UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
REGION II
26 FEDERAL PLAZA
NEW YORK, NEW YORK 10007

documentation for

HARO3

"A computer program for the modeling of water quality parameters in steady state multi-dimensional natural aquatic systems"

SECOND EDITION

prepared by

Steven Chapra
George A. Nossa

Systems Analysis Section
Data Systems Branch

October, 1974

TABLE OF CONTENTS

ABSTRACT

INTRODUCTION

THEORY

One dimensional analysis of a single substance T-2
One dimensional analysis of coupled substances T-10
Multi-dimensional analysis T-13
Application of the theory via a computer program T-15

THE COMPUTER PROGRAM

Flow chart of the main program C-3
Description of the main program and the subroutines C-4
Listing of the source deck C-12

USERS MANUAI

Description of input U-4
Example problem U-15
Job Control Language U-46
Restrictions U-47

APPENDIX

References
List of figures and tables
Nomenclature

ACKNOWLEDGEMENTS

ABSTRACT

definition : HAR03 is a computer program which can be used to model the steady-state distribution of water quality variables for multi-dimensional bodies of water. The technique underlying the program is based on the conservation of mass and up to two variables reacting in a feed forward fashion with first order kinetics may be modeled.

computer : HAR03 has been designed for an IBM 370 computer, requires approximately 184 K of core to compile and was written for a Fortran IV G or H Level compiler.

application : HAR03 has been designed with the Biochemical Oxygen Demand (BOD) - Dissolved Oxygen system in mind.

With minor modification the program may be used to model other variables which are analogous to this system such as chlorides, polyphosphate - orthophosphate, coliform bacteria, etc.

In order to save core for small systems, HAR03 has been compiled in three versions. The first version handles a system of up to 50 segments and is designated HAR50; similarly HAR100 and HAR200 handle a maximum of 100 and 200 segments correspondingly. The only difference between these versions is in the size of the arrays defined in the programs.

INTRODUCTION

```
C     ************************************************************************
C
C     THIS PROGRAM CONVERTS SYSTEMS PARAMETERS OF HAR03 VERSION 1  (APRIL 197
C     TO CONFORM WITH INPUT REQUIREMENTS OF HAR03 VERSION 2.0 (SEPT. 1974)
C     FOR AREA, DISPERSION, FLOW AND INTERFACE NUMBER
C
C     ************************************************************************
      DEFINE FILE 5(400,42,U,KK)
      DIMENSION AREA(6),E(6),Q(6),IARAY(6)
      KK=1
    2 READ(2,1)(AREA(J),E(J),Q(J),IARAY(J), J=1,6)
    1 FORMAT(3(3F6.1,I2))
      IF(AREA(1))10,11,10
   11 KSW=KSW + 1
      IF(KSW-2)12,99,99
   10 KSW = 0
   12 WRITE(5'KK)(AREA(J),E(J),Q(J),IARAY(J),J=1,6)
      GO TO 2
   99 KXK=KK-1
      KK=1
      DO 15 I=1,KXK
      READ(5'KK)(AREA(J),E(J),Q(J),IARAY(J),J=1,6)
      WRITE(2,3)(AREA(J),E(J),Q(J),IARAY(J),J=1,6)
    3 FORMAT(3(2F6.2,F6.0,I3))
   15 CONTINUE
      CALL EXIT
      END
```

The formulation of a mathematical model of any system is greatly determined by two factors: the nature of the system itself and the purposes and perspective of the investigator. The modeler must strike a balance between objective reality and the subjectivity of his needs to attain a successful analysis. This problem is further compounded when dealing with the high complexity of the natural world.

One of the more prevalent misconceptions among neophytes in the field of water quality modeling is that there is one analytical technique which is superior in depicting the water quality in a natural body of water. This may be partially due to the fact that the field straddles several more or less hard sciences and engineering disciplines and as such can be perceived from a variety of perspectives.

For instance, hydrodynamicists, who are essentially interested in the movement of fluids, often tend to emphasize the obviously important effect of water motion on the transport of matter in a system. Ecologists and aquatic biologists on the other hand stress the equally important reactions between the community of organisms which populate the system. The danger in these or in any particular approach comes from the automatic exclusion or underestimation of viewpoints outside the area of expertise of the modeler.

One of the older approaches to water quality modeling which rather effectively incorporates a number of perspectives in representing the causal relationships of stream pollution is that of the sanitary engineering profession. Due to their interest in designing waste treatment facilities, sanitary engineers were rather early introduced to the problems of wastes and their impact on the environment. A classic study in this profession was that done by Streeter and Phelps[1] on the Ohio River in 1925.

By making a variety of simplifying assumptions in the hydrodynamic and biological areas, these investigators arrived at a very utilitarian approach to water quality analysis which still stands as a viable technique for answering many questions about the relationship between pollution and the aquatic environment of a stream. In the hydrodynamic area, they assumed that the waste load was delivered by a pipe into a channel which could be described as having constant geometrical dimensions and constant flow. As well it was assumed that the pollutant was instantaneously mixed in the lateral and vertical directions and that the simple continuity equation, $Q=AV$, applied. From the biological standpoint, it was decided that a chemical parameter upon which most species depend for life, namely, dissolved oxygen could be modeled as an indicator of the health of the biota. To do this, they had to use a measure of the oxygen demand of the waste, the biochemical oxygen demand (BOD), as the input to the system and formulated relationships between dissolved oxygen and BOD in terms

of first order kinetics. The result is what is now called the Streeter-Phelps equation which in its basic form is:

$$D = \left[\frac{K_d}{K_a - K_r}\left(e^{-K_r x/u} - e^{-K_a x/u}\right)\right] L_o + D_o e^{-K_a x/u} \quad \text{.....(I-1)}$$

where:

D=dissolved oxygen deficit= DO_s -DO

DO_s =saturation concentration of dissolved oxygen

DO =actual concentration of dissolved oxygen

Lo=initial concentration of BOD at point of introduction of waste

Do=initial concentration of dissolved oxygen deficit at point of introduction of waste

Kr=BOD removal rate

Kd=Deoxygenation rate

Ka=Reaeration rate

U=stream velocity

x=distance downstream from point of introduction of waste

The results of the Streeter-Phelps equation is called the "D.O. sag" and is illustrated in figure I-1.

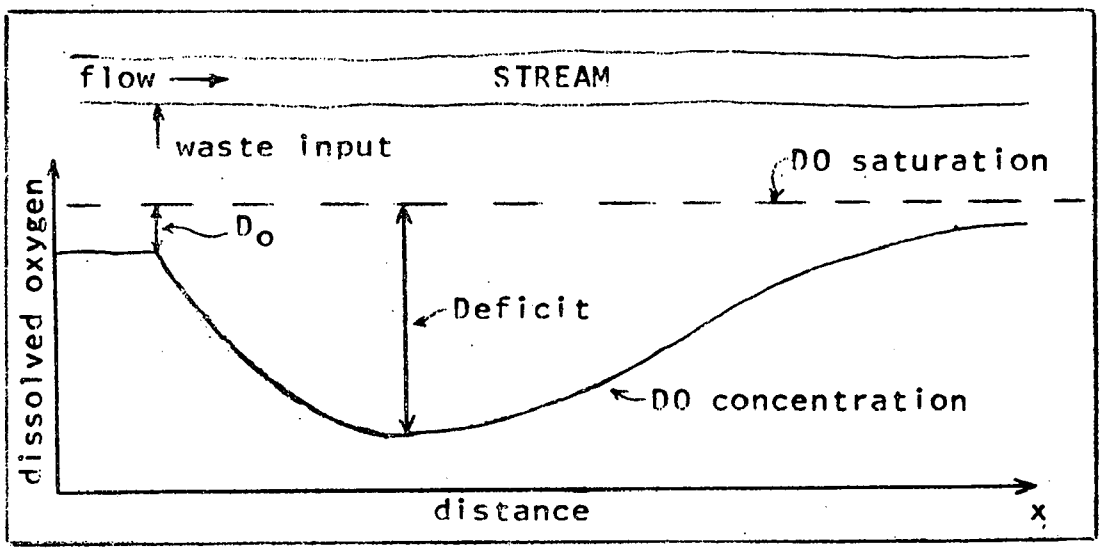

Figure I-1: D.O. sag generated by Streeter Phelps equation

This environmental model is ideal for the evaluation of various treatment schemes as its basic control variable is the waste input. This emphasis on relating man's waste inputs to the aquatic environment with the express purpose of managing the inputs and thus the water quality is what typifies the sanitary engineering approach. This can be contrasted with an aquatic biologist who might be more interested in the interraction between the organisms with a mind to prediction and description rather than control.

Since 1925 various contributions and technical developments have advanced the state of the art of water quality modeling. Particularly, the advent of the digital computer has expanded study beyond the limits experienced by Streeter and Phelps. One of the major influences from the modern sanitary engineering perspective is Robert Thomann, formerly technical director of the Delaware Estuary Comprehensive Study (DECS) and now a professor at Manhattan College. The program documented here, HARO3, is basically a representation of an approach developed by Thomann for the Delaware[2] which has proven itself to be particularly effective as a tool for water quality management.

As with Streeter and Phelps, the body of water is assumed to be at a steady state in time, and BOD and dissolved oxygen are the water quality variables (however, the program is general enough so that other variables such as temperature, coliform bacteria, etc. which are analagous to BOD and dissolved oxygen, could be modeled). Then the system is broken up into finite volumes which are completely mixed. By stipulating a priori the flow and dispersion across the interfaces between volumes, the hydrodynamics of the system are input to the model and by using mathematics the concentrations of the variables in each volumn can be calculated. Thus the approach allows extension from streams to estuaries and lakes.

This report is in several parts. First, the theoretical background is developed. This is done in terms of an estuary since it is one of the more complex bodies of water to which this program can be applied. However, the concepts expressed can be readily applied to lakes, streams and other bodies of water. Then, a description of the computer program is given followed by a user's guide.

As a final note, HARO3 is meant to be used to either furnish insight into a particular phenomena, or as a predictive device for use in water quality planning. Care must be taken at all times to consider all the assumptions underlying its formulation and by no means could it ever be construed to apply to any and every aquatic system or problem. With this in mind it is an excellent tool for the use of those interested in applying rational approaches to the problems of the deterioration of the environment.

THEORY

One dimensional analysis of a single substance...

Thomann's approach [1,2] to steady-state estuarine water quality modeling essentially applies a numerical solution technique to a convective-diffusion equation for mass transport including decay and source terms. This equation, in time variable form for a one-dimensional channel of uniform cross-section, may be written as:

$$\frac{\partial c}{\partial t} + U\frac{\partial c}{\partial x} = E\frac{\partial^2 c}{\partial x^2} - kc + (\text{sources} - \text{sinks}) \quad \ldots\ldots (T\text{-}1)$$

where

x=distance
t=time (on the scale of minutes to hours)
c=concentration of a constituent
E=longitudinal dispersion coefficient
U=tidal velocity
k=first order decay rate of constituent

A steady state in an estuary is a somewhat difficult concept to define. The term literally means that the constituent under investigation and the parameters which describe it do not vary in the time scale of the problem. As an estuary, in this report, is defined as a body of water subject to tidal oscillations, it can be argued that such a system will never reach a steady state but will continuously be in transition. Several investigators [3,4,5] have treated the subject and one approach is to take average values over a tidal cycle. The following relationship results:

$$\frac{\partial \bar{c}}{\partial t_t} + U_f\frac{\partial \bar{c}}{\partial x} = \bar{E}\frac{\partial^2 \bar{c}}{\partial x^2} - k\bar{c} + (\text{sources} - \text{sinks}) \quad \ldots\ldots\ldots (T\text{-}2)$$

where

the bar above a parameter or variable designates that the value is averaged over a tidal cycle

U_f = net advective velocity. In many cases this can be interpreted as the velocity caused by fresh water flow entering the estuary

t_t = time (on the scale of the time of a tidal cycle - several hours)

$\bar{E}$ = a tidally averaged dispersion coefficient which includes the dispersive effect caused by tidal motion

A "steady state" for equation T-2 therefore means that the variables and parameters in question do not vary from tidal cycle to tidal cycle. This situation is often approximated in the summer and early fall when low rainfall minimizes the transients caused by fresh water runoff to an estuary and meteorological conditions are relatively constant. These periods are often critical when investigating such a classical gauge of pollution as dissolved oxygen.

With this as background, Thomann suggested that the system be divided into volumes or segments as in the one dimensional system depicted in figure T-1.

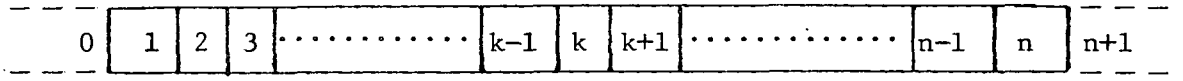

Figure T-1: Estuary of n segments

Equation T-2 can then be written in the following form for Section k (from this point on the bar which was used to designate that a parameter or variable was tidally average will be dropped for convenience):

$$Vol_k \frac{\partial c_k}{\partial t_t} + Q_{fk} l_k \frac{\partial c_k}{\partial x} = E_k A_k l_k \frac{\partial^2 c_k}{\partial x^2} - K_k Vol_k c_k + W_k \quad \text{.....(T-3)}$$

where

k= a subscript which denotes that the variable or parameter is an average for the segment k.

Q_{fk}= average net advective flow for the segment.

$=U_{fk} A_k$

Vol_k= volume of the segment = $A_k l_k$

A_k= cross-sectional area of the segment

l_k= length of the segment

W_k= waste input to the segment

At steady state $c_k/\ t$ would equal zero and equation T-3 could be written as:

$$0 = E_k A_k l_k \frac{d^2 c_k}{dx^2} - Q_{fk} l_k \frac{dc_k}{dx} - K_k Vol_k c_k + W_k \quad \text{.....(T-4)}$$

Each term containing a differential in equation T-4 can be approximated mathematically, as:

$$E_k A_k l_k \frac{d^2 c_k}{dx^2} = E'_{k-1,k}\ (c_{k-1} - c_k) + E'_{k,k+1}\ \ (c_{k+1} - c_k) \quad \text{.....(T-5)}$$

$$Q_{fk} l_k \frac{dc_k}{dx} = Q_{k,k+1}(\alpha_{k,k+1} c_k + \beta_{k,k+1} c_{k+1}) - Q_{k-1,k}(\alpha_{k-1,k} c_{k-1} + \beta_{k-1,k} c_k) \quad \text{...(T-6)}$$

where

$$E'_{ij} = E_{i,j}\ A_{i,j}/l_{i,j} \quad \text{.....(T-6a)}$$

i,j = subscript designating the interface between segments i and j.

$l_{i,j}$ = The average length of adjacent segments

=)1i + 1j)/2

$\alpha_{i,j}$ and $\beta_{i,j}$ = weighting coefficients used to correct the approximations of equation T-6 for adjacent segments of unequal length.

$$\alpha_{i,j} = \frac{l_j}{l_i + l_j} \quad \text{.....(T-7)}$$

$$\beta_{i,j} = 1 - \alpha_{i,j} = \frac{l_j}{l_i + l_j} \quad \ldots\ldots\ldots\ldots \quad (T\text{-}8)$$

As well, the above weighting coefficients can be used to insure meaningful results. It can be shown that for all positive solutions

$$\alpha > 1 - E'/Q \quad \ldots\ldots\ldots\ldots \quad (T\text{-}8a)$$

It must be stated here that Thomann's approximation in equation T-5 is strictly correct for segments of equal length (and in two and three dimensions for orthogonally shaped segments). Therefore, care should be exercised to avoid the placement of segments of highly different lengths adjacent to each other or the construction of very irregularly shaped segments. In the event that this is unavoidable, adjustment of E may be necessary to reflect the true effect represented by equation T-5.

As well, it has been shown that a numerical dispersion effect is implicit in this particular difference scheme which can be estimated by

$$E_{numerical} = Ul\ (\alpha - 1/2) \quad \ldots\ldots\ldots \quad (T\text{-}8b)$$

This can be particularly significant in streams where velocities are high and distortion may result. It is therefore imperative that the modeller be always aware of the numerical dispersion in this model and make appropriate correction for it.

Equations T-5 and T-6 can be substituted into (T-4):

$$0 = E'_{k-1,k}\,(c_{k-1}-c_k) + E'_{k,k+1}\,(c_{k+1}-c_k)$$

$$+ Q_{k-1,k}\,(\alpha_{k-1,k}c_{k-1} + \beta_{k-1,k}c_k)$$

$$- Q_{k,k+1}\,(\alpha_{k,k+1}c_k + \beta_{k,k+1}\,c_{k+1})$$

$$- K_k Vol_k C_k + W_k \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(T-9)}$$

By grouping terms

$$(-Q_{k-1,k}\alpha_{k-1,k} - E_{k-1,k})\,c_{k-1}$$

$$+ (Q_{k,k+1}\alpha_{k,k+1} - Q_{k-1,k}\beta_{k-1,k} + E'_{k-1,k} + E'_{k,k+1} + Vol_k K_k)c_k$$

$$+ (Q_{k,k+1}\beta_{k,k+1} - E_{k,k+1})\,c_{k+1} = W_k \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{(T-10)}$$

Letting

$$a_{k,k-1} = -Q_{k-1,k}\alpha_{k-1,k} - E_{k-1,k} \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{(T-11)}$$

$$a_{k,k} = Q_{k,k+1}\alpha_{k,k+1} - Q_{k-1,k}\beta_{k-1,k} + E'_{k-1,k} + E'_{k,k+1} + Vol_k K_k \quad \ldots \text{(T-12)}$$

$$a_{k,k+1} = Q_{k,k+1}\beta_{k,k+1} - E_{k,k+1} \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{(T-13)}$$

The general equation for the kth segment is therefore

$$a_{k,k-1}\, c_{k-1} + a_{k,k} c_k + a_{k,k+1} c_{k+1} = W_k \quad \ldots\ldots\ldots\ldots\ldots \text{(T-14)}$$

For the furthest upstream section (figure T-1, section 1), which is a boundary of the system, a similar procedure would yield:

$$a_{11}\, c_1 + a_{12}\, c_2 = W_1 + (Q_{01}\, \alpha_{01} + E'_{01})\, c_o \quad \ldots\ldots\ldots\ldots \text{(T-15)}$$

or

$$a_{11}\, c_1 + a_{12} c_2 = W'_1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(T-16)}$$

where

o = a subscript to denote the values of the variables and parameters beyond the upstream boundary.

For the furthest downstream segment (in figure T-1, segment k):

$$a_{n-1,n} c_{n-1} + a_{n,n} c_n = W_n + (-Q_{n,n+1} \beta_{n,n+1} + E'_{n,n+1})\, c_{n+1} \quad \ldots\ldots \text{(T-17)}$$

or

$$a_{n-1,n} c_{n-1} + a_{n,n} c_n = W'_n \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(T-18)}$$

where

n + 1 = a subscript to denote the values of the variables and parameters beyond the downstream boundary.

The complete set of equations for the system can be written as

$$
\begin{aligned}
a_{11}c_1 + a_{12}c_2 + 0 + 0 + \cdots + 0 &= W_1 \\
a_{21}c_1 + a_{22}c_2 + a_{23}c_3 + 0 + \cdots + 0 &= W_2 \\
0 + a_{32}c_2 + a_{33}c_3 + a_{34}c_4 + \cdots + 0 &= W_3 \\
&\vdots \\
0 \quad 0 \quad 0 + \cdots + 0 + a_{n,n-1}c_{n-1} + a_{nn}c_n &= W_n
\end{aligned}
\qquad \ldots\ldots\ldots\ldots \text{(T-19)}
$$

At this point there are n equations with n unknowns. Solutions may be obtained either by solving the equations simultaneously or by writing T-19 in matrix form and inverting it. This latter technique is employed in HARO3. Equation T-19 in matrix form is

$$\begin{bmatrix} a_{11} & a_{12} & 0 & . & . & . & . & 0 \\ a_{21} & a_{22} & a_{23} & 0 & . & . & . & 0 \\ 0 & a_{32} & a_{33} & a_{34} & 0 & . & . & 0 \\ . & & & & & & & \\ . & & & & & & & \\ . & & & & & & & \\ 0 & . & . & . & 0 & & a_{n,n-1} & a_{nn} \end{bmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ . \\ . \\ . \\ c_n \end{pmatrix} = \begin{pmatrix} W_1' \\ W_2 \\ W_3 \\ . \\ . \\ . \\ W_n' \end{pmatrix} \quad \ldots\ldots\ldots\ldots \text{(T-20)}$$

or

$$[A]\ (c) = (W) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(T-21)}$$

By inverting [A] a solution vector can be obtained:

$$(c) = [A]^{-1}\ (W) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(T-22)}$$

Besides allowing solution of equation (T-21), equation T-22 offers an additional feature. The elements of the matrix $[A]^{-1}$, which is called the single system response matrix, represent the unit response of a particular section to the addition of a unit load to another section. For instance, the element a_{23} of $[A]^{-1}$ is the unit concentration response of section 2 to a unit input of waste into section 3. This is a particularly useful feature of Thomann's technique in that once a system response matrix is generated for a particular set of parameters, the effect of various load changes reflecting treatment strategies can be directly determined from this matrix. It, therefore, provides a quick technique for judging the trade-offs of various treatment alternatives.

One dimensional analysis of coupled substances...

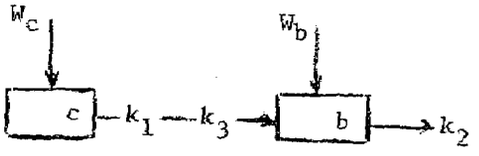

Figure T-2: Schematic for a coupled system of two reactants

Figure T-2 is a schematic for a coupled system of two constituents, c and b, which react with first order kinetics. W_c and W_b are waste inputs of the constituents and the k's are the reaction rates which link them.

In classical sanitary engineering practice the BOD-dissolved oxygen system corresponds to the schematic with c = BOD and b = dissolved oxygen deficit. The rate k_1 is referred to as the BOD removal rate, k_2 is a rate of reaeration from the atmosphere and k_3 is a deoxygenation rate. In the case where there is no loss of BOD via sedimentation or another non-oxygen demanding removal, k_1 would equal k_3.

The equations describing the conservation of mass for c would be exactly like that developed in the previous section. The equation for b in a segment k at steady state would be

$$0 = E'_{k-1,k}(b_{k-1}-b_k) + E'_{k,k+1}(b_{k+1}-b_k)$$

$$+ Q_{k-1,k}(\alpha_{k-1,k}b_{k-1} + \beta_{k-1,k}b_k)$$

$$- Q_{k,k+1}(\alpha_{k,k+1}b_k + \beta_{k,k+1}b_{k+1})$$

$$- Vol_k K_{k2} b_k + Vol_k K_{k3} c_k + W_b \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(T-23)}$$

where

c_k = BOD in section k

b_k = dissolved oxygen deficit in section k

K_{k2} = reaeration rate in segment k

K_{k3} = deoxygenation rate in segment k

W_b = waste load or source of dissolved oxygen deficit to section k

By grouping terms this results in a matrix equation similar to T-21:

$$[B](b) = [\text{Vol } K_3](c) + (W_b) \quad \text{(T-24)}$$

where

[Vol K_3] = an nxn diagonal matrix

[B] = similar in form to [A] with the exception that the diagonal elements contain the expression Vol K_{k2} rather than Vol K_{k1}

(c) = solution vector to the single system (in this case BOD).

The solution is therefore

$$(b) = [B]^{-1}[\text{Vol } K_3](c) + [B]^{-1}(W_b) \quad \text{(T-25)}$$

In this program, three sources of dissolved oxygen deficit will be considered:

$$W_b = -PMR + BD*H + Y \quad \text{(T-26)}$$

where

PMR = the dissolved oxygen deficit due to the photosynthesis and respiration of algae.

BD*H = the dissolved oxygen deficit due to bottom deposits.

Y - direct oxygen deficit from waste sources

One additional feature of the analysis should be noted. By substituting equation T-22 into equation T-25 the following relationship results

$$(b) = [B]^{-1} (\text{Vol } K_3) [A]^{-1} (W_c) + [B]^{-1} (W_b) \quad \text{(T-26a)}$$

or after expressing the term (Vol K_3) as a diagonal matrix and multiplying the matrixes the following results

$$(b) = [C]^{-1} (W_c) + [B]^{-1} (W_b) \quad \text{(T-26b)}$$

As can be seen $[C]^{-1}$ is a particularly useful entity in that it directly relates the deficit in the body of water to the waste load of BOD being discharged. It is called the total system response matrix.

For boundaries where flow enters the section with a concentration c_b:

$$a_{ii} = \sum_k (Q_{ik}\alpha_{ik} + E'_{ik}) + V_i K_i + Q_{ii}\beta_{ii} + E'_{ii} \quad \text{(T-33)}$$

$$W_i \equiv W_i + (E'_{ii} - Q_{ii}\alpha_{ii})\, c_b \quad \text{(T-34)}$$

For boundaries where flow leaves the section to an area with a concentration c_b:

$$a_{ii} = \sum_\kappa (Q_{ik}\alpha_{ik} + E'_{ik}) + V_i + Q_{ii}\alpha_{ii} + E'_{ii} \quad \text{(T-35)}$$

$$W \equiv W_i + (E'_{ii} - Q_{ii}\beta_{ii})\, c_b \quad \text{(T-36)}$$

Similar to the previous analysis for one dimension, equation T-30 together with the appropriate boundary conditions can be incorporated into a matrix.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & & & \\ a_{n1} & a_{n2} & \cdots & \cdots & a_{nn} \end{bmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \\ \\ c_n \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ \\ \\ w_n \end{pmatrix}$$

or $$[A](c)=(W) \quad \text{(T-37)}$$

with a solution

$$(c)=[A]^{-1}(W) \quad \text{(T-38)}$$

This may be extended to a coupled system of two reactants (e.g. BOD-DO deficit) in a fashion analagous to that done on page

As well, the criterion that

$$\alpha > 1 - E'/Q \quad \text{(T-39)}$$

to insure positive results also applies.

Application of the theory via a computer program...

At this point, the solution has been reduced to two equations: T-22 and T-26b for BOD and DO deficit, respectively.

$$(c) = [A]^{-1} (W_c) \quad \text{(T-22)}$$

$$(b) = [C]^{-1} (W_c) + [B]^{-1} (W_b) \quad \text{(T-26b)}$$

For HARO3, equation T-26b will not be used due to storage requirements for the additional matrix, [C].* In its place, a form of equation T-25 will be used

$$(b) = [B]^{-1} ((Vol * K_3) (c) + (W_b)) \quad \text{(T-40)}$$

Aside from input-output considerations, the major part of the algorithm of HARO3 therefore reduces to the setting up of the matrices [A] and [B] and their respective forcing functions.

Since the only difference between matrix [A] and matrix [B] is that their diagonals contain different reaction terms, a matrix excluding reaction terms can be formulated. Called the system matrix, [AB], it can be transformed into [A] or [B] by merely adding the appropriate reaction term to the diagonal.

* note: another version of the program will be available in the future which uses T-26b and which generates the valuable total system response matrix, $[C]^{-1}$.

With this as a background, a very simple representation of the major steps of HARO3 can be presented:

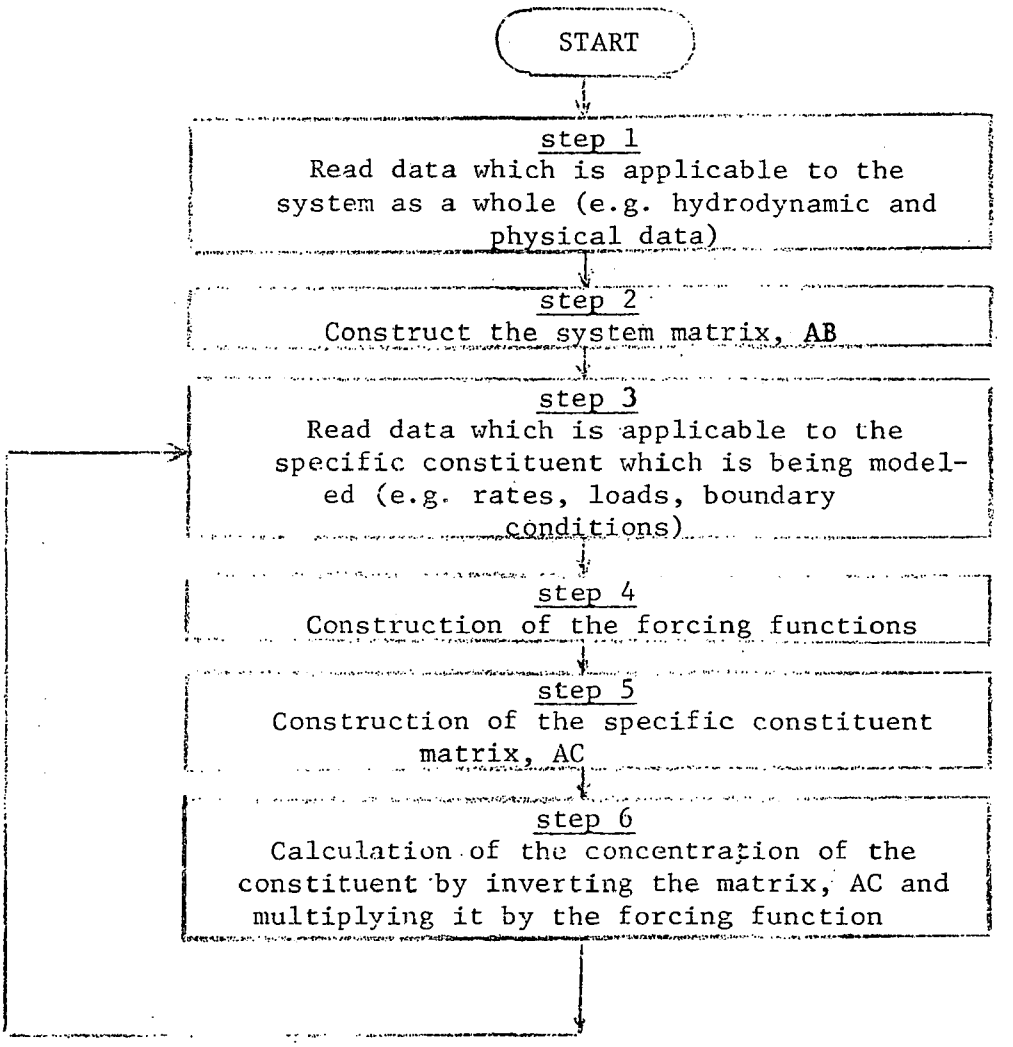

Figure T-4: Simple Schematic of HARO3 computer program

Of the six major parts, step 2 and step 4 require further discussion:

Step 2 - The construction of the system matrix [AB]

When constructing [AB], each section is considered individually with each of its interfaces treated one at a time. The following sequence is undertaken for each interface:

1) calculation of E' as in equation T-6a

$$E'_{ij} = \frac{E_{ij} A_{ij}}{\frac{(l_i + l_j)}{2}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(T-41)}$$

2) a value of alpha is calculated to adjust the advective approximation for segments of unequal length.

for flow across the interface and into the section ($Q < o$)

$$\alpha_{ji} = \frac{l_i}{l_i + l_j} \quad \ldots\ldots\ldots\ldots \text{(T-42)}$$

for flow across the interface and out of the section ($Q > o$)

$$\alpha_{ij} = \frac{l_j}{l_i + l_j} \quad \ldots\ldots\ldots\ldots \text{(T-43)}$$

3) the criterion for positive solutious is applied to alpha (equation T-39) and if it is not met, an adjustment is made.

This is accomplished by the following test:

if $\alpha > 1 - E'/Q$ do not alter

if $\alpha < 1 - E'/Q$ recalculate α as

$$\alpha = 1 - E'/2Q \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(T-44)}$$

which places it well within the criterion.

3) calculation of the elements of the system matrix

The elements are calculated as in equation T-31 (excluding the reaction term) and T-32.

When each of these three steps is done for each interface, the procedure is repeated for each segment until all have been incorporated into the matrix.

Step 4 - The construction of the forcing function

At present, the forcing functions which are calculated are peculiar to the BOD - DO deficit system. From the theory, equations T-34 and T-36 give their most basic form:

for $Q > o$ $\quad W_i' = W_i + (E'_{ii} - Q_{ii}\,\beta_{ii})\,c_b$ (T-36)

for $Q < o$ $\quad W_i' = W_i + (E'_{ii} - Q_{ii}\,\alpha_{ii})\,c_b$ (T-34)

As can be seen, the forcing function consists of two basic parts:

1) Boundary condition forcing function

of the form, $[E'_{ii} - Q_{ii}\ (\alpha_{ii}$ or $\beta_{ii})]c_b$

where

E'_{ii} = the bulk dispersion across the interface between the section and the boundary.

Q_{ii} = the flow across that interface

α_{ii} or β_{ii} = assumed initially to be 0.5 and then subsequently tested as in equation T-39 and if necessary recalculated according to equation T-44.

c_b = the concentration across the boundary. Aside from this, this part of the forcing function does not vary as to the constituent being modeled.

2) Other forcing functions

of the form, W_i

for BOD or chlorides

W_i = any load to the section (e.g. waste loads, runoff loads, etc.)

for DO deficit

W_i = -PMR + BD*H + loads

as explained in equation T-26.

THE COMPUTER PROGRAM

The computer program from which HARO3 evolved was developed by Hydroscience, Inc. for the Massachusetts Water Resources Commission[1]. It was originally written for an IBM 1130 computer in FORTRAN IV and utilized a Gauss-Seidel technique to solve the set of simultaneous equations generated by the Thomann technique. The program was never adequately documented and as such was not appropriate for general distribution.

A source deck of the program was made available to the Environmental Protection Agency by the State of Massachusetts and after modification two new versions were published by E.P.A.[2] entitled HARO1 and HARO2. Compatible with an IBM 370 and an IBM 1130, respectively, these versions retained the Gauss-Seidel solution technique and had the advantage of being thoroughly documented.

The major difference between HARO3 and the previous versions is in its use of a matrix inversion solution technique. As well, parts of the algorithm have been redesigned to more accurately reflect Thomann's technique and to improve its clarity and ease of use.

In its present form HARO3 is programmed for an IBM 370/158 computer. The core requirements for each version have been summarized below:

Version	Maximum No. of segments	Storage Requirements
HAR50	50	76K
HAR100	100	148K
HAR200	200	410K

The compile and link time for any of these versions is .46 minutes, which includes optimization by the Fortran IV H level compiler. Without optimization, compilation and linkage takes .24 minutes. Each of these versions are stored as load modules at OSI in Bethesda, Md. (EPA computer facility).

An estimate of the actual execution time is given in table C-1. Each time is for what is called an estuarine additive coupled system (described on page U-3) which analyzes the chlorides, carbonaceous and nitrogenous BOD and the D.O. deficit of a system. This type of run performs five matrix inversions which in larger systems takes up the largest amount of time.

Number of segments	execution time (minutes)
8	.02
30	.10
45	.27
64	.60

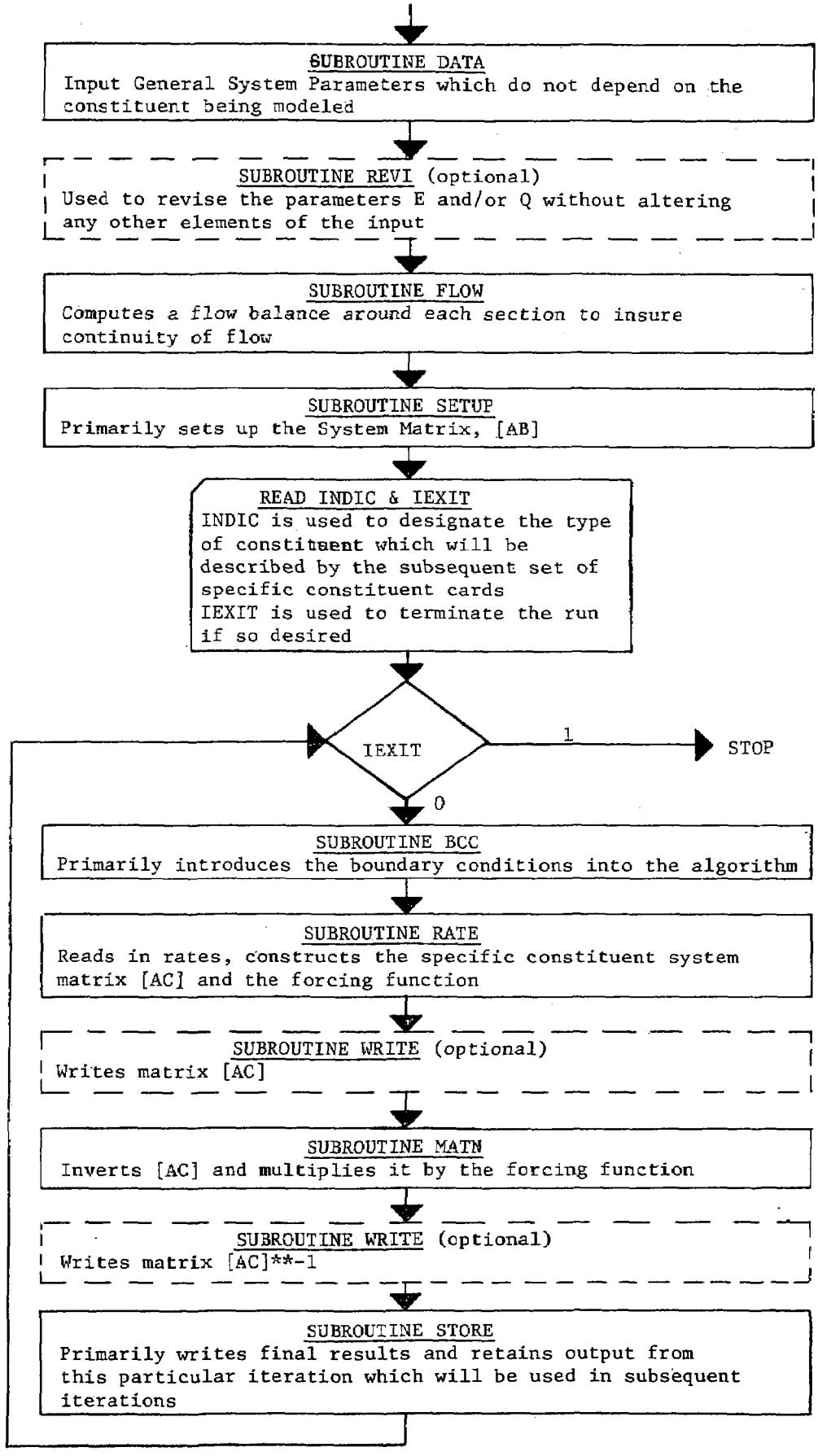

FIGURE C-1: Flow chart of the main program of HAR03

DESCRIPTION OF THE MAIN PROGRAM AND EACH OF THE SUBROUTINES

The Main Program-HARO3...

HARO3 has been programmed using a modular approach. Wherever possible, an attempt has been made to give each subroutine a unique function. The main program was designed to coordinate the subroutines to effectively analyze the water quality of a system. This coordination is depicted in figure C-1 which represents a development of the simple schematic given in figure T-3. In figure C-1 certain details have been omitted and if the user requires more detail he should consult the code itself.

Subroutine DATA...

The primary purpose of this subroutine is to input parameters which do not depend on the particular substance being modeled. These include system parameters (number of sections, indicators, etc.) and certain physical parameters such as cross-sectional area, net advective flow, etc., which are only input once when modeling a water body. These are in contrast to parameters such as loads and reaction rates which would vary with the constituent being modeled.

The particular tasks accomplished by DATA are:

1) read and write system parameters
2) initialize certain matrices to zero
3) read and write interface parameters AREA, E and Q and place them into the storage matrix ARRAY.
4) read and write the characteristic lengths (LA)
5) read the general section parameters depth (H), temperature (T), and volume (VOL)
6) Apply the following conversion factors:

$$Q(MGD) = Q\ (CFS) * .6463\ MGD/cfs$$

$$VOL(MG) = VOL\ (10^6 ft^3) * 7.48\ gal/ft^3$$

At this point, a further word about the function of ARRAY is in order. ARRAY is an N X 18 sized matrix which is used to save space when storing the interface parameters AREA, E and Q. Used in conjunction with IARAY, ARRAY avoids the use of three N X N matrices when handling these variables. It does this as follows:

IARAY(I,J)	contains the section number of the section which forms the interface with section I.
ARRAY(I,J)	contains the area of interface (I,IARAY(I,J))
ARRAY(I,J + 1)	contains the E of interface (I,IARAY(I,J))
ARRAY(I,J + 2)	contains the Q of interface (I,IARAY(I,J))
J	is an integer variable which is incremented by 3 from 1 to 16

For computational purposes, every interface must be accounted for in terms of the section to which it applies. This would involve duplicate input - for instance, AREA (I,J) is the same as AREA (J,I). To simplify input, only one of the values must be input and subroutine DATA (statements DATA 075 to 091) fills out the remaining positions in ARRAY.

Subroutine REVI...

This subroutine is used to revise the parameters E and/or Q without altering any other elements of the input. This can often be an advantage when doing sensitivity analyses. The indicator ICON is employed to call the subroutine and designates which parameter is to be changed.

REVI also prints the revised values in a labelled tabular form.

Subroutine FLOW

FLOW basically computes a balance of the flows into and out of a section. If there is more or less than a zero balance around the section, the resultant value is printed as an excess flow.

The purpose of this exercise is to both keep track of flows and to insure that no mistakes were made when inputting the flow regime. This can be particularly useful when working in two or three dimensions.

Subroutine SETUP

The primary purpose of this subroutine is to set up the system matrix, AB, as was generally outlined on pages T-17 to T-18 . The tasks performed by SETUP are:

1) calculation of the bulk dispersion coefficient, E'

According to the theory:

$$E'_{i,j} = \frac{E_{i,j} A_{i,j}}{l_{i,j}}$$

where

$E_{i,j}$ = the dispersion coefficient across the interface

$A_{i,j}$ = the cross-sectional area of the interface

$l_{i,j}$ = the average length of the adjoining sections = $(l_i + l_j)/2$

The program calculates E' (EPRIM) as follows:

```
EPRIM(J)=ARRAY(I,JJ)*ARRAY(I,JJ+1)*417.166/(L(K,I)+L(I,K))
```

where

ARRAY(I,JJ) = $A_{i,j}$ (ft^2)

ARRAY(I,JJ+1) = $E_{i,j}$(mi^2/day)

L(I,K) = l_i

L(K,I) = l_k

417.1166 - **conversion** factor which is computed as follows:

Original Units	Conversion Factor	New Units
$\frac{ft^2 * mile^2}{day * ft^2}$	$\frac{(5280\ ft)^2}{(mile)^2} \times \frac{7.481\ gal}{ft^3} \times 2 \times \frac{MG}{10^6\ gal}$	MGD

417.1166

2) A value of alpha is calculated to adjust the advective term for sections of different length. This value is checked against the positivity criterion (equation T-39) and if it is not met, alpha is recalculated (as in equation T-44)

Alpha(J) = 1.0-(EPRIM(J)/(2*ABS(ARRAY(I,JJ+2))))

which corresponds to

= 1 - E'/2Q

3) The elements of the system matrix [AB] are calculated

4) The values for EPRIM and ALPHA are printed.

5) The section parameters: depth, volume and temperature, are printed.

Subroutine BCC

The primary purpose of this subroutine is to introduce boundary conditions into the algorithm. BCC marks the point at which the type of constituent being modeled becomes relevant. At the present time the program distinguishes between three types of constituents:

specific type in HARO3	which is analogous to
Chlorides	any conservative substance
BOD	a single constituent reacting with first order kinetics
D.O. deficit	the second constituent of a coupled, feed-forward system reacting with first order kinetics.

BCC performs the following tasks:

1) Reads the number of boundary conditions

2) If there are none it writes the message
   ZERO BOUNDARY CONDITION

3) if there are boundary conditions it reads and writes them with the appropriate labels (CL, BOD or DEF)

4) it calculates the part of the forcing function due to the boundary condition as described on page

   for Q < 0

```
FRCBC(I)=8.34*BC(KLAST)*(EPRIM(J)-ALPHA(J)*ARRAY(I,JJ+2))
```

   for Q > 0

```
FRCBC(I)=8.34*BC(KLAST)*(EPRIM(J)-(1.0-ALPHA(J)*ARRAY(I,JJ+2))
```

where

FRCBC(I) = forcing function due to the boundary condition
BC = boundary concentration (mg/l)
EPRIM = E' (MGD)
ARRAY(I,JJ+2) = Q (MGD)
8.34 = conversion factor of (MGD*mg/l) to #/day

Subroutine RATE

RATE finishes preparation for calculation by completing the construction of the specific constituent system matrix [AC] and the appropriate forcing function. It does this by:

1) Reading the reaction rates (@ T=20° C) which are to be added to the diagonal of [AB] to transform it to [AC]. In the BOD-DO deficit system these rates are equal to

   a) The removal rate, Kr or Kn, for CBOD and NBOD, respectively
   b) the reaeration rate, Ka, for deficit
   c) zero for chlorides

At the same time it inputs the appropriate temperature correction factors which are used to adjust the rates via the following general formula

$$K_t = K_{20} * \theta^{(T-20)}$$

where some typical values are [3]

rate	θ
Kr	1.047
Ka	1.024

2) the rates are multiplied by the section volumes and added to the diagonal of [AB] converting it to [AC]

3) for BOD or chlorides the forcing function is formed by reading the waste loads (LOAD) and adding them to FRCBC which was calculated in subroutine BCC

4) for D.O. deficit the forcing function is completed by first reading the deoxygenation rate (K_d) the benthal demand (BD) and the net photosynthesis effect (PMR). Appropriate temperature correction factors are read and applied to Kd and BD. The loads are read and then the forcing function is formed (as in equation)

```
XK1(I)=8.34*VOL(I)*(XK1(I)*KD(I)*FL-PMR(I)
          +BD(I)*H(I))+Y(I)
```

where

XK1(I) on the left hand side of the equation is the deficit forcing function
XK1(I) on the right hand side of the equation is the BOD concentration in the body of water as calculated during the BOD iteration.
FL is a ratio relating ultimate to 5-day BOD
Y is the same as FRCBC as calculated in subroutine BCC

then the deficit waste load is added.

```
XK1 = XK1 + LOAD
```

Subroutine WRITE

This subroutine writes the basic matrices before and after inversion if this is required.

Subroutine MATN

This subroutine inverts the matrix [AC] and multiplies it by the forcing function.

Subroutine STORE...

Subroutine STORE performs a number of miscellaneous tasks:

1) writes the final concentrations for each constituent being modeled
2) calculates the saturation value of oxygen [4]

```
CS(I) = (1.-.000009*CL(I))*(14.652-.41022*T(I)
        + .0079910 * T(I) ** 2.-.000077774 * T(I)**3.)
```

where T(I) = the temperature of a section
CL(I) = the chloride concentration of the section

3) performs what can be called miscellaneous bookkeeping functions. For instance, it retains values of a particular constituent which would be necessary when calculating another constituent. (e.g. the BOD is an input to the calculation of DO deficit and therefore must be retained). It therefore plays an important role in the coordination of a particular configuration of reactants.

```
C*********************************************************************** HAR03000
C*********************************************************************** HAR03001
C*********************************************************************** HAR03002
C                                                                       HAR03003
C                          COMPUTER PROGRAM HAR03                       HAR03004
C                                                                       HAR03005
C                                                                       HAR03006
C     HAR03 IS A COMPUTER PROGRAM WHICH CAN BE USED TO MODEL THE        HAR03007
C     STEADY-STATE DISTRIBUTION OF WATER QUALITY VARIABLES FOR          HAR03008
C     MULTI-DIMENSIONAL BODIES OF WATER.  THE TECHNIQUE UNDERLYING      HAR03009
C     THE PROGRAM IS BASED ON THE PRINCIPLE OF THE CONSERVATION         HAR03010
C     OF MASS AND UP TO TWO WATER QUALITY VARIABLES REACTING            HAR03011
C     IN A FEED FORWARD FASHION WITH FIRST ORDER KINETICS MAY BE        HAR03012
C     MODELED                                                           HAR03013
C                                                                       HAR03014
C     HAR03 HAS BEEN DESIGNED FOR AN IBM 370/155 COMPUTER, REQUIRES     HAR03015
C     APPROXIMATELY 150 K OF CORE AND WAS WRITTEN FOR A FORTRAN IV      SEPT 74
C     G OR H LEVEL COMPILER                                             SEPT 74
C                                                                       HAR03018
C     THIS VERSION 'HAR100' CAN ACCOMODATE UP TO 100 COMPLETELY MIXED   HAR03019
C     SEGMENTS                                                          HAR03020
C                                                                       HAR03021
C*********************************************************************** HAR03022
C*********************************************************************** HAR03023
C*********************************************************************** HAR03024
      COMMON/ALGAE/NX,N,INDIC,ICON,MMM,NMK                              HAR03025
      COMMON/BASS/MX,IPRNT,FRCBC(100)                                   HAR03026
    1 FORMAT(I2)                                                        HAR03027
   12 FORMAT(8E10.4)                                                    HAR03028
   20 FORMAT(3I2)                                                       HAR03029
      MX=5                                                              HAR03030
      NX=6                                                              HAR03031
      NMK=0                                                             HAR03032
      CALL DATA                                                         HAR03033
   25 READ(MX,1)ICON                                                    HAR03034
      IF(ICON)57,57,26                                                  HAR03035
   26 CALL REVI                                                         HAR03036
      GO TO 25                                                          HAR03037
   57 CONTINUE                                                          HAR03038
      CALL FLOW                                                         HAR03039
      CALL SETUP                                                        HAR03040
      MMM=1                                                             HAR03041
   93 READ(MX,20)INDIC,IEXIT                                            HAR03042
      IF(IEXIT)8,8,1000                                                 HAR03043
    8 CALL BCC                                                          HAR03044
      CALL RATE                                                         HAR03045
      IF(IPRNT)10,10,1003                                               HAR03046
 1003 CALL WRITE                                                        HAR03047
   10 CALL MATN                                                         HAR03048
      IF(IPRNT)11,11,1013                                               HAR03049
 1013 CALL WRITE                                                        HAR03050
   11 CALL STORE                                                        HAR03051
      GO TO 93                                                          HAR03052
 1000 STOP                                                              HAR03053
```

```
      END                                                                HAR03054
      SUBROUTINE DATA                                                    DATA 000
C*************************************************************************DATA 001
C*************************************************************************DATA 002
C                                                                        DATA 003
C                              SUBROUTINE DATA                           DATA 004
C                                                                        DATA 005
C     INPUTS DATA WHICH WOULD NOT VARY WITH THE PARTICULAR               DATA 006
C     CONSTITUENT BEING MODELED, I.E. PRIMARILY PHYSICAL AND             DATA 007
C     HYDRODYNAMIC DATA                                                  DATA 008
C                                                                        DATA 009
C*************************************************************************DATA 010
C*************************************************************************DATA 011
      REAL LA(100,6)                                                     SEPT 74
      DIMENSION AREA(6),E(6),Q(6),TITLE(10)                              DATA 013
       COMMON/ALGAE/NX,N,INDIC,ICON,MMM,NMK                              DATA 014
      COMMON/BASS/MX,IPRNT,FRCBC(100)                                    DATA 015
      COMMON/COHOE/LA,SCALE(4),ARRAY(100,18),IARAY(100,6),NLA(100,6)     SEPT 74
      COMMON/PIKE/JCON,H(100),T(100),VOL(100),AB(100,100)                DATA 017
   31 FORMAT(8F10.0)                                                     DATA 018
  100 FORMAT(10A4,3I3,3X,4F7.0)                                          SEPT 74
  101 FORMAT (///28X, 'ZERO SEGMENT NUMBER IN INTERFACE ', I3, '-', I3,  DATA 020
     1' COMPUTATION DISCONTINUED')                                       DATA 021
  200 FORMAT(6(I4,' -',I4,2X,F7.0,2X))                                   DATA 022
  201 FORMAT(6(I3,F10.0))                                                DATA 023
  300 FORMAT(                 '1                                         DATA 024
     1 CHARAC. LENGTHS OF SEGMENTS (FT) '///' INTERFACE  LENGTH    INTERDATA 025
     1FACE  LENGTH    INTERFACE  LENGTH    INTERFACE  LENGTH    INTERFACDATA 026
     2E  LENGTH    INTERFACE  LENGTH')                                   DATA 027
 1111 FORMAT(3(3F6.0,I3))                                                DATA 029
 2000 FORMAT('1'//40X,10A4////10X,'**SYSTEM PARAMETERS**',//10X,         DATA 031
     1'NUMBER OF SECTIONS = ',I4,25X,'IPRNT = ',I4/10X,'JCON = ',I4)     SEPT 74
 2002 FORMAT(//10X,'**SCALE FACTORS**'//10X,                             DATA 036
     1'SCALE(1) = ',F10.3,4X,'SCALE(2) = ',F10.3,4X,                     DATA 037
     2'SCALE(3) = ',F10.3,4X,'SCALE(4) = ',F10.3)                        DATA 038
 2200 FORMAT('1',2X,'INTERFACE',5X,'AREA',7X,'E',8X,'Q ',2(6X,'INTERFACESEPT 74
     1',5X,'AREA',9X,'E',7X,'Q')/,4X,'ROW-SEG',4X,'(FT**2)', 2X,         SEPT 74
     2'(MI**2/D)',2X,'(CFS)',   6X,'ROW-SEG',5X,'(FT**2)',2X,            SEPT 74
     3'(MI**2/D)',2X,'(CFS)',5X,'ROW-SEG',5X,'(FT**2)',2X,'(MI**2/D)',   SEPT 74
     4 2X,'(CFS)' )                                                      SEPT 74
 3000 FORMAT(  '   (',I3,'-',I3,')',2X,F8.0,4X,F6.3,F8.1,2(5X,'(',I3,    SEPT 74
     1 '-',I3,')',2X,F8.0,4X,F6.3,F8.1))                                 SEPT 74
C*************************************************************************DATA 042
C                                                                       *DATA 043
C     READ AND WRITE THE SYSTEM PARAMETERS AND SCALE FACTORS            *DATA 044
C                                                                       *DATA 045
C*************************************************************************DATA 046
      READ(MX,100)TITLE,N,IPRNT,JCON,       SCALE                        SEPT 74
      WRITE(NX,2000)TITLE,N,IPRNT,JCON                                   SEPT 74
      WRITE(NX,2002)SCALE                                                DATA 049
C*************************************************************************DATA 050
C                                                                       *DATA 051
C     INITIALIZE MATRICES TO ZERO                                       *DATA 052
C                                                                       *DATA 053
```

```
C*************************************************************************** DATA 054
      DO 1001 I=1,N                                                         DATA 055
      DO 1021 J=1,18                                                        DATA 058
 1021 ARRAY(I,J)=0.0                                                        DATA 059
      DO 1001 J=1,6                                                         DATA 060
      LA(I,J)=0.0                                                           SEPT 74
 1001 IARAY(I,J)=0                                                          DATA 061
C*************************************************************************** DATA 062
C                                                                          *DATA 063
C     READ AND WRITE INTERFACE PARAMETERS AREA, E AND Q AND PLACE          *DATA 064
C     THEM INTO ARRAY                                                      *DATA 065
C                                                                          *DATA 066
C*************************************************************************** DATA 067
      DO 1002 I=1,N                                                         SEPT 74
      READ(MX,1111) (AREA(J),E(J),Q(J),     IARAY(I,J),J=1,6)             DATA 069
      DO 1002 J=1,6                                                         SEPT 74
      IF(AREA(J))72,71,72                                                   SEPT 74
   72 IF(IARAY(I,J))74,74,71                                                SEPT 74
   74 WRITE (NX,101)I,IARAY(I,J)                                            SEPT 74
      CALL EXIT                                                             SEPT 74
   71 JJ=(J-1)*3+1                                                          SEPT 74
      ARRAY(I,JJ)=AREA(J)*SCALE(1)                                          DATA 072
      ARRAY(I,JJ+1)=E(J)*SCALE(2)                                           DATA 073
 1002 ARRAY(I,JJ+2)=Q(J)*SCALE(3)                                           DATA 074
      DO 1008 I=1,N                                                         DATA 075
      DO 1008 JJ=1,6                                                        DATA 076
      IF(IARAY(I,JJ))1008,1008,1088                                         DATA 077
 1088 IF(IARAY(I,JJ)-I)1012,1008,1012                                       DATA 078
 1012 JJJ=(JJ-1)*3+1                                                        DATA 079
      DO 1004 J=1,6                                                         DATA 080
      JK=(J-1)*3+1                                                          DATA 081
      NK=IARAY(I,JJ)                                                        DATA 082
      IF(IARAY(NK,          J))1005,1006,1005                               DATA 083
 1006 ARRAY(NK,          JK)=ARRAY(I,JJJ)                                   DATA 084
      ARRAY(NK,          JK+1)=ARRAY(I,JJJ+1)                               DATA 085
      ARRAY(NK,          JK+2)=-ARRAY(I,JJJ+2)                              DATA 086
      IARAY(NK,          J)=I                                               DATA 087
      GO TO 1008                                                            DATA 088
 1005 IF(IARAY(NK,          J)-I           )1004,1008,1004                  DATA 089
 1004 CONTINUE                                                              DATA 090
 1008 CONTINUE                                                              DATA 091
      WRITE (NX,2200)                                                       DATA 092
      LINE=0                                                                SEPT 74
      DO 1032 I=1,N                                                         DATA 093
      WRITE(NX,3000)  I,IARAY(I,1),ARRAY(I,1),ARRAY(I,2),ARRAY(I,3),        DATA 094
     1                I,IARAY(I,2),ARRAY(I,4),ARRAY(I,5),ARRAY(I,6),        DATA 095
     1                I,IARAY(I,3),ARRAY(I,7),ARRAY(I,8),ARRAY(I,9)         DATA 096
      WRITE(NX,3000)  I,IARAY(I,4),ARRAY(I,10),ARRAY(I,11),ARRAY(I,12),     DATA 097
     1                I,IARAY(I,5),ARRAY(I,13),ARRAY(I,14),ARRAY(I,15),     DATA 098
     1                I,IARAY(I,6),ARRAY(I,16),ARRAY(I,17),ARRAY(I,18)      DATA 099
      LINE=LINE+2                                                           SEPT 74
      IF(FLOAT(LINE/54)-FLOAT(LINE)/54.)1032,1099,1099                      SEPT 74
 1099 WRITE(NX,2200)                                                        SEPT 74
      LINE=0                                                                SEPT 74
```

```
 1032 CONTINUE                                                          DATA 100
C*************************************************************************** DATA 101
C                                                                       *DATA 102
C     READ AND WRITE INTERFACE PARAMETER- CHARACTERISTIC LENGTH         *DATA 103
C                                                                       *DATA 104
C*************************************************************************** DATA 105
      WRITE(NX,300)                                                     DATA 106
      LINE=0                                                            SEPT 74
      DO 500 I=1,N                                                      DATA 107
      READ(MX,201)(NLA(I,J),LA(I,J), J=1,6)                             SEPT 74
      DO 77 J=1,6                                                       SEPT 74
   77 LA(I,J)=LA(I,J)*SCALE(4)                                          SEPT 74
      WRITE(NX,200)(I,NLA(I,J),LA(I,J),J=1,6)                           SEPT 74
      LINE=LINE+1                                                       SEPT 74
      IF(FLOAT(LINE/54)-FLOAT(LINE)/54.)500,1098,1098                   SEPT 74
 1098 WRITE(NX,300)                                                     SEPT 74
      LINE=0                                                            SEPT 74
  500 CONTINUE                                                          DATA 118
C*************************************************************************** DATA 119
C                                                                       *DATA 120
C     READ SECTION PARAMETERS- DEPTH, VOLUME AND TEMPERATURE            *DATA 121
C                                                                       *DATA 122
C*************************************************************************** DATA 123
      READ(MX,31)(H(I),I=1,N)                                           DATA 124
      READ(MX,31)  (T(I),VOL(I),I=1,N)                                  DATA 125
      IF(T(2))14,13,14                                                  DATA 126
   13 DO 14 I=1,N                                                       DATA 127
      T(I)=T(1)                                                         DATA 128
   14 CONTINUE                                                          DATA 129
C*************************************************************************** DATA 130
C                                                                       *DATA 131
C     APPLY CONVERSION FACTORS                                          DATA 132
C                                                                       *DATA 133
C*************************************************************************** DATA 134
      DO 12 I=1,N                                                       DATA 135
      DO 23 J=1,6                                                       DATA 136
      JJ=(J-1)*3+1                                                      DATA 137
   23 ARRAY(I,JJ+2)=ARRAY(I,JJ+2)*.6463                                 DATA 138
   12 VOL(I)=VOL(I)*7.48                                                DATA 139
      RETURN                                                            DATA 140
      END                                                               DATA 142
      SUBROUTINE REVI                                                   REVI 000
C*************************************************************************** REVI 001
C*************************************************************************** REVI 002
C                                                                       REVI 003
C                          SUBROUTINE REVI                              REVI 004
C                                                                       REVI 005
C     THIS SUBROUTINE CAN BE USED TO CHANGE THE PARAMETERS E AND/OR     REVI 006
C     Q WITHOUT ALTERING THE REST OF THE INPUT                          REVI 007
C                                                                       REVI 008
C*************************************************************************** REVI 009
C*************************************************************************** REVI 010
      REAL LA(100,6)                                                    SEPT 74
      DIMENSION Q(6),E(6),REVIS(12)                                     REVI 011
```

```
      COMMON/ALGAE/NX,N,INDIC,ICON,MMM,NMK                                REVI 012
      COMMON/BASS/MX,IPRNT,FRCBC(100)                                     SEPT 74
      COMMON/COHOE/LA,SCALE(4),ARRAY(100,18),IARAY(100,6),NLA(100,6)      SEPT 74
    2 FORMAT(1H1,40X'REVISED PARAMETER LIST'//)                           REVI 014
    5 FORMAT(12F5.0)                                                      REVI 015
   10 FORMAT(//50X,'NEW DISPERSIONS',/6X,5('INTERFACE',3X,'E',7X),'INTERREVI 016
     1FACE',3X,'E'/  15X,5(                                             '(MREVI 017
     7I**2/DAY)',9X)     ,'(MI**2/DAY)'//)                              REVI 018
   13 FORMAT(3X,6(3X,I4,'-',I3,F9.3))                                     REVI 019
   14 FORMAT(//50X'NEW FLOWS'/7X,6('INTERFACE',4X'Q',6X)/18X,5('(CFS)',1REVI 020
     65X),'(CFS)'//)                                                    REVI 021
      WRITE (NX,2)                                                       REVI 022
      GO TO(8,9),ICON                                                    REVI 023
C*************************************************************************REVI 024
C                                                                       *REVI 025
C     CHANGE E'S                                                        *REVI 026
C                                                                       *REVI 027
C*************************************************************************REVI 028
    8 DO 6 I=1,N                                                         REVI 029
      IF(I-(I/2)*2)21,21,20                                              REVI 030
   20 READ(MX,5)REVIS                                                    REVI 031
      IPOS=0                                                             REVI 032
   21 DO 6 J=1,6                                                         REVI 033
      IPOS=IPOS+1                                                        REVI 034
      JJ=(J-1)*3+1                                                       REVI 035
    6 ARRAY(I,JJ+1)=REVIS(IPOS)*SCALE(2)                                 REVI 036
      WRITE (NX,10)                                                      REVI 037
      DO 11 I=1, N                                                       REVI 038
      DO 12 J=1,6                                                        REVI 039
      JJ=(J-1)*3+1                                                       REVI 040
   12 E(J)=ARRAY(I,JJ+1)/SCALE(2)                                        REVI 041
   11 WRITE (NX,13)(I, IARAY (I,J), E(J), J=1, 6)                        REVI 042
      GO TO 99                                                           REVI 043
C*************************************************************************REVI 044
C                                                                       *REVI 045
C     CHANGE Q'S                                                        *REVI 046
C                                                                       *REVI 047
C*************************************************************************REVI 048
    9 DO 4 I=1,N                                                         REVI 049
      IF(I-(I/2)*2)23,23,24                                              REVI 050
   24 READ(MX,5)REVIS                                                    REVI 051
      IPOS=0                                                             REVI 052
   23 DO 4 J=1,6                                                         REVI 053
      IPOS=IPOS+1                                                        REVI 054
      JJ=(J-1)*3+1                                                       REVI 055
    4 ARRAY(I,JJ+2)=REVIS(IPOS)*SCALE(3)*0.6463                          REVI 056
      WRITE (NX,14)                                                      REVI 057
      DO 15 I=1, N                                                       REVI 058
      DO 16 J=1,6                                                        REVI 059
      JJ=(J-1)*3+1                                                       REVI 060
   16 Q(J)=ARRAY(I,JJ+2)/(SCALE(3)*0.6463)                               REVI 061
   15 WRITE (NX,13)(I, IARAY(I,J), Q(J), J=1, 6)                         REVI 062
   99 RETURN                                                             REVI 063
      END                                                                REVI 064
```

```
      SUBROUTINE FLOW                                                    FLOW 000
C*****************************************************************     FLOW 001
C*****************************************************************     FLOW 002
C                                                                        FLOW 003
C                           SUBROUTINE FLOW                              FLOW 004
C                                                                        FLOW 005
C     THIS SUBROUTINE CALCULATES A FLOW BALANCE AROUND EACH SECTION     *FLOW 006
C     TO INSURE THAT THE FLOW REGIME HAS BEEN INPUT CORRECTLY           *FLOW 007
C                                                                       *FLOW 008
C*****************************************************************     FLOW 009
C*****************************************************************     FLOW 010
      REAL LA(100,6)                                                     SEPT 74
      COMMON/ALGAE/NX,N,INDIC,ICON,MMM,NMK                               FLOW 012
      COMMON/COHOE/LA,SCALE(4),ARRAY(100,18),IARAY(100,6),NLA(100,6)     SEPT 74
      NMN=0                                                              FLOW 014
  214 FORMAT('       SECTION ',I3,' HAS AN EXCESS FLOW   OF ',F12.5,     FLOW 015
     1' CFS')                                                            FLOW 016
  302 FORMAT('1')                                                        FLOW 017
      DO 213 I=1,N                                                       FLOW 018
      QQ=ARRAY(I,3)+ARRAY(I,6)+ARRAY(I,9)+ARRAY(I,12)+ARRAY(I,15)        FLOW 019
     1+ARRAY(I,18)                                                       FLOW 020
      QQ=QQ/.6463                                                        FLOW 021
      IF(QQ-.001)216,37,37                                               FLOW 022
  216 IF(QQ+.001)37,37,213                                               FLOW 023
   37 IF(NMN)29,29,27                                                    FLOW 024
   29 NMN=1                                                              FLOW 025
      WRITE(NX,302)                                                      FLOW 026
   27 WRITE(NX,214)I,QQ                                                  FLOW 027
  213 CONTINUE                                                           FLOW 028
      RETURN                                                             FLOW 029
      END                                                                FLOW 030
      SUBROUTINE SETUP                                                   SETUP000
C*****************************************************************     SETUP001
C*****************************************************************     SETUP002
C                                                                        SETUP003
C                           SUBROUTINE SETUP                             SETUP004
C                                                                        SETUP005
C     THE PRIMARY TASK OF SETUP IS TO CONSTRUCT THE MATRIX, AB           SETUP006
C                                                                        SETUP007
C*****************************************************************     SETUP008
C*****************************************************************     SETUP009
      REAL LA(100,6)                                                     SEPT 74
      COMMON/ALGAE/NX,N,INDIC,ICON,MMM,NMK                               SETUP011
      COMMON/COHOE/LA,SCALE(4),ARRAY(100,18),IARAY(100,6),NLA(100,6)     SEPT 74
      COMMON/PIKE/JCON,H(100),T(100),VOL(100),AB(100,100)                SETUP013
      COMMON/BREAM/EPRIM(100),ALPHA(100)                                 SETUP014
   23 FORMAT(6(I3,'-',I3,F6.0,F6.3,2X))                                  SEPT 74
   24 FORMAT('1',49X,'VALUES OF ALPHA AND EPRIM',///,6(' INTRFC EPRIM ALSETUP016
     1PHA  '),/,6(8X,'(MGD)',8X),//)                                     SETUP017
   42 FORMAT(4X,I7,3F10.2)                                               SETUP018
   43 FORMAT('1',3X,'SECTION TEMPERATURE  VOLUME     DEPTH'/             SETUP019
     115X,'(C)    (10**6GAL)    (FT)'//)                                 SETUP020
      WRITE (NX,24)                                                      SETUP021
      DO 28 I=1,N                                                        SETUP022
```

```
      DO 99 J=1,N                                                        SETUP023
   99 AB(I,J)=0.0                                                        SETUP024
C*************************************************************************SETUP025
C                                                                       *SETUP026
C     CALCULATE EPRIM AND ALPHA FOR EACH INTERFACE                      *SETUP027
C                                                                       *SETUP028
C*************************************************************************SETUP029
      DO 20 J=1,6                                                        SETUP030
      EPRIM(J)=0.0                                                       SETUP031
      ALPHA(J)=0.0                                                       SETUP032
      JJ=(J-1)*3+1                                                       SETUP033
      K=IARAY(I,J)                                                       SETUP034
      IF(K)20,20,10                                                      SETUP035
C*************************************************************************SEPT 74
C                                                                        SEPT 74
C     GENERATE (I,K) LENGTH ELEMENT                                      SEPT 74
C                                                                        SEPT 74
C*************************************************************************SEPT 74
   10 DO 77  KK=1,6                                                      SEPT 74
      IF(NLA(I,KK)-K)77,81,77                                            SEPT 74
   81 XLEN1=LA(I,KK)                                                     SEPT 74
      GO TO 82                                                           SEPT 74
   77 CONTINUE                                                           SEPT 74
C*************************************************************************SEPT 74
C                                                                        SEPT 74
C     GENERATE (K,I) LENGTH ELEMENT                                      SEPT 74
C                                                                        SEPT 74
C*************************************************************************SEPT 74
   82 DO 78 KK=1,6                                                       SEPT 74
      IF(NLA(K,KK)-I)78,84,78                                            SEPT 74
   84 XLEN2=LA(K,KK)                                                     SEPT 74
      GO TO 83                                                           SEPT 74
   78 CONTINUE                                                           SEPT 74
   83 EPRIM(J)=ARRAY(I,JJ)*ARRAY(I,JJ+1)*417.1166/(XLEN2+XLEN1)          SEPT 74
      IF(ARRAY(I,JJ+2))300,301,301                                       SETUP037
  300 ALPHA(J)=XLEN1/(XLEN1+XLEN2)                                       SEPT 74
      GO TO 200                                                          SETUP039
  301 ALPHA(J)=XLEN2/(XLEN1+XLEN2)                                       SEPT 74
  200 CONTINUE                                                           SETUP041
      IF(ARRAY(I,JJ+2))14,17,14                                          SETUP042
   14 IF(ALPHA(J)-1.0+EPRIM(J)/ABS(ARRAY(I,JJ+2)))15,15,17               SETUP043
   15 ALPHA(J)=1.0-(EPRIM(J)/(2.*ABS(ARRAY(I,JJ+2))))                    SETUP044
C*************************************************************************SETUP045
C                                                                       *SETUP046
C     SETUP THE SYSTEM MATRIX - AB                                      *SETUP047
C                                                                       *SETUP048
C*************************************************************************SETUP049
   17 IF(ARRAY(I,JJ+2))18,19,19                                          SETUP050
   18 AB(I,I)=AB(I,I)+EPRIM(J)+(1.0-ALPHA(J))*ARRAY(I,JJ+2)              SETUP051
      IF(I-K)21,20,21                                                    SETUP052
   21 AB(I,K)=ALPHA(J)*ARRAY(I,JJ+2) - EPRIM(J)                          SETUP053
      GO TO 20                                                           SETUP054
   19 AB(I,I)=AB(I,I)+ALPHA(J)*ARRAY(I,JJ+2)+ EPRIM(J)                   SETUP055
      IF(I-K) 22,20,22                                                   SETUP056
```

```
   22 AB(I,K)=(1.-ALPHA(J))*ARRAY(I,JJ+2)-EPRIM(J)                    SETUP057
   20 CONTINUE                                                        SETUP058
C*****************************************************************SETUP059
C                                                                    *SETUP060
C     WRITE VALUES OF EPRIM AND ALPHA                                *SETUP061
C                                                                    *SETUP062
C*****************************************************************SETUP063
      WRITE(NX,23)                                                   SETUP064
     1I,IARAY(I,1),EPRIM(1),ALPHA(1),I,IARAY(I,2),EPRIM(2),ALPHA(2),  SETUP065
     2I,IARAY(I,3),EPRIM(3),ALPHA(3),I,IARAY(I,4),EPRIM(4),ALPHA(4),  SETUP066
     2I,IARAY(I,5),EPRIM(5),ALPHA(5),I,IARAY(I,6),EPRIM(6),ALPHA(6)   SETUP067
      LINE=LINE+1                                                    SEPT 74
      IF(FLOAT(LINE/54)-FLOAT(LINE)/54.)28,1028,1028                 SEPT 74
 1028 WRITE(NX,23)                                                   SEPT 74
      LINE=0                                                         SEPT 74
   28 CONTINUE                                                       SETUP068
C*****************************************************************  SETUP069
C                                                                    *SETUP070
C     WRITE SECTION PARAMETERS                                       *SETUP071
C                                                                    *SETUP072
C*****************************************************************  SETUP073
      WRITE(NX,43)                                                   SETUP074
      LINE=0                                                         SEPT 74
      DO 399 I=1,N                                                   SEPT 74
      WRITE(NX,42) I,T(I),VOL(I),H(I)                                SEPT 74
      LINE=LINE+1                                                    SEPT 74
      IF(FLOAT(LINE/54)-FLOAT(LINE)/54.)399,1029,1029                SEPT 74
 1029 WRITE(NX,43)                                                   SEPT 74
      LINE=0                                                         SEPT 74
  399 CONTINUE                                                       SEPT 74
      DO 32 I=1,N                                                    SETUP076
   32 H(I)=3.281/H(I)                                                SETUP077
      RETURN                                                         SETUP078
      END                                                            SETUP079
      SUBROUTINE BCC                                                 BCC  000
C*****************************************************************  BCC  001
C*****************************************************************  BCC  002
C                                                                    BCC  003
C                              SUBROUTINE BCC                        BCC  004
C                                                                    BCC  005
C     THIS SUBROUTINE INCORPORATES BOUNDARY CONDITIONS INTO THE      BCC  006
C     SYSTEM                                                         BCC  007
C                                                                    BCC  008
C*****************************************************************  BCC  009
C*****************************************************************  BCC  010
      REAL LA(100,6)                                                 SEPT 74
      DIMENSION ICOL(100),BC(100)                                    BCC  012
      COMMON/ALGAE/NX,N,INDIC,ICON,MMM,NMK                           BCC  013
      COMMON/BASS/MX,IPRNT,FRCBC(100)                                BCC  014
      COMMON/BREAM/EPRIM(100),ALPHA(100)                             BCC  016
      COMMON/COHOE/LA,SCALE(4),ARRAY(100,18),IARAY(100,6),NLA(100,6) SEPT 74
    7 FORMAT(I2)                                                     BCC  017
   44 FORMAT('1',40X,'ZERO BOUNDARY CONCENTRATIONS')                 BCC  018
  209 FORMAT('1SEGMENT',5X,'CL  BOUNDARY CONDITION(MG/L)             BCC  019
```

```
     1                          /(I4,19X,F10.2))                                  BCC  020
  210 FORMAT('1SEGMENT',5X,'BOD BOUNDARY CONDITION(MG/L) '                      BCC  021
     1                          /(I4,19X,F10.2))                                  BCC  022
  211 FORMAT('1SEGMENT',5X,'DEF BOUNDARY CONDITION(MG/L) '                      BCC  023
     1                          /(I4,19X,F10.2))                                  BCC  024
 1150 FORMAT(6( F8.0,I4))                                                       BCC  025
C*************************************************************************** BCC  026
C                                                                              *BCC  027
C     READ AND WRITE BOUNDARY CONDITIONS                                       *BCC  028
C                                                                              *BCC  029
C*************************************************************************** BCC  030
    8 READ(MX,7)NUMBC                                                          BCC  031
      IF(NUMBC)31,31,36                                                        BCC  032
   31 WRITE(NX,44)                                                             BCC  033
      DO 103 I=1,50                                                            BCC  034
      ICOL(I)=0                                                                BCC  035
  103 BC(I)=0.                                                                 BCC  036
      GO TO 213                                                                BCC  037
   36 READ(MX,1150)(BC(J),       ICOL(J),J=1,NUMBC)                          BCC  038
      IF(INDIC)206,207,208                                                     BCC  039
  206 WRITE(NX,209)(ICOL(J),BC(J),       J=1,NUMBC)                          BCC  040
      GO TO 213                                                                BCC  041
  207 WRITE(NX,210)(ICOL(J),BC(J),       J=1,NUMBC)                          BCC  042
      GO TO 213                                                                BCC  043
  208 WRITE(NX,211)(ICOL(J),BC(J),       J=1,NUMBC)                          BCC  044
C*************************************************************************** BCC  045
C                                                                              *BCC  046
C     CALCULATE THE PART OF THE FORCING FUNCTION DUE TO BOUNDARY COND.         *BCC  047
C                                                                              *BCC  048
C*************************************************************************** BCC  049
  213 KLAST=1                                                                  BCC  050
      DO 60  I=1,N                                                             BCC  051
      FRCBC(I)=0.0                                                             BCC  052
      IF(ICOL(KLAST)-I)60,25,60                                                BCC  053
   25 DO 30 J=1,6                                                              BCC  054
      JJ=(J-1)*3+1                                                             BCC  055
      K=IARAY(I,J)                                                             SEPT 74
      IF(K-I) 30,35,30                                                         SEPT 74
   30 CONTINUE                                                                 BCC  058
C*************************************************************************** SEPT 74
C                                                                              SEPT 74
C     GENERATE (I,K) LENGTH ELEMENT                                            SEPT 74
C                                                                              SEPT 74
C*************************************************************************** SEPT 74
   35 DO 77  KK=1,6                                                            SEPT 74
      IF(NLA(I,KK)-K)77,81,77                                                  SEPT 74
   81 XLENI=LA(I,KK)                                                           SEPT 74
      GO TO 82                                                                 SEPT 74
   77 CONTINUE                                                                 SEPT 74
C*************************************************************************** SEPT 74
C                                                                              SEPT 74
C     GENERATE (K,I) LENGTH ELEMENT                                            SEPT 74
C                                                                              SEPT 74
C*************************************************************************** SEPT 74
```

```
   82 DO 78 KK=1,6                                                          SEPT 74
      IF(NLA(K,KK)-I)78,84,78                                               SEPT 74
   84 XLEN2=LA(K,KK)                                                        SEPT 74
      GO TO 83                                                              SEPT 74
   78 CONTINUE                                                              SEPT 74
   83 ALPHA(J)=0.5                                                          SEPT 74
      EPRIM(J)=ARRAY(I,JJ)*ARRAY(I,JJ+1)*417.1166/(XLEN2+XLEN1)             SEPT 74
      IF(-0.5+EPRIM(J)/ABS(ARRAY(I,JJ+2)))40,45,45                          BCC  061
   40 ALPHA(J)=1.0-EPRIM(J)/(2.0*ABS(ARRAY(I,JJ+2)))                        BCC  062
   45 IF(ARRAY(I,JJ+2))50,55,55                                             BCC  063
   50 FRCBC(I)=8.34*BC(KLAST)*(EPRIM(J)-ALPHA(J)*ARRAY(I,JJ+2))             BCC  064
      KLAST=KLAST+1                                                         BCC  065
      GO TO 60                                                              BCC  066
   55 FRCBC(I)=8.34*BC(KLAST)*(EPRIM(J)-(1.0-ALPHA(J))*ARRAY(I,JJ+2))       BCC  067
      KLAST=KLAST+1                                                         BCC  068
   60 CONTINUE                                                              BCC  069
 1000 RETURN                                                                BCC  070
      END                                                                   BCC  071
      SUBROUTINE RATE                                                       RATE 000
C*************************************************************************** RATE 001
C*************************************************************************** RATE 002
C                                                                           RATE 003
C                          SUBROUTINE RATE                                  RATE 004
C                                                                           RATE 005
C     THE PRIMARY PURPOSES OF THIS SUBROUTINE ARE TO CONSTRUCT THE          RATE 006
C     SPECIFIC CONSTITUENT SYSTEM MATRIX, AC  AND TO COMPLETE THE           RATE 007
C     APPROPRIATE FORCING FUNCTIONS                                         RATE 008
C                                                                           RATE 009
C*************************************************************************** RATE 010
C*************************************************************************** RATE 011
      REAL KA(100),KD(100),KSTOR(100),LOAD                                  RATE 012
      DIMENSION Y(100),BD(100),PMR(100)                                     RATE 013
      COMMON/ALGAE/NX,N,INDIC,ICON,MMN,NMK                                  RATE 014
      COMMON/BASS/MX,IPRNT,FRCBC(100)                                       RATE 015
      COMMON/PIKE/JCON,H(100),T(100),VOL(100),AB(100,100)                   RATE 016
      COMMON/TROUT/W(100),XK1(100),B(100,1),AC(100,100)                     RATE 017
    1 FORMAT(16F5.0)                                                        RATE 018
   12 FORMAT(F10.0,I3)                                                      RATE 019
   16 FORMAT(' BOD LOAD =',F10.0,' POUNDS/DAY FOR SECTION ',I3)             RATE 020
   17 FORMAT(' DEF LOAD =',F10.0,' POUNDS/DAY FOR SECTION ',I3)             RATE 021
   18 FORMAT('1')                                                           RATE 022
   34 FORMAT('1 SECTION   CHLORIDE BOUNDARY LOAD (MGD*MG/L)'///)            RATE 023
   35 FORMAT(2X,I5,12X,F14.2)                                               RATE 024
   36 FORMAT('1   THE TEMPERATURE CORRECTION FACTOR FOR THE BOD REMOVAL     RATE 025
     1RATE = ',F7.3,///,'  SECTION   BOD REM RATE  BOD WASTE LOAD   BOD     RATE 026
     1BOUNDARY LOAD'/12X,'*TEMP CORR*',5X,'(MGD*MG/L)',7X,'(MGD*MG/L)')     RATE 027
   37 FORMAT(2X,I5,F12.3,7X,F12.3,5X,F12.3)                                 RATE 028
   38 FORMAT('1  THE TEMPERATURE CORRECTION FACTOR FOR THE REAERATION RARATE 029
     1TE = ',F7.3,/'   THE BOD CORRECTION FACTOR (FL) = ',F7.3,/'   THE     RATE 030
     1TEMPERATURE CORRECTION FACTOR FOR THE BENTHAL DEMAND = ',F7.3,/'      RATE 031
     1 THE TEMP CORR FACTOR FOR KD = ',F7.3,///'   SECTION   DEOX RATE     RATE 032
     1 REAER RATE   BOUND COND LOAD   TOTAL LOAD     PMR          BD'/,12RATE 033
     1X,2('*TEMP CORR*  '),2('  (MGD*MG/L)',4X),'MG/L/D  *TEMP CORR*')     RATE 034
   39 FORMAT(2X,I6,5X,F6.3,7X,F6.3,6X,F12.3,3X,F12.3,2X,F9.3,3X,F9.3)       RATE 035
```

```
C*************************************************************************RATE 036
C                                                                         *RATE 037
C     READ IN DECAY RATE FOR BOD SYSTEM, REAERATION RATE FOR DO DEFICIT*RATE 038
C     SYSTEM                                                               RATE 039
C     IF THE K'S ARE THE SAME FOR EVERY SECTION THEN JUST INPUT K(1)    *RATE 040
C     READ AND APPLY TEMPERATURE CORRECTION FACTORS                     *RATE 041
C                                                                         *RATE 042
C*************************************************************************RATE 043
      IF(MMM-5)8,386,386                                                  RATE 044
    8 READ(MX,1)FAC1,(KA(I),I=1,15)                                       RATE 045
      IF(KA(2))202,3,202                                                  RATE 046
    3 DO 64 I=2,N                                                         RATE 047
   64 KA(I)=KA(1)                                                         RATE 048
      GO TO 334                                                           RATE 049
  202 LMM=N-15                                                            RATE 050
      IF(LMM)334,334,203                                                  RATE 051
  203 READ(MX,1)      (KA(I),I=16,N)                                      RATE 052
  334 IF(MMM-3)2,388,2                                                    RATE 053
  388 DO 389 I=1,N                                                        RATE 054
  389 KSTOR(I)=KA(I)                                                      RATE 055
      FAC4=FAC1                                                           RATE 056
      GO TO 2                                                             RATE 057
  386 DO 387 I=1,N                                                        RATE 058
  387 KA(I)=KSTOR(I)                                                      RATE 059
      FAC1=FAC4                                                           RATE 060
    2 DO 6 I=1,N                                                          RATE 061
      KA(I)=KA(I)*FAC1          **(T(I)-20.)                              RATE 062
C*************************************************************************RATE 063
C                                                                         *RATE 064
C     CALCULATE THE SPECIFIC CONSTITUENT SYSTEM MATRIX, AC                *RATE 065
C                                                                         *RATE 066
C*************************************************************************RATE 067
      DO 801 J=1,N                                                        RATE 068
  801 AC(I,J)=AB(I,J)                                                     RATE 069
    6 AC(I,I)=AB(I,I)+KA(I)*VOL(I)                                        RATE 070
      IF(INDIC)10,10,54                                                   RATE 071
C*************************************************************************RATE 072
C                                                                         *RATE 073
C     CALCULATE THE BOD OR CHLORIDE FORCING FUNCTION                      *RATE 074
C                                                                         *RATE 075
C*************************************************************************RATE 076
   10 DO 9 I=1,N                                                          RATE 077
      W(I)=FRCBC(I)                                                       RATE 078
    9 FRCBC(I)=FRCBC(I)/8.34                                              RATE 079
      LLL=0                                                               RATE 080
 1100 READ(MX,12)LOAD,ISEC                                                RATE 081
      IF(ISEC)13,22,13                                                    RATE 082
   13 IF(LLL)28,28,14                                                     RATE 083
   28 WRITE(NX,18)                                                        RATE 084
   14 W(ISEC)=W(ISEC)+LOAD                                                RATE 085
      LLL=LLL+1                                                           RATE 086
      WRITE(NX,16)LOAD,ISEC                                               RATE 087
      GO TO 1100                                                          RATE 088
C*************************************************************************RATE 089
```

```
C                                                                       *RATE 090
C     READ IN DEOXYGENATION RATE                                        *RATE 091
C     IF THE KD'S ARE THE SAME FOR EVERY SECTION THEN JUST INPUT KD(1)  *RATE 092
C                                                                       *RATE 093
C     READ IN PHOTOSYNTHESIS MINUS RESPIRATION(PMR),  MG./L./DAY        *RATE 094
C     READ IN BOTTOM DEMAND(BD), GM./SQ. M./DAY                         *RATE 095
C     VALUES MUST BE READ IN FOR EACH SECTION                           *RATE 096
C     READ AND APPLY TEMPERATURE CORRECTION FACTORS                     *RATE 097
C                                                                       *RATE 098
C*********************************************************************RATE 099
   54 READ(MX,1)FAC2,FL,(KD(J),J=1,14)                                   RATE 100
      DO 15 I=1,N                                                        RATE 101
      Y(I)=FRCBC(I)                                                      RATE 102
   15 FRCBC(I)=FRCBC(I)/8.34                                             RATE 103
      IF(KD(2))48,48,199                                                 RATE 104
   48 DO 47 J=2,N                                                        RATE 105
   47 KD(J)=KD(1)                                                        RATE 106
      GO TO 19                                                           RATE 107
  199 LII=N-14                                                           RATE 108
      IF(LII)19,19,310                                                   RATE 109
  310 READ(MX,1)(KD(I),I=15,N)                                           RATE 110
   19 IF(MMM-5)55,58,58                                                  RATE 111
   55 READ(MX,1)(PMR(I),I=1,N)                                           RATE 112
      READ(MX,1)FAC3,(BD(I),I=1,N)                                       RATE 113
      DO 56 I=1,N                                                        RATE 114
   56 BD(I)=BD(I)*FAC3  **(T(I)-20.)                                     RATE 115
      GO TO 57                                                           RATE 116
   58 DO 53 I=1,N                                                        RATE 117
      PMR(I)=0.                                                          RATE 118
   53 BD(I)=0.                                                           RATE 119
   57 DO 33 I=1,N                                                        RATE 120
      KD(I)=KD(I)*FAC2  **(T(I)-20.)                                     RATE 121
C*********************************************************************** RATE 122
C                                                                       *RATE 123
C     READ AND WRITE LOADS                                              *RATE 124
C     CALCULATE THE DEFICIT FORCING FUNCTION                            *RATE 125
C                                                                       *RATE 126
C*********************************************************************** RATE 127
      XK1(I)=8.34*VOL(I)*(XK1(I)*KD(I)*FL -PMR(I)+BD(I)*H(I))+Y(I)       RATE 128
   33 CONTINUE                                                           RATE 129
      LLL=0                                                              RATE 130
  101 READ(MX,12)LOAD,ISEC                                               RATE 131
      IF(ISEC)21,22,21                                                   RATE 132
   21 IF(LLL)326,326,325                                                 RATE 133
  326 WRITE(NX,18)                                                       RATE 134
  325 XK1(ISEC)=XK1(ISEC)+LOAD                                           RATE 135
      LLL=LLL+1                                                          RATE 136
      WRITE(NX,17)LOAD,ISEC                                              RATE 137
      GO TO 101                                                          RATE 138
   22 IF(INDIC)29,30,31                                                  RATE 139
   29 DO 303 I=1,N                                                       RATE 140
  303 W(I)=W(I)/8.34                                                     RATE 141
      WRITE(NX,34)                                                       RATE 142
      WRITE(NX,35)(I,W(I),I=1,N)                                         RATE 143
```

```
      GO TO 40                                                          RATE 144
   30 DO 304 I=1,N                                                      RATE 145
  304 W(I)=W(I)/8.34                                                    RATE 146
      WRITE(NX,36)FAC1                                                  RATE 147
      DO 853 I=1,N                                                      RATE 148
  853 W(I)=W(I)-FRCBC(I)                                                RATE 149
      WRITE(NX,37)(I,KA(I),W(I),FRCBC(I),I=1,N)                         RATE 150
      DO 854 I=1,N                                                      RATE 151
  854 W(I)=FRCBC(I)+W(I)                                                RATE 152
      GO TO 40                                                          RATE 153
   31 DO 305 I=1,N                                                      RATE 154
  305 XK1(I)=XK1(I)/8.34                                                RATE 155
       WRITE(NX,38)FAC1,FL,FAC3,FAC2                                    RATE 156
      WRITE(NX,39)(I,KD(I),KA(I),FRCBC(I),XK1(I),PMR(I),BD(I),I=1,N)    RATE 157
   40 IF(INDIC)843,843,288                                              RATE 158
  288 DO 842 I=1,N                                                      RATE 159
  842 W(I)=XK1(I)                                                       RATE 160
  843 DO 841 I=1,N                                                      RATE 161
  841 B(I,1)=W(I)                                                       RATE 162
      RETURN                                                            RATE 163
      END                                                               RATE 164
      SUBROUTINE WRITE                                                  WRITE000
C*********************************************************************** WRITE001
C*********************************************************************** WRITE002
C                                                                       WRITE003
C                              SUBROUTINE WRITE                         WRITE004
C                                                                       WRITE005
C     THIS SUBROUTINE OUTPUTS MATRIX AC AND MATRIX AC**-1               WRITE006
C                                                                       WRITE007
C*********************************************************************** WRITE008
C*********************************************************************** WRITE009
      COMMON/ALGAE/NX,N,INDIC,ICON,MMM,NMK                              WRITE010
      COMMON/TROUT/W(100),XK1(100),B(100,1),AC(100,100)                 WRITE011
  800 FORMAT('1',50X,'INITIAL SYSTEM MATRIX')                           WRITE012
  803 FORMAT(///48X,'INVERTED SYSTEM MATRIX')                           WRITE013
 1000 FORMAT(    /4X,9I13/)                                             WRITE014
 1005 FORMAT('1',51X,'INITIAL BOD MATRIX')                              WRITE015
 1007 FORMAT('1',48X,'INITIAL DO DEFICIT MATRIX')                       WRITE016
 1012 FORMAT(///50X,'INVERTED BOD MATRIX')                              WRITE017
 1015 FORMAT(///47X,'INVERTED DO DEFICIT MATRIX')                       WRITE018
 2000 FORMAT(I3,1X,9E13.5/(4X,9E13.5))                                  WRITE019
      JMAX=N                                                            WRITE020
      MM=1                                                              WRITE021
      IF(NMK)1003,1003,1010                                             WRITE022
 1003 IF(INDIC)1018,1004,1006                                           WRITE023
 1018 WRITE(NX,800)                                                     WRITE024
      GO TO 1008                                                        WRITE025
 1004 WRITE(NX,1005)                                                    WRITE026
      GO TO 1008                                                        WRITE027
 1006 WRITE(NX,1007)                                                    WRITE028
 1008 WRITE(NX,1000) (J,J=1,JMAX)                                       WRITE029
      DO 1001 I=1,JMAX                                                  WRITE030
 1001 WRITE(NX,2000)I,(AC(I,J),J=1,JMAX)                                WRITE031
      NMK=NMK+1                                                         WRITE032
```

```
      RETURN                                                            WRITE033
 1010 IF(INDIC)1019,1011,1014                                           WRITE034
 1019 WRITE(NX,803)                                                     WRITE035
      GO TO 1016                                                        WRITE036
 1011 WRITE(NX,1012)                                                    WRITE037
      GO TO 1016                                                        WRITE038
 1014 WRITE(NX,1015)                                                    WRITE039
 1016 WRITE(NX,1000) (J,J=1,JMAX)                                       WRITE040
      DO 1002 I=1,JMAX                                                  WRITE041
 1002 WRITE(NX,2000)I,(AC(I,J),J=1,JMAX)                                WRITE042
      NMK=0                                                             WRITE043
      RETURN                                                            WRITE044
      END                                                               WRITE045
      SUBROUTINE MATN                                                   MATN 000
C*********************************************************************** MATN 001
C*********************************************************************** MATN 002
C                                                                       MATN 003
C                              SUBROUTINE MATN                          MATN 004
C                                                                       MATN 005
C     THIS SUBROUTINE INVERTS THE MATRIX AC AND MULTIPLIES              MATN 006
C     IT BY THE FORCING FUNCTION                                        MATN 007
C                                                                       MATN 008
C*********************************************************************** MATN 009
C*********************************************************************** MATN 010
      DIMENSION IPIVO(100),INDEX(100,2),PIVOT(100)                      MATN 011
      COMMON/ALGAE/NX,N,INDIC,ICON,MMM,NMK                              MATN 012
      COMMON/TROUT/W(100),XK1(100),B(100,1),AC(100,100)                 MATN 013
      DO 900 I=1,N                                                      MATN 014
      B(I,1)=B(I,1)/1000000.                                            MATN 015
      DO 900 J=1,N                                                      MATN 016
  900 AC(I,J)=AC(I,J)/1000000.                                          MATN 017
      MM=1                                                              MATN 018
      DO 20 J=1,N                                                       SEPT 74
   20 IPIVO (J)=0                                                       MATN 021
      DO 550 I=1,N                                                      SEPT 74
C                                                                       MATN 023
C     SEARCH FOR PIVOT ELEMENT                                          MATN 024
C                                                                       MATN 025
      AMAX=0.0                                                          SEPT 74
      DO 105 J=1,N                                                      SEPT 74
      IF (IPIVO (J)-1) 60, 105, 60                                      SEPT 74
   60 DO 100 M=1,N                                                      MATN 029
      IF (IPIVO (M)-1) 80, 100, 740                                     SEPT 74
   80 IF (ABS(AMAX)-ABS(AC(J,M)))85, 100, 100                           MATN 031
   85 IROW=J                                                            MATN 032
      ICOLU =M                                                          SEPT 74
      AMAX=AC(J,M)                                                      SEPT 74
  100 CONTINUE                                                          MATN 035
  105 CONTINUE                                                          MATN 036
      IPIVO (ICOLU )=IPIVO (ICOLU )+1                                   SEPT 74
C                                                                       MATN 038
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL                 MATN 039
C                                                                       MATN 040
      IF (IROW-ICOLU ) 150, 260, 150                                    SEPT 74
```

```
  150 DO 200 L=1,N                                                    MATN 043
      SWAP=AC(IROW,L)                                                 SEPT 74
      AC(IROW,L)=AC(ICOLU,L)                                          SEPT 74
  200 AC(ICOLU,L)=SWAP                                                MATN 046
      IF(MM)260, 260, 210                                             SEPT 74
  210 DO 250 L=1, MM                                                  MATN 048
      SWAP=B(IROW,L)                                                  SEPT 74
      B(IROW,L)=B(ICOLU ,L)                                           SEPT 74
  250 B(ICOLU ,L)=SWAP                                                MATN 051
  260 INDEX(I,1)=IROW                                                 MATN 052
      INDEX(I,2)=ICOLU                                                SEPT 74
      PIVOT(I)=AC(ICOLU,ICOLU)                                        SEPT 74
C                                                                     MATN 056
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT                               MATN 057
C                                                                     MATN 058
      AC(ICOLU,ICOLU)=1.0                                             SEPT 74
      DO 350 L=1,N                                                    SEPT 74
  350 AC(ICOLU,L)=AC(ICOLU,L)/PIVOT(I)                                MATN 061
      IF(MM)380, 380, 360                                             SEPT 74
  360 DO 370 L=1,MM                                                   MATN 063
  370 B(ICOLU ,L)=B(ICOLU ,L)/PIVOT(I)                                MATN 064
C                                                                     MATN 065
C     REDUCE NON-PIVOT ROWS                                           MATN 066
C                                                                     MATN 067
  380 DO 550 L1=1,N                                                   MATN 068
      IF(L1-ICOLU ) 400, 550, 400                                     SEPT 74
  400 S=AC(L1,ICOLU)                                                  MATN 070
      AC(L1,ICOLU)=0.0                                                SEPT 74
      DO 450 L=1,N                                                    SEPT 74
  450 AC(L1,L)=AC(L1,L)-AC(ICOLU,L)*S                                 MATN 073
      IF(MM)550, 550, 460                                             SEPT 74
  460 DO 500 L=1,MM                                                   MATN 075
  500 B(L1,L)=B(L1,L)-B(ICOLU ,L)*S                                   MATN 076
  550 CONTINUE                                                        MATN 077
C                                                                     MATN 078
C     INTERCHANGE COLUMNS                                             MATN 079
C                                                                     MATN 080
      DO 710 I=1,N                                                    SEPT 74
      L=N+1-I                                                         SEPT 74
      IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630                        SEPT 74
  630 JROW=INDEX(L,1)                                                 MATN 084
      JCOLU =INDEX(L,2)                                               SEPT 74
      DO 705 M=1,N                                                    SEPT 74
      SWAP=AC(M,JROW)                                                 SEPT 74
      AC(M,JROW)=AC(M,JCOLU)                                          SEPT 74
      AC(M,JCOLU)=SWAP                                                SEPT 74
  705 CONTINUE                                                        MATN 090
  710 CONTINUE                                                        MATN 091
  740 CONTINUE                                                        MATN 092
      DO 901 I=1,N                                                    MATN 094
      DO 901 J=1,N                                                    MATN 095
  901 AC(I,J)=AC(I,J)/1000000.                                        MATN 096
      RETURN                                                          MATN 097
      END                                                             MATN 098
```

```
      SUBROUTINE STORE                                                  STORE000
C*****************************************************************     STORE001
C*****************************************************************     STORE002
C                                                                       STORE003
C                          SUBROUTINE STORE                             STORE004
C                                                                       STORE005
C*****************************************************************     STORE006
C*****************************************************************     STORE007
      DIMENSION CL(100),CS(100),R(100)                                  STORE008
      COMMON/ALGAE/NX,N,INDIC,ICON,MMM,NMK                              STORE009
      COMMON/PIKE/JCON,H(100),T(100),VOL(100),AB(100,100)               STORE010
      COMMON/TROUT/W(100),XK1(100),B(100,1),AC(100,100)                 STORE011
   42 FORMAT('1 SECTION       DEFICIT(MG/L)')                         STORE012
   44 FORMAT('1 SECTION         BOD(MG/L)')                           STORE013
   45 FORMAT('1 SECTION     CHLORIDES(MG/L)')                         STORE014
   48 FORMAT(3X,I4,3X,F14.3)                                          STORE015
  807 FORMAT(10X,I3,5(4X,F9.2))                                       STORE016
  808 FORMAT('1        SECTION    CHLORIDES     TEMPERATURE     DEFICIT     STORE017
     1SATURATION        DO',//)                                       STORE018
  817 FORMAT('1','                          RESULTS FOR COMBINED (CARBONACEOUSTORE019
     4S AND NITROGENOUS ) RUN'//)                                     STORE020
  828 FORMAT('         SECTION    CHLORIDES     TEMPERATURE     DEFICIT     STORE021
     1SATURATION        DO',/,19X,'(MG/L)           (C)           (MG/L)     SEPT 74
     1    (MG/L)        (MG/L)')                                      SEPT 74
      DO 805 I=1,N                                                     STORE023
  805 XK1(I)=B(I,1)                                                    STORE024
C*****************************************************************     STORE025
C                                                                       STORE026
C     WRITES FINAL CONCENTRATIONS                                       STORE027
C                                                                       STORE028
C*****************************************************************     STORE029
   11 IF(INDIC)850,840,41                                              STORE030
  850 WRITE(NX,45)                                                     STORE031
      GO TO 643                                                        STORE032
  840 WRITE(NX,44)                                                     STORE033
      GO TO 643                                                        STORE034
   41 WRITE(NX,42)                                                     STORE035
  643 WRITE(NX,48)(I,XK1(I),I=1,N)                                     STORE036
 9999 GO TO(1800,1801,1802,1801,1803),MMM                              STORE037
 1800 GO TO(1199,1199,813,813),JCON                                    STORE038
C*****************************************************************     STORE039
C                                                                       STORE040
C     CALCULATES THE SATURATION VALUE OF DO                             STORE041
C                                                                       STORE042
C*****************************************************************     STORE043
  813 DO 815 I=1,N                                                     STORE044
      CL(I)=XK1(I)                                                     STORE045
  815 CS(I)=(1.-.000009*CL(I))*(14.652-.41022*T(I)+.0079910*           STORE046
     1T(I)**2.-.000077774*T(I)**3.)                                    STORE047
      MMM=MMM+1                                                        STORE048
      RETURN                                                           STORE049
 1199 DO 1139 I=1,N                                                    STORE050
      CL(I)=0.                                                         STORE051
 1139 CS(I)=(1.-.000009*CL(I))*(14.652-.41022*T(I)+.0079910*           STORE052
```

```
     1T(I)**2.-.000077774*T(I)**3.)                                    STORE053
      MMM=MMM+2                                                        STORE054
      RETURN                                                           STORE055
1801  MMM=MMM+1                                                        STORE056
      RETURN                                                           STORE057
1802  WRITE(NX,808)                                                    STORE058
      DO 806 I=1,N                                                     STORE059
      W(I)=CS(I)-XK1(I)                                                STORE060
      WRITE(NX,807)I,CL(I),T(I),XK1(I),CS(I),W(I)                      STORE061
 806  CONTINUE                                                         STORE062
      GO TO(93,1804,93,1804),JCON                                      STORE063
  93  RETURN                                                           STORE064
1804  DO 829 I=1,N                                                    STORE065
 829  R(I)=XK1(I)                                                      STORE066
      MMM=MMM+1                                                        STORE067
      RETURN                                                           STORE068
1803  WRITE(NX,817)                                                    STORE069
      WRITE(NX,828)                                                    STORE070
      DO 881 I=1,N                                                     STORE071
      XK1(I)=XK1(I)+R(I)                                               STORE072
      W(I)=CS(I)-XK1(I)                                                STORE073
      WRITE(NX,807)I,CL(I),T(I),XK1(I),CS(I),W(I)                      STORE074
 881  CONTINUE                                                         STORE075
      RETURN                                                           STORE076
      END                                                              STORE077
```

USER'S MANUAL

The manner in which the data deck for HARO3 is structured is shown in figure U-1. As can be seen, the cards can be divided into two basic groups:

1) General System Cards (GS cards) which are the same for any constituent or substance being modeled and need only be input once. They primarily contain the physical and hydrodynamic parameters of the system.

2) Specific Constituent Cards (SC cards) which as the name implies, are specific to the particular constituent being modeled. These include boundary concentrations and biological or chemical parameters such as rates and loads.

After reading the single set of GS cards, a variable number of sets of SC cards are read depending on the particular combination of reactions of the constituents being modeled. Two indicators are used to designate which combination is to be modeled:

1) INDIC: As can be seen from figure U-1, this indicator is read in prior to each set of SC cards. It can presently take on three values, -1, 0 and 1 to designate chlorides, BOD and DO deficit respectively

2) JCON: This indicator is read on the first GS card: the Title Card. It designates a particular combination or configuration of reactions and is utilized by subroutine STORE to direct the flow of the program.

Since HARO3 has been designed with the BOD-DO deficit system in mind, it is capable of handling reaction configurations which are to some extent peculiar to that system. However, it should be noted that there are many substances which are analagous to BOD and dissolved oxygen and can therefore be modeled using HARO3. With this in mind, the following are the particular combinations of reactants which can be presently modeled using the program:

1) Conservative substances: (Chlorides; total dissolved solids; some pesticides and herbicides; etc) any substance which does not decay or can be approximated by zero decay.

2) Single reactive substance: (BOD; coliform bacteria, ammonia; etc) any substance which decays according to first order kinetics

3) Coupled reactive substances: (BOD-DO deficit; polyphosphate - orthophosphate; ammonia-nitrate). A feed forward system of two substances reacting with first order kinetics.

4) Additive coupled substances: (CBOD, NBOD-DO deficit) this is peculiar to the BOD-DO deficit system where the BOD can be broken down into carbonaceous (CBOD) and nitrogenous (NOD) components. Both components deplete dissolved oxygen but at different rates so each is computed separately and their deficits added to determine the total deficit.

5) Estuarine coupled reactive substances (Chlorides, BOD-deficit) HARO3 has the additional option for those estuaries where chlorides have been modeled. A chloride run can procede the BOD-deficit so that the resulting chloride concentrations can be used to determine the saturation value of dissolved oxygen.

6) Estuarine additive coupled system (Chlorides, NBOD, CBOD-deficit). The same as 5 but for the additive system.

Figure U-2 summarizes the various deck setups for each of the above combinations with the appropriate values of INDIC and JCON.

A description of input and output is then followed by an example problem including sample input and output to further clarify the procedure.

Finally, some restrictions on the use of the program are given.

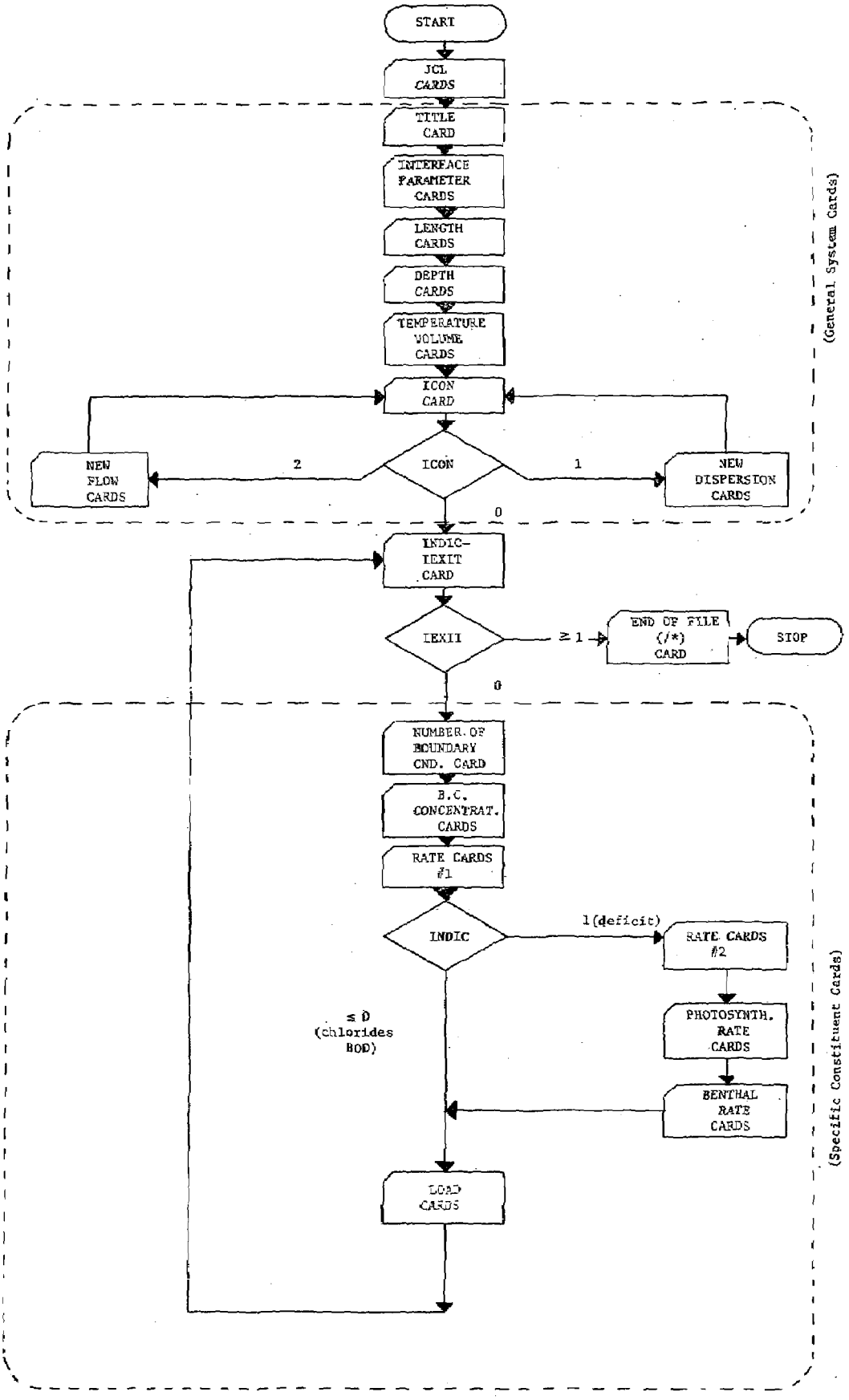

Figure B-1: *Structure of data deck for HARO3*

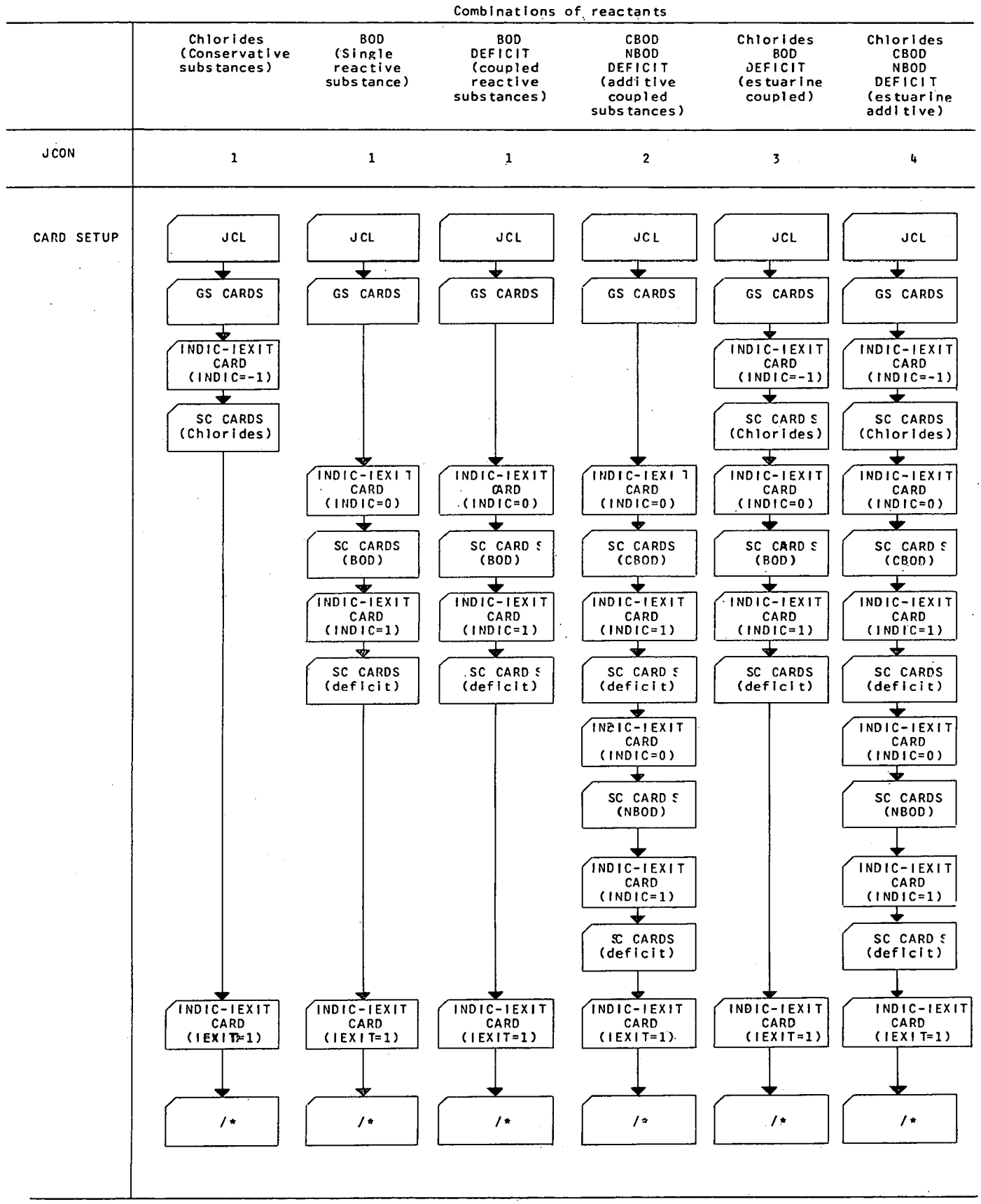

Figure U-2 : Deck setups for various combinations of reactants using program HAR03.

GENERAL SYSTEM CARDS

[TITLE CARD]

Columns	Variable	Description	Units	Format
(1-40)	TITLE	To be used to label the run		A40
(41-43)	N	The number of sections in the model		I3
(44-46)	IPRNT	An indicator, IPRNT=1 for printout of matrices IPRNT=0 to suppress the printout of matrices		I3
(47-49)	JCON	An indicator used to designate the type of system being modeled JCON=1 BOD Deficit JCON=2 CBOD NBOD Deficit JCON=3 Chlorides BOD Deficit JCON=4 Chlorides CBOD NBOD Deficit		I3
(53-59)	SCALE(1)	Area scale factor	ft^2/ou*	F7
(60-66)	SCALE(2)	Dispersion scale factor	mi^2/day/ou	F7
(67-73)	SCALE(3)	Flow scale factor	cfs/ou	F7
(74-80)	SCALE(4)	Length scale factor	ft/ou	F7

*note: ou stands for the "original units" in which the parameter to be converted by the scale factor is expressed. The purpose of the scale factors is to allow the user to input parameters in units which are different from those specified on the following pages. For instance, according to this program length should be input as feet. However, it may be more convenient to enter it in miles and set SCALE(4) to 5280. The program would then internally convert the length from miles to feet

[INTERFACE PARAMETER CARDS (2 per section; total number = 2*N)]

Columns	Variable	Description	Units	Format
(1-6)	AREA	Cross sectional area of interface with section	ft^2	F6
(7-12)	E	Dispersion coefficient of first interface with section	mi^2/day	F6
(13-18)	Q	Flow * across the first interface with section	cfs	F6
(19-21)	IARAY	Section forming the first interface with section		I3
(22-27)	AREA	Cross sectional area of second interface	ft^2	F6
(28-33)	E	Dispersion coefficient of second interface	mi^2/day	F6
(34-39)	Q	Flow across the second interface	cfs	F6
(40-42)	IARAY	Section forming the second interface		I3
(43-48)	AREA	Cross sectional area of third interface	ft^2	F6
(49-54)	E	Dispersion coefficient of third interface	mi^2/day	F6
(55-60)	Q	Flow across the third interface	cfs	F6
(61-63)	IARAY	Section forming the third interface		I3

(note: The second interface parameter card is identical to above, but for the fourth, fifth and sixth interfaces. If the section in question has an interface which forms a boundary across which mass is transported (that is, a boundary condition) interface parameters must be input for it. To do this, input the appropriate AREA, E and Q for the interface and input the section's number as the IARAY.

It is only necessary to input the parameters for a particular interface once. In other words after inputting the parameters of the interface of section 1 with section 2, it is not necessary to input the parameters of the interface of section 2 with section 1.)

* flow out of a section is positive; flow into a section is negative.

[CHARACTERISTIC LENGTHS](1 card per section; 1 length for each interface; therfore up to 6 lengths per card)

Columns	Variable	Description	Units	Format
1-3	JI	First section which forms an interface with section I		I3
4-13	LA(I,JI)	Length of section I with respect to section JI*	feet	F10
14-16	JJ	Second section which forms an interface with section I.		I3
17-26	LA(I,JJ)	Length of section I with respect to section JJ	feet	F10
27-29	JK	Third section which forms an interface with section I		I3
30-39	LA(I,JK)	Length of section I with respect to section JK	feet	F10
40-42	JL	Fourth section which forms an interface with section I		I3
43-52	LA(I,JL)	Length of section I with respect to section JL	feet	F10
53-55	JM	Fifth section which forms an interface with section I		I3
56-65	LA(I,JM)	Length of section I with respect to section JM	feet	F10
66-68	JN	Sixth section which forms an interface with section I		I3
69-78	LA(I,JN)	Length of section I with respect to section JN	feet	F10

* note: the following figure illustrates the interpretation of characteristic lengths as used in this program.

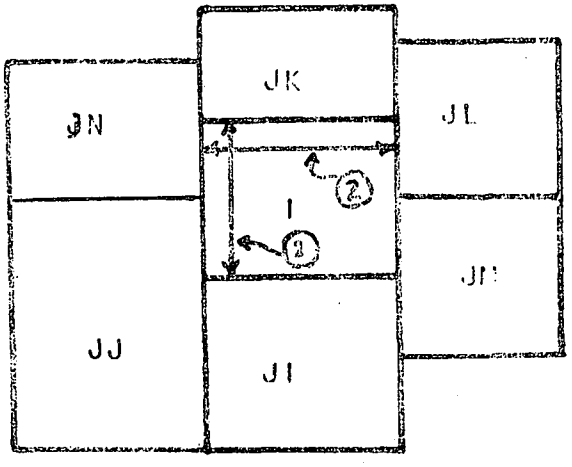

figure U-1: Depiction of characteristic lengths

where

① - LA(I,JK) and LA(I,JI)

② - LA(I,JJ), LA(I,JM) LA(I,JL) and LA(I,JN)

As can be seen, any other than orthogonally shaped segments would present a problem as to the measurement of LA. It is therefore reiterated that oddly shaped segmentation schemes should be avoided except when absolutely necessary to depict the natural geometry of the system.

[DEPTH CARDS]

Columns	Variable	Description	Units	Format
1-10	H(1)	Depth of 1st section	ft	F10
11-20	H(2)	Depth of 2nd section	ft	F10
-	-	-	-	
-	-	-	-	
-	-	-	-	
-	-	-	-	
-	H(N)	Depth of Nth section	ft	F10

[TEMPERATURE-VOLUME CARDS]

1-10	T(1)	Water temperature of 1st section	°C	F10
11-20	VOL(1)	Volume of 1st section	$10^6 ft^3$	F10
21-30	T(2)	Temperature of 2nd section	°C	F10
31-40	VOL(2)	Volume of 2nd section	$10^6 ft^3$	F10
-				
-				
-				
-	T(N)	Temperature of Nth section	°C	F10
-	VOL(N)	Volume of Nth section	$10^6 ft^3$	F10

[ICON CARD]

Columns	Variable	Description	Units	Format
(1-2)	ICON	An indicator. If desired at this point changes in the E's or Q's which were read in previously, may be made using ICON If ICON=0: No changes are called for and the program would continue by reading the INDIC card as described on page U-11 IF ICON=1: Would allow a new set of dispersion coefficients to be read as on "New E cards" (see below) If ICON=2: Would allow a new set of flows to be read as described on page U-11 (new Q Cards")		I2

[NEW E CARDS (OPTIONAL-MUST BE PRECEDED BY ICON CARD=1)]

Columns	Variable	Description	Units	Format
(1-5)	Revised E's	For 1st Interface with first section	mi^2/day	F5
(6-10)	"	" 2nd " " " "	"	"
(11-15)	"	" 3rd " " " "	"	"
(16-20)	"	" 4th " " " "	"	"
(21-25)	"	" 5th " " " "	"	"
(26-30)	"	" 6th " " " "	"	"
(31-35)	"	" 1st " " second "	"	"
(36-40)	"	" 2nd " " " "	"	"
(41-45)	"	" 3rd " " " "	"	"
(46-50)	"	" 4th " " " "	"	"
(51-55)	"	" 5th " " " "	"	"
(56-60)	"	" 6th " " " "	"	"

IN SIMILAR FASHION, REVISIONS ARE INPUTTED FOR THE THIRD TO Nth SECTIONS

note: These cards must be followed by another "ICON CARD" (as explained above). If revisions are to be made in flow, ICON must equal 2, and the new values can be read as on page U-11. If no more revisions are necessary, ICON must equal 0 and the program will proceed to read the "INDIC CARD" as explained on page U-11

[NEW Q CARDS (OPTIONAL-MUST BE PRECEDED BY ICON CARD=2)]*

Columns	Variable	Description	Units	Format
(1-5)	Revised Q's	For 1st Interface with First Section	cfs	F5
(6-10)	"	" 2nd " " " "	"	"
(11-15)	"	" 3rd " " " "	"	"
(16-20)	"	" 4th " " " "	"	"
(21-25)	"	" 5th " " " "	"	"
(26-30)	"	" 6th " " " "	"	"
(31-35)	"	" 1st " " Second "	"	"
(36-40)	"	" 2nd " " " "	"	"
(41-45)	"	" 3rd " " " "	"	"
(46-50)	"	" 4th " " " "	"	"
(51-55)	"	" 5th " " " "	"	"
(56-60)	"	" 6th " " " "	"	"

IN SIMILAR FASHION, REVISIONS ARE INPUTTED FOR THE THIRD TO Nth SECTIONS

These cards must be followed by another "ICON CARD" (as explained on page U-10). If revisions are to be made in dispersion, ICON must equal 1 and the new values can be read as described on page U-10. If no more revisions are necessary, ICON must equal 0 and the program will proceed to read the "INDIC CARD" as explained below.

[INDIC CARD]

Columns	Variable	Description	Units	Format
(1-2)	INDIC	An indicator used to denote whether subsequent DATA will be for Chloride, BOD or D.O. deficit INDIC = -1 for Chloride INDIC = 0 for BOD INDIC = 1 for D.O. deficit		I2
(3-4)	IEXIT	An indicator which can either continue or terminate the run If IEXIT>0 The run will Terminate IEXIT≤0 The run will Continue		I2

* on both the "New E Cards" and the "New Q Cards" the scale factors input on the title card (page U-6) apply.

SPECIFIC CONSTITUENT CARDS

[NUMBER OF BOUNDARY CONDITIONS]

Columns	Variable	Description	Units	Format
(1-2)	NUMBC	Number of boundary interfaces across which mass is transported		I2

[BOUNDARY CONDITIONS]

(1-8)	BC(1)	Concentration of water quality parameter at first boundary	mg/l	F8
(9-12)	ICOL(1)	Section adjoining first boundary		I4
	BC(2)	Concentration of water quality parameter at second boundary	mg/l	F8
	ICOL(2)	Section adjoining second boundary		I4
-	-	-		-
-	-	-		-
-	-	-		-
-	-	-		-
(61-68)	BC(6)	Concentration of water quality parameter at 6th boundary	mg/l	F8
(69-72)	ICOL(6)	Section adjoining 6th boundary		I4

Proceed with cards until a total of NUMBC boundary conditions have been read

[RATE CARD #1]***

(1-5)	FAC1	Temperature correction factor for first reaction rate	1/day	F5
(6-10)	K(1)*	First reaction rate for first section ** for Chlorides =0.0 for CBOD =CBOD removal rate(Kr) for NBOD =NBOD removal rate(Kn) for D.O. deficit =Reaeration rate	1/day	F5
	K(2)	First reaction rate for second section	1/day	F5
	-		-	-
	-		-	-
	-		-	-
	-		-	-
(76-80)	K(15)	First reaction rate for Nth section	1/day	F5

Proceed until N reaction rates have been input

* note: If all the rates are the same, only one value need be input as K(1) and the remaining values input as zero. The program has an internal test to see if K(2) is zero. If this is so, the program automatically sets all values of K equal to K(1). Only one card should be input when exercising this option and besides the value of K(1) the temperature correction factor should still be input.

** note: rates at 20° C

*** see page U-13

[RATE CARD #2] (Optional–for D.O. deficit only)

Columns	Variable	Description	Units	Format
(1-5)	FAC2	Temperature correction factor for second reaction rate (deoxygenation rate)		F5
(6-10)	FL	Ratio of ultimate to 5-day BOD		F5
(11-15)	KD(1)*	Deoxygenation rate for section 1**	1/day	F5
(16-20)	KD(2)	Deoxygenation rate for section 2	1/day	F5
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
(76-80)	KD(14)	Deoxygenation rate for section 14	1/day	F5
	(proceed until N rates have been input)			

[PHOTOSYNTHESIS RATE] (Optional for D.O. deficit only)***

(1-5)	PMR(1)	Net photosynthesis rate for section 1	mg/l/day	F5
(6-10)	PMR(2)	Net photosynthesis rate for section 2	mg/l/day	F5
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
(76-80)	PMR(16)	Net photosynthesis rate for section 16	mg/l/day	F5
	(proceed until N rates have been input)			

[BENTHAL DEOXYGENATION RATE] (Optional for D.O. deficit only)***

(1-5)	FAC3	Temperature correction factor for benthal demand		
(6-10)	BD(1)	Benthal demand for section 1**	$gm/M^2/day$	F5
(11-15)	BD(2)	Benthal demand for section 2	$gm/M^2/day$	F5
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
(76-80)	BD(15)	Benthal demand for section 15	$gm/M^2/day$	F5
	(proceed until N rates have been input)			

* As described on the previous page, if all rates are equal only KD(1) need be input and all other KD's can be left blank.

** rates at T=20° C

*** note: In an additive coupled run which calculates the deficit due to CBOD, the deficit due to NBOD and then adds them it is unnecessary to input the reaeration rates, the photosynthesis rate and the benthal demand with the NBOD part of the run. All such cards should be omitted rather than inputting them a second time. This is illustrated in the example problem which follows.

Columns	Variable	Description	Units	Format
(1-10)	LOAD	LOAD to a section	#/day	F10
(11-13)	ISEC	Section to which LOAD is applied		I3

(note - Individual LOAD CARDS are needed for each LOAD input. The last LOAD should be followed by a blank card which indicates that the previous card was the last LOAD to be input. If there are no loads merely input a blank card).

EXAMPLE PROBLEM

The following example problem is primarily intended to illustrate the input and output of HARO3. It should be noted that several simplifications have been made which would never be used in an actual modeling study and as such this example should not be used as a strict guide to applications of the program.

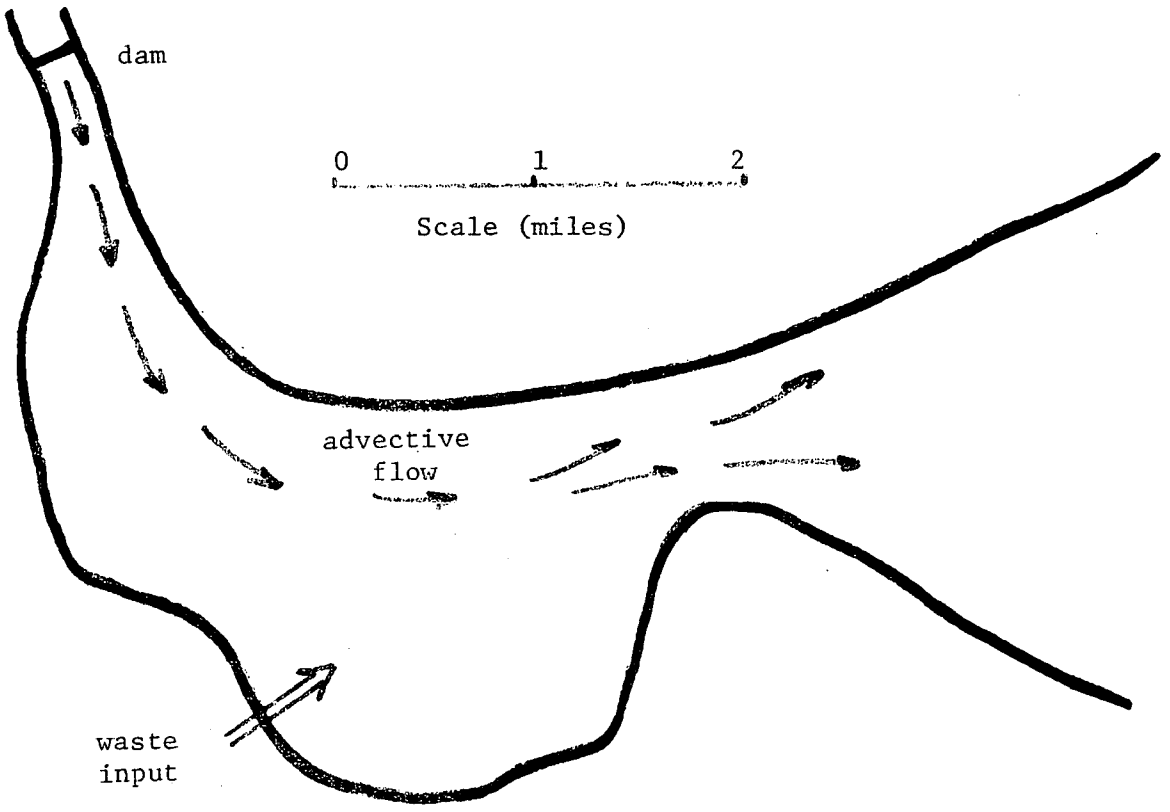

figure U-3: Overview of a fictitious tidal bay

The tidal bay depicted in figure U-3 has a dam situated at its head end and opens onto a large estuary at its mouth. As shown in figure U-4, the bay can be generally divided into a deep section on the north and shallow flats on the south. An investigation of the bays hydrodynamics has indicated that the fresh water flow entering the system over the dam makes its way to the sea along the northern coast and that relatively small amounts of water are transferred between the flats and the deeper section.

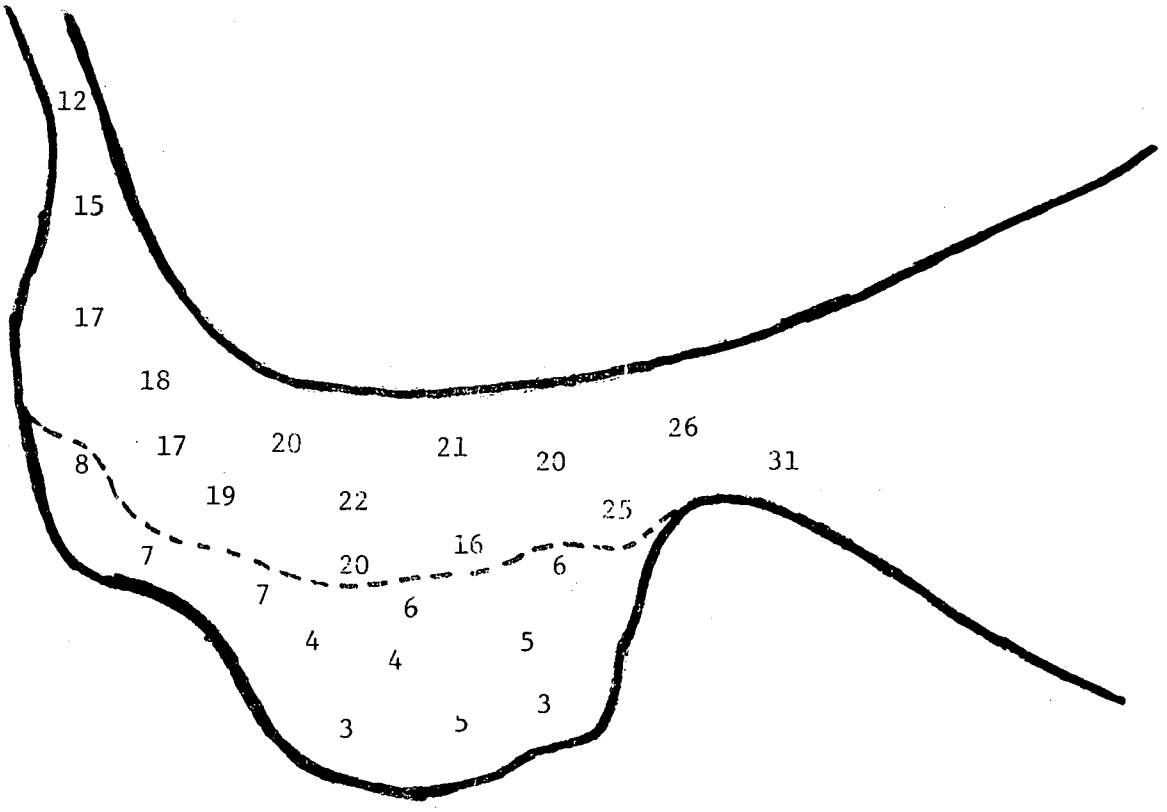

figure U-4: Depth readings (feet) for the tidal bay

At present, a large municipality discharges raw sewage directly to the southwest corner of the bay resulting in critical depression of dissolved oxygen levels as shown in figure U-5. One question which might be addressed using HARO3 would be whether discharging the waste into the deeper area to the north would significantly improve the water quality?

Utilizing the available data, the segmentation scheme shown in figure U-6 was developed. Data for the segments is given in tables U-1 and U-2.

The results of the analysis is shown in figure U-7. As can be seen, discharge into the northern part of the bay significantly improves water quality by taking advantage of the additional dilution, transport and mixing which exist there.

A printout of the input cards is given in figure U-8 for the run in which the load is discharged in the northern part of the bay. The corresponding output is also attached.

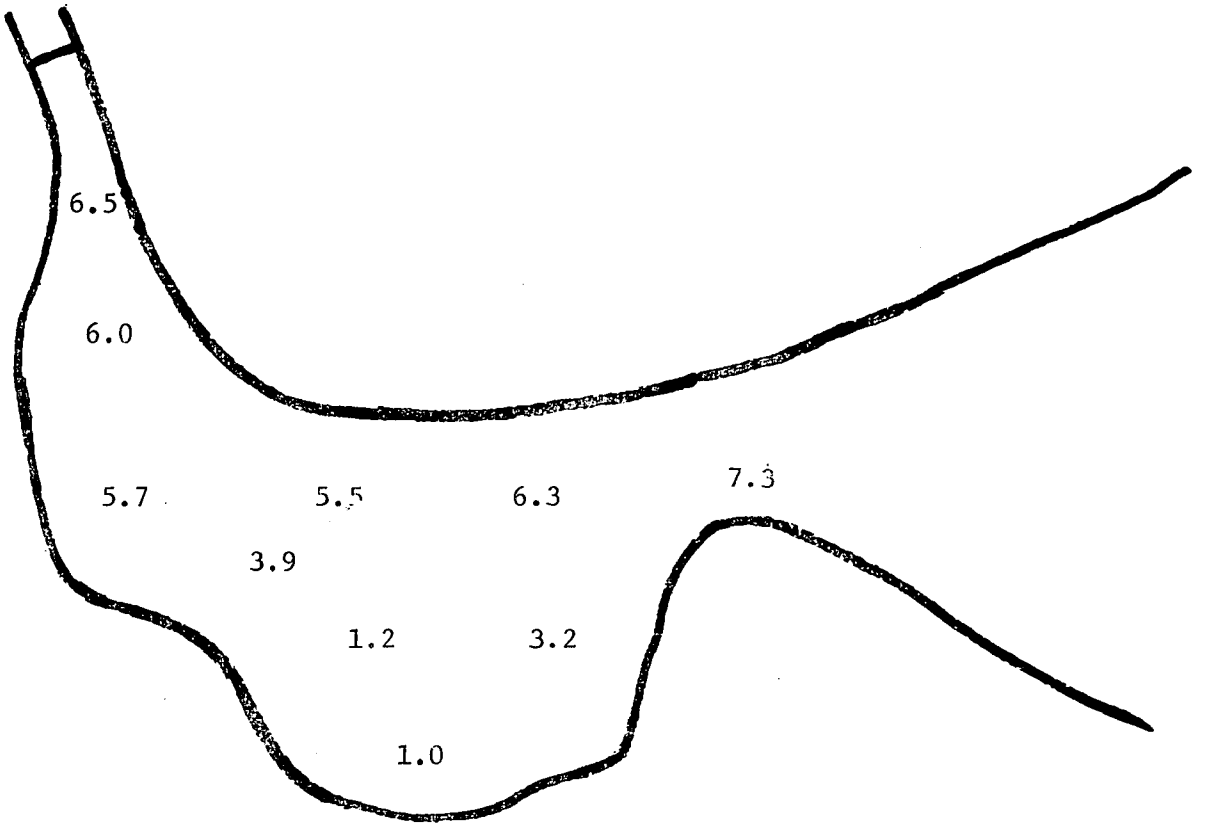

Figure U-5: Dissolved oxygen concentrations (mg/l)

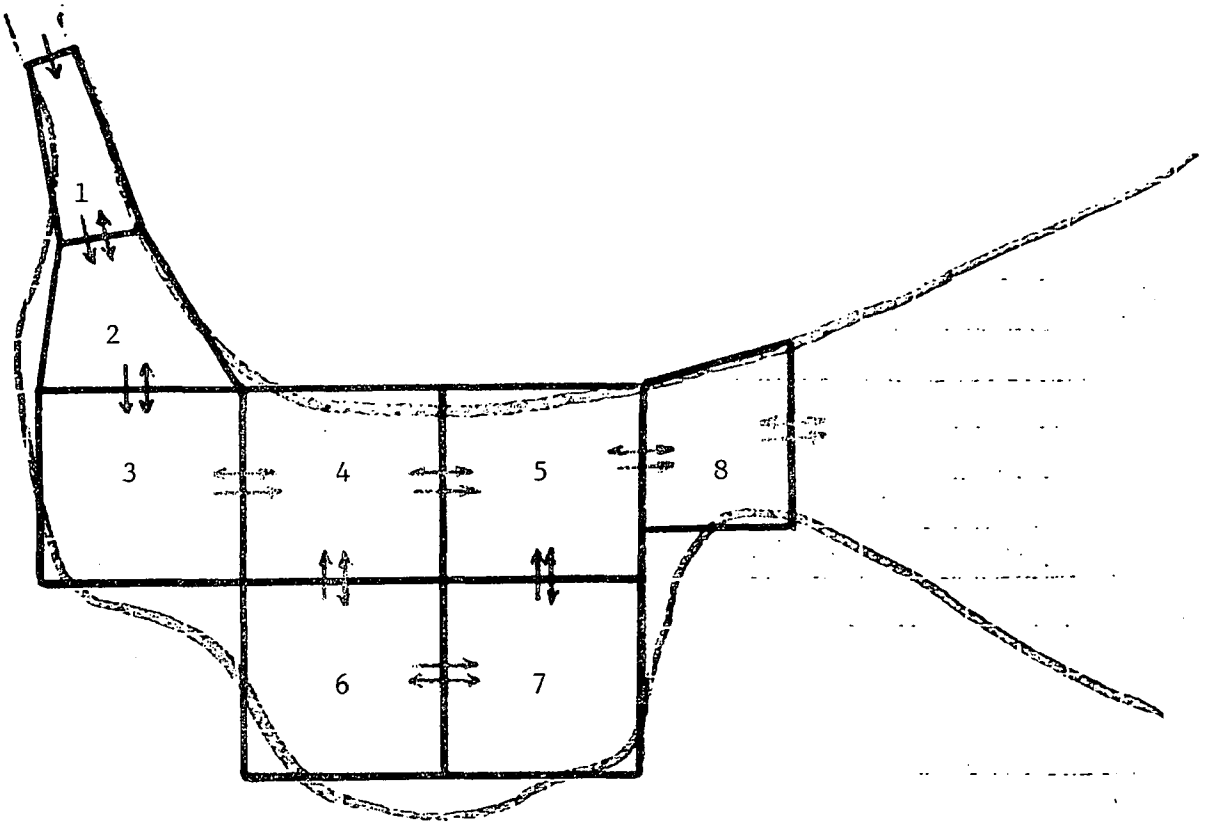

Figure U-6: Segmentation scheme for the bay

Interfaces

	1-1	1-2	2-3	3-4	4-5	4-6	6-7	5-7	5-8	8-8
Area (ft^2)		39600.	95040.	100320.	105600.	36960.	21200.	31680.	99000.	1584
Dispersion ($mile^2$/day)		0.5	.5	1.0	1.0	0.5	0.5	0.5	1.5	1.5
Flow (cfs)	150.	150.	150.	150.	243.				243.	243.
Boundary condition (mg/l)										
Chlorides										1000
CBOD	2.									.5
NBOD	2.									
DO deficit	1.									.5

Table U-1: Interface data for run with load to section 4

Sections

	1	2	3	4	5	6	7	8
Temperature (°c)	21.0	22.0	22.0	22.0	22.0	4.0	24.0	22.0
Depth (feet)	12.0	13.0	15.0	20.0	22.0	4.0	5.0	30.0
Volume (10^6 $feet^3$)	83.64	271.8	418.18	557.57	613.32	111.51	139.30	836.35
CBOD removal rate (1/day)	.3	.3	.3	.3	.3	.3	.3	.3
NBOD removal rate (1/day)	.3	.3	.3	.3	.3	.2	.3	.3
reaeration rate (1/day)	.24	.23	.22	.17	.15	.16	.18	.12
CBOD deoxygenation rate (1/day)	.1	.1	.1	.1	.1	.1	.1	.1
NBOD deoxygenation rate (1/day)	.1	.1	.1	.1	.1	.1	.1	.1
CBOD load (#/day)				100000.				
NBOD load (#/day)				100000.				
Benthal load (gm/M^2/day)						.05		
Algal load (mg/l/day)							.25	
Waste load flow (cfs)				93.				

Table U-2: Section data with load to section 4

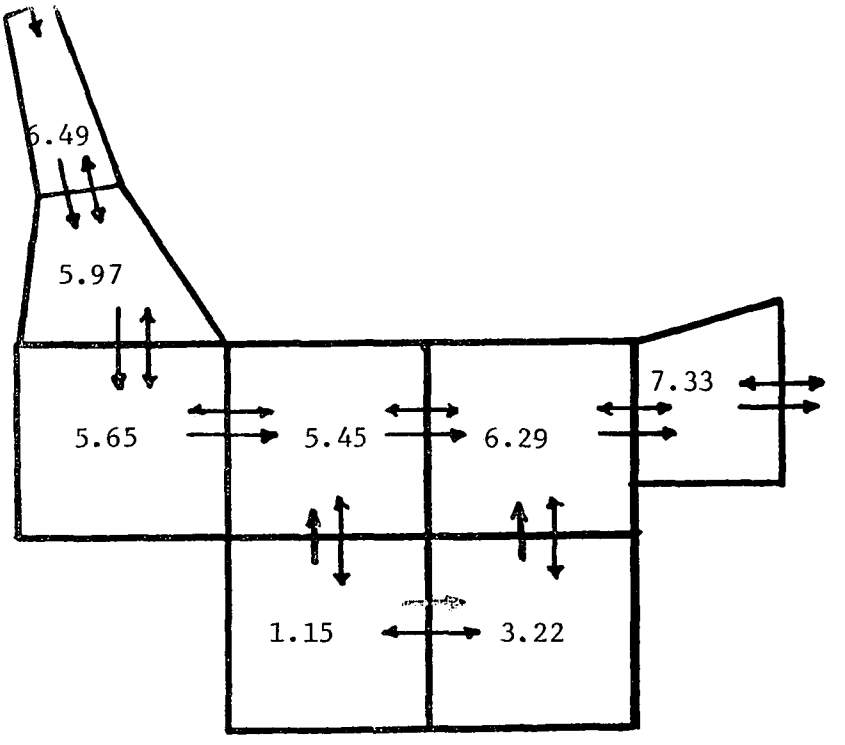

Figure U-7a: Results of model run with load to segment 6.

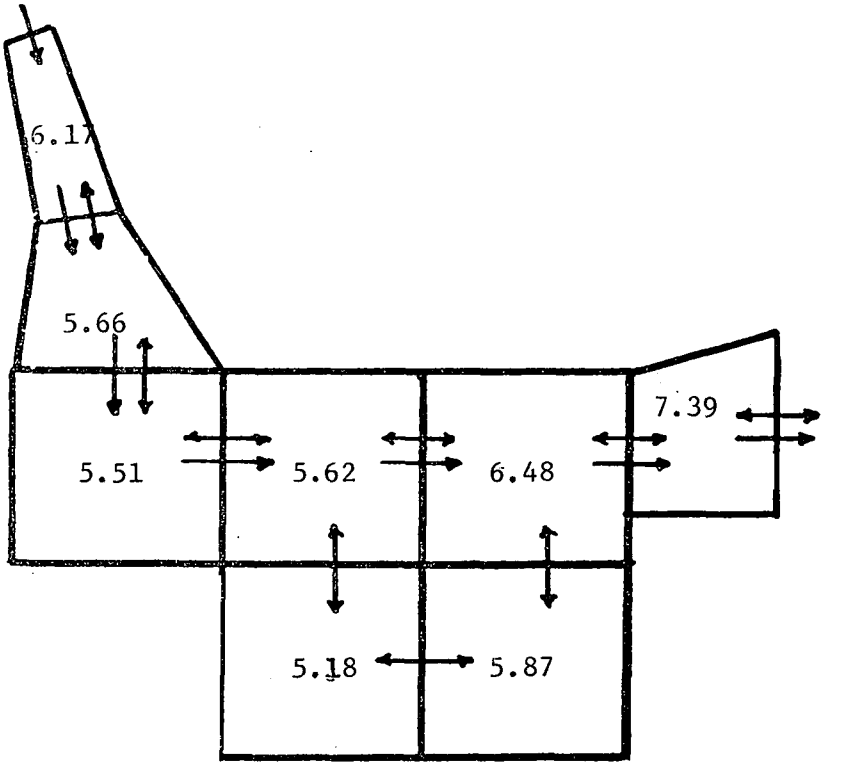

Figure U-7b: Results of model run with load to segment 4.

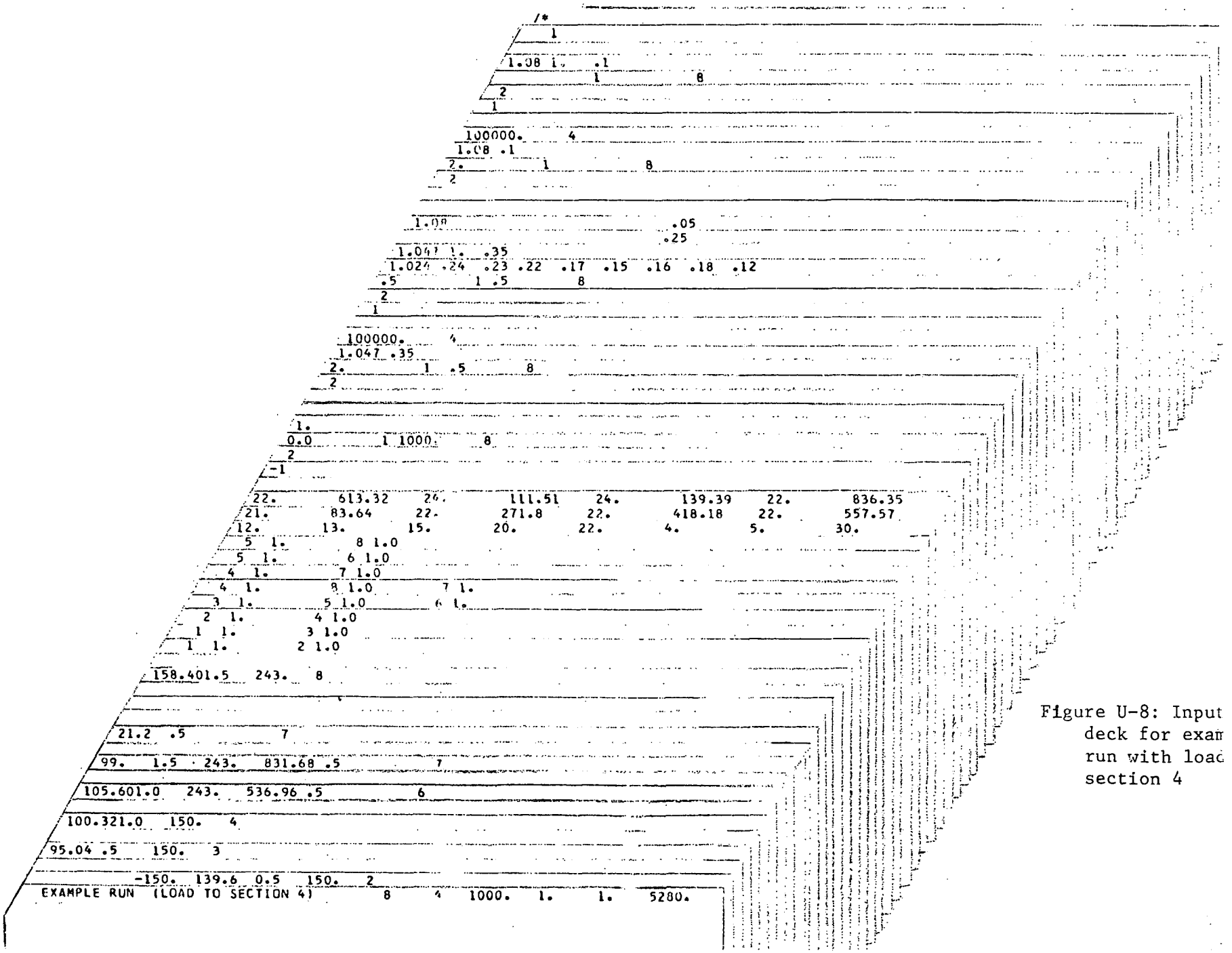

Figure U-8: Input deck for exam run with load section 4

EXAMPLE RUN (LOAD TO SECTION 4)

SYSTEM PARAMETERS

NUMBER OF SECTIONS = 8 IPRNT = 0
JCON = 4 IPNCH = 0

SCALE FACTORS

SCALE(1) = 1000.000 SCALE(2) = 1.000 SCALE(3) = 1.000 SCALE(4) = 5280.000

INTERFACE ROW-SEG	AREA (FT**2)	E (MI**2/D)	Q (CFS)	INTERFACE ROW-SEG	AREA (FT**2)	E (MI**2/D)	Q (CFS)	INTERFACE ROW-SEG	AREA (FT**2)	E (MI**2/D)	Q (CFS)
(1- 1)	0.	0.0	-150.0	(1- 2)	39600.	0.500	150.0	(1- 0)	0.	0.0	0.0
(1- 0)	0.	0.0	0.0	(1- 0)	0.	0.0	0.0	(1- 0)	0.	0.0	0.0
(2- 3)	95040.	0.500	150.0	(2- 1)	39600.	0.500	-150.0	(2- 0)	0.	0.0	0.0
(2- 0)	0.	0.0	0.0	(2- 0)	0.	0.0	0.0	(2- 0)	0.	0.0	0.0
(3- 4)	100320.	1.000	150.0	(3- 2)	95040.	0.500	-150.0	(3- 0)	0.	0.0	0.0
(3- 0)	0.	0.0	0.0	(3- 0)	0.	0.0	0.0	(3- 0)	0.	0.0	0.0
(4- 5)	105600.	1.000	243.0	(4- 6)	36960.	0.500	0.0	(4- 3)	100320.	1.000	-150.0
(4- 0)	0.	0.0	0.0	(4- 0)	0.	0.0	0.0	(4- 0)	0.	0.0	0.0
(5- 8)	99000.	1.500	243.0	(5- 7)	31680.	0.500	0.0	(5- 4)	105600.	1.000	-243.0
(5- 0)	0.	0.0	0.0	(5- 0)	0.	0.0	0.0	(5- 0)	0.	0.0	0.0
(6- 7)	21200.	0.500	0.0	(6- 4)	36960.	0.500	0.0	(6- 0)	0.	0.0	0.0
(6- 0)	0.	0.0	0.0	(6- 0)	0.	0.0	0.0	(6- 0)	0.	0.0	0.0
(7- 5)	31680.	0.500	0.0	(7- 6)	21200.	0.500	0.0	(7- 0)	0.	0.0	0.0
(7- 0)	0.	0.0	0.0	(7- 0)	0.	0.0	0.0	(7- 0)	0.	0.0	0.0
(8- 8)	158400.	1.500	243.0	(8- 5)	99000.	1.500	-243.0	(8- 0)	0.	0.0	0.0
(8- 0)	0.	0.0	0.0	(8- 0)	0.	0.0	0.0	(8- 0)	0.	0.0	0.0

CHARAC. LENGTHS OF SEGMENTS (FT)

INTERFACE	LENGTH	INTERFACE	LENGTH	INTERFACE	LENGTH	INTERFACE	LENGTH	INTERFACE	LENGTH	INTERFACE	LENGTH
1- 1	5280.	1- 2	5280.	1- 0	0.	1- 0	0.	1- 0	0.	1- 0	0.
2- 1	5280.	2- 3	5280.	2- 0	0.	2- 0	0.	2- 0	0.	2- 0	0.
3- 2	5280.	3- 4	5280.	3- 0	0.	3- 0	0.	3- 0	0.	3- 0	0.
4- 3	5280.	4- 5	5280.	4- 6	5280.	4- 0	0.	4- 0	0.	4- 0	0.
5- 4	5280.	5- 8	5280.	5- 7	5280.	5- 0	0.	5- 0	0.	5- 0	0.
6- 4	5280.	6- 7	5280.	6- 0	0.	6- 0	0.	6- 0	0.	6- 0	0.
7- 5	5280.	7- 6	5280.	7- 0	0.	7- 0	0.	7- 0	0.	7- 0	0.
8- 5	5280.	8- 8	5280.	8- 0	0.	8- 0	0.	8- 0	0.	8- 0	0.

SECTION 4 HAS AN EXCESS FLOW OF 92.99998 CFS

VALUES OF ALPHA AND EPRIM

INTRFC	EPRIM (MGD)	ALPHA	INTRFC	EPRIM (MGD)	ALPHA	INTRFC	EPRIM (MGD)	ALPHA	INTRFC	EPRIM (MGD)	ALPHA	INTRFC	EPRIM (MGD)	ALPHA	INTRFC	EPRIM (MGD)	ALPHA
1- 1	0.	1.000	1- 2	782.	0.500	1- 0	0.	0.0	1- 0	0.	0.0	1- 0	0.	0.0	1- 0	0.	0.0
2- 3	1877.	0.500	2- 1	782.	0.500	2- 0	0.	0.0	2- 0	0.	0.0	2- 0	0.	0.0	2- 0	0.	0.0
3- 4	3963.	0.500	3- 2	1877.	0.500	3- 0	0.	0.0	3- 0	0.	0.0	3- 0	0.	0.0	3- 0	0.	0.0
4- 5	4171.	0.500	4- 6	730.	0.500	4- 3	3963.	0.500	4- 0	0.	0.0	4- 0	0.	0.0	4- 0	0.	0.0
5- 8	5866.	0.500	5- 7	626.	0.500	5- 4	4171.	0.500	5- 0	0.	0.0	5- 0	0.	0.0	5- 0	0.	0.0
6- 7	419.	0.500	6- 4	730.	0.500	6- 0	0.	0.0	6- 0	0.	0.0	6- 0	0.	0.0	6- 0	0.	0.0
7- 5	626.	0.500	7- 6	419.	0.500	7- 0	0.	0.0	7- 0	0.	0.0	7- 0	0.	0.0	7- 0	0.	0.0
8- 8	9385.	0.500	8- 5	5866.	0.500	8- 0	0.	0.0	8- 0	0.	0.0	8- 0	0.	0.0	8- 0	0.	0.0

SECTION	TEMPERATURE (C)	VOLUME (10**6GAL)	DEPTH (FT)
1	21.00	625.63	12.00
2	22.00	2033.06	13.00
3	22.00	3127.99	15.00
4	22.00	4170.62	20.00
5	22.00	4587.63	22.00
6	24.00	834.09	4.00
7	24.00	1042.64	5.00
8	22.00	6255.89	30.00

SEGMENT	CL BOUNDARY CONDITION(MG/L)
1	0.0
8	1000.00

SECTION	CHLORIDE BOUNDARY LOAD (MGD*MG/L)
1	0.0
2	0.0
3	0.0
4	0.0
5	0.0
6	0.0
7	0.0
8	9306581.00

SECTION	CHLORIDES(MG/L)
1	755.943
2	855.837
3	901.211
4	923.532
5	957.424
6	932.199
7	947.311
8	983.405

SEGMENT	BOD BOUNDARY CONDITION(MG/L)
1	2.00
8	0.50

BOD LOAD = 100000. POUNDS/DAY FOR SECTION

THE TEMPERATURE CORRECTION FACTOR FOR THE BOD REMOVAL RATE = 1.047

SECTION	BOD REM RATE *TEMP CORR*	BOD WASTE LOAD (MGD*MG/L)	BOD BOUNDARY LOAD (MGD*MG/L)
1	0.366	0.0	193.890
2	0.384	0.0	0.0
3	0.384	0.0	0.0
4	0.384	11990.406	0.0
5	0.384	0.0	0.0
6	0.421	0.0	0.0
7	0.421	0.0	0.0
8	0.384	0.0	4653.289

SECTION	BOD(MG/L)
1	0.899
2	1.035
3	1.538
4	2.257
5	1.112
6	1.335
7	0.846
8	0.638

SEGMENT	DEF BOUNDARY CONDITION(MG/L)
1	0.50
8	0.50

THE TEMPERATURE CORRECTION FACTOR FOR THE REAERATION RATE = 1.024
THE BOD CORRECTION FACTOR (FL) = 1.000
THE TEMPERATURE CORRECTION FACTOR FOR THE BENTHAL DEMAND = 1.080
THE TEMP CORR FACTOR FOR KD = 1.047

SECTION	DEOX RATE *TEMP CORR*	REAER RATE *TEMP CORR*	BOUND COND LOAD (MGD*MG/L)	TOTAL LOAD (MGD*MG/L)	PMR MG/L/D	BD *TEMP CORR*
1	0.366	0.246	48.472	254.649	0.0	0.0
2	0.384	0.241	0.0	807.254	0.0	0.0
3	0.384	0.231	0.0	1845.740	0.0	0.0
4	0.384	0.178	0.0	3611.526	0.0	0.0
5	0.384	0.157	0.0	1957.236	0.0	0.0
6	0.421	0.176	0.0	514.864	0.0	0.068
7	0.421	0.198	0.0	110.375	0.250	0.0
8	0.384	0.126	4653.289	6184.867	0.0	0.0

SECTION	DEFICIT(MG/L)
1	1.643
2	1.858
3	2.012
4	1.987
5	1.469
6	2.000
7	1.492
8	0.930

SECTION	CHLORIDES	TEMPERATURE	DEFICIT	SATUP.TION	DO
1	755.94	21.00	1.64	8.78	7.14
2	855.84	22.00	1.86	8.60	6.74
3	901.21	22.00	2.01	8.60	6.58
4	923.53	22.00	1.99	8.59	6.61
5	957.42	22.00	1.47	8.59	7.12
6	932.20	24.00	2.00	8.26	6.26
7	947.31	24.00	1.49	8.26	6.77
8	983.40	22.00	0.93	8.59	7.66

SEGMENT	BOD BOUNDARY CONDITION(MG/L)
1	2.00
8	0.0

BOD LOAD = 100000. POUNDS/DAY FOR SECTION 4

THE TEMPERATURE CORRECTION FACTOR FOR THE BOD REMOVAL RATE = 1.080

SECTION	BOD REM RATE *TEMP CORR*	BOD WASTE LOAD (MGD*MG/L)	BOD BOUNDARY LOAD (MGD*MG/L)
1	0.108	0.0	193.890
2	0.117	0.0	0.0
3	0.117	0.0	0.0
4	0.117	11990.406	0.0
5	0.117	0.0	0.0
6	0.136	0.0	0.0
7	0.136	0.0	0.0
8	0.117	0.0	0.0

SECTION	BOD(MG/L)
1	2.369
2	2.635
3	3.098
4	3.615
5	1.832
6	2.731
7	1.930
8	0.681

SEGMENT	DEF BOUNDARY CONDITION(MG/L)
1	0.0
8	0.0

THE TEMPERATURE CORRECTION FACTOR FOR THE REAERATION RATE = 1.024
THE BOD CORRECTION FACTOR (FL) = 1.000
THE TEMPERATURE CORRECTION FACTOR FOR THE BENTHAL DEMAND = 1.080
THE TEMP CORR FACTOR FOR KD = 1.080

SECTION	DEOX RATE *TEMP CORR*	REAER RATE *TEMP CORR*	BOUND COND LOAD (MGD*MG/L)	TOTAL LOAD (MGD*MG/L)	PMR MG/L/D	BD *TEMP CORR*
1	0.108	0.246	0.0	160.041	0.0	0.0
2	0.117	0.241	0.0	624.963	0.0	0.0
3	0.117	0.231	0.0	1130.468	0.0	0.0
4	0.117	0.178	0.0	1758.574	0.0	0.0
5	0.117	0.157	0.0	980.310	0.0	0.0
6	0.136	0.176	0.0	309.920	0.0	0.0
7	0.136	0.198	0.0	273.810	0.0	0.0
8	0.117	0.126	0.0	497.244	0.0	0.0

SECTION	DEFICIT(MG/L)
1	0.965
2	1.077
3	1.075
4	0.983
5	0.639
6	1.085
7	0.902
8	0.268

RESULTS FOR COMBINED (CARBONACEOUS AND NITROGENOUS) RUN

SECTION	CHLORIDES (MG/L)	TEMPERATURE (C)	DEFICIT (MG/L)	SATURATION (MG/L)	DO (MG/L)
1	755.94	21.00	2.61	8.78	6.17
2	855.84	22.00	2.94	8.60	5.66
3	901.21	22.00	3.09	8.60	5.51
4	923.53	22.00	2.97	8.59	5.62
5	957.42	22.00	2.11	8.59	6.48
6	932.20	24.00	3.08	8.26	5.18
7	947.31	24.00	2.39	8.26	5.87
8	983.40	22.00	1.20	8.59	7.39

The following JCL statements apply to users of HARØ3 at OSI computer facilities:

```
//INTHARØ3 JOB (ACNT,INT,2,5),PROGNAME

//JOBLIB DD UNIT=333Ø,VOL=SER=BCSØ25,DISP=SHR,DSN=CHAPRA.xxxxx

//STEP1 EXEC PGM=MODEL,REGION=nnnK

//GO.FTØ7FØØ1 DD SYSOUT=B

//GO.FTØ6FØØ1 DD SYSOUT=A

//GO.FTØ5FØØ1 DD *

   Place data cards here

/*
```

Note:

```
   xxxxx - HA1ØØ for 100 segment model
           HA2ØØ for 200 segment model
            HA5Ø for  50 segment model

     nnn -  76 K for  50 segment model
           148 K for 100 segment model
           41Ø K for 200 segment model

     INT - OSI user initials
    ACNT - OSI account
```

Restrictions

1) HARO3 is presently limited to a system of 200 segments

2) A section may have only one interface act as a boundary

3) Each section may have up to six interfaces

APPENDIX

References

Introduction

1. Streeter, H.W. and Phelps, E.B., A Study of the Pollution and Natural Purification of the Ohio River, III. Factors Concerned in the Phenomena of Oxidation and Reaeration. U.S. Public Health Service, Public Health Bulletin Number 146, February, 1925, 75 pp. Reprinted by U.S. Department of Health, Education and Welfare, PHS, 1958.

2. Thomann, R.V. Mathematical Model for Dissolved Oxygen Journal of the Sanitary Engineering Division, ASCE, Vol.89 No. SA5, Oct. 1963, pp. 1-30.

Theory

1. Thomann, R.V. same as 2 above.

2. Thomann, R.V. Systems Analysis and Water Quality Management, "Environmental Research and Applications, Inc., New York, 1971.

3. Arons, A.B. and Stommel, H., "A Mixing Length Theory of Tidal Flushing", Transactions, American Geophysical Union, Vol. 32, No. 3, June 1951, pp. 419-421.

4. Stommel, H., "Computation of Pollution in a Vertically Mixed Estuary", Sewage and Industrial Wastes, Vol. 25, No. 9, September, 1953, pp. 1065-1071.

5. O'Connor, D.J., "Estuarine Distribution of Non-Conservative Substances", Journal of the Sanitary Engineering Division, ASCE, Vol. 91, No. SA1, February, 1965, pp. 23-42.

6. Thomann, R.V. System Analysis and Water Quality Management, p. 180

The Computer Program

1) Development of Water Quality Model of Boston Harbor Prepared for Massachusetts Water Resources Commission by Hydroscience, Inc., Westwood, N.J., July, 1971, 173 pp.

2) HARO1: A Steady State Estuarine Water Quality Computer Model prepared by Data Systems Branch U.S.E.P.A., Region II, N.Y., N.Y.

3) Thomann, R.V. Systems Analysis and Water Quality Management p.96 & 103

4) Committee on Sanitary Engineering Research, Solubility of Atmospheric Oxygen in Water, J. Sanit. Eng. Div., Am Soc. Civil Engrs., SA 4, 41 (1960).

LIST OF FIGURES

Introduction . . . . .

figure		page
I-1	D.O. sag generated by the Streeter - Phelps equation	I-3

Theory . . . . . .

figure		page
T-1	One dimensional Estuary of N segments	T-3
T-2	Schematic for a coupled system of two reactants	T-10
T-3	Schematic of generalized two-dimensional system	T-13
T-4	Simple schematic of HARO3 computer program	T-16

Computer Program . . . .

figure		page
C-1	Flow chart of the main program of HARO3	C-3

Users Manual . . . .

figure		page
U-1	Structure of data deck for HARO3	U-4
U-2	Deck setups for various combinations of reactants using program HARO3	U-5
U-2a	Depiction of characteristic lengths	U-8
U-3	Overview of fictitious tidal bay	U-15
U-4	Depth readings for the tidal bay	U-16
U-5	Dissolved oxygen concentrations	U-17
U-6	Segmentation scheme for the bay	U-17
U-7a	Results of model run with load to segment 6	U-19
U-7b	Results of model run with load to segment 4	U-19
U-8	Input deck for example run with load to section 4	U-20

LIST OF TABLES

U-1	Interface data for run with load to section 4	U-18
U-2	Section data with load to section 4	U-18

NOMENCLATURE

SYMBOL		DEFINITION	UNITS *
Documentation	Computer Program		
A	AREA (J) or ARRAY (I,JJ), where JJ=1,4,7,10,13,16 J=1,6 I=1,N	Cross-sectional area	L^2
[A]	AC(I,J) I=1,N J=1,N	Single system matrix. A matrix which represents a single substance which decays with first order kinetics (see page T-9)	L^3/T
$[A]^{-1}$	AC(I,J) I=1,N J=1,N	Single system response matrix. The inverse of [A], this matrix represents the response of the system to the input of a load of a single substance decaying with first order kinetics (see page T-10)	$[(M/L^3)/(M/T)]$
[AB] or AB	AB(I,J) I=1,N J=1,N	The system matrix. Since matrices [A] and [B] differ only in a reaction term in their diagonals, a matrix without such a term has been formulated and is called the system matrix (see page T-15)	L^3/T
AC	AC(I,J) I=1,N J=1,N	The specific constituent matrix. When the appropriate reaction term is added to the diagonal of the system matrix the matrix for a specific constituent is generated. In the theory they were called [A] or [B].	L^3/T
		As well, the term AC is used in the computer program to represent the matrix after it has been inverted.	$[(M/L^3)/(M/T)]$
[B]	AC(I,J) I=1,N J=1,6	A matrix which represents the second substance of coupled set of constituents which react with first order kinetics (see page T-11)	M/L^3
$[B]^{-1}$	AC(I,J) I=1,N J=1,6	The inverse of [B], this matrix represents the response of the system to the input of a load of the second substance of a coupled set of constituents.	$[(M/L^3/(M/T)]$
BD	BD(I) I=1,N	Benthal oxygen demand	$M/L^2/T$
BOD		Biochemical oxygen demand	M/L^3
c		concentration of a constituent	M/L^3

* units; T=time, L=length, M=mass

SYMBOL		DEFINITION	UNITS
Documentation	Computer Program		
$[C]^{-i}$		The total system response matrix relates input of the first substance in a coupled set of reactants to the response of the second constituent.	$[(M/L^3)/(M/T)]$
CBOD		Component of BOD due primarily to oxygen utilized by suprophytic organisms as they utilize the carbonaceous matter in a waste	M/L^3
D		Dissolved oxygen deficit = the saturation value minus the actual concentration of dissolved oxygen = DO_s - DO	M/L^3
D_o		The initial concentration of dissolved oxygen deficit at the point introduction of waste into a stream.	M/L^3
DO		Concentration of dissolved oxygen	M/L^3
DO_s	CS	Saturation concentration of dissolved oxygen	M/L^3
E	E(J) ARRAY(I,JJ+1) where I=1,N J=1,6 JJ=1,4,7,10, 13,16	Longitudinal dispersion coefficient	L^2/T
E'	EPRIM(J) J=1 6	Bulk dispersion coefficient =EA/1	L^3/T
H	H	depth	L
k or K		First order reaction coefficient, (also used to designate a section)	1/T
K_a	KA	Reaeration rate (first order)	1/T
K_d	KD	Deoxygenation rate (first order)	1/T
K_r	KA	BOD removal rate (first order)	1/T

SYMBOL		DEFINITION	UNITS
Documentation	Computer Program		
l		section length	L
L_o		Initial concentration of BOD at point of introduction of a waste into a stream.	M/L^3
NBOD		Nitrogenous component of BOD primarily caused be autotrophic bacteria utilizing nitrogen in waste.	M/L^3
PMR	PMR	Net algal oxygen rate = photosynthesis effect - respiration effect of algae on oxygen	$M/L^3/T$
Q	Q(J) ARRAY(I,JJ+2) where J=1,6 I=1,N JJ=1,4,7,10, 13,16	net advective flow	L^3/T
t		time	T
T	T(I)	temperature	° C
U		advective velocity	L/T

SYMBOL		DEFINITION	UNITS
Documentation	Computer Program		
vol	VOL	Volume	L
W	W	Waste load of a constituent	M/T
W_b		Deficit waste load	M/T
W_c		BOD waste load	M/T
x		Distance	L
Y	Y	Dissolved oxygen deficit waste load	M/L^3
α	ALPHA	weighting coefficient used in the approximation scheme of the model.	
β	1-ALPHA	weighting coefficient = $1-\alpha$	

Acknowledgements

The material presented in this report is based on extensions and revisions of the first version and that original work is gratefully acknowledged. Steve Chapra has moved from E.P.A. and is presently an Environmental Engineer for the Great Lakes Environmental Research Laboratories at Ann Arbor, Michigan.

Thanks must be extended to Richard Winfield of the Manhattan College Department of Environmental Engineering and Science and to Robert Braster, Salvatore Nolfo and Kevin Bricke of E.P.A. for their thoughtful reviews of the material contained in this report.

The typing of the report was very patiently and professionally handled by Ms. Maryann LaBarbera.

G. A. N