modules PLOTH and associated subroutines. Options are written for a CALCOMP 31" incremental X-Y plotter through the system 4 interface program GRAFOR that also controls off line plotting options.

4. Dealing with Bay Data

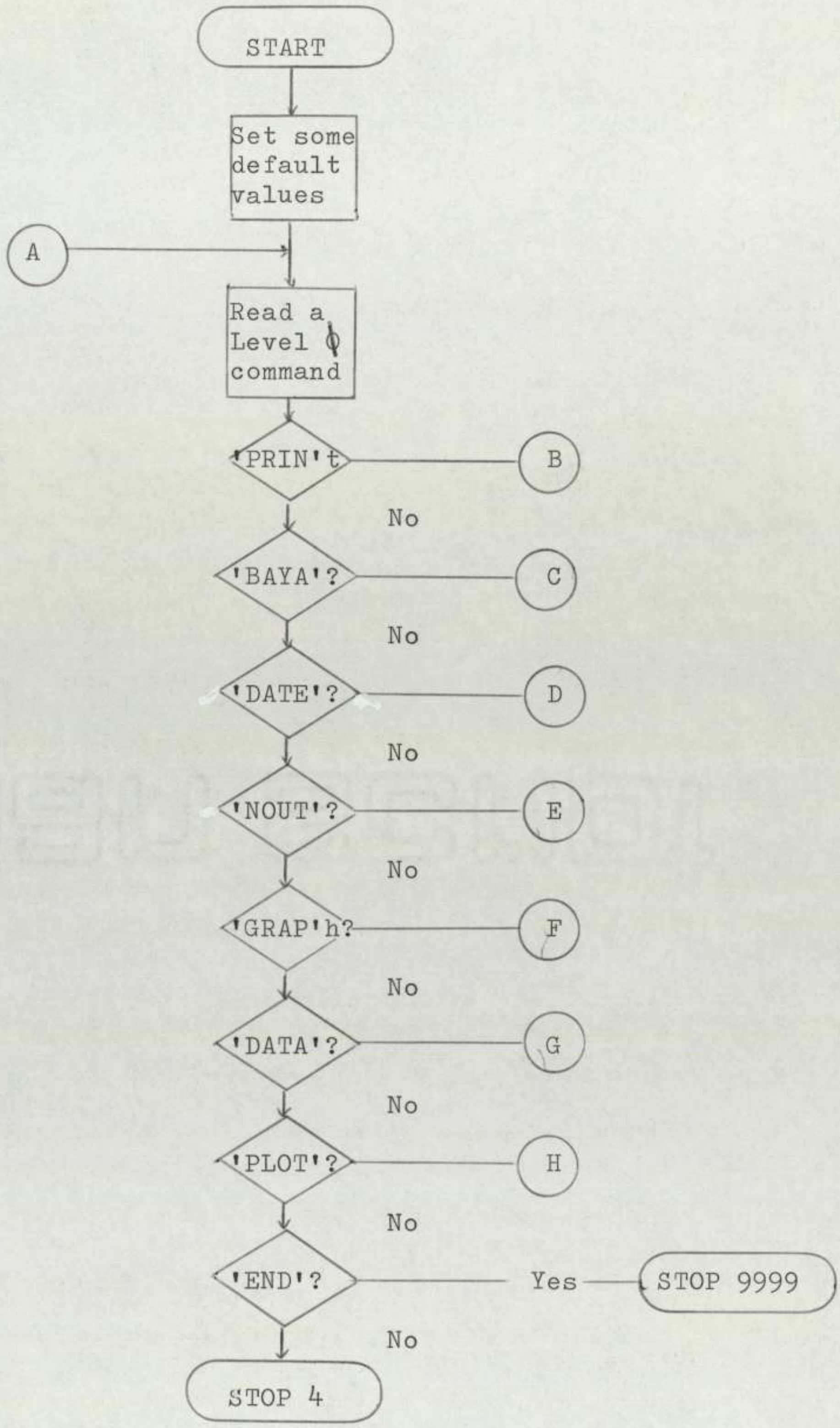
Routine BAYP deals with the overall presentation of a set of bay data as an interpretive picture rather than specific items. Several options are available.

FLOW LOGIC

The flow logic following gives a comprehensive outline and, with a listing, should be sufficient for program understanding.

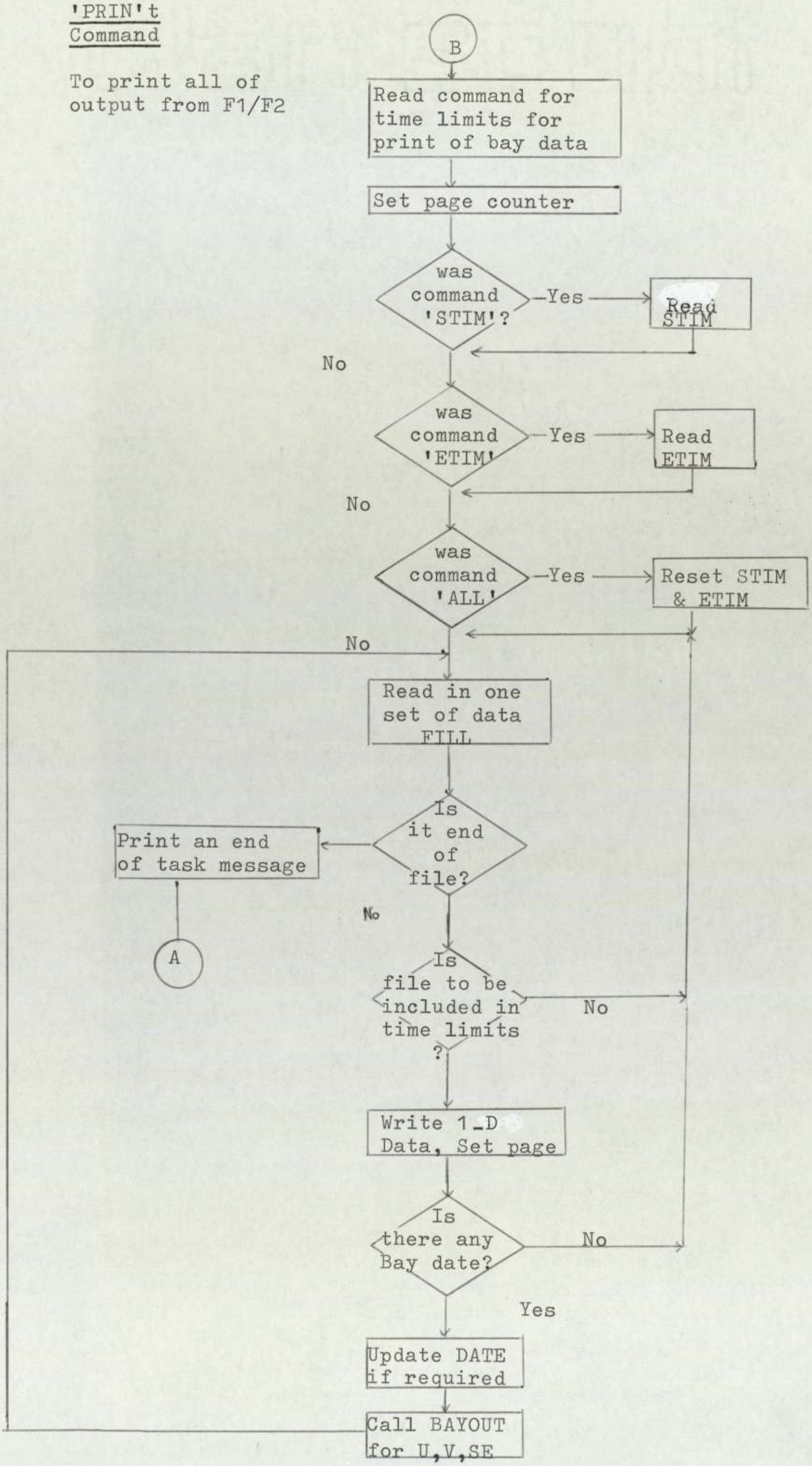
Initial Phase

Read in and test a level ɸ command.

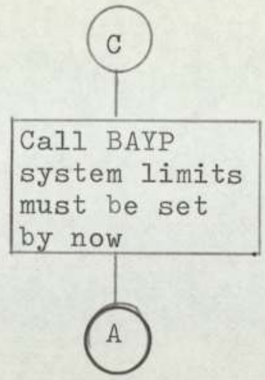


'PRIN't
Command

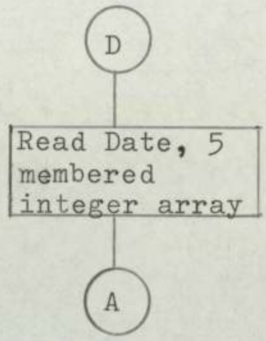
To print all of output from F1/F2



'BAYA'11 Command

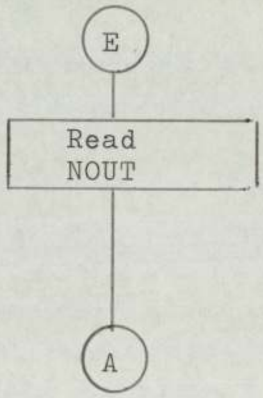


'DATE' Command



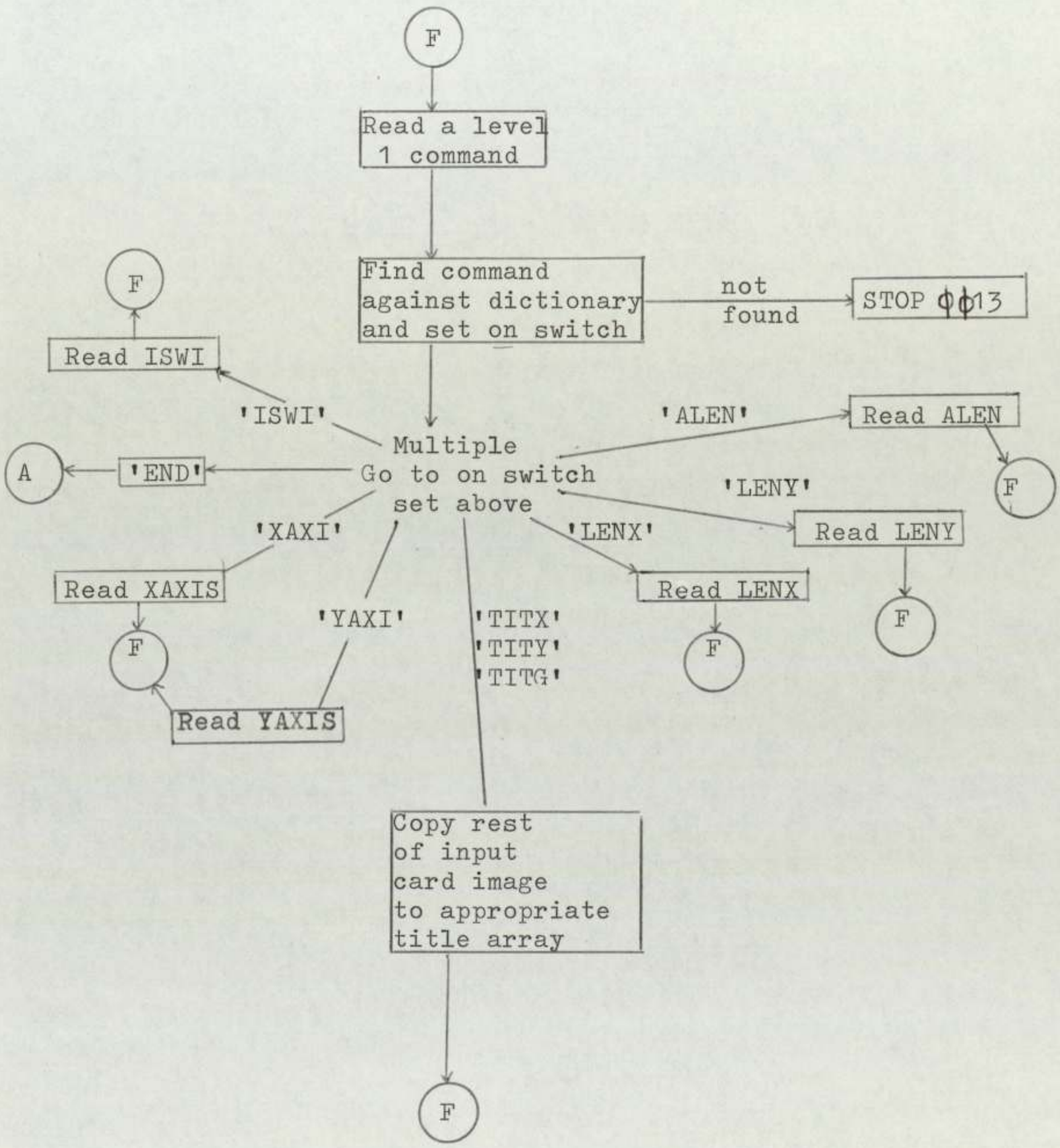
'NOUT' Command

Date set reference number of F1/F2 output file

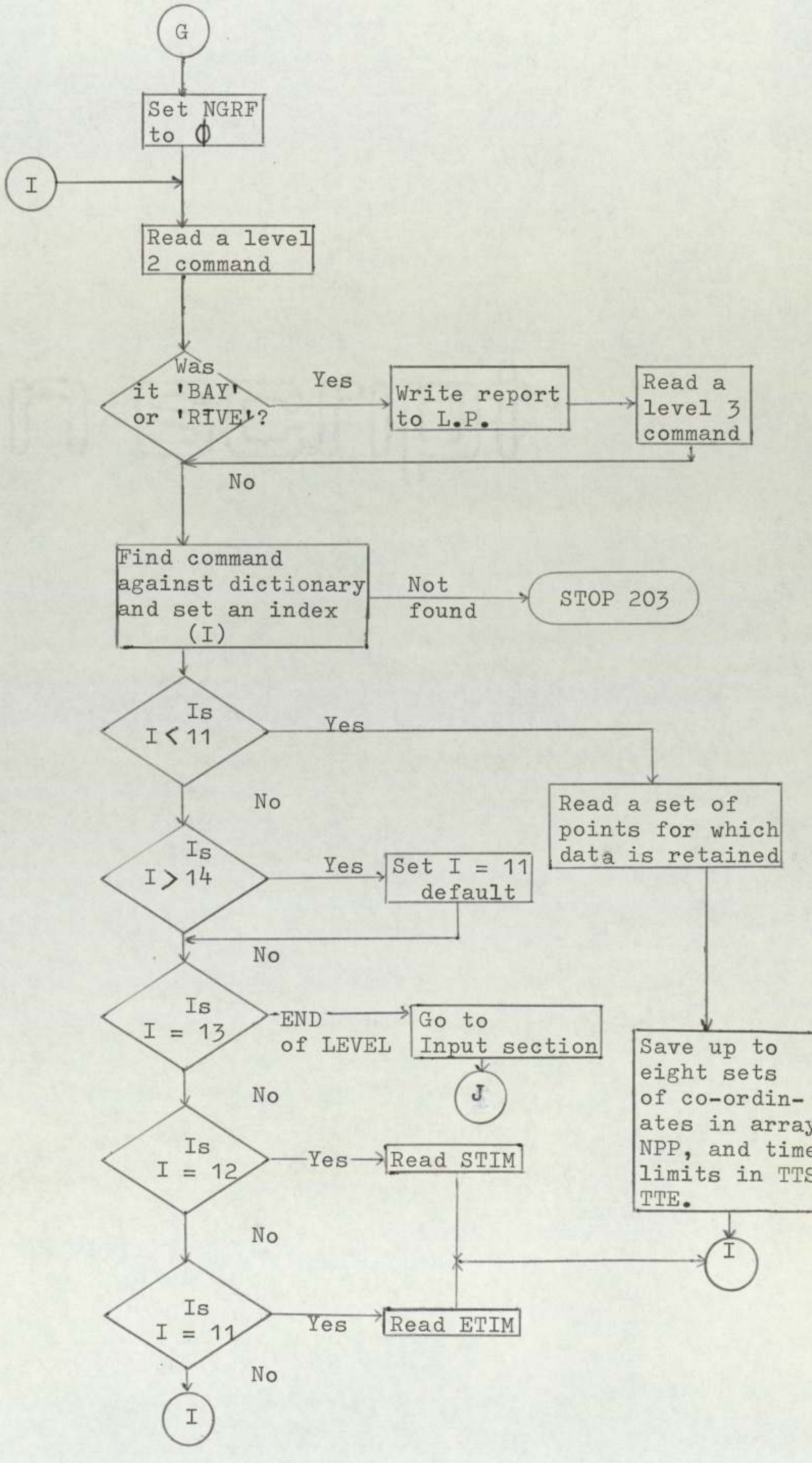


'GRAP'h command

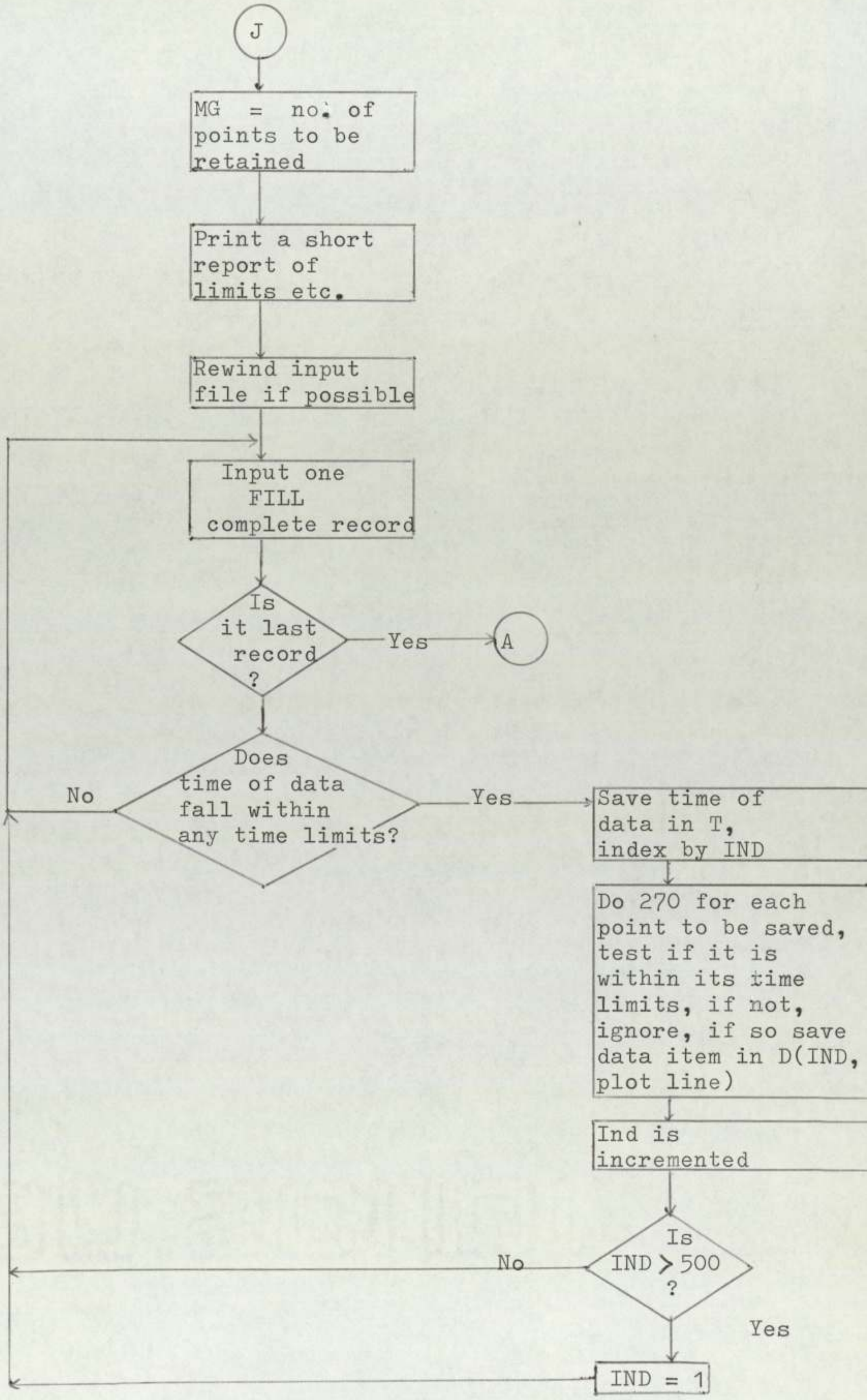
Input parameters for general plotting routine through Level 1 commands



'DATA' Commands for delimiting the data to be retained for plotting

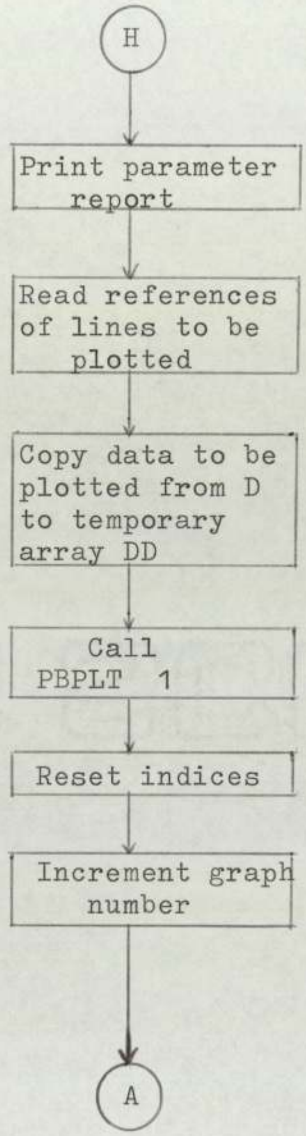


Section to input data
from F1/F2 output file



'PLOT' Command

To plot a selected subset of points retained through 'DATA' command



Subroutine BAYP

Function	:	To control a whole bay data display as a subtask of FISHPT. It duplicates much of FISHPT but then has an easy conversion to be redesigned as an individual program if required.
System Functions	:	READ/WRITE, I/O, END & ERR options, SQRT, TIME
User Functions	:	PLOTH, PLOT, INP (in file FILL)
Calling Statement	:	CALL BAYP (NGPT, MMAX1T, NMAX2T, MMAX2T)
Common Areas	:	/SPECS/, /CTCH22/, /IO/
Common Variables Reset	:	Area /SPECS/, is read in if bay outline is required, NOBD, MBD in /CTCH22/ used as work areas, /IO/, has some initialisation for I/O peripherals.
Error messages	:	STOP 2 No bay data present, model F1 probably input file STOP 4444 Command out of range

---

Description

The calling parameters are, in the order of the calling list:

(A) Number of internal grid points per segment
(B) Number of segments
(C) Number of columns in bay
(D) Number of rows in bay

The program tests if these are blank, ie the level ɸ command NOUT has not appeared in FISHPT, and if they are zero they have to be entered from the F1/F2 file.

There are a series of commands to set up plot options:

'VELP' Plot bay velocities
'SEPL' Plot bay elevations
'VELW' Write bay velocities to line printer
'SEWR' Write bay elevations to line printers
'PLOP' Plot option for routine PLOTH. Next card has a digit in range 1 to 5.
'ENDB' End of bay plot (write commands)

Up to 5 of these commands will usually occur. Initially all of these options are switched off and 'PLOP' set to ϕ. The first occurrence of any of the commands switch the option on, subsequent occurrences of the same command will always reverse the current state of the switch. If any of the plot options are on, 'PLOP' must occur as a command. 'ENDB' command must terminate the command section.

The next record must contain some of

NO the data set reference number of the file from F1/F2, if NOUT has not appeared in routine FISHPT.

NEST Total number of points required to specify the estuary outline (up to 52 points, ie 26 X-Y co-ordinates allowed)

EST(I) In sequence the points of the estuary outline, using as many records as required

NOTE: If NEST > 0 then NO must appear, identical to NOUT from FISHPT.

The program tests to see whether the F1/F2 file leader cards with title and problem parameters have already been processed. The run parameters are eventually those variable names from the input list without the final identifier T.

The next record contains the following:

NP the number of time points at which bay data is to be plotted or printed. Up to 50 times allowed in one call of BAYP.

TIMES(I) the NP values of print/plot times

The main input/processing segment starts at the double call to routine INP. At any time there are two sets of data present; U,V,SE for time TIMEA and UP, VP, SEP for TIMEB.

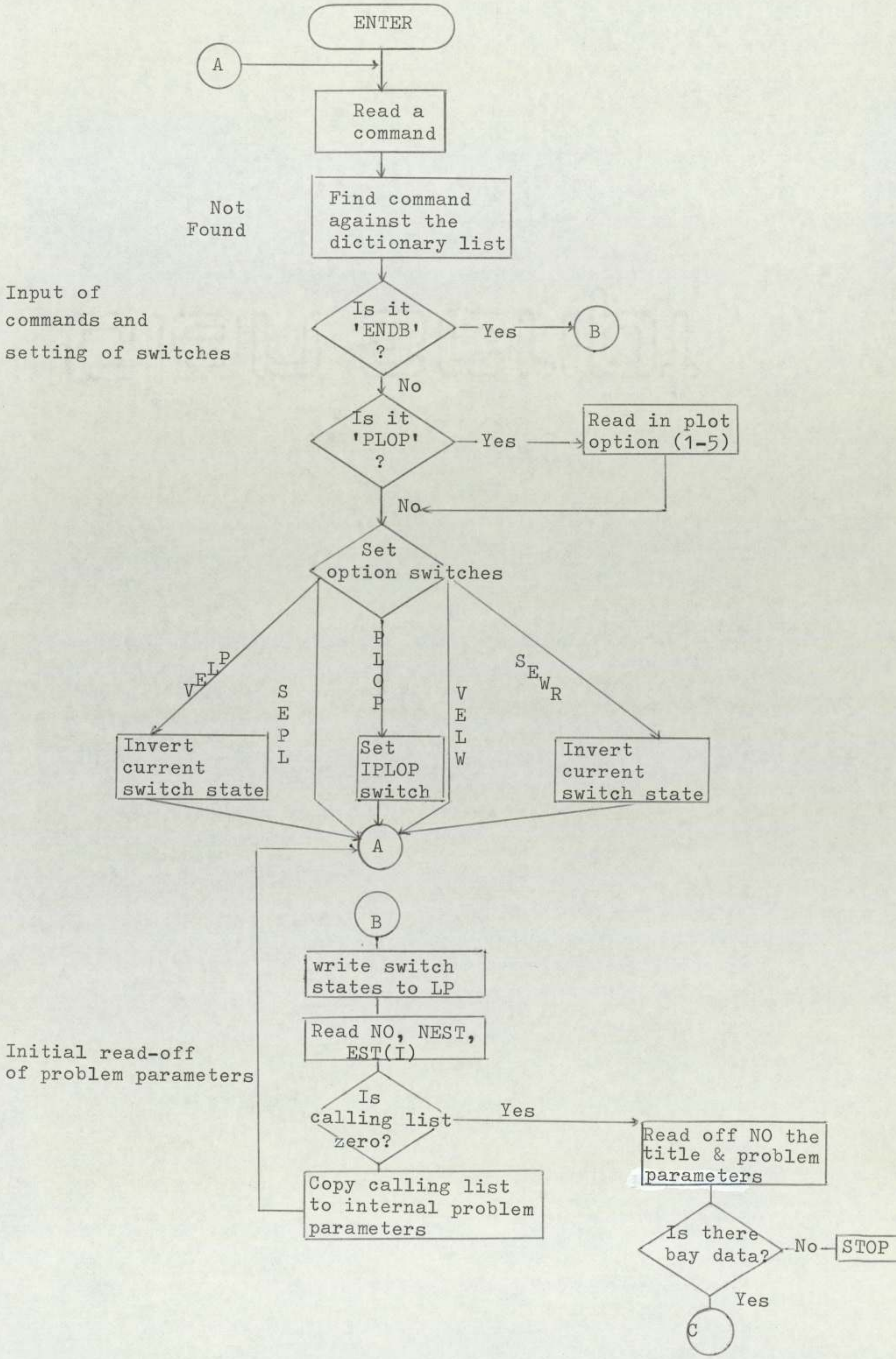
If the next time of print/plot falls within these times, the value of the arrays U, V, SE are interpolated for the time required. If not, the arrays are pushed down and a new time and new arrays read in and tested. Values of the print/plot time must be in ascending temporal order.

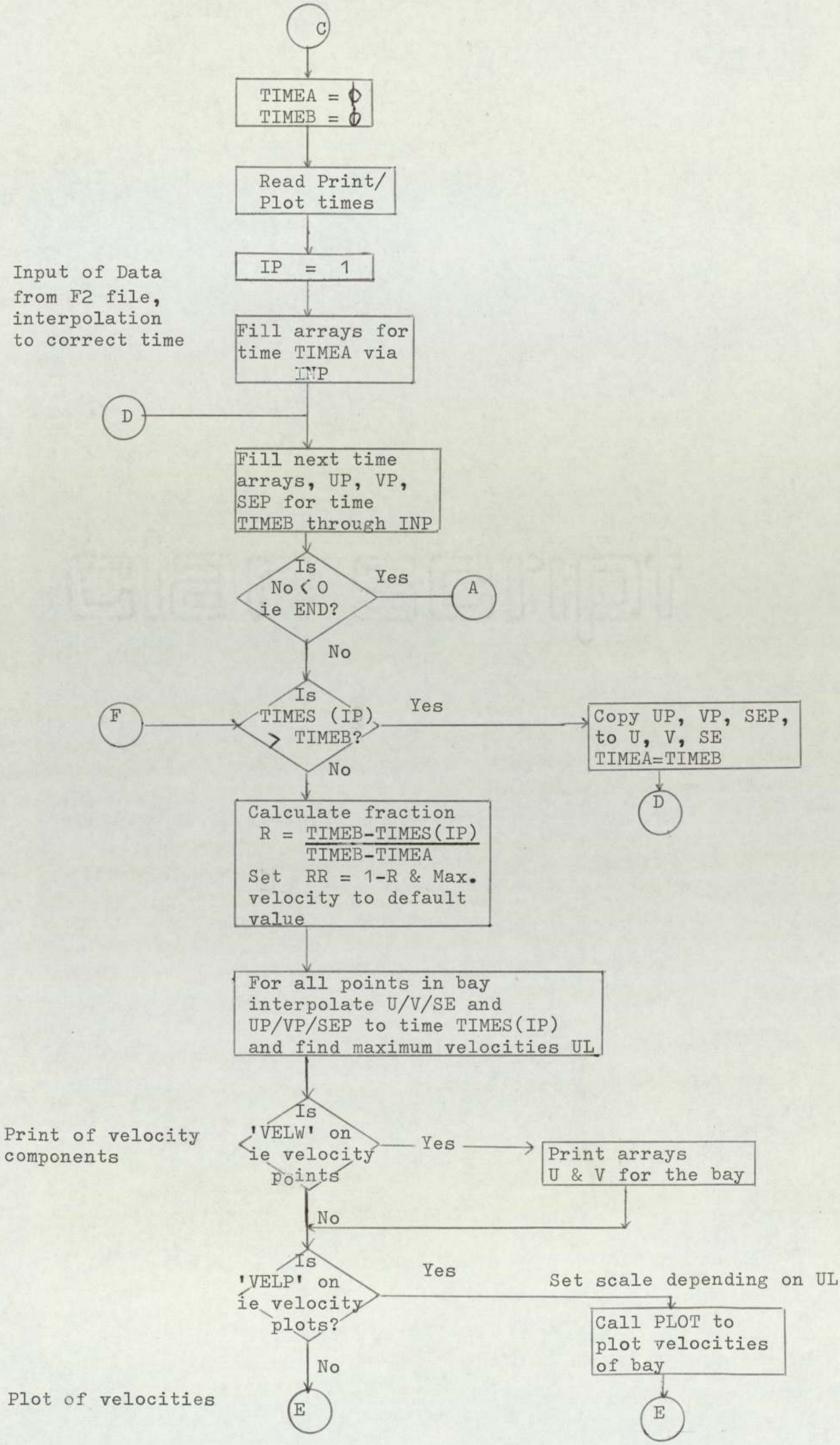
If a time is found between TIMEA and TIMEB and arrays calculated, the arrays are printed if required. Then if the plot options are on, either PLOT is called for velocities, or PLOTH for elevations.

This loop continues until all input has been read (in which case INP returns NO as - NO) or until all print/plot requests have been achieved.

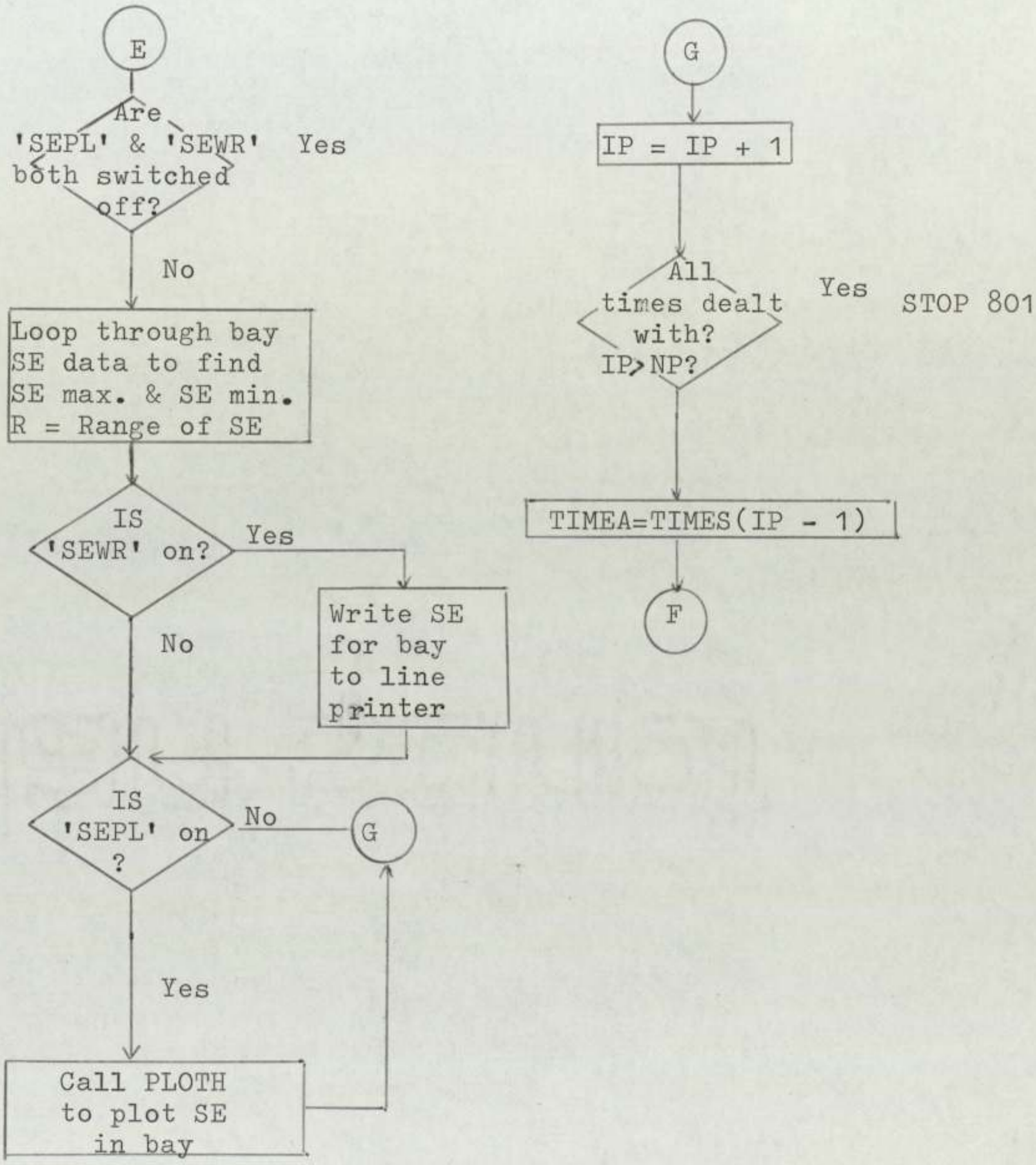
<u>PRINT SEQUENCE</u> for each column, five lines are printed
Line 1 - column number Line 2 - magnitude of velocity
Line 3 - Direction of velocity Line 4 - U velocity component
Line 5 - V velocity component

Flow Logic


ENTER
A
Read a command
Not Found
Find command against the dictionary list
Input of commands and setting of switches
Is it 'ENDB' ?
Yes
B
No
Is it 'PLOP' ?
Yes
Read in plot option (1-5)
No
Set option switches
VELP
SEPL
PLOP
VELW
SEWR
Invert current switch state
Set IPLOP switch
Invert current switch state
A
B
write switch states to LP
Read NO, NEST, EST(I)
Initial read-off of problem parameters
Is calling list zero?
Yes
Read off NO the title & problem parameters
Copy calling list to internal problem parameters
Is there bay data?
No
STOP
Yes
C


C
TIMEA = 0
TIMEB = 0
Read Print/
Plot times
IP = 1
Input of Data
from F2 file,
interpolation
to correct time
Fill arrays for
time TIMEA via
INP
D
Fill next time
arrays, UP, VP,
SEP for time
TIMEB through INP
Is
No < 0
ie END?
Yes
A
No
F
Is
TIMES (IP)
> TIMEB?
Yes
Copy UP, VP, SEP,
to U, V, SE
TIMEA=TIMEB
D
No
Calculate fraction
R = TIMEB-TIMES(IP) / TIMEB-TIMEA
Set RR = 1-R & Max.
velocity to default
value
For all points in bay
interpolate U/V/SE and
UP/VP/SEP to time TIMES(IP)
and find maximum velocities UL
Print of velocity
components
Is
'VELW' on
ie velocity
points
Yes
Print arrays
U & V for the bay
No
Is
'VELP' on
ie velocity
plots?
Yes
Set scale depending on UL
Call PLOT to
plot velocities
of bay
No
Plot of velocities
E
E

Elevation
Print and Plot
options
Updating TIMES
indexing.



Subroutine PLOTH

Function	:	To provide options for plotting elevation data in bay.
System Functions	:	GRAFOR, COS, SIN, FLOAT
User Functions	:	INTP, PLOT4
Calling Statement	:	CALL PLOTH (N, M, H, T, SU, SL, IBOXOP)
Common Areas	:	/SPECS/
Common Variables Reset	:	None
References	:	Users Guide to Grafor, University of Bath, Computer Unit Handbook

---

Description

There are 5 options, indicated by the values of IBOXOP = 0,1,2,3 or 4

IBOXOP	DESCRIPTION OF OPTION
0	A box is drawn to represent the bay/elevation axis and each column is plotted in perspective.
1	Similar to ϕ but lines are drawn every ϕ.1 columns and points calculated every ϕ.1 row. Takes much longer to prepare in run time terms.
2	Plot bay by rows, each row having a separate axis and divisions are referred to a local zero only
3	Similar to option 1 but a non linear interpolation makes the resultant curve smoother
4	The graph is layed down and +ve y is to east, +ve x to south. It is a smaller plot to consider how data changes from row to row. The plot is prepared in routine PLOT4.

On the first call to the routine for some of the options the start/end of actual data in the bay is retained to allow for variable geometry and also make the plotting algorithum more efficient.

The projection angle is PA and set implicitly at the start of the routine. The constants COSPA = COS(PA) and SINPA= SIN(PA). The plot options ϕ, 1 and 3 are scaled to FLCC inches in the x, FLC inches in the y (vertical) direction. Currently PA = 15°, a fairly low projection, ie a low profile/viewing point. The arrays ICS and ICE hold, in element I, the first row and last row of bay data for that column. They are calculated on the first call only. Much of the detailed calculation is as a consequence of preparing parameters for transfer to GRAFOR, and as this is a specific routine for the ICL4-70 CALCOMP only an outline will be given.

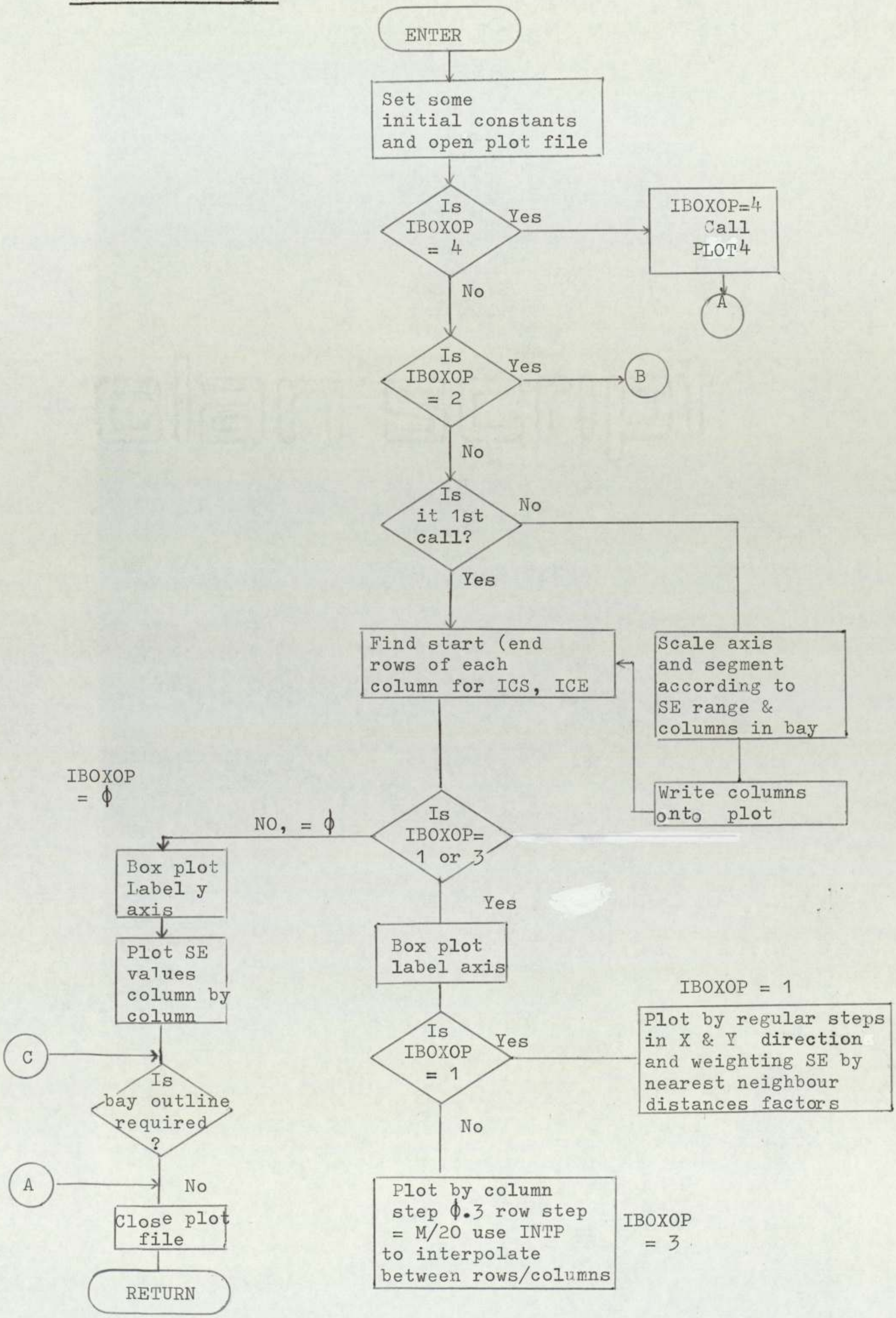
The calling parameters are

N, M, H : H(N,M) is the bay array with row/column dimensions like that of the bay, holding elevation data.

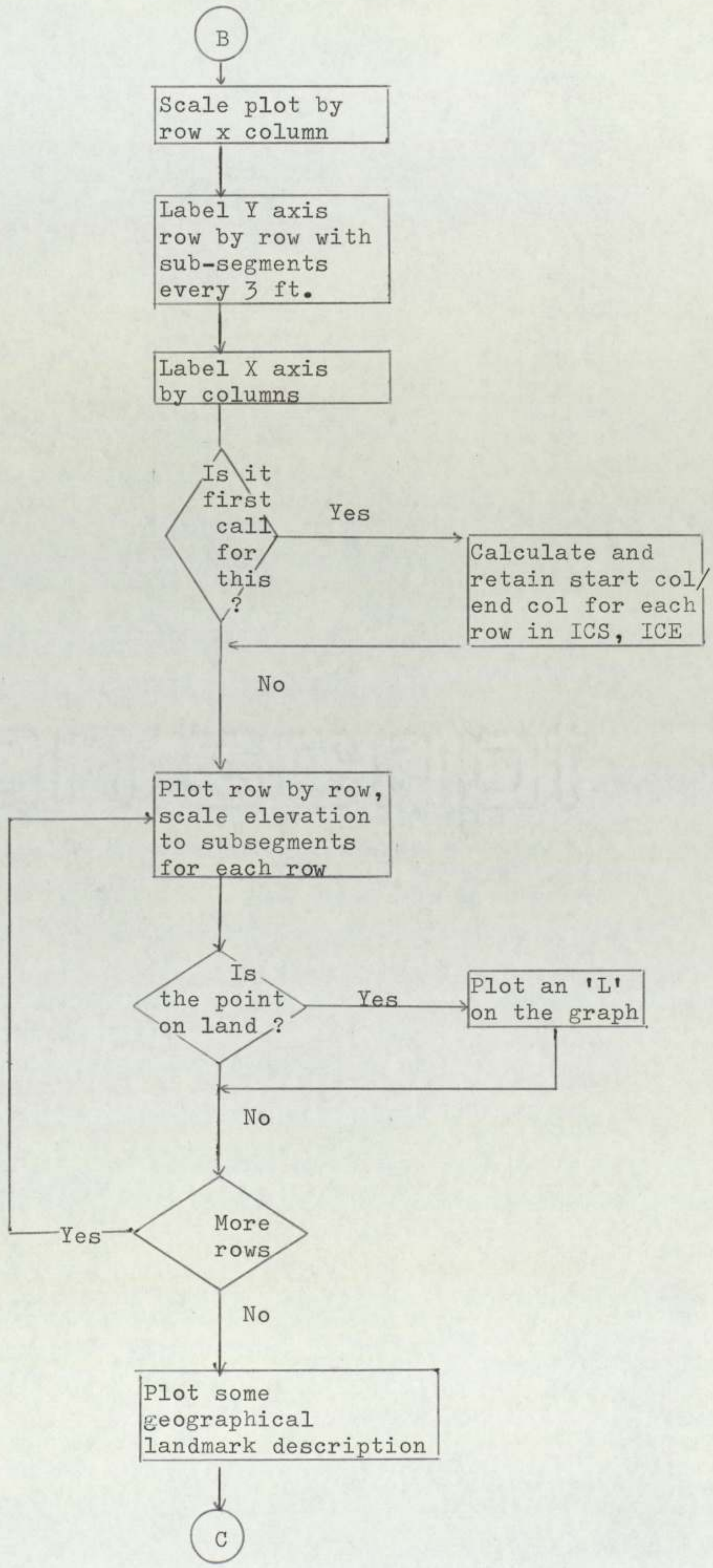
T is the time of the data in H in hours

SU/SL is the maximum / minimum elevation found in array SE.

Outline Flow Logic



IBOXOP = 2


B
Scale plot by row x column
Label Y axis row by row with sub-segments every 3 ft.
Label X axis by columns
Is it first call for this ?
Yes
Calculate and retain start col/ end col for each row in ICS, ICE
No
Plot row by row, scale elevation to subsegments for each row
Is the point on land ?
Yes
Plot an 'L' on the graph
No
More rows
Yes
No
Plot some geographical landmark description
C

Subroutine Plot 4

Function	:	To control option 5 in routine BAYP for plot of bay elevation data
System Functions	:	GRAFOR
User Functions	:	None
Calling Statement	:	CALL PLOT 4 (N, M, H, T, SU, SL)
Common Areas	:	/SPECS/
Common Variables Reset	:	None

Description

Calling parameter list as in BAYP call to PLOTH. This option is a condensed form with the rows running north-south on the graph plot paper. No labelling is done other than identifying rows/columns and the plot is scaled to 4"x8". The estuary outline can be overlayed if required. The plot sequence is to fix a column, plot the row, repeat for all columns.

F.4

PROGRAM SUITE FOR

LEXICAL ANALYSIS OF

OBJECT DECKS.

INTRODUCTION

THIS SUITE OF PROGRAMS ARE DESIGNED TO ASSIST THE PROGRAMMER BY ANALYSING VARIABLE LENGTH CHARACTER STRINGS ON SOURCE STATEMENTS, TO ASSIST DATA PREPARATION,COLLATION AND DOCUMENTATION.

OUTLINE

The package consists of 13 routines, either independant removable sections, or subroutines bound to another higher level routine. The three initial advantages through use were:

1. Complete lexical analysis of object decks improved error detection
2. Use of BLOWUP reduced card usage by 65 per cent
3. Docu mentation standards can be maintained easily and variations introduced into the source deck with minimum error.

It was written for an ICL4-70 using MJ but a version for 1904 Fortran now exists.

SUITE COMPOSITION

ROUTINE NAME	LEVEL	FUNCTION	REQUIRES ROUTINES
1. XTSD	Program	To input a high level language source deck and analyse where variable length source strings occur and recur. Tabulation in order of first occurrence by card number	FILL72 PA (optional)
2. XTDD	Program	To input an 80 column data deck and analyse occurrence of strings on it. No fixed delimiters but to recognise split strings, an intervening space is required. Explanatory alphanumeric characters are also included. It is a version of XTSD, but slightly quicker	FILL 8Ø PA (optional)
3. FILL72	Subroutine	To input a 72 character high level source string from an input card file. Sets up a buffer with required format for XTSD. Returns number of characters in card used.	NUM
4. FILL80	Subroutine	Similar to FILL72 except it reads an 80 column card, usually a data file. No delineator tests	NUM
5. PA	Subroutine	To print arrays used in programs in this suite for aid in possible error analysis	
6. NUM	Integer Function	Returns a non zero value if a specified column contains a numeric character	
7. IEQUAL	Integer Function	Returns non zero value if two specified strings are strictly equal	
8. ALPHAS	Subroutine	To execute and sort on a variable string length array with pointers set up in routines XTSD & XTDD	PA (optional)

ROUTINE NAME				REQUIRES ROUTINES
9.	BLOWUP	Program	To allow FORTRAN or fixed format data to be punched on packed format	
10.	CODECN	Program	To convert ELLIOTT 803 code from paper tape input to E.B.D.I.C. Code	BIPIN
11.	ALPHAN	Program	To analyse column of files for occurrence of certain characters.	
12.	ALPHAT	Subroutine	To execute a sort on character string using binary pattern as integer equivalent. Slightly faster than ALPHAS.	PA (optional)
13.	POLISH	Program	To convert a standard arithmetic statement into Polish Notation.	

PROGRAM XTSD

FUNCTION

To read a high level language source deck, disseminate it into logically seperate strings, build up a table of cross references of where strings occur, then sort this table into order and print it down. Assissts in documentation and debugging.

METHOD

The logic is built around the handling of 2 arrays, SYM and TRE . Both are set to 5000 words length,and a work area uses a further 400 words. The array SYM has a mixed content of integers and characters,and it holds the strings found in the input. This array is structured as follows :

Element I	Length of string to follow (say L)
" I+1	1st character of string
" I+L	Last character of string
" I+L+1	Element of TRE containing first card number where string occurs.

The first element of SYM is then a numeric length and is used to initiate the search pattern.

<u>TRE</u> This array contains the cross referencing card numbers so that

Element I	Card number where string occurs
I+1	Element of TRE where card number of next occurrence of string occurs.

There are two common areas, named

/AA/ Contains only the array A

/E/ Contains the end of input file flag ENDL and current card counter

A data statement sets all values in SYM & TRE to a negative number to indicate absence of valid strings and cross references. ISYM & ITRE are pointers always pointing to the next free element of SYM & TRE arrays respectively. The end of input flag is set to .FALSE., the card counter to zero and NLIM to a little less than the array limits.

Should the pointers exceed NLIM at any stage, string tabulation and searches are abandoned and existing data printed as if end of input file is reached. Then pointers and arrays are reset and input continued. This allows any size file to be analysed by automatic segmentation into blocks. If one complete listing is required the store requirements would become unreasonable.

Statement 10 calls the FILL72 Subroutine. The routine fills the common area /AA/ and set the value of N to point to the first space after the final string.

Statements 14 to 18 ensure that the first element in A is a blank.

The next section processes elements 1 to N now in the form blank-string-blank-string etc and inserts lengths of strings into the element preceeding the string. This element then contains a numeric value.

The next section (statement numbers 809-850) analyse the card to see if any strings within the card are equal. As a multiple occurrence should only generate one reference, elimination of the situation at this stage will create considerable economies later.

Primarily, the lengths of respective strings are tested. Only if these are equal and the length is greater than one is there any need to go to a do loop.

The card is now in the form

lengthA - stringA - lengthB - stringB .... lengthx - stringx

where no string is equal to any other string on the same card image

A search of SYM now has to be made for each string on the card image. As SYM soon becomes much larger than A, it is desirable to only search SYM once and complete multiple passes on A for string equality. ISS is the pointer to the length element in SYM array, so SYM(ISS) is length of string. The string then occurs in element ISS+1 to ISS+1+ (SYM(ISS)-1). ICDR is pointing to the length element of the input image A.

The entire card image is tested to see if any of the individual strings are identical to the SYM string.

Whenever such an equality condition is met, control drops down to statement no. 42.

The statements after 42 perform two functions; one is to search along array TRE to link the current card number with previous references to the same string. The second function is to delete this string from the array A and collapse the rest of the card image to continue the image format (statements 52 to 62). This could possibly mean that the card is subsequently empty, in which case a further input is requested through FILL72.

Should the string not match the established string in SYM, the SYM pointers are advanced and the tests repeated. Once the entire symbol table has been searched, any remains in A are strings that have not occurred in the symbol table. Also, TRE array is indexed with the current card image sequence number.

At relevant times, the pointers are tested to see if they are going out of range. If they are, control jumps to statement 8ØØ1.

Statement numbers 8000 to 150 are the output phase of the tabulation. For each string in the symbol table, the card references are collected in array W. If there are more than 400 references in any block of 5000 symbol strings, only the last 400 occurrences will be printed. 40 strings are printed per page. To save output, if a string is less than 8 characters, then the references are printed on the same line.

The execution time is now printed (Mill time in E.T.U.'s).

If this is the final block, execution can now cease and STOP 5001 marks a normal EOJ condition.

Should the print out only be because the pointers are going out of range, the indeces of SYM & TRE are reset to numeric -1111 and control loops to the card analysis starting at statement number 25. This way the card being analysed at the time of going out of range will provide a continuity link between two segments.

TIME: Over a test of 500 source cards, total run time was 245 etu's or 61 seconds CPU time on a 4-70 (this includes an amount of I/O bound phases).

PROGRAM XTDD

FUNCTION

To read a data deck ,usually of 80 characters,and to disseminate into strings,and record where identical strings occur. In many respects identical to XTSD. The strings of input have to be spaced by one column,leading and trailing zeroes are included in string structures.

METHOD

Identical to that employed in XTSD except that there are far fewer delimiters required, unless the input is of a structured nature (eg index texts).

POSSIBLE MODIFICATIONS TO

XTSD & XTDD

1. STRING SORT In the current versions, the symbol table is printed in order of their occurrence. If desired, a call can be made to the routine ALPHAS just after statement 8ØØØ. This will use a work array of length comparable to the length of SYM. This routine will perform a variable length alphanumeric string sort as defined by a data list internal to ALPHAS. The time overhead is in the region of 30% and in testing XTSD & XTDD was not included. Testing on ALPHAS was run independantly. If this option is included, the strings in SYM are returned in alphabetic order. Non alphabetic symbols are strictly of a lower priority. See ALPHAS for other details.
2. Other Length Records The basic modification that has to be made is the size of the array A in common area /AA/. Some modification will be necessary to input/output strings to take maximum sizes into account.
3. Smaller Core Store Version Savings on core store can be made in two ways: do not include the ALPHAS routine, and reduce the size of SYM & TRE to largest possible bearing in mind the system limits.
4. Traced Version Should errors develop, the user may wish a trace dump. To do this, search through the master file and where :CALL PA(: replace, in column 1, a letter C by a blank. This will activate relevant calls of PA at all critical phases of the program. It should be noted that this will generate much information and the job become I/O bound with times of runs about 4 times that of normal version.
5. Full Tabulation If it is desired to also see how often a string occurs on one card, this can be achieved by deleting the entire section looking for identical strings. This will produce an occurrence tabulation for every occurrence of a string, even if it occurs more than once per card.

If this is done, it is advisable to extend size of TRE as practice shows about 50% more tabulations are made in this case.

Subroutine FILL72

FUNCTION

The routine reads in a standard 72 column record of a language deck. The current version is set up to input a FORTRAN IV deck.Some preliminary functions are performed on the record prior to analysis by XTSD.

METHOD

The array uses two common areas, /A/ to hold the input buffer A and /E/ to hold ENDL, IC (the end of file flag and current card counter). DLIM is a 12 element array allowing up to twelve delimiters to be declared. For the test version on Fortran IV, the following set were defined.

( ) = * ** - , ' / ' + (** is defined through *)

NLIM is set to the number of delimiters defined.

A card image is input. If this read attempt meets the EOF, control jumps to 99. The input file is printed straightaway. A control is kept on cards printed to ensure adequate paging control. A test is made to see if this is a comment card (is the first character a 'C' in the case of FORTRAN IV). If so, no buffering is needed and the next card can be input. Also, a test is needed to remove a possible continuation character (in column 6 in case of FORTRAN IV). The first 'DO' loop searches from column 72 in to find the last column used. The 'DO' 15 loop searches for delimiters. If any are encountered, they are replaced by blanks on the image. If the delimiter encountered is a point (.) then function INUM is called to see if the preceeding or next characters are numeric. If so, it can be assumed that the point is used in the context of a decimal point and so not as a deliniter.

The final section buffers the image so that one blank precedes every string and the value of N returned to the calling routine points to the blank following the last string.

The 99 branch sets ENDL to .TRUE. and there is a 100 branch for blank cards which are merely flagged as being blank. As no string analysis is required, control can go internally back to the read statement.

Subroutine FILL80

FUNCTION

The routine fills the first 80 characters of a record into a work area and prepare it for analysis by the program XTDD.

METHOD

The module is a version of FILL72 as the logic is almost identical.It is somewhat quicker as there are no continuations,comments or delimiters as in standard FORTRAN IV, to about 25 per cent of total CPU time. By altering the size of the common area for the work array,and resetting the limit, longer or shorter records may be analysed.

POSSIBLE MODIFICATIONS TO

FILL72 & FILL8Ø

1. Re-definition of delimiters. The current set are specific for FORTRAN IV and may not include delimiters for other languages. To change them the DATA statement needs to be modified and NLIM re-defined to reduce time in the search loop.

2. Re-definition of continuation. The current test (A(6) ≠ blank) is again specific for FORTRAN. The appropriate column needs to insert for other languages. If there is no continuation facility the test can be omitted altogether.

3. Re-definition of non program statements. The current test (is A(1) = letter C?) is for FORTRAN and only the column number and character defining comments needs to be identified.

4. Selective analysis of input records. By variation of the input format (number 4), selective reading of records can be made to perhaps improve efficiency for larger records and large files.

Subroutine PA

FUNCTION

To provide 1. A trace facility for XTSD and XTDD

2. To print a mixture of arrays holding integers/character strings

METHOD

There are 3 types of arrays used in the analysis programs. All are half word in the original versions.

SYM has a structure of length-string-pointer (numeric-alphanumeric variable length-numeric)

A has a structure of length-string (numeric-alphanumeric variable length)

TRE has a structure of card number-pointer (numeric-numeric)

This routine must be able to handle all three types. The variable ITYP is a type code.

The array to be printed is considered as a pattern and the numeric part given a 1 and alphanumeric given at Ø. A single number is built up as follows:-

ITYP	Array contains	Example
0	Pure alphanumeric	Input card buffer array A
1	Pure numeric	'TRE' array
10	Length-string	'A' when ready for analysis in XTSD & XTDD
101	Length-string-pointer	'SYM' array

N should point to the first vacant element after the valid data.

R is an 8 character alphanumeric identifier.

IR is a full word numeric reference.

A is the array (assumed to be half word in the standard version)

The complete calling statement is PA(A,N,ITYP,R,IR)

The two types 'Ø' and '1' are best handled as specific cases and control for these goes to statements 267 and 2Ø respectively.

The types 10 and 101 can be printed in a common section as only the pointer element is different and this is completed in Statements '40' to '3 FORMAT'. If negative numbers appear it will be because addressing is to an alphanumeric value instead of numeric and a branch to DEBUG with suitable error message stops program execution.

As the array is a calling parameter and not common, it can be attached to any of the other modules in the suite.

Integer Function NUM(N)

FUNCTION

To return a value of 1 if the element passed ,A(N), is numeric and zero if it is non numeric. This is an expensive necessity in terms of CPU time but is necessary for the 4-70 system as there is no user controlled de-buffering available, which would allow optional input format structures and so allow this routine to be dispensed with. The 10 arabic numerals are specified. The most common should be specified early on in the defining array. Zero should be in the first element as it is the most commonly used digit.

METHOD

The array A is supplied by a named common area /AA/, and the calling parameter is the element to be inspected.The value is set appropiately. This routine contributes about 5 per cent of the run time and a more efficient version is being developed.

Integer Function IEQUAL

FUNCTION

To return a value of 1 if the string in one array is equal to the second string passed. Else zero is returned.

METHOD

The calling statement parameter list is (L1,IS1, L2,IS2, A1,A2) .A1,A2 are arrays containing strings, and the test is between a string in A1 starting at element IS1 of length L1 and one in A2 commencing at element IS2 of length L2. An initial test of whether L1 = L2 causes most tests to be shorted and return control. This test could be made in the main calling routine at a saving of 1 to 3 per cent of CPU time.

Subroutine ALPHAS

FUNCTION

To sort an array of the type SYM (Program XTDD,XTSD) into order specified by a data statement. Variants on alphabetical order are allowed. This routine requires about the same amount of store as XTSD/2 .

METHOD

All arrays are half word to conserve one store, as one work array of size identical to SYM is required. 'CHAR' is an array containing NCHAR characters (the alphabet - 26, the numerals - 10, characters not defined as delimiters - 5), 41 in the current version.

The 'DO 20' loop takes two elements from the character list and searches along SYM for the identical character. When one is found, the character is replaced by the index it occupied in CHAR array. All strings in SYM are handled so. The initially numeric constituents of SYM remain unaltered. If any characters are not converted a message is output to that effect.

The sort has now been made essentially a numeric sort of variable lengths. Throughout the length and pointers for each string have to be observed.

The first string of SYM is copied into the work array SZ. Then this string is compared against all other strings in SYM. If a string was numerically lower in SYM it was copied into SZ and used as the definitive string. At the end of one pass of SYM the string in SZ is the numerically lowest string found in SYM. This is then deleted from SYM, and SYM collapsed.

Then the first string of SYM is copied into the next area of SZ and a pass of SYM made. The end will see the lowest remaining string in SYM as the second string in SZ. As many passes of SYM have to be made as there are strings, but SYM becomes shorter at each stage so an effective rate of (strings in SYM)/2 passes only are made. SZ is slowly built up in alphabetic sequence (represented by number strings).

The remaining problem is to reconvert the number strings back to alphanumeric strings. This is easily accomplished although it has to be considered that some characters may have retained their bit representation and so a test is made to see if the number string elements fall in 1 to NCHAR as defined for array CHAR.

TIMING: Very variable but adds about 40% to run time of XTSD or XTDD.

APPENDIX TO ALPHAS : HIERARCHY OF SEARCH LISTS

'CHAR' Index	CHAR Element	HIER Rating
1	E	5
2	A	1
3	I	9
4	O	15
5	S	19
6	Ø	27
7	9	7
8	T	20
9	Q	17
10	.	37
11	C	3
12	1	28
13	2	29
14	3	30
15	4	31
16	5	32
17	6	33
18	7	34
19	8	35
20	9	36
21	X	24
22	B	2
23	D	4
24	F	6
25	H	8
26	J	10
27	K	11
28	L	12
29	M	13
30	N	14
31	P	16
32	R	18
33	U	21
34	V	22
35	W	23

CHAR Index	CHAR Element	HIER Rating
36	Y	25
37	Z	26
38	*	38
39	=	39
40	-	40
41	/	41
42	,	42
43	(	43
44	)	44
45	<	45
46	>	46
47	!	47
48	"	48
49	$	49
50	%	50
51	@	51
52	£	52

PROGRAM BLOWUP

FUNCTION

To allow cards to be prepared in a compressed mode and expand them into a logically correct structure for a FORTRAN IV deck or a sparse data deck

METHOD

The first card in the data deck contains program parameters. Column 1 must contain the delimiter symbol. If left blank, the delimiter default symbol is '<'.

The other 79 columns have the following values in order

LIST	If line printer listing of input and output is required, set to 1. Otherwise no line printer file established unless through NOUT.
IN	Input data set reference number (80 column record expected)
NOUT	Output data set reference (80 columns expected)
INDX	Reference number given to first card image (default value 100;000)
ISTP	Increment for card image reference numbers (default value 10)

There are two buffer arrays, GET & PUT. GET is an 80 character holding array that retains the compact card image. PUT is a receiving area holding a maximum of one card image at a time. Continuations need not be written on the input deck as the program will itself handle these.

The current version is written to handle FORTRAN IV compact code. The ANSI standard demands continuation character in col. 6 (up to 19 allowed), label in col. 1-5 and program statements commencing in column 7.

If the array GET contains two consecutive delimiter symbols, the indent demanded by the Standard will be invoked.

Cards are sequentially numbered in col. 73-80 ready for input as source deck.

The method used is a simple one pass scan of GET looking for the deliniting symbol or a card size in excess of the permitted limit.

Finding a delimiter causes the current card image to be dumped to NOUT and a test of the next character will determine start of next card. If more than permitted columns accrue in PUT, a card image is dumped and the branch dealing with continuation is invoked.

<u>TIMING</u> On average 3-4 card images could be punched onto one 80 column card. This implies that as long as the cost of running the program for a program of N FORTRAN card images does not exceed the cost of cards saved, the program is viable.

If £C is cost of cards per 1,000 then the program is viable if and only if

$$\pounds \frac{C \times N}{500} > \pounds(\text{Cost of Run})$$

At University College Cardiff ICL 4-70 the program is viable if 'LIST' is switched off and $N > 50$.

PROGRAM CODECN

FUNCTION

To convert an input paper tape in ELIOT AUTOCODE 803 to standard EBCDIC suitable for the 4-70.

A version of this is available to for conversion to BCD for the 1900 series computers.

METHOD

The code is entered from 5 track paper tape in blocks of 500 symbols into array ID.

In the loop 'DO 100' the code is translated through the use of a defined set of characters in array LT. There are two problems in that the code has a letter shift and numeric shift, so a switch is set to differentiate between the two conditions. The value of input over 29 means a control character is found. This triggers a print and punch (current data set reference numbers 6 & 98) command of the line of converted code so far. The buffering format of the input records is copied to the output file in this way.

Should any character be greater than 2^5-1 (i.e. 31) a print of the whole block is made and execution is stopped.

BIPIN is a systems supplied machine language program to input paper tape in binary representation. The calling parameters are holding array, block input size and 2 for half word option or 4 for full word option.

PROGRAM ALPHAN

FUNCTION

To compute a character occurence record of selected areas in any type of input record. This routine is to detect sporadic characters or misalignment,particularly within formatted data structures.

METHOD

The array CHR allows up to 100 different characters to be stored for analysis. Up to 800 character files can be handled, or more if the read in is selective. The number of characters to be located multiplied by the number of columns to be searched must not exceed 3ØØØ.

The first card of a data input file contains in free format:

NCHR Number of characters to be tested. If program is run frequently on large similar files, these characters could be defined through a data statement.

NFMT Number of words of store required for the format statement, up to 60 allowed (i.e. up to 3 cards to specify an object time format.)

NCOL Number of column to be analysed.

ICR-ILP Input - output data set references.

The next card has in its first NCHR columns the characters to be considered. The most frequently occurring character ought to be entered first as this cuts time for searching considerably.

The next one to three cards contain the object - time formats. Again if this is constant it can be included explicitly at compilation.

For each column, the frequency with which each character occurs is counted. A count is also made of the number of characters not found per column. If an interactive system is available, a section of the program will allow an on line edit of a non located character to be made. When the EOF condition or ERR condition is met, the frequencies are dumped to the line printer.

Subroutine ALPHAT

FUNCTION

The programs XTSD,XTDD produce an array SYM which is to be sorted. This routine will do a fast sort of the array in terms of the binary structure used to store the characters. It is about 30 per cent faster than ALPHAS .

METHOD

The program is identical to ALPHAS for the sort phase, but modifications of the first and last sections allow this routine to be considerably faster than ALPHAS.

In ALPHAS the first step is to replace each character by a priority rating in a list of all characters to be sorted (arrays CHAR, HIER & CHAR2). This allows the user to determine sort order (e.g. reverse alphabetic and forward numeric or ordinary alphabetic and reverse numeric) and the last phase reconverts back to strings from array SZ to SYM.

It is reasonable to assume the majority of use would be for a normal forward sort (i.e. A,B,C .... Z, 0, 1, 2 .... 9, special symbols). Each character is represented by a bit formation specific to the code of the system running the program. Whatever size word is used, only one byte stores the character, the others all filled by an identical filler.

For example, on a 4-70 or a 360, it is irrelevant whether the letter X 'D1' (is = 'J') is stored

X '4Ø 4Ø 4Ø D1'	or	X 'D1 4Ø
(full word, right justified)		(half word, left justified)

as all characters are padded simultaneously.

Consider the characters S → Z, +, *, Ø 4 on a 360 machine (table 1)

Table 1 Character Representation on a IBM 360 (EBCDIC)

Character	Max.	Decimal	Bit Representation		Lower eight (HW, L. Justified)		Integer Equivalent (2 bytes)	
S	E2	226	1110	0010	0100	0000	578 56	64
T	E3	227	1110	0011	0100	0000	581 12	64
U	E4	228	1110	0100	0100	0000	583 68	64
V	E5	229	1110	0101	0100	0000	586 24	64
W	E6	230	1110	0110	0100	0000	588 80	64
X	E7	231	1110	0111	0100	0000	591 36	64
Y	E8	232	1110	1000	0100	0000	594 92	64
Z	E9	233	1110	1001	0100	0000	597 48	64
+	4E	78	0100	1110	0100	0000	199 68	64
*	5C	92	0101	1100	0100	0000	235 52	64
Ø	FØ	240	1111	0000	0100	0000	614 40	64
1	F1	241	1111	0010	0100	0000	616 96	64
2	F2	242	1111	0011	0100	0000	619 52	64
3	F3	243	1111	0100	0100	0000	622 08	64
4	F4	244	1111	0101	0100	0000	264 64	64

If the above characters were an exhaustive set, and a sort carried out, the order would be - + - * - S-T-U-V W-X-Y-Z-1-2-3-4, i.e., in increasing order of magnitude. The magnitude of a character is its integer value when the bit configuration storing the character is read as if it were an integer number. On an ICL 4-70 this order of magnitude coincides with the natural order alphabetic - numeric - special symbols. If the code is similar to above, it is possible that the most significant bit (on everywhere except on '+', '*') is used as a sign bit, in which case the rest of the number would be complemented when read and this might alter the sorting order.

Timing.

On extensive tests it appears that for a 4-70, 768k store configuration, a 1,000 character sort takes between 0.5 and 1.0 seconds mill time, depending on number of characters per strings.

F.5 A GRAPHICS PACKAGE.

Introduction.

Since the inception of the project, a series of subroutines have been developed to assist in the production of graphical displays, either via the CALCOMP plotter at Cardiff U.C. or some V.D.U. The main aim of the package is to allow modification of data for display and the overlaying of plots, the repetitious use of standard symbols for plotting and the addition of comments to plots.

Reference

(1) 'Users' Guide to the Multi Job GRAFOR Routine', University of Bath Computer Unit, 1976.

(2) 'GINO-F User Manual', Computer Aided Design Centre, Cambridge.

Structure.

Composed of subroutines which are virtually independent of each other, grouped into files with identifier DISPaa where aa are optional and DISP indicates 'part of DISPlay package'.

Number.	Name of Subroutine.	In File (MJ)	Function of Subroutine
1	CJRC	DISPE	To plot a circle.
2	EXTRME	DISP	To calculate extremes of vectors.
3	GRID	DISP	To lay a grid on a plot
4	PLOTS	DISP	To plot a line.
5	ROTATE	DISP	To rotate about axis
6	ELLIPS	DISPE	To plot an ellipse
7	SPIRAL	DISPE	To plot a spiral
8	SETAXS	DISPAX	To create a new file and set up axis
9	RFLECT	DISPG	To reflect about a line
10	WRIT	DISPA	To write a string onto a plot
11	RANDOM	DISPA	To generate a random vector
12	PRJ3T2	DISPH	To project 3 dimensional data to 2 dimensions
13	POLCAR	DISPH	To convert cartesian to polar co-ordinates.
14	CARPOL	DISPH	To convert polar to cartesian co-ordinates.
15	TRNSFM	DISPLY	To transform data
16	COPY	DISPLY	To copy plotting arrays
17	LETTER	DISPLY	To generate alphabetic characters in blocked form
18	PLOT3	DISPLY	To plot a line (cf. PLOTS)
19	RECTGL	DISPG	To plot a rectangle
20	RVLVE	DISPA	To revolve about axis (cf. ROTATE)
21	PLYGN	DISPA	To plot a polygon

1. Subroutine CIRC (N,X,Y,RADIUS).

Function: Plots a circle of radius 'RADIUS' data units, centred on (X,Y)

Uses: System I/O, FLOAT, GRAFOR.

Description: N is an integer defining the number of incremental steps to be used to achieve a 6.284 radian circle. If N is zero, it is set to 100, if it is negative, it is replaced by its absolute value. The pen is moved in the up position to the point (X+RADIUS,Y) and thepen placed on paper. Then incremental points are plotted until the pen returnes to starting point. A one line message is output to line printer.

2. Subroutine EXTRME (A,N,RM,RN,IP).

Function: Analyses the real vector A, or N elements (N<1000), and returns the maximum value in RM, the minimum value in RN. IP is a six figure integer, the three most significant digits pointing to the index of the maximum value, the three least significant pointing to the minimum value found.

Description: Set maximum, minimum to first element of A, and consequently IP to 001001. Each remaining element of A is then examined and RM,RN,IP redefined as required. If N<2, defaults are set to value addressed by A(1).

3. Subroutine GRID(XS,DE,XST,YS,YE,YST,CROSS).

Function: To lay a grid over in plot in the region (XS,YS) - (XS,YE) - (XE,YE) - (XS,YE) every XST units on the X axis and every YST units on the Y axis. If CROSS is zero or negative, a mesh will be drawn over the specified area, if it is positive only the points of intersections are marked by a vertical cross of 'CROSS' units.

Uses: System I/0, GRAFOR.

Description: The pen is raised and moved to the lower left hand corner of the grid box (XS,YS). The pen then moves up the Y axis either writing a line or drawing crosses on intersection points as determined by the value of 'CROSS'. The routine is written to minimize pen movement and is very fast and a useful aid in graph plotting.

4. Subroutine PLOTS (X,Y,PEN,NPTS).

Function: To plot the line defined by the vector points Xi, Yi, i = 1,NPTS. A third array PEN has integer values of 5 or 6 depending on whether for point X_{j-1}, Y_{j-1} to X_j, Y_j the pen is required in the up (value 5) or in the down/write (value 6) position.

Uses: GRAFOR

Description: Each point is passed to GRAFOR in turn through a 'do' loop, if NPTS <1 then a plot of X(1), Y(1) will be made.

5. Subroutine ROTATE (X,Y,NPTS,THETA).

Function: To consider the points X_i, Y_i and rotate them through 'THETA' radians.

Uses: SIN,COS,System I/0

Description: The values of COS(THETA) and SIN(THETA) are calculated and stored temporaril Then a 'do' loop recalulates each point and the returned arrays are the revolved points. A one line statement to line printer confirms execution. Useful for superimposing or splitting curves on a graph.

6. Subroutine ELLIPS(DC,YC,A,B,THETAS,THETAE,THETAI,NSTDS,ANGINC)

Function: To draw a curve of the form $x^2/a^2 + y^2/b^2 = 1$. To be centred on the point xc,yc or major axis 'A', minor axis 'B'. Inclination to the x axis of 'ANGINC' radians. The curve to be plotted from angle inclination 'THETAS' to 'THETAE' either in NSTPS steps or in incremental 'THETAI' radian steps.

Uses: System I/0, COS,SIN,GRAFOR,ROTATE(see above),FLOAT.

Description: ARCL is an internally supplied function to calculate the arc length between point 0 and 0 + d0. The first point is calculated and the pen moved to it in the up position. The ellipse is calculated as if it has an inclination of 0 radians, then rotated through use of ROTATE. Partial ellipses can be plotted using the angular limits. Useful for outlining areas around a point. 2 lines are output, one with the basic calling parameters, the second to notify successful completion.

7. Subroutine SPIRAL (XC,YC,NREVS,DEG,R0,C1,C2).

Function: To plot a spiral centred on (DC,YC), of 'NREVS' revolutions, one plot point every 'DEG' degrees and following the equation $R = R_o(1+ C_1 0^{C_2})$ in polar co-ordinates.

Uses: System I/0, GRAFOR, COS, SIN.

Description: The angular co-ordinate is set to zero and the pen is moved to (XC,YC). Then R is calculated and pen moved to write line to the appropriate (X,Y) co-ordinate. The angular co-ordinate is incremented until its value exceeds 6.28 * NREVS. Large values may lead to poor estimates of COS/SIN. A brief report of parameters and then the last point plotted is written to the printer to signal correct execution.

8. Subroutine SETAXS

Function: To close any open plot files, create a new file, set up axis etc., or just close last file prior to end of job.

Uses: System I/0, GRAFOR,

Common Areas: /AX/ FMT,XSF,YSF,SLX,SLY,DRAWAX,IORIEN,IED,IWID,ILEN,NXP,NYP,NXN,NYN,IOPT, QUEUE.

Description: When SETAXS is called, all the above parameters in labelled common area /AX/ must be set for the particular application required. For the first call however, an option is provided to use this routine to input the list (if IOPT is set to 0). Then subsequent calls (when IOPT > 0) use the values in the common location.

The input list is:

Card	Col. From	Col. To	Variable Name.	Description.
1	1	80	FMT	Heading in the graph in the form of an object time format.
2	1	10	XSF	Xaxis scale factor, i.e. inches per data unit.
3	1	10	YSF	Y axis scale factor.
4	1	10	SLX	X axis segment length, i.e. number of data units per segment.
5	1	10	SLY	Y axis segment length.
6	1		DRAWAX	Logical variable (T or F) to decide whether axis to be drawn (T) or not drawn (F)
7	1	10	ORIEN	Orientation of axis relative to paper (see ref. [1]
8	1	10	IED	Indent in 10ths of an inch from 'left' hand side of paper.
9	1	10	IWID	Width of plot in inches.
10	1	10	ILEN	Length of plot in inches.
11	1	10	NXP	Number of segments in positive arm of x axis
12	1	10	NXN	" " " " negative " " " "
13	1	10	NYP	" " " " positive " " " "
14	1	10	NYN	" " " " negative " " " "

Additional to the IOPT parameter there is the option QUEUE. If it is zero or positive the plot file, when closed, will be automatically added to the multijob plot queue, and then after plotting automatically deleted. If negative, the file is saved when closed.

If IOPT is set to 99 the only action of the routine is to close the current plot file.

A report of the essential parameters is written to the line printer file. Graphs so created are serially numbered to allow cross reference to the graph plot or visual display.

9. Subroutine RFLECT(X,Y,NPTS,THETA)

Function: To reflect the set of points defined by the pairs X_i, Y_i about a line passing through (0,0) and at an angle of THETA radians with X axis.

Uses: COS, SIN, System I/o

Description: Cos 2θ, Sin 2θ are calculated and stored. Then each set of points is repositioned and the contents of X and Y are the new co-ordinates. A one line message is output on successful completion.

10. Subroutine WRIT (N,R2).

Function: To write a string onto a plot. Depending on N two methods are possible. If N is -1, then R2(1) is the X, R2(2) the Y co-ordinates of the start of the string and the string is is found in R2(3) onwards. If N is positive, then a string is read from a card with it's co-ordinates, in the order Xc, Yc, ISCAL, IWAY. Each parameter to take 5 columns. ISCAL is height of lettering in tenths of an inch, IWAY is orientation (see IORIEN in SETAXS).

Uses: System I/O, GRAFOR

Common: /AX/

Description: Basic two paths as outlined in 'Function'. If a card is input, then the details are copied to the line printer file, but not so if N = -1 and an internal call is made.

11. Subroutine RANDOM (X,NPTS, FACTOR).

Function: To fill the vector X, or NPTS locations with random numbers in the range 0 to FACTOR.

Uses: Double precision, IMFRIN, IMFRDF

Description: The system routine IMFRDF supplies a double precision pseudo random number once initialised. This is achieved by calling IMFRIN only on the first call of the routine RANDOM. 'NPTS' calls of IMFRDF are made, the result of each scaled by FACTOR and then truncated to single precision and stored in X.

12. Subroutine PRJ3T2(X3,Y3,73,X2,Y2,NPTS,THETA)

Function: The project 3 dimensional points ($X3_i$, $Y3_i$, 73_i) to a 2 dimensional plane for plotting or viewing ($X2_i$, $Y2_i$) $X2_1$ X3 and Y2 and Y3 can be coincident, X2 must not coincide with 73. THETA is the angle between the 3 dimensional planes as sketched on a 2 dimensional surface (usually 30^o or $\pi/6$ rads).

Uses: SIN,COS

Description SIN and COS of theta are calculated and stored temporarily, then each 73 co-ordinate is resolved into additional X and Y displacements, the new co-ordinates being stored in X2 and Y2.

13. Subroutine POLCAR (X,Y,R,THETA)

Function: To convert cartesian co-ordinate pairs to polar co-ordinates.

Uses: SQRT, ATAN.

Description: To map (x,y) -> (r, θ), r is always defined as $(x^2 + y^2)^{\frac{1}{2}}$. θ is usually defined as atan (x/y). However if y = 0 this will fail and cause the execution to halt. In this case θ is defined as 0.7854 (i.e. $\pi/4$ rads). If X and Y are zero, tha mapping is direct (0,0) -> (0,0).

14. **Subroutine CARPOL (R,TH,X,Y).**

Function: To convert polar co-ordinates (R,TH) to cartesians (X,Y)

Uses: COS, SIN.

Description: X is defined as R * COS(TH) and Y as R * SIN(TH).

15. **Subroutine TRNSFM (P1, P2, P3, NP1, NP2, NP3, PHASE)**

Function: Arrays P1, P2 define an outline, the subroutine generating an outline P3, being a transformation from P1 to P2 the extent of which is determined by PHASE.

Uses: SQRT, INT, COPY (routine 16 graphic package)

Description: P1, P2, P3 are arrays of dimension (100,3). In each array, the point (i,1) is a pen up (value '5') or pen down (value '6') location, (i,2) the x co-ordinate and (i,3) the y co-ordinate. NP1, NP2, NP3 are the number of points in P1, P2 and P3. NP3 is set on return to either NP1 or NP2. 'PHASE' is a real variable between 0 and 1 to determine the degree of transformation from P1 to P2. If it is $\leq$, P3 = P1; if $\geq$ 1, P3 = P2. If PHASE = 0.5, P3 is a 'halfway' shape between P1 and P2. The new co-ordinates are calculated as follows. "Consider a point in P1. Calculate it's nearest neighbour in space array P2. Move along an imaginary connecting line between these two points depending on the value of PHASE". The complication is for the 'pen up/down' component (i,1). As the transmitted values are either 5 or 6, the final transformed values will be between these limits. By definition, if the final value is <5.5, then the pen remains in the 'up' position, if > 5.5 the pen is brought down to the write position. If PHASE is not >0 or <1, then appropriate calls of 'COPY' can be made.

16. **Subroutine COPY (P,R,N).**

Function: P and R are arrays dimensioned (100,3), the routine copying P(1,1) to P(N,3) into the equivalent elements in array R.

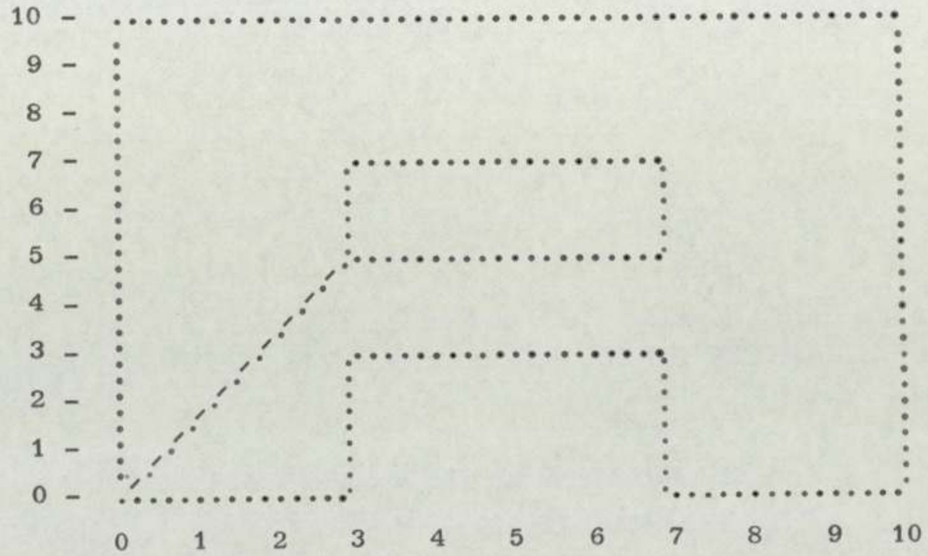
Description: A 'do' loop loops from 1 to N for the main index, with the 3 elements being moved explicitly.

17. Subroutine LETTER (LET,P,NP,XSC,YSC,XC,YC,NDP).

Function: The alphabetic character LET is 'blown up' and defined as a series of plot/jump co-ordinates into array P.

Uses: System I/O

Description: Each alphabetic character is defined on a 10 x 10 grid. For example, the letter A is defined in



The letter 'A' is defined on a 10 x 10 grid.

.....Plotting line

-.-.- Pen 'up'line

14 sets of points (0,0), (3,0), (3,3), (7,3), (7,0), (10,0), (10,10), (0,10), (0,0), jump with pen up to (3,5), then plot (7,5), (7,7), (3,7), (3,5).

On the first call of this routine some pointers are set to make the definition of 960 co-ordinates simpler. An array NCHS is built so that for the i th letter, the co-ordinate data starts at LETCO (K) where K = NCHS(i)*2+1. The array NCHR holds the number of co-ordinate pairs required to define each letter.

On all but the first call, the first action is to test whether LET is a valid letter of the alphabet. If not, a message is output to the line printer and NP set to -99 to indicate an error condition. When the letter is located, the limits of it's co-ordinates are calculated to refer to array LETCO which hold all definitions in vector form.

When the limits (IST to ILST) have been located, they can be copies into array P, dimension (NDP3). Usually though the 10 x 10 grid is not used in actual routines, so the appropriate co-ordinates are scaled by factor XSC or YSC, and displaced by XC and YC before saving in array P.

NP is set to the number of (pen,x,y)sets of data, to be found in array P. The maximum dimension of P has to be transmitted from the main program and this is NDP.

A negative number in a grid definition of a letter implies that the pen should be in the 'up' position when moving to absolute value of this grid point from the previous grid point.

18. <u>Subroutine PLOT3 (P,NP,I1,I2)</u>.

Function: P(NP, 3) is an array of (pen,x,y) values, and it is to be plotted from P(I1,1-2-3) to P(I2, 1-2-3).

Uses: GRAFOR,INT

Description: The appropriate calls to GRAFOR are made via a 'do' loop. The pen position P(i,1) is reset as an integer before the calling list is linked.

19. <u>Subroutine RECTGL(XC,YC,XL,YL,THETA)</u>

Function: To plot a rectangle with bottom left hand corner at (XC,YC), length XL, height YL and inclination THETA radians to the X axis.

Uses: SIN,COS,GRAFOR,System I/O

Description: The pen is moved to the point (XC,YC) while 'up'. Then while down, each of the other three co-ordinates are calculated and the pen moved around in sequence until it is returned to (XC,YC). A one line report goes to the printer file for each call of this routine.

20. Subroutine RVLVE (X,Y,NPTS,THETA).

Function: To revolve the co-ordinate pairs X_i, Y_i, more accurate than 'ROTATE'. The amount of rotation is THETA radians for the first 'NPTS' pairs.

Uses: ATAN, SQRT, COS, SIN.

Description: Converts the pairs to polar co-ordinates, adds an angle, then converts back to cartesian co-ordinates. Slightly longer execution time than 'ROTATE'.

21. Subroutine PLYGN(XS,YS,XLNGTH,NSIDES,ANGLE).

Function: To plot a 'NSIDES' sided polygon of length of sides XLNGTH, with the left hand side of the base line on the point (XS,YS). The base line is at an inclination of 'ANGLE' radians to the x axis.

Uses: FLOAT,COS,SIN,System I/O

Description: The pen is moved to (XS,YS), then the co-ordinates of successive points are calculated and the pen moved respectively. When the pen returns to (XS,YS) a one line report is issued to the line printer file.

F.6 **The Date-Time Package.**

Function: To handle all aspects of date-time computations.

Use: Call to various routines and functions using a 5 member integer array to pass date.

Method: The principal array is a 5 membered integer array holding the date in the form:

YY MM DD HH MM

e.g. midnight on 3rd July, 1965 is

65 07 03 24 00

When reference is made to 'date' in terms of parameters of call or return, a 5 membered integer array is required (usually referred to as ICD - current date)

Summary of options available.

1. REAL FUNCTION DAYMNT(ICD)

   Returns the current date as a fraction of the current month.

2. INTEGER FUNCTION ICONB(ICD,N)

   Generates the YY/MM/DD/HH/MM format from a compound digit YYMMDD. Assumes HH and MM are zero. N is the digit, ICD the return date

3. REAL FUNCTION RESET (ICD,ISET)

   When ISET is -1 the date is stored. Subsequent calls with ISET at zero will return the number of minutes elapsed between current date and last base date retained.

4. INTEGER FUNCTION ICHDT (ICD1,ICD2)

   Tests to see if date 2 is after date 1, and if so swaps them over and returns value -1, else zero.

5. INTEGER FUNCTION IDFDTE (ICD1,ICD2)

   Assuming ICD1 is chronologically after ICD2, returns the time difference between them in minutes.

6. INTEGER FUNCTION IDFDT2(ICD1,ICD2)

Returns difference between two dates as no5., however, if ICD1 is before ICD2 the difference is returned as -ve. The values of ICD1 and ICD2 are preserved.

7. REAL FUNCTION DFDTD(ICD1,ICD2,n)

Returns the difference between two dates irrespective of sequence in hours if N = 1, days if N = 2 or months if N = 3 (1 month = 30.25 days)

8. INTEGER FUNCTION IPLAUS(ICD)

Returns 0 if a date is plausible. If the year is entered with the century, then the century digits are removed.

Error conditions as follows, in a 5 digit return number.

Digit	Value	Error
1 (RHmost digit)	1	Minutes <0
	2	Minutes > 59
2	1	Hours <0
	2	Hours >24
3	1	Days <0
	2	Days >31
	3	Days > days in month
4	1	Month <0
	2	Month >0
5 (LH Most digit)	1	Year < 1960 or 60
	2	Year > 1980 or 80

9. REAL FUNCTION YERPRT(ICD)

Calculates fraction of current year passed at current date.

Returns fraction between 0 and 1.

10. SUBROUTINE ADDMIN(ICD,IDDMIN)

Adds IDDMIN minutes to the current date.

11. SUBROUTINE COPYDT(ICD1,ICD2)

Copies date 1 to area for date 2.

12. REAL FUNCTION HRSDAY(ICD)

Converts HH/MM to fraction of a day.

13. REAL FUNCTION HRSMIN(ICD)

Converts HH/MM to hours and decimal minutes, e.g. 12.30 to 12.5

14. INTEGER FUNCTION ICON(ICD)

Converts the first 3 elements of date (YY/MM/DD) to a 6 figure integer YYMMDD. If year is specified as '19'77 the 19 part is truncated.

15. INTEGER FUNCTION IFDATE(ICD1,ICD2)

Set to +1 if ICD1 is after ICD2

0 if ICD1 = ICD2

-1 if ICD1 is before ICD2

16. INTEGER FUNCTION ICHNDT(ICD,IDDMIN)

Set to 0 normally to return, but set to 1 if the addition of IDDMIN minutes will change the date.

17. INTEGER FUNCTION IMNTH(IM,IY)

IMG the current month (ICD(2)) and IY the year (ICD(1))

Returns the number of days in the current month.

18. SUBROUTINE DATIME(IO)

For 1900 Series only, prints current realtime date and time to top of a new page on peripheral channel no IO. Printed as 'TO-DAYS DATE AND CURRENT TIME = DD/MM/YY - HH/MM/SS'

19. INTEGER FUNCTION ICNVDT(ICD)

Calculates minutes elapsed from start of current year.

20. INTEGER FUNCTION IDATED(ICDS,ICD,ICDE)

Returns 0 if the current date (ICD) does fall within the time interval from ICDS (start-date) to ICDE(end-date).

F.7 Brief Description of additional routines

Contents

1. Program LINREG

Linear Regression Analysis.

Although only two sets of readings are admitted at any one time, the flexibility of the routine have made it very useful for all routine data analysis.The normal statistical parameters for both data sets are computed. The regression line of the y set on the x set is computed , and also the confidence limits on the line using Student t-values.Using the philosophy of the BMD package, Problems and Sub-problems are permitted. A comprehensive trans-generation package is attached to the routine.Input formats are flexible as they are defined at run-time only.There are 9 input variants , briefly :

a. Pairs of x-y values, unsorted, are read in.Internal sort follows.

b. Well ordered x-y pairs are entered. Ascending order.

c. NX values of x are read in, then one line image for each x holds a variable number of associated y values.

d. Each line has an initial x value, followed by an integer indicating the number of associated y values, followed by these y values.May require more than one line image per set.

e. Each x value is followed by the number of y-values and the average y value and its variance.This disposes with the need for an explicit list of y values.

f. Using the current internal data set, transform it using the transgeneration subroutine and repeat all calculations

g. Using the current data, read in a set points to be deleted from the data set as a whole.

h. Using the current data set , delete the maximum deviant from the current regression line and repeat the fit. Comparison of improvement in fitting is easily seen.

i. Add individual points back into the set of data.

2. Subroutine SORT

Sorts at least one array and at most four arrays into ascending or descending order. The algorithm is based on the highly efficient Shell sort method. Reference CACM , July 1959 , page 30. The routine is independant and uses dummy arrays.

3. Subroutine EIGEN

To compute the Eigenvalues and Eigenvectors of a real symmetric matrix. The transferred matri must be real,symmetric and will be destroyed in computation. The method is diagonalisation as defined by Jacobi and adapted by van Neumann . Reference A. Ralston and H.S. Wilf ,John Wiley and Sons, New York , 1961, Chapter 7. One matrix of the same size as the data matrix is required as work space.The eigenvalues are returned in the diagonal elements of the initial data matrix.The work area holds eigenvectors upon return.

4. Subroutine MULTMA

To compute the n^{th} power of a square matrix,m x m . There are also options included to use the routine to set a matrix to unity or clear it.A work array of the same size as the main array is also required.Upon return this array holds the previous convolution. A test program is included and the routine is independant.

5. Program SERIES

To compute power and corss-spectra and correlations of time series. This major suite of routines was to be used on the time-base analysis of data. The source reference is Computer Programs for Spectral Analysis of Economic Time Series, H. F. Karreman .Formats are object-time and multiple series are permitted.Data however must be complete for successful analysis. Apart from statistics , some projection is also possible . The whole suite is run together under a steering segment and requires large core-store.

6. Program CUMULA

Analysis of Cross-sectional data for input to models etc.

Input consists of three phases :

a. A set of cross-section at known distances from a common zero

b. A segmentation scheme , regular or irregular, for the system

c. A tidal profile , or a series of profiles.

Having entered and validated a set of cross-sections, they are sorted into ascending order from the zero of data. The next item should normally be a segmentation scheme, if none is found the estuary is regarded as one segment. Either an integer n is input to specify n+1 equidistant segment boundaries, or an integer -m to specify a table of m+1 boundaries. The latest segmentation remains active until replaced. A series input of tidal profiles are processed as follows :
Interpolation of an order dependant on the accuracy of the profile gives profile heights at cross-section points. This is input to XSCT to give area and width and wetted perimeter. Then by reference to the segmentation scheme interpolation estimates parameter values at boundaries. Using the spacing data for the boundaries, volumes and surface areas are calculated. Uses subroutines SETUP,XSECT,WETTED,CHNWTP,RINTER,INTP,SORT

7. Program TEXT

To compute total travel and tidal displacement of a set of discrete particles.

A set of positions for input of discrete particles at various phases of the tidal cycle are entered.Velocity data is abstracted from the prepared data input for the Stochastic Model from the Fischer Model via routine INPTAP.The progressive positions of the particles are plotted out on the graph plotter. If any actual data is available,this can be superimposed on the plot.
Uses routines GRAFOR,TIDEX,INPTAP and INTP.

F.8 **Program HYDOR.**

Function:	To act as a processing program for cross-sectional data and enable segmentation of a stretch to be carried out.
System Functions:	READ/WRITE I/O
User Functions:	SPLIT, CALC, PRINTX
Error Messages:	A. STOP 1 - Cross sedtions not in sequence B. STOP 12 - More points in cross section than permitted C. 'EXTRAPOLATION NOT ALLOWED', continue to next data.
Common Areas:	None
Common Variables Reset:	None

Description.

Input consists of two phases, one being the basic cross sectional data that is to be used as being representative of the reach, and the other describing the type of segmentation required to be performed by the program.

Because of the variety of data formats used for this kind of data, the first card of input is an object time format statement describing the layout of the x-y data pairs. The first program segment will read in a header card for each cross section, followed by the x-y (distance from bank-depth) data points in the layout previously entered through the object time format.

All cross sections should have common units for distance from bank and depths although they need not be the same. There is - facility for re-zeroing depths to a new datum.

The zero for distances of each cross section should commence at the upstream limit of the reach. These distances are saved in array DIST. Cross-sections must appear in strict sequence moving from upstream to downstream end.

The final cross section is labelled by a card following the useful data, containing "-1 -1 -1".

Two arrays are now established, X and Y, for which the element (I,J) holds the x (distance from bank) and y (depth) characteristics of cross section I, data pair J. These are printed to the line printer NOUT (set to 6 via data statements) via the routing PRINTX.

The next input card reads a set of scaling variables

CONVD to convert the units of distance

CONVDP to convert depths

CONVX to convert distances from bank

NOUT2 additional output stream reference

SHFT to realign depths prior to conversion by CONVDP

The 'DO 900' loop rescales the data according to the above factors. If NOUT2 is non zero, the newly scaled data is copied to that peripheral number (usually for purposes of a hard copy).

The program is now ready to consider the primary options available. The first four columns of the next card must contain either

'TIDE' or

'SEGM'

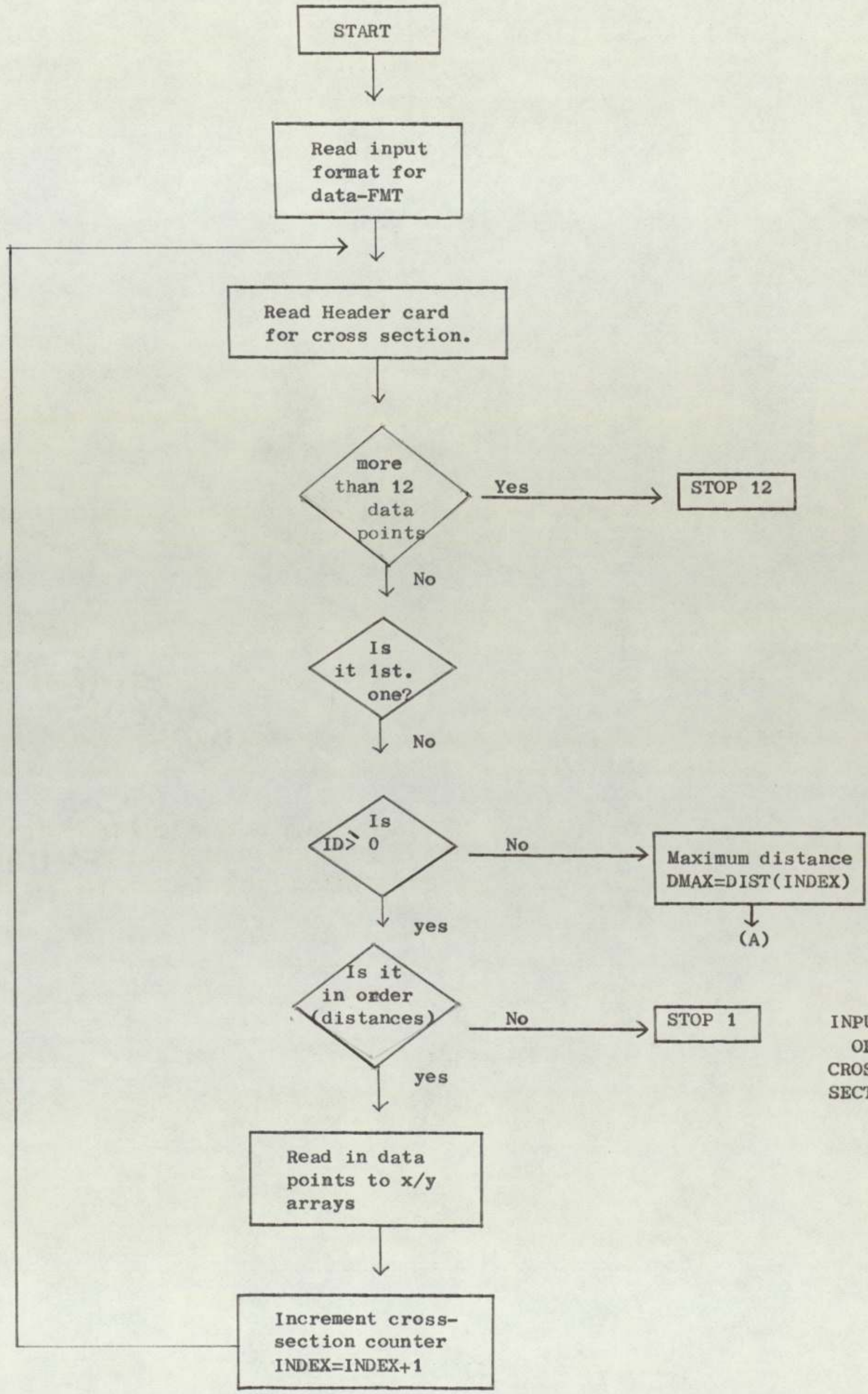
and the first time this card is read, the contents are 'TIDE'. This signals input of a tide profile of M points, each point (X-Y: distance from zero - height of tide) referring to a tide or river height above bed level at a given instant in time.

A set of volume/surface area tables are then calculated through routine 'CALC' and stored.

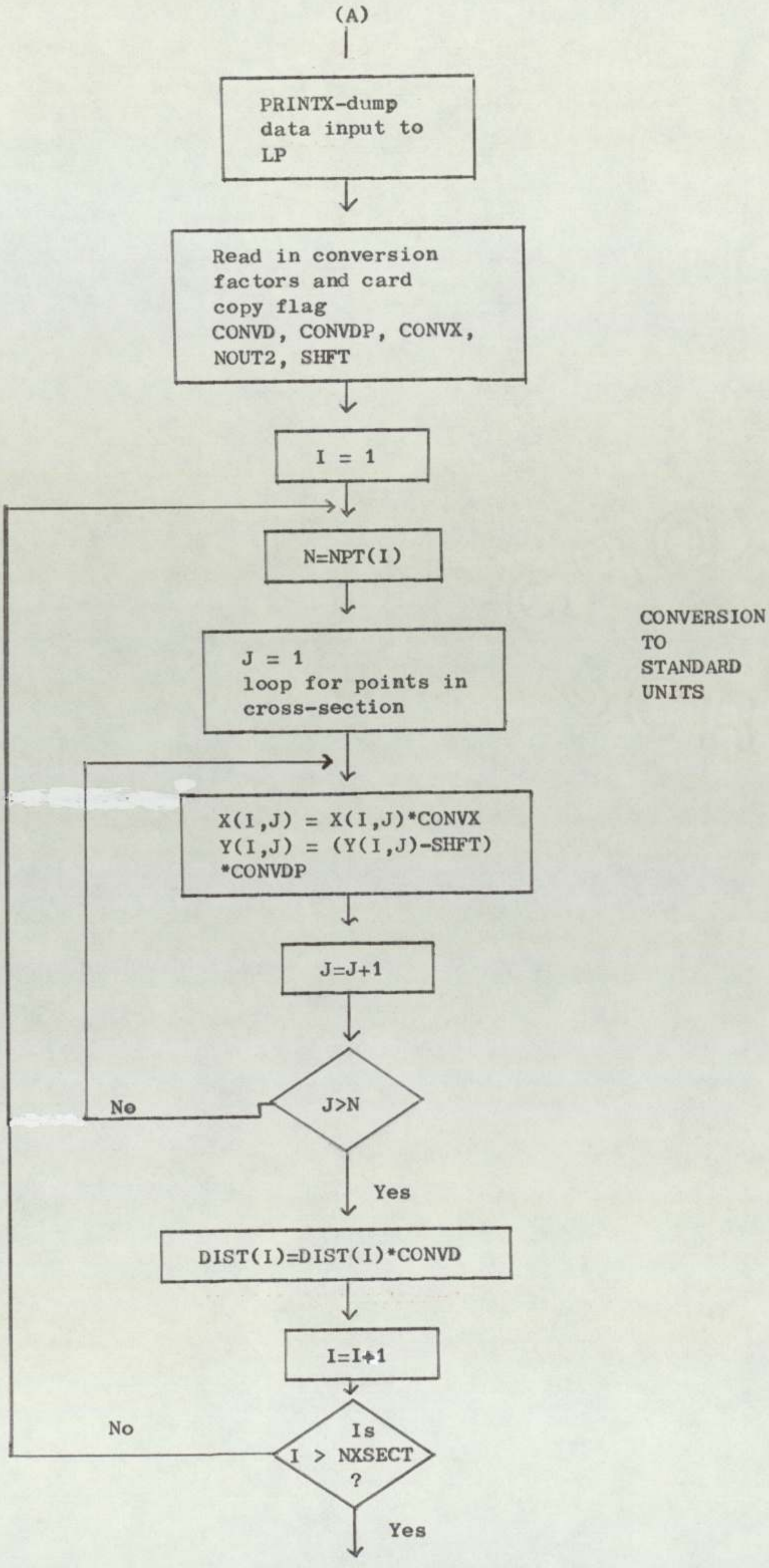
Further options can then be 'SEGM', in which case a set of distances (M of them) are read in and the estuary segmented using these as segment boundaries.

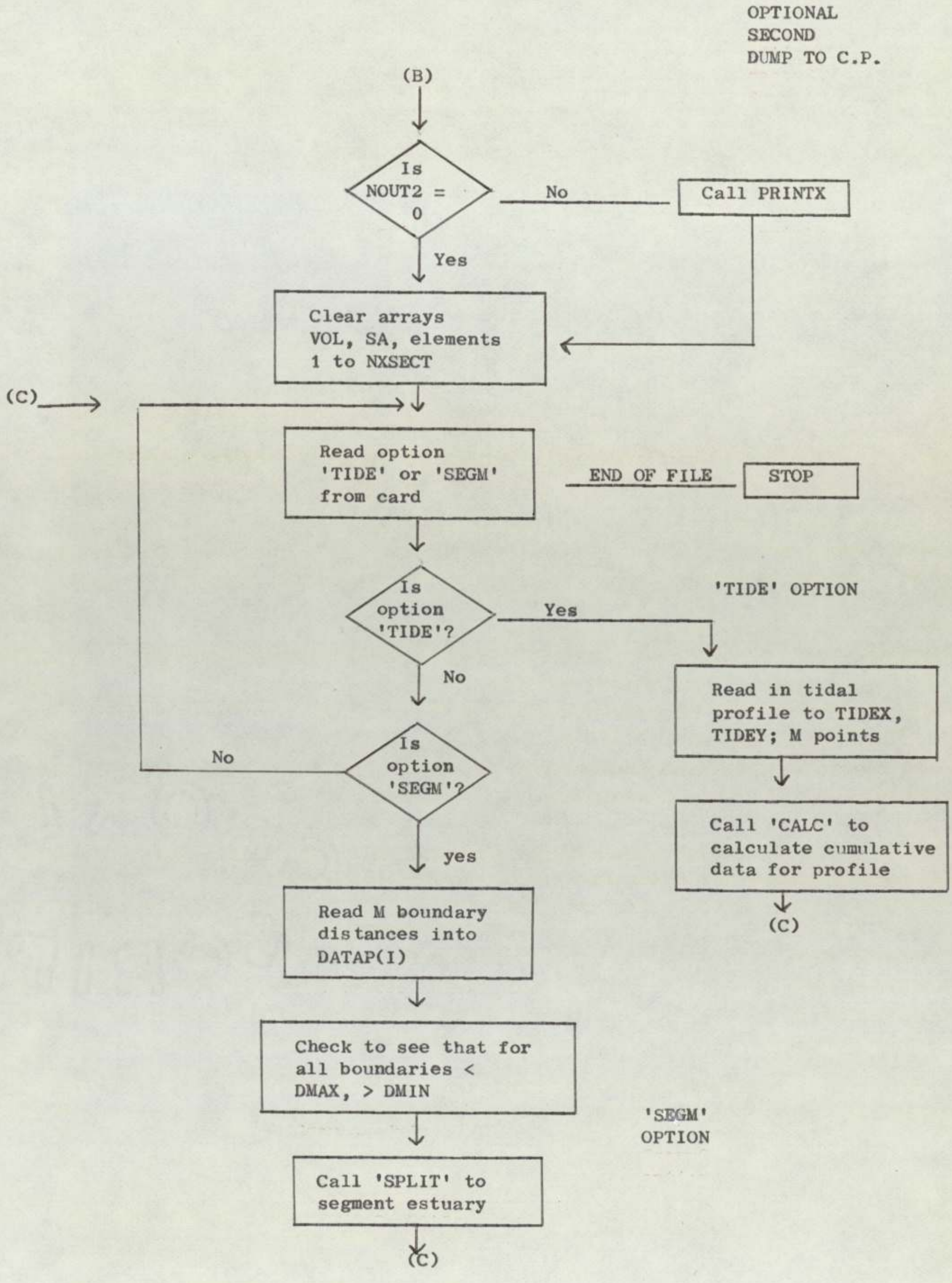
As many 'TIDE' or 'SEGM' options as desired may appear, although the first option must be a 'TIDE' option. If no further data input, normal termination occurs

Flow Logic.



INPUT OF CROSS-SECTIONS


(A)
PRINTX-dump data input to LP
Read in conversion factors and card copy flag CONVD, CONVDP, CONVX, NOUT2, SHFT
I = 1
N=NPT(I)
J = 1 loop for points in cross-section
X(I,J) = X(I,J)*CONVX
Y(I,J) = (Y(I,J)-SHFT) *CONVDP
J=J+1
J>N
No
Yes
DIST(I)=DIST(I)*CONVD
I=I+1
Is I > NXSECT ?
No
Yes
(B)
CONVERSION TO STANDARD UNITS



OPTIONAL
SECOND
DUMP TO C.P.
(B)
Is
NOUT2 =
0
No
Call PRINTX
Yes
Clear arrays
VOL, SA, elements
1 to NXSECT
(C)
Read option
'TIDE' or 'SEGM'
from card
END OF FILE
STOP
Is
option
'TIDE'?
Yes
'TIDE' OPTION
Read in tidal
profile to TIDEX,
TIDEY; M points
Call 'CALC' to
calculate cumulative
data for profile
(C)
No
Is
option
'SEGM'?
No
yes
Read M boundary
distances into
DATAP(I)
Check to see that for
all boundaries <
DMAX, > DMIN
'SEGM'
OPTION
Call 'SPLIT' to
segment estuary
(C)

Data Requirements and Formats for Program HYDOR.

Record	Col from	Col to	Format	Variable	Description
1	1	80	20A4	FMT	Variable object time format.
2	Free			ID NPT(INDEX) DIST(INDEX)	Cross-section identifier Points in cross-section Distance of cross-section from zero.
3	As given by	FMT		X(INDEX,J) Y(INDEX,J) J=1,N	Distance from bank/ depths for N points
*	Record 2 and 3 Repeated for each cross-section				
4	1	9		TEXT ' -1 -1 -1'	End of cross-sections flags
5	Free			CONVD CONVDP CONVX NOUT2 SHFT	Conversion factors for 1.Distances DIST 2.Depths 3.Distances X 4.Alternative output 5.Datum shift for depths
6	1	4	A4	Option:tide	'TIDE' Flags tide profile
7	1	4	A4	Option:tide profile or segmentation	'TIDE' or 'SEGM'
8	Free			M TIDEX(I),TIDEY(I) I = 1, M DATAP(I) I = 1, M	Data points in either a tidal profile or segmented estuary. Tidal profile if option is 'TIDE If segmentation is option.
*	Records 7 and 8 repeated as required.				

Subroutine CALC

Function: To calculate cumulative volumes and surface areas.

System Functions: WRITE

User Functions: SETUP, RINTER, XSECT

Calling Statement: CALL CALC(X, Y, DIST, NXSECT, NPT, TX, TY, M, XF, XB, YF, YB, VOL, SA)

Common Areas: None

Description.

Input list

X, Y are the two 2-dimensional 50x12 arrays holding cross sectional data.

DIST is the distance, from zero, of cross sections array.

NXSECT is the number of cross sections available.

NPT is the array holding data points per cross-section.

TX,TY holding area for tidal profile data pairs : distance-height above bed.

XF,XB,YF,YB work areas of min. dimension MAX(NPT)

VOL array holding cumulative volumes to each cross sectional area.

SA array holding cumulative surface areas to each cross section.

The routine starts between cross-section 1 and 2, the back and forward reference (IF, IB) pointers indicating the location of the method, working from the tidal limit to the seaward limit.

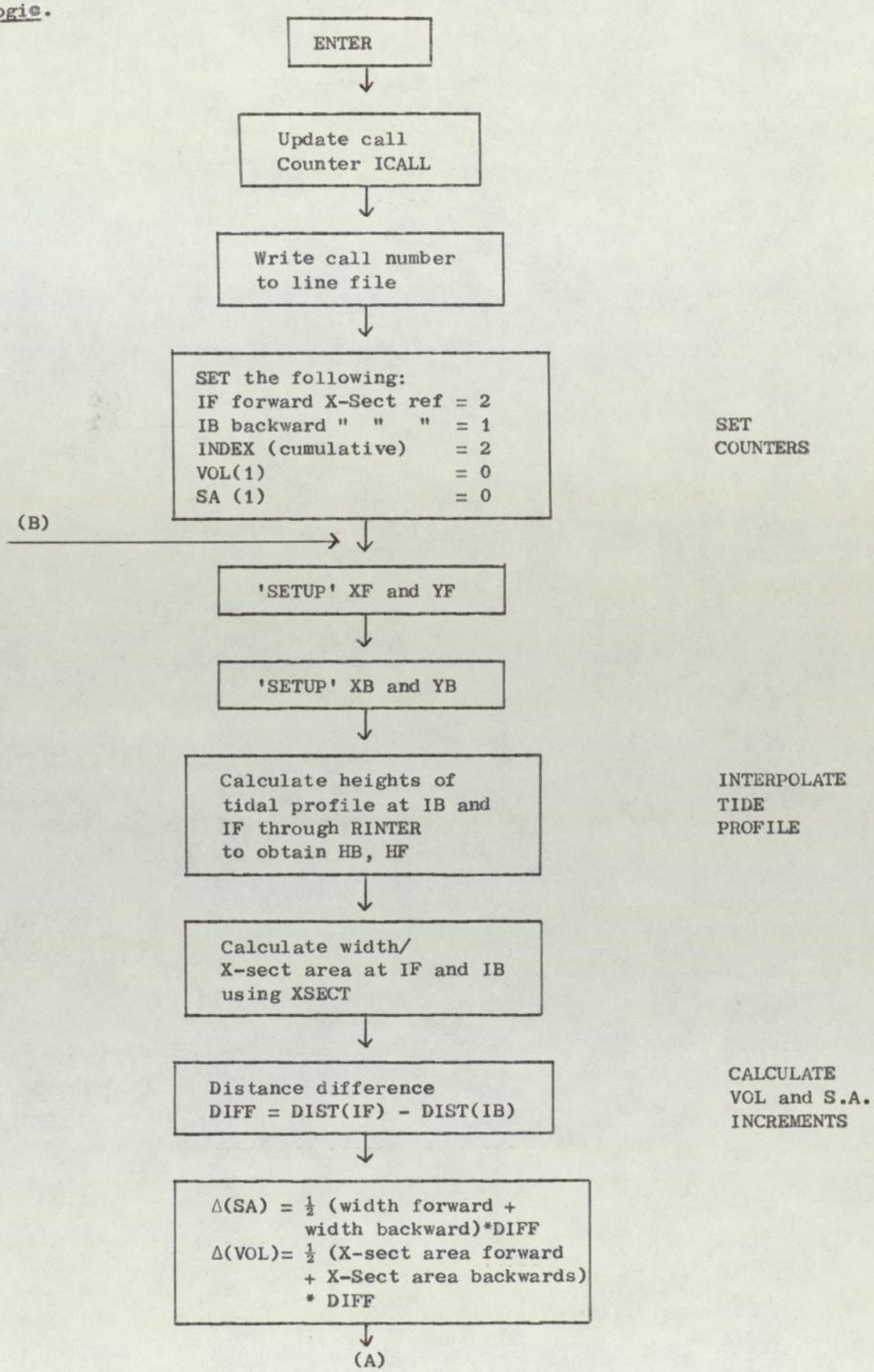
Two calls to routine SETUP copy the cross-section from X/Y arrays to XF/YF, XB/YB vectors. Then two calls of RINTER interpolate the input tidal profile to a height at cross-sections IF and IB. Using these heights as inputs to XSECT gives four outputs:

XSECAF	Cross	sectional	area	for	height	HF	at	cross	section	IF
XSECAB	"	"	"	"	"	HB	"	"	"	IB
WDTHF	Width	"	"	"	"	HF	"	"	"	IF
WDTHB	"	"	"	"	"	HB	"	"	"	IB

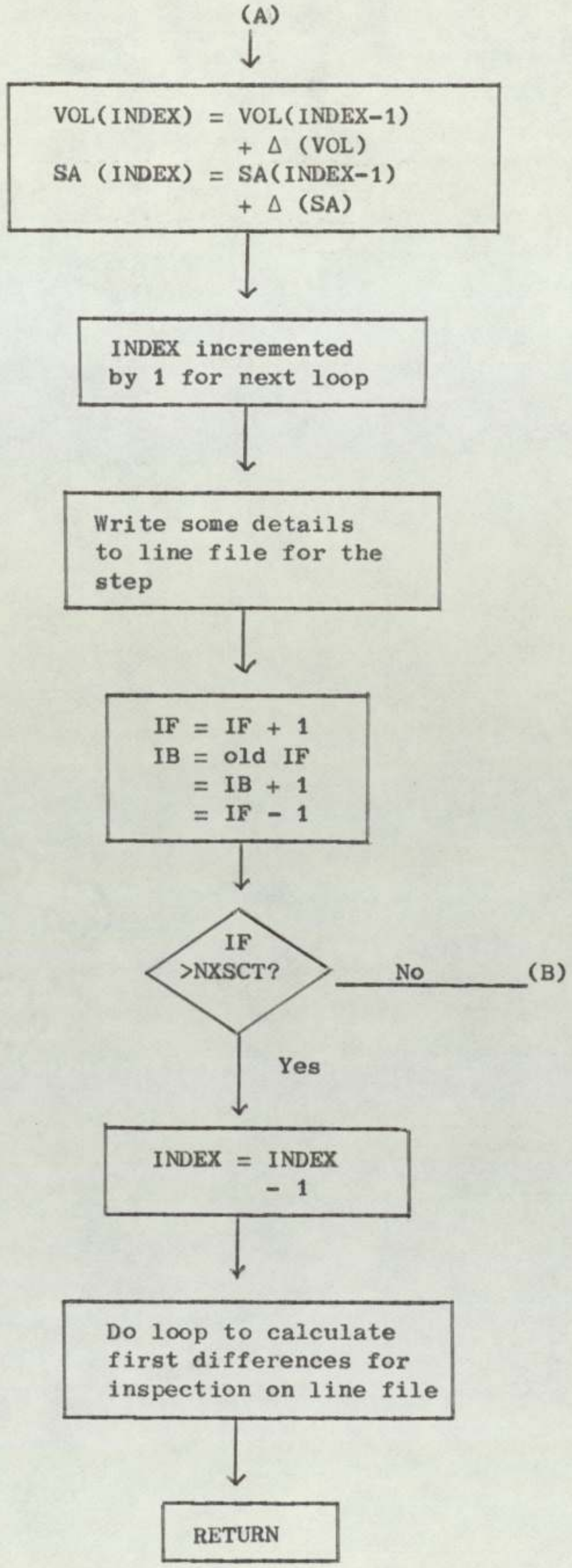
The distance between the two cross-sections, IB and IF is DIFF and so an estimate of the volume and surface area between the two sections can be made.
These estimates are used to update the cumulative arrays VOL and SA. The algorithm then steps one cross section further and IF becomes IB etc. and the process repeated until all cross sections have been processed.

A tabulated output of volumes/surface areas and first differences is now produced for visual inspection prior to segmentation.

Flow Logic.



ENTER
Update call Counter ICALL
Write call number to line file
SET the following:
IF forward X-Sect ref = 2
IB backward " " " = 1
INDEX (cumulative) = 2
VOL(1) = 0
SA (1) = 0
SET COUNTERS
(B)
'SETUP' XF and YF
'SETUP' XB and YB
Calculate heights of tidal profile at IB and IF through RINTER to obtain HB, HF
INTERPOLATE TIDE PROFILE
Calculate width/ X-sect area at IF and IB using XSECT
Distance difference DIFF = DIST(IF) - DIST(IB)
CALCULATE VOL and S.A. INCREMENTS
Δ(SA) = ½ (width forward + width backward)*DIFF
Δ(VOL)= ½ (X-sect area forward + X-Sect area backwards) * DIFF
(A)


(A)
VOL(INDEX) = VOL(INDEX-1) + Δ (VOL)
SA (INDEX) = SA(INDEX-1) + Δ (SA)
INDEX incremented by 1 for next loop
Write some details to line file for the step
IF = IF + 1
IB = old IF
= IB + 1
= IF - 1
IF >NXSCT?
No
(B)
Yes
INDEX = INDEX - 1
Do loop to calculate first differences for inspection on line file
RETURN

Subroutine PRINTX

Function:	To output cross-sectional data to line printer or card punch (or other off line device)
System Functions:	WRITE
User Functions:	None
Calling Statements:	CALL PRINTX(X, Y, L1, L2, INDEX, NOUT, DIST, NPT)
Error Message:	None
Common Areas:	None

Description.

Input List.

X, Y are the data arrays of dimension (L1, L2) with INDEX rows of data. DIST is the vector of distances corresponding to data rows. NPT contains the number of data pairs in X, Y per row. NOUT is the output peripheral

= 6 Line Printer

= 98 Card Punch.

If NOUT is not 6 or 98, no output is made and control returns to calling program.
If NOUT is 6 a listing of cross-sections is made.
If NOUT is 98 the input transformed data is punched in the format:

	Col.from	Col.to	Identifier.	Description.
HEADER RECORD	1	6	ID	Serial identifier (from 100 upwards).
	7	12	NPT(K)	Points in cross-section
	13	27	DIST(K)	Distance from zero.
DATA RECORD	1	15	X(K,J)	Distance from bank
	16	30	Y(K,J)	Depth at X(K,J)
	31	70	-	Blank
	71	80	ID	Identifier

All parameters are returned uncorrupted.

Subroutine RSPLIT.

Function: Given a set of cumulative volumes/surface areas, this subroutine segments an estuary into M segments.

System Functions: WRITE

User Functions: RINTER

Calling Statement: CALL SPLIT (X, Y, N, DARG, MARGS, TYPE)

Error Messages -

Common Areas -

Description.

Input List:

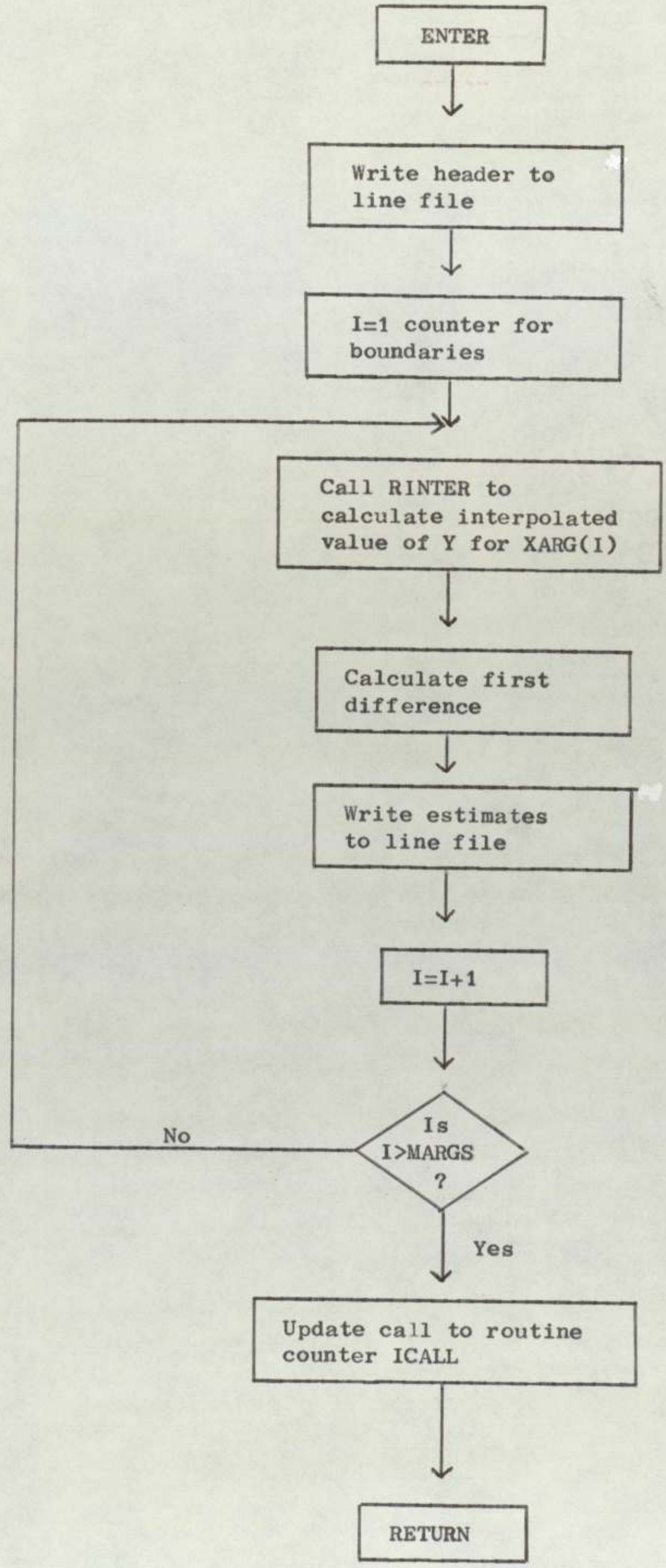
X Array holding distances from zero at which cumulative parameter is known.

Y Array holding value of cumulative parameter.

XARG Array of MARGS elements holding distances of the segment boundaries.

TYPE 8-letter identifier of the parameter.

The 'DO loo' loop interpolates the known data for a segmentation described by the boundaries in XARG. at each cycle the estimates are printed and also the first differences as a quick visual guide to the physical dimensions of the system. In many cases, one aims to obtain a constant first difference segmentation. Control returns after 'MARGS' loops without corruption of the calling list.



ESTIMATION
OF
SEGMENTS

Subroutine SETUP.

Function: To copy data from two dimensional array store arrays to a vector for use by 'CALC'

Error Messages: STOP 23546 More than allowed points in a cross-section

Calling Statement: CALL SETUP (IFB, XFB, YFB, X, Y, NPT, NUPP)

Description.

Calling list:

X,Y are the two dimensional arrays of data (1st dimension being NUPP) and each cross-section in X and Y have NPT (I) points.

IFB is the reference number of the cross-section to be copies to the vectors XFB, YFB.

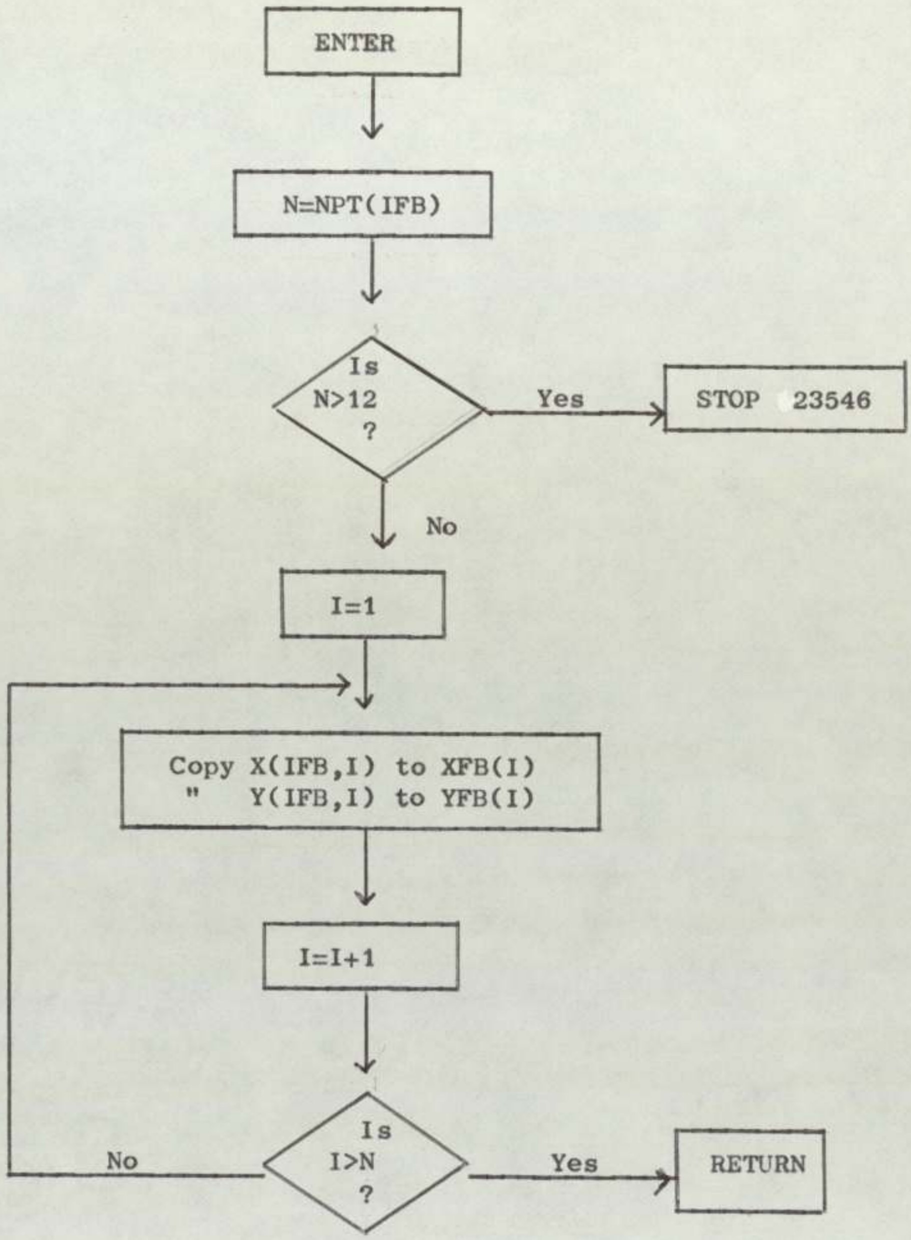
The 'DO 1' loop copies each element of

X(IFB, i) ——> XFB (i)

and Y(IFB, i) ——> YFB (i)

A check is made to see that the number of points does not exceed the limit. The input list has XFB, YFB reset but otherwise all variables are returned unaltered.

Flow Logic.


ENTER
N=NPT(IFB)
Is
N>12
?
Yes
STOP 23546
No
I=1
Copy X(IFB,I) to XFB(I)
" Y(IFB,I) to YFB(I)
I=I+1
Is
I>N
?
No
Yes
RETURN

Subroutine SPLIT.

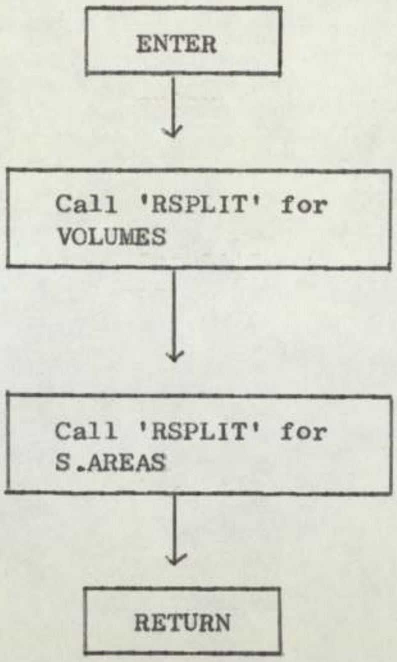
Function:	To split the calling list from HYDOR into those parameters required for volumes and those for surface areas.
System Functions:	None
User Functions:	RSPLIT
Calling Statement:	CALL SPLIT (D1, D2, NXSECT, DATAP, M, DIST)
Error Messages:	None
Common Areas:	None

Description.

Input List:

NXSECT	Number of cross sections in memory.
DATAP	Array holding number of data points per cross-section
DIST	Array holding distance of each cross-section from zero
D1) D2)	2-D arrays holding X-Y data of cross-section
M	Number of segmentation points requested.

The routine initiates two calls to RSPLIT, one to evaluate segment volumes, one for surface areas. Control is returned without corrupting any of the transferred parameters.


ENTER
Call 'RSPLIT' for VOLUMES
Call 'RSPLIT' for S.AREAS
RETURN

Subroutine XSECT

Function:	Given a set of points x-y representing distance from one bank and depth, and a height, the routine calculates cross-sectional areas and width of the cross sections.
System Functions:	ABS
User Functions:	None
Calling Statement:	CALL XSECT (X, Y, NDATAP, D, XSECTA, WIDTH)
Error Messages:	None
Common Areas:	None

Description

Input list:

X,Y	Two vectors holding distance from bank-depth data for a cross-section
NDATAP	Number of points in the cross-section.
D	the depth of interest
XSECTA	Cross-sectional area of section at depth D
WIDTH	width of section at depth D

The main loop in the program 'DO 200' considers adjacent x-y points in relation to depth (fig. 1).

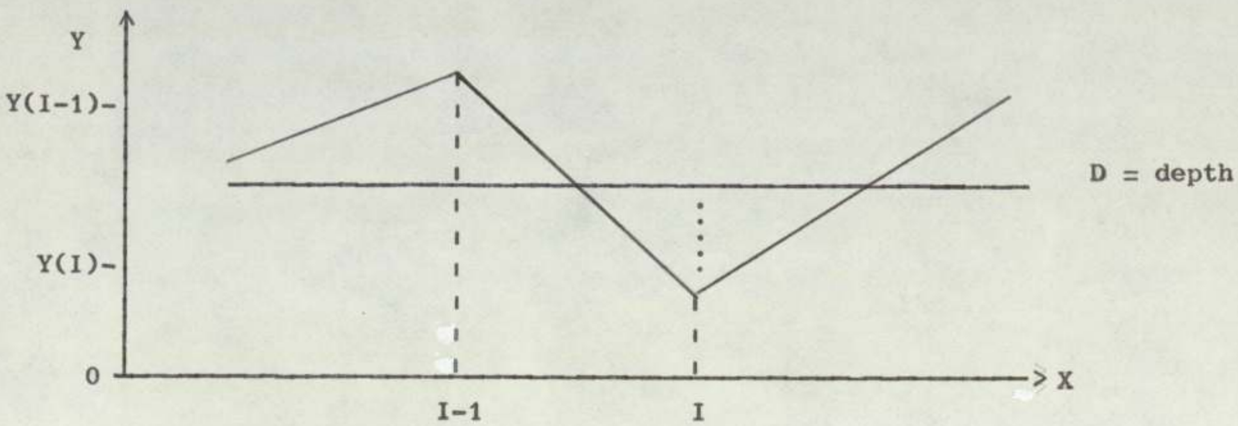


Fig 1: A Cross-section Configuration.

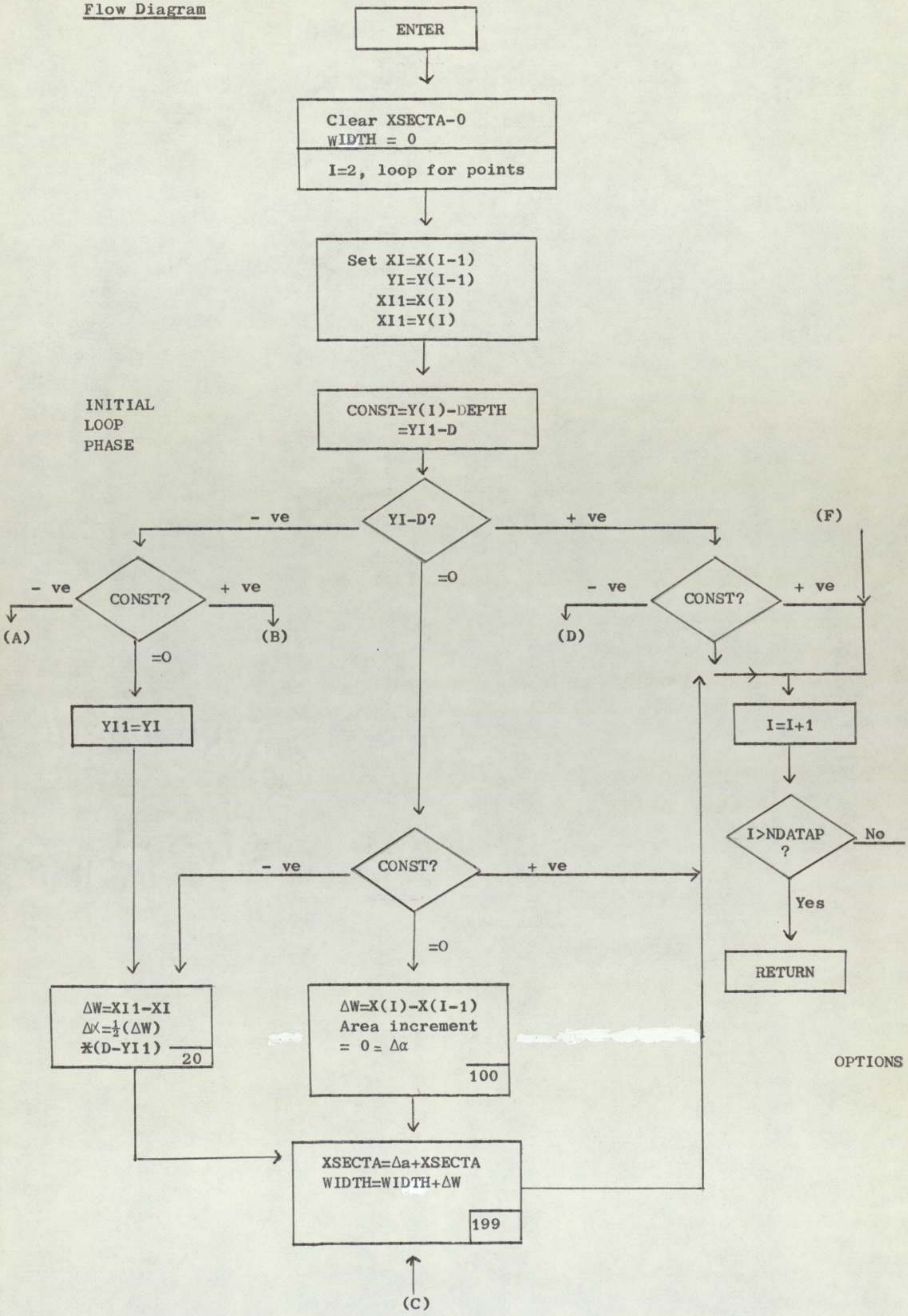
There are a combination of possible configurations for any two points in relation to the level of D:

Y(I) > D, Y(I-1) > D	therefore	No contributions by this section.				
Y(I) = D, Y(I-1)<D	"	Contributions to area and width				
Y(I) = D, Y(I-1)<D	"	"	"	"	"	
Y(I) < D, Y(I-1)>D	"	"	"	"	"	
				(as figure 1)		
Y(I) < D, Y(I-1) = D	"	"	"	"	"	
Y(I) < D, D> Y(I-1)	"	"	"	"	"	
Y(I) < D, Y(I) > Y(I-1)	"	"	"	"	"	

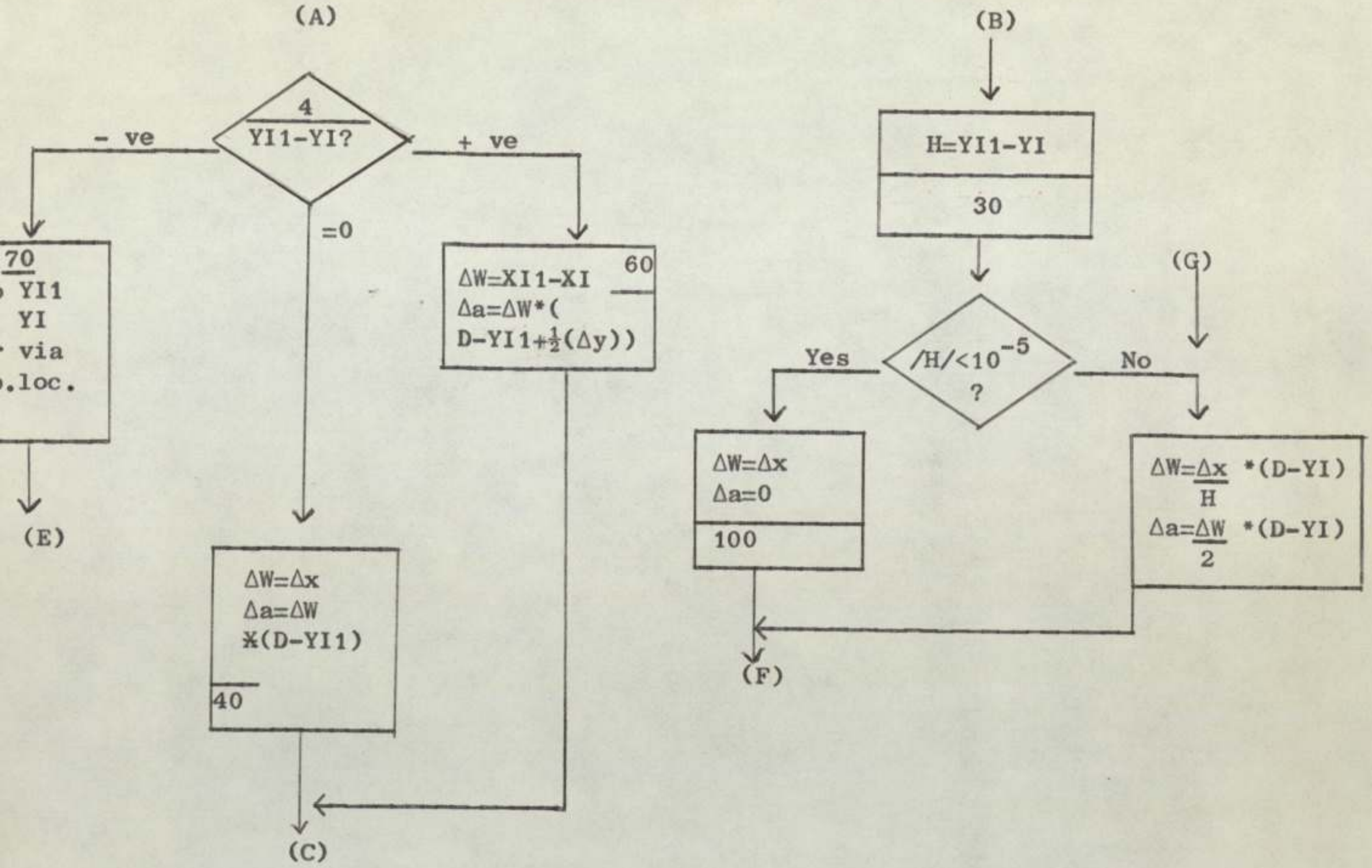
The specific contributions are calculated using similar triangles and the area of a triangle formula. Each contribution is cumulatively added to XSECTA and WIDTH.

Flow Diagram

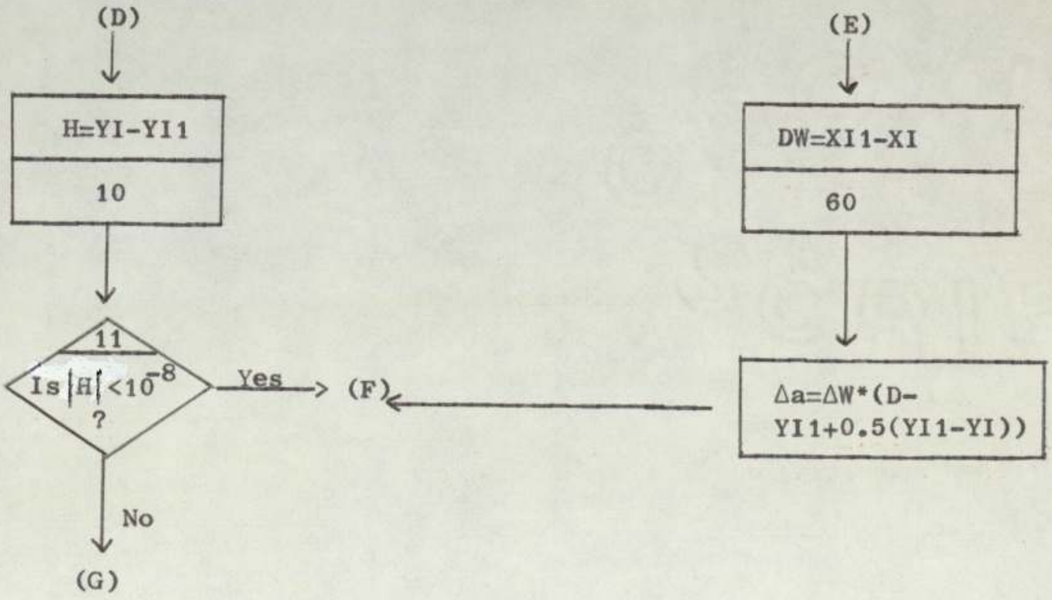
INITIAL
LOOP
PHASE



OPTIONS


(A)
4
YI1-YI?
- ve
+ ve
=0
70
YI1
YI
via
.loc.
(E)
ΔW=XI1-XI
60
Δa=ΔW*(
D-YI1+½(Δy))
ΔW=Δx
Δa=ΔW
*(D-YI1)
40
(C)
(B)
H=YI1-YI
30
(G)
/H/<10^-5
?
Yes
No
ΔW=Δx
Δa=0
100
ΔW=Δx *(D-YI)
H
Δa=ΔW *(D-YI)
2
(F)

OPTIONS


(D)
H=YI-YI1
10
11
Is |H| <10^-8
?
Yes
(F)
No
(G)
(E)
DW=XI1-XI
60
Δa=ΔW*(D-
YI1+0.5(YI1-YI))